

## Genera 8.3 Software Installation Guide for UX Family Machines

### CD-ROM Distribution Media for All Sites

All sites with Ivory-based machines receive the following CD-ROM distribution media:

*Genera 8.3 Distribution Worlds, Sources, and Bundled Products, all platforms*, 1 CD, P.N. 995854 (This CD contains Genera 8.3 software for all machine types, including 3600 Family microcodes, Ivory FEP Kernel and Flods, MacIvory Macintosh software, UX UNIX software, both Netboot cores, all Distribution worlds, system sources, examples, fonts, documentation and loadable systems. In addition, the CD contains sources and binaries for the layered products Symbolics C, Fortran, and Pascal.) This CD also contains the Symbolics Color layered product. **Note:** You must be under license for Symbolics Color in order to restore the Color product. Contact your sales representative for more information on the color product.

### Disk Space Requirements for Genera

#### Worlds:

All new machines are shipped from the factory with the most recent version of Genera worlds already on disk.

#### Ivory:

Netboot-Core-from-Genera-8-3.ilod.1	269 [Netboot core: Genera 8.3]
Genera-8-3.ilod.1	44402 [Genera 8.3]
Genera-8-3-UX-Delivery.ilod.1	19325 [Genera 8.3 UX Delivery]
Genera-8-3-MacIvory-Delivery.ilod.1	18367 [Genera 8.3 MacIvory Delivery]

#### 3600:

Netboot-Core-from-Genera-8-3.load.1	161 [Netboot core: Genera 8.3]
Genera-8-3.load.1	38367 [Genera 8.3]

#### Sources and loadable systems:

Note that the Genera 8.3 CD-ROM can be used as a SYS host, eliminating the need to restore sources and loadable systems to disk.

- Using the Genera 83 CD-ROM as a SYS host requires only about 260 LMFS records (for the SYS:SITE; directory).

- Restoring the Genera 8.3 sources in their entirety requires approximately 41,250 LMFS records (equivalent to approximately 165,000 Ivory FEP blocks or 204,000 KBytes for UNIX).

### Overview of CD-ROM Installation

This section describes the general procedure for installing Genera from CD-ROM. **Note:** In all cases, make sure that machine is powered off before connecting or removing the CD-ROM drive.

- Connect the CD-ROM drive to one of the following:
  - An NXP1000 or a Symbolics XL Family machine. See the section "Installing a SCSI Device". Symbolics has qualified the Apple CD-ROM drive (Sony model CDU-8001) for use with NXP1000s and XLs, however, other SCSI CD-ROM drives may also work.
  - A Symbolics MacIvory. Follow the manufacturer's instructions for configuring and connecting a SCSI CD-ROM drive.
  - A Sun workstation. Follow the manufacturer's instructions for configuring and connecting a SCSI CD-ROM device.
- When the CD-ROM drive is connected to a Sun, you must make the CD-ROM drive accessible to other machines on the network.

### Setting up Network Access for a CD-ROM Drive on a Sun Workstation

1. Check the UNIX permissions on `/dev/sr0`. These permissions will be propagated by SunOS to all files on the Genera Distribution CD-ROM. At a minimum, the read and execute bits must be set for the user who will perform the installation (often root). For more information on setting UNIX file permissions, see the SunOS Reference Manual entry for `chmod`. For more information on the Sun CD-ROM driver, see the SunOS Reference Manual entry on `sr`.
2. Insert the CD-ROM in the Sun CD-ROM drive, and turn on the drive. Mount the CD-ROM into the Sun filesystem using the UNIX `mount` command (the `mount` command must be run by the super-user):

```
% su
# mount -o ro -t hsfs /dev/sr0 /cdrom
```

For more information on the UNIX `mount` command, see the SunOS Reference Manual entry on `mount`.

3. On the machine with the CD-ROM drive, make sure that the directory the CD-ROM is mounted onto is exported (see the SunOS Reference Manual entry on `exportfs`).

### CD-ROM Installation Steps - an Overview

1. *(Sites with MacIvories only)*  
Install the Genera 8.3 MacIvory Macintosh software on all MacIvories which will run Genera 8.3. Since installing this software from CD-ROM requires a CD-ROM drive on the local MacIvory, we suggest that you install this from the Genera 8.3 MacIvory floppies. See the section "Installing the MacIvory Diskettes".
2. *(Sites with UX Family machines only)*  
Install the Genera 8.3 UX UNIX software on all UX Family machines which will run Genera 8.3. Run the UX UNIX software installation tool, to install the Genera 8.3 UX UNIX software from the tape. See the section "Upgrading the Symbolics UX UNIX Software".
3. Restore the Genera 8.3 Distribution worlds from CD-ROM, see the section "Restoring the Genera Distribution Worlds From CD-ROM".
4. Make a site-configured Genera 8.3 world for each machine-type. See the section "Making a Genera Site-Configured World".

If your CD-ROM drive is connected to a MacIvory or an XL running Genera 8.3, it is possible to use the CD-ROM as a `SYS` host. If you wish to store any portion of the Genera 8.3 documentation, sources, examples, fonts, and non-loaded systems on CD-ROM instead of in a LMFS, See the section "Using a CD-ROM as a `SYS` Host" and use that section as a guide when updating your `sys.translations` file.

5. If there are any Genera 8.3 documentation, sources, examples, fonts, or non-loaded systems that you do *not* plan to access from CD-ROM, you must restore them into a Lisp Machine File System (LMFS) or a UNIX File System. For instructions on how to do this, see the section "Restoring the Genera Sources".
6. On each machine that is now running a site-configured Genera 8.3 world, copy the new flod files. See the section "Copying New Flod Files" for each machine type. **Note:** The Flod files for 3600-family machines are the same as in previous Genera releases. Therefore, this step is not necessary on 3600-family machines.
7. Cold boot each machine with the new flods.

8. Make customized user and server worlds for your site. See the section "Making Customized Genera User and Server Worlds".
9. Back up your world(s) to tape. See the section "Backing Up Worlds to Tape".

### Upgrading the Symbolics UX UNIX Software

Upgrading the Symbolics UX UNIX Software for a new release of Genera is subtly different from installing a new UX system. For more information on UX installation, use the online documentation for the *Genera 8.1 Software Installation Guide*.

If the Sun CD-ROM is not connected to the machine on which the UX UNIX software is to be installed, the CD-ROM must be mounted onto the UX machine's filesystem. On the machine with the CD-ROM drive, make sure that the directory the CD-ROM is mounted onto is exported (see the SunOS Reference Manual entry on `exportfs`). On the machine on which the UX UNIX software is to be installed, mount the CD-ROM onto the filesystem:

```
% su
# mkdir /cdrom
# mount OtherSun:/cdrom /cdrom
```

In order to upgrade the Symbolics UX UNIX Software, follow the installation procedure described in *Installing and Configuring the Symbolics UX UNIX Software*, but answer No to the questions in "Phase 4 (FEP Partition File Configuration) of Symbolics UX Installer Program" and some of the questions in "Phase 5 (UNIX Startup Configuration) of Symbolics UX Installer Program".

More specifically, the usual "Phase 5 (UNIX Startup Configuration) of Symbolics UX Installer Program" answers for an upgrade will be as follows:

Do you wish to install the UX400S/UX1200S kernel?

**Yes.** *Upgrading the UX UNIX software requires building a new UNIX kernel which will include the new release of the UX device drivers. This question controls whether the new kernel will be installed (the kernel is actually built in Phase 3).*

Do you wish to create the UX400S/UX1200S configuration file(s)?

**No.** *The UX configuration files only need to be updated in order to add or remove network interfaces of FEP partition files, or to change the directory where IFEP PROM files are stored, or when a new UX board is being installed. Upgrading the UX UNIX software doesn't usually involve any of these situations.*

Do you wish to create/update the UX400S/UX1200S device files in /dev?

**Yes.** *The UX device files should be updated whenever a new UNIX kernel*

*is built, in order that they specify the correct major device number for the UX device driver.*

Do you wish /etc/rc.local to start UX400S/UX1200S life support?

**No.** /etc/rc.local *only needs to be updated when a new UX board is added to the system.*

Do you wish to reboot UNIX after installation?

**Yes.** *UNIX should be rebooted whenever a new kernel is installed.*

A sample upgrade dialogue would look something like the following.

% **su**

Password:

# **cd /cdrom/sun**

# **extract\_cdrom**

The following product will be installed:

Symbolics UX400S/UX1200S UNIX Software for SunOS 4.1.2, Genera 8.3

Copyright (c) 1993 Symbolics, Inc. All rights reserved.

Portions of font library Copyright (c) 1984 Bitstream, Inc.

All Rights Reserved.

#### TRADEMARK NOTICE

Symbolics, Ivory, Symbolics UX400S, and Symbolics UX1200S are trademarks and Genera is a registered trademark of Symbolics, Inc.

UNIX(R) is a registered trademark of AT&T. X Window System is a trademark of the Massachusetts Institute of Technology. Open Windows, Sun, Sun-3, Sun-4, and SunOS are trademarks, and Sun Microsystems is a registered trademark of Sun Microsystems, Inc.

Use the Genera Show Legal Notice command to see important legal notices.

Do you want to continue [y|n]? **y**

Do you wish to restore UX-family sun3 life support? **y**

Do you wish to restore UX-family life support sources? **y**

Do you wish to restore the Genera fonts? **y**

extract\_cdrom: Restoring files

x ./sun3/etc/test-ivory, 139264 bytes, 272 tape blocks  
 x ./ifep/solstice-device.prom, 130717 bytes, 256 tape blocks  
 x ./src/Makefile, 58127 bytes, 114 tape blocks  
 x ./src/autoboot.c, 4193 bytes, 9 tape blocks

*[many other files here ...]*

extract\_cdrom: Running installation tool

Do you wish to install the UX400S/UX1200S software (default Yes)?

The ivory-life program has to be run by root, the super-user.

Do you wish ivory-life to be installed setuid to root (default Yes)?

Certain services provided by the ivory-servicesd program require that it run as root.

Do you wish ivory-servicesd to be installed setuid to root (default Yes)?

Several of the UX400S/UX1200S life-support programs want to be able to read /dev/kmem.

Do you wish these programs to be installed setgid to kmem (default Yes)?

Where should application programs go (default /usr/bin)?

Where should maintenance and installation programs go (default /usr/etc)?

Where should include files go (default /usr/include/ivory)?

Where should libraries go (default /usr/lib)?

Do you wish to install sources to the UX400S/UX1200S software (default Yes)?

Where should sources go (default /usr/share/symbolics/src)?

Where should IFEP proms go (default /usr/share/symbolics/ifep)?

Do you wish to install the Genera fonts for the Sun Open Windows X server (default Yes)?

Where do the Open Windows programs live (default /usr/openwin)?

Where should the Open Windows Genera fonts go (default /usr/openwin/lib/genera-fonts)?

WARNING: directory /usr/openwin/lib/genera-fonts is not in the search path defined by environment variable FONTPATH. Make sure to update FONTPATH to include /usr/openwin/lib/genera-fonts.

Do you wish to install the Genera fonts for the MIT X server (default No)?

Do you wish to build a UNIX kernel with support for the UX400S/UX1200S (default Yes)?

How many UX400S boards should this kernel support (default 1)? **0**

How many UX1200S boards should this kernel support (default 1)?

Where is your kernel build directory (default /usr/share/sys)?

What kernel architecture does this machine use (default sun3)?

What kernel usually runs on this machine (default GENERIC)?

Do you wish to create FEP partition file(s) (default Yes)? **no**

Do you wish to install the UX400S/UX1200S kernel (default Yes)? **yes**

Do you wish to create the UX400S/UX1200S configuration file(s) (default Yes)? **no**

Do you wish to create/update the UX400S/UX1200S device files in /dev (default Yes)? **yes**

Do you wish /etc/rc.local to start UX400S/UX1200S life support (default Yes)? **no**

Do you wish to reboot UNIX after installation (default Yes)? **yes**

About to begin UX400S/UX1200S software installation, this will probably take a while.  
Okay (default Yes)?

### Restoring the Genera Distribution Worlds From CD-ROM

**Note:** CD-ROM installation only works properly in Genera 8.1.1 or later versions. Be sure to install the Genera 8.3 worlds using a machine running Genera 8.1.1. or later.

Restore the Genera 8.3 world load files into your FEP file system, using the CD-ROM labeled GENERA 8.3 DISTRIBUTION WORLDS, SOURCES, AND BUNDLED PRODUCTS.

1. Insert the Genera 8.3 CD-ROM into the CD-ROM drive at your site. Type the CP command Select Activity FEP-Tape. In the FEP-Tape activity, type the command:

```
Read Image File
```

We recommend that you type this command, rather than choose it from the FEP Tape menu, because the typed command gives you more control over the tape specification. Supply the pathname of a tape image on the Genera 8.3 CD-ROM according to the following:

CD-ROM drive on a Lisp machine

```
Lisp-host|CDROMn:>DISTRIBUTION>BASIC_WORLD.TAPE
```

CD-ROM drive on a Sun

```
Sun-host:/cdrom/distribution/basic_world.tape
```

where "n" is the SCSI address of the ROM Drive.

2. On Ivory-based machines, you are asked if you want to restore the FEP kernel and flod files. Answer Y (for Yes) and accept the default pathname.

**Note:** Do not delete your old FEP kernel and flod files. We recommend that you always save the previous version of the kernel and flod files for a backup. See the section "Saving Previous FEP Kernels and FLOD Files".

3. You are asked if you want to restore the Genera 8.3 Netboot Core. Answer Y (for Yes) and accept the default pathname.
4. You are asked if you want to restore the Genera 8.3 world load file. Answer Y (for Yes) and accept the default pathname, or specify a FEP pathname that

indicates an Ivory world load, such as:

```
FEP0:>Genera-8-3.ilod
```

The FEP-Tape loading program then loads the world from the CD-ROM.

5. Update the boot.boot file on your machine to load the new world. Your updated boot.boot file should look similar to this:

```
Load World
Start
```

This automatically loads the most recently created world on your disk (excluding netboot cores). **Note:** The next time you boot your machine, you must cold boot the FEP in order to use the new FEP kernel and FLOD files.

See the section "Contents of Hello.Boot Files and Boot.Boot Files".

6. For the remaining Genera 8.3 distribution worlds, use the Copy File command to copy the appropriate Genera worlds from the "worlds" directory on the CD-ROM to your local FEP filesystem.

If the CD-ROM drive is connected to an NXP1000, XL or a MacIvory, the world pathnames will look similar to this (where *host* is the name of the CD-ROM host, and *n* in *CDROMn* is the SCSI address of the CD-ROM drive):

```
cdrom-host|CDROMn:>WORLDS>GENERA_8_3.ILOD
```

If the CD-ROM drive is connected to a Sun workstation, the world pathnames will look similar to this (where *host* is the name of the Sun CD-ROM host, and */cdrom* is the directory the CD-ROM is mounted onto).

```
cdrom-host:/cdrom/worlds/genera_8_3.load
```

**Note:** The ISO9660 pathnames on the CD-ROM contain underscores which should be changed to hyphens when you copy the files to your FEP.

For example, when copying the Genera 8.3 world from CD-ROM to a 3600, with the CD-ROM drive connected to a Sun, you would type the following to a Lisp Listener (where *3600-host* is the name of the 3600):

```
Copy File cdrom-host:/cdrom/worlds/genera_8_3.load 3600-host|FEP0:>genera-8-3.load
```

Similarly, with the CD-ROM drive, SCSI address 3, connected to a MacIvory or XL, you would type the following:

```
Copy File CDROM-HOST|CDROM3:>WORLDS>GENERA_8_3.LOAD (to)
3600-host|FEP0:>genera-8-3.load
```

The worlds directory on the CD-ROM contains the following worlds:

```

GENERA_8_3.ILOD
GENERA_8_3.LOAD
GENERA_8_3_MACI_DELIVERY.ILOD
GENERA_8_3_UX_DELIVERY.ILOD
NETBOOT_CORE_GENERA_8_3.ILOD
NETBOOT_CORE_GENERA_8_3.LOAD

```

## Completing the Installation

### Creating a New Site for Genera

The steps in this section should be performed at new sites that have multiple Symbolics (3600-family, MacIvory, XL, NXP, or UX) machines.

### Choosing Names, Addresses, and Servers for the New Site

1. Select a name for the site. For information about choosing site names, see the section "Choosing a Site Name".
2. Select a name for each machine at the site, including non-Symbolics machines.
3. Select a network address for each Symbolics computer. Symbolics UX machines must have both a Chaosnet address and an Internet address.

Remember that each machine must have a unique network address. Machines that are not Symbolics computers might already have Internet addresses for IP-TCP networks. If not, select the appropriate network addresses for them. For information about network addresses, see the section "Choosing Machine Names and Network Addresses".

**Note:** All preconfigured machines come from Symbolics with Chaosnet address 52525. *Unless you have a standalone machine, this is not a desirable Chaosnet address.*

4. Choose a Symbolics computer to be your namespace server. You cannot use a non-Symbolics machine for a namespace server. For information about server machines, see the section "Machines and Worlds".
5. Choose a machine to be the file server and, if necessary, the print server, mail server, and Domain server. Symbolics recommends that you use your namespace server machine for these, provided it has enough disk space. More information is available about choosing server machines. See the section "Machines and Worlds".

## Contents of Hello.Boot Files and Boot.Boot Files

This section briefly describes the philosophy of which FEP commands belong in the Hello.boot file and which belong in the Boot.boot file on both 3600-family and Ivory-based machines.

### Hello.boot File

This file should contain anything that holds constant for this machine, regardless of the world you are running. For example, this includes your flod files, your paging files, your network address, and so on.

First, this file must scan all the flods from which you want to use commands. This includes the lisp, loaders, info, and debug flods.

Next, the file should include the Initialize Hardware Tables command, so that any errors it encounters will be seen early in the booting process. (This command is completely optional in Ivory-based machines. It is automatically invoked by the Ivory FEP when needed, but errors it signals may be less clear when automatically invoked.)

Then, the Hello.boot file should declare any paging files you always want to use; this saves you having to manually Add Paging Files, or to say the same in your boot file. Do not declare any "spare" paging you want to save for an emergency, because Load World automatically uses all declared paging files, and thus the spare paging file would not be available in case of an emergency. It is harmless to declare paging files that do not exist (for example, one you might delete every now and then to make room for Copy World); nonexistent ones give a warning, but are not used.

### Ivory machines only:

Starting with Genera 8.1, the hello.boot file for an Ivory machine should contain two lines:

```

Hello Innn
Hello Local (or hostname)

```

Each of these corresponds to a boot file. The *nnn* refers to the IFEP number, which is 328 for Genera 8.3.

### Hello Innn Boot File

The *Innn*.boot file (where *nnn* is the IFep version number, which is 328 for Genera 8.3) should contain the commands to scan the flod files and initialize things.

```

Scan I328-lisp.flod
Scan I328-loaders.flod
Scan I328-info.flod
Scan I328-debug.flod
Initialize Hardware Tables

```

### Hello Local Boot File

The local.boot file should contain those commands that set up this specific machine, declaring paging files, setting the network address, and any other boot options.

```
Declare Paging Files FEP0:>Paging-1.page
Declare More Paging Files FEP0:>Paging-2.page,Paging-3.page
Set Boot Options :Network Address Chaos|52525 :IDS Enable
```

Boot options are required for standalone sites. They are optional for other sites, but they do save manually invoking them or having them duplicated or inconsistent in boot files (if you have multiple boot files). See the section "Set Boot Options FEP Command".

The boot options replace other FEP commands. For compatibility, some of the boot options have corresponding commands. Note that some of these commands (such as Enable/Disable IDS on 3600-family machines), if present, must appear in the Boot.boot file, since they must follow Load World. (In the Ivory-based FEP, these commands warn if you try to use them at the wrong time, while in the 3600-family FEP they silently perform no operation if used at the wrong time). We recommend using the Set Boot Options command instead of the corresponding commands, so you can set all the boot options in one spot.

Normally, you set your network address with Set Boot Options, but you could use the Set Network Address command instead. Whichever way you choose to set the network address, it should be done in the Local.boot file, and should not be duplicated in your Boot.boot files.

### 3600 and Ivory Machines

Note that, in previous releases, you had to set the network address after loading the world, but this restriction no longer holds on either architecture.

Here is a sample Hello.boot file for a 3600-family machine:

#### Sample 3600-Family Machine Hello.Boot File

```
Scan v127-info.flod
Scan V127-loaders.flod
Scan v127-lisp.flod
Scan v127-debug.flod
Initialize Hardware Tables
Declare Paging-Files FEP0:>Paging-1.page
Declare More Paging-Files FEP0:>Paging-2.page,Paging-3.page
Set Chaos-Address 52525
```

(For a sample Hello.boot file for an Ivory-based machine, see the section "Sample Ivory-Based Machine Hello.Boot File".)

## Boot.boot Files

These files should contain any setting that pertains only to the particular world (and microcode, on a 3600-family machine) this file loads and starts. (Note that this excludes the address of the machine.) There is a slim possibility that you might want to have some explicit paging file commands here (for example, a boot file to boot with just one paging file explicitly added for when you want to do world copying).

### Sample 3600-Family Boot.Boot Files

Here is the sequence of commands for a Symbolics 3600-family machine that is cold booting a world from the local disk:

```
Clear Machine
Load Microcode FEP0:>3640-fpa-mic.mic.430
Load World FEP0:>genera-8-3.load
Enable IDS
Start
```

Here is the sequence of commands for a Symbolics 3600-family machine that is cold booting a world from a remote disk (netbooting):

```
Clear Machine
Load Microcode FEP0:>3640-fpa-mic.mic.430
Netboot inc-site-genera-8-3
Enable IDS
Start
```

The information in the rest of this section applies to Ivory-based machines only.

Boot.boot files for Ivory-based machines should contain only the following:

```
Load World
Start
```

Note that you can set your default world to boot in boot options. If you choose to load the default world, there is no need to give an explicit pathname argument to Load World.

It is unnecessary to do a Clear Machine before the Load World since this command does nothing on Ivory-based machines. On 3600 family machines, it serves a purpose. To avoid confusion and make boot files easier to understand and maintain, if Clear Machine is left in the boot file of an Ivory machine, it is ignored.

It is possible, although not recommended, to give the Enable/Disable IDS command here, but the recommended way is to do all that in your Hello.boot file, via Set Boot Options :IDS [Enable/Disable]. You should also use the Set Network Address commands in your Hello.boot file. (Note that if someone copies a boot file with an address in it to another machine, confusion will result.)

Note also that Disable IDS saves only an inconsequential amount of memory and overhead, so there is usually no advantage in doing it.

Here is the recommended sequence of commands for a Symbolics Ivory-based machine that is cold booting a world from the local disk:

```
Load World
Start
```

This loads the most recent world on your local disk, which is usually the one you want to boot.

For a netbooted machine, the recommended sequence is:

```
Netboot inc-site-genera-8-3
Start
```

At this point you are ready to boot Genera.

### Defining the New Site

This step should be performed at new sites with multiple machines only. Sites with a standalone XL or a standalone MacIvory do not need to define a site.

On the namespace server machine, use the Define Site command to create the initial namespace database. Once you have entered data into this database, the namespace files are stored permanently.

Define Site enables the local machine to access the permanent namespace database files, and at the same time, it creates a host namespace object for the local machine.

### Define Site Dialogue for XL, 3600, and MacIvory Machines

In a Lisp Listener, issue the Define Site command and give a site name. This might be the name of your company, or it can be related to the machine names you have chosen.

```
Command: Define Site (site name) Downunder
```

A menu comes up with with the necessary fields to be filled in, as in Figure 59 :

```
Command: Define Site (site name) Downunder
Defining site DOWNUNDER with the local host as the Primary Namespace Server
Namespace Server Name: the name of the primary namespace server
<ABORT> aborts, <END> uses these values
```

Figure 61. The basic Define Site menu

Enter the name of your namespace server. The Define Site dialogue expects an actual name here, so do not enter Local for the Namespace Server name.

After you have entered the name of your namespace server, additional menu slots are displayed, with the appropriate default information filled in. You can, of course, change them.

```

Command: Define Site (site name) downunder
Defining site DOWNUNDER with the local host as the Primary Namespace Server
Namespace Server Name: wombat
System File Directory: wombat:>sys>site>
Namespace Descriptor File: wombat:>sys>site>DOWNUNDER-namespace.text
Default Login: Lisp-Machine
Host for Bug Reports: wombat
Local Timezone: EDT
Standalone Site: Yes No
<ABORT> aborts, <END> uses these values

```

Figure 62. An expanded Define Site menu

#### Namespace Server Name

The name of the local host in the new site.

#### System File Directory

The physical directory that will correspond to the logical directory SYS:SITE; in the new site. For more information, see the section "Logical Pathnames and the SYS Host".

#### Namespace Descriptor File

The name of the file that will hold the namespace description that is used to find all the files that make up the namespace database.

#### Default Login

The "Lisp name" of the user object "Lisp-Machine".

#### Host for Bug Reports

The name of the host that will accept bug-report mail. For information about the mailer, see the section "Installing and Configuring the Mailer".

#### Local Timezone

The abbreviation of the local timezone or an integer indicating the hour offset from GMT. For more information about time zones, see the section "Specifying a Time Zone for Your Site".

#### Standalone Site

Whether the local site consists of just the local machine or not. A broadcast is used to verify the identity of the local host and get the current time when booting at non-standalone sites. Saying "Yes" means that these broadcasts will not be performed.

Press END to define your site.

Now you are ready to add your machines to your site.

## **Registering a New Symbolics XL, MacIvory Machine, or 3600**

### **Choosing a Name and Network Address for the New Machine**

Select a name and network address(es) for the new machine. Symbolics UX machines must have both a Chaosnet address and an Internet address. For information about network addresses, see the section "Choosing Machine Names and Network Addresses".

**Note:** All preconfigured machines come from Symbolics with Chaosnet address 52525. *Unless you have a standalone machine, this is not a desirable Chaosnet address.*

### **Creating a New Host Object for a 3600, XL, or MacIvory Machine**

1. Use the Namespace Editor on the namespace server to set up the namespace. For information about using the Namespace Editor, see the section "Using the Namespace Editor".
2. Create a host object for each new 3600-family, XL, and MacIvory machine at the site. The system type for all machines is "LispM". The machine type should be one of "3600", "XL400", "XL1200", or "MacIvory". Supply the appropriate Chaosnet address for each machine.
3. Create namespace objects that represent users, other computers and printers at the site (whether Symbolics or non-Symbolics machines).

### **Making a Genera Site-Configured World**

**Note:** You must build a site-configured world for each machine type (3600-Family or Ivory) at your site, with the exception of standalone sites that consist of only one MacIvory or one XL machine. We do not recommend saving a site-configured world in which the Define Site command has been issued. When building a site-configured world, use the Set Site command on a freshly booted machine.

Perform these steps to make a site-configured world:

1. Edit (update) the SYS:SITE;SYS.TRANSLATIONS file. For instructions on how to do this, see the section "Updating the Sys.Translations File for Genera 8.3".

Once you have updated the translations file, return to this section and finish making the Genera 8.3 site-configured world.

2. Cold boot the Genera 8.3 distribution world on your machine.

3. Use the Set Site command to configure the machine for your site. For information about the Set Site dialogue, see the section "Set Site Dialogue".
4. Save the new world, using either the Save World Complete command or the Save World Incremental command, depending on your circumstances. Information is available to help you make this decision. See the section "Save World Command".

This example shows how to save an incremental version of the world for Genera 8.3 on Ivory-based machines:

```
Save World Incremental FEP0:>Inc-Site-from-Genera-8-3.load
```

This site-configured world will serve as the base world for all Ivory worlds you build at your site, except for the namespace server's world. Similarly, a site-configured 3600 world will serve for all 3600-Family worlds you build at your site (except for namespace server worlds). Information is available about making user and server worlds for your site. See the section "Making Customized Genera User and Server Worlds".

### Updating the Sys.Translations File for Genera 8.3

Before making a site-configured world, edit the file SYS:SITE:SYS.TRANSLATIONS so that Genera 8.3 will "know" where its sources reside. Once you have finished editing the SYS.TRANSLATIONS file for your site, save and load the file.

Symbolics suggests that you use a Symbolics NXP1000, XL, MacIvory, or 3600 Family machine to hold the distribution sources. Alternatively, you can use a non-Symbolics machine to store sources. For further information, see the section "Storing Genera Sources on a Non-Symbolics Machine".

For sites that use NXP, MacIvory, XL, or 3600-series machines, it is preferable to store sources on NXP, XL, or 3600-series machines, rather than MacIvories.

If the machines at your site will be running only Genera 8.3, use this example as a model for your SYS.TRANSLATIONS file:

```
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: USER -*-
(fs:set-logical-pathname-host "SYS" :translations
 '("sys:**;*.*.*" "ACME-SMBX:>rel-8-3>sys>**>*.*.*))
```

If the machines at your site will be running multiple versions of Genera, see the section "A Sample SYS.TRANSLATIONS File That Stores Genera Sources for Multiple Releases".

### A Sample SYS.TRANSLATIONS File That Stores Genera Sources for Multiple Releases

This example shows a SYS.TRANSLATIONS file that has been conditionalized to allow Genera 8.1.1, and Genera 8.2 NXP to coexist with Genera 8.3. In this example, ACME-SMBX is the name of the Symbolics computer on whose Lisp Machine File System (LMFS) the sources will be stored.

```
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: USER -*-
(multiple-value-bind (major-version minor-version)
  (sct:get-release-version)
  (select major-version
    ;; Translations for Genera 8.1.1, 8.2 NXP, and 8.3
    (8 (selector minor-version equalp
      ("1.1")
      (fs:set-logical-pathname-host "SYS" :translations
        '(("sys:**;*. *.*" "ACME-SMBX:>rel-8-1>sys>**>*. *.*"))))
      ("2 NXP" "2.1 NXP")
      (fs:set-logical-pathname-host "SYS" :translations
        '(("sys:**;*. *.*" "ACME-SMBX:>rel-8-2>sys>**>*. *.*"))))
      (3)
      (fs:set-logical-pathname-host "SYS" :translations
        '(("sys:**;*. *.*" "ACME-SMBX:>rel-8-3>sys>**>*. *.*"))))
      (otherwise (error "~A ~A unknown release or system version.~2T
        Edit SYS:SITE;SYS.TRANSLATIONS to add this system."
          major-version minor-version))))
    (otherwise (error "~A ~A unknown release or system version.~2T
      Edit SYS:SITE;SYS.TRANSLATIONS to add this system."
        major-version minor-version))))))
```

### Managing System Source Files When Running Multiple Releases

If you are planning to run more than one release at the same time, we recommend keeping sources for all of them. Keeping all sources available may not be possible because of disk space limitations on your SYS host (or hosts), but the lack of certain source files may cause problems. This means that if you wish to delete some of the Genera 8.3 sources, you may have to employ a workaround.

The workaround is to change the SYS.TRANSLATIONS file so that Genera 8.1 and 8.2 can find files in the Genera 8.3 directory. *This workaround is limited to Genera fonts, and FEP flod file directories. Do not use the workaround on documentation, X-fonts, Lisp sources, or binaries.*

For example, if you have to delete the Genera 8.3 font files from your SYS host, modify the translations file as shown in bold (this translation allows users of Genera 8.1 and 8.2 systems to find screen fonts in the >rel-8-3> directory.

```

;;; -*- Mode: LISP; Syntax: Common-lisp; Package: USER -*-
(multiple-value-bind (major-version minor-version)
  (sct:get-release-version)
  (select major-version
    ;; Translations for Genera 8.1.1, 8.2 NXP, and 8.3
    (8 (selector minor-version equalp
      ("1.1")
      (fs:set-logical-pathname-host "SYS" :translations
        '(("sys:fonts;**;*.*.*"
          "ACME-SMBX:>rel-8-3>sys>fonts>**>*.*.*"))
        ("sys:**;*.*.*" "ACME-SMBX:>rel-8-1>sys>**>*.*.*")))))
      ("2 NXP" "2.1 NXP")
      (fs:set-logical-pathname-host "SYS" :translations
        '(("sys:fonts;**;*.*.*"
          "ACME-SMBX:>rel-8-3>sys>fonts>**>*.*.*"))
        ("sys:**;*.*.*" "ACME-SMBX:>rel-8-2>sys>**>*.*.*")))))
      (3)
      (fs:set-logical-pathname-host "SYS" :translations
        '(("sys:**;*.*.*" "ACME-SMBX:>rel-8-3>sys>**>*.*.*"))))
      (otherwise (error "~A ~A unknown release or system version.~2T
        Edit SYS:SITE;SYS.TRANSLATIONS to add this system."
          major-version minor-version))))
    (otherwise (error "~A ~A unknown release or system version.~2T
      Edit SYS:SITE;SYS.TRANSLATIONS to add this system."
        major-version minor-version))))))

```

If you are using a non-Symbolics machine to store your system documentation, sources, examples, fonts, and non-loaded systems, you would make such a modification in the appropriate file system syntax.

Once you have made a site-configured world, you are ready to restore the Genera 8.3 documentation, sources, examples, fonts, and non-loaded systems.

### Restoring the Genera Sources

This section describes how to restore the Genera documentation, sources, examples, fonts, and non-loaded systems. This step is not necessary for sites that use the CD-ROM as a SYS host. See the section "Using a CD-ROM as a SYS Host".

Use the CD-ROM labeled *Genera 8.3 Distribution Worlds, Sources, and Bundled Products*.

Restoring the Genera sources in their entirety requires approximately 41,250 LMFS records (equivalent to approximately 165,000 FEP blocks or 204,000 KBytes for UNIX). If you have limited disk space (for example, if you have only one machine at your site), you may want to restore the sources selectively, to conserve disk space.

You can use a non-Symbolics machine to store the Genera documentation, sources, examples, fonts, and non-loaded systems. For information on how to do this, see the section "Storing Genera Sources on a Non-Symbolics Machine".

1. Boot the a site-configured Genera world and put the Genera 8.3 CD-ROM into the CD-ROM drive.
2. Use the following command to restore the files from the distribution image on the CD-ROM:

```
Restore Distribution :Menu yes :Use Disk yes
```

The system responds by displaying a menu, as shown in Figure 47.

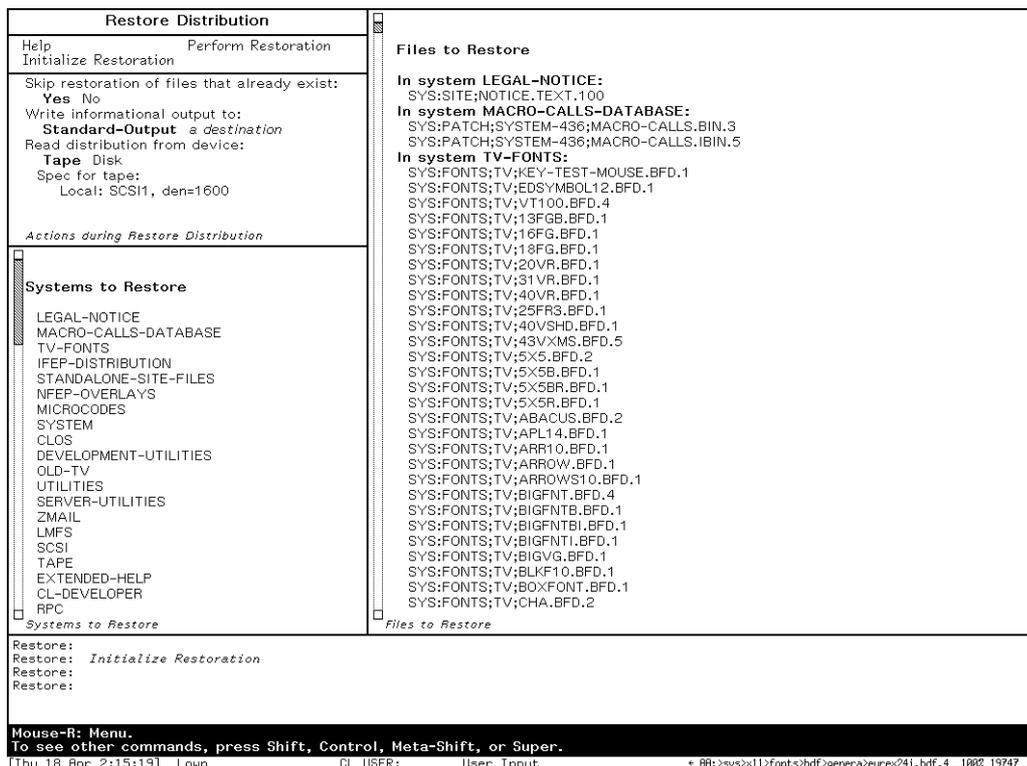


Figure 63. System Selection Menu for the Restore Distribution Command

In the restore distribution frame, specify a pathname for a distribution image on the CD-ROM from the following list:

```
8_3_SOURCE_DIST_1.TAPE ; Genera sources and documentation
8_3_SOURCE_DIST_2.TAPE ; Genera non-loaded systems
8_3_FPC_DIST.TAPE ; Symbolics Fortran, Pascal, and C
```

For example, on a Lisp host:

```
LISPM-HOST|CDROMn:>DISTRIBUTION>8_3_SOURCE_DIST_1.TAPE
```

and on a Sun host:

```
sun-host:/cdrom/distribution/8_3_source_dist_1.tape
```

3. For each distribution image, click on [Initialize Restoration], and use the mouse to deselect systems that you do not want to restore from the menu. For more information on using the Restore Distribution frame, see the section "Restore Distribution Frame".

Once you have deselected those systems that you don't want to restore, click on [Perform Restoration]. This restores the selected systems from the distribution image. Note that restoring the FPC distribution image is optional.

All sites must restore the following systems:

LEGAL-NOTICE  
MACRO-CALLS-DATABASE  
TV-FONTS

All sites that include an Ivory-based machine (NXP, MacIvory, XL, or Symbolics UX) must restore the following systems:

IFEP-DISTRIBUTION (Ivory FEP Kernel and IFEP Overlays)

All Standalone XL and MacIvory sites must restore the following systems:

STANDALONE-SITE-FILES

All sites that include a 3600 Family machine must restore the following systems:

NFEP-OVERLAYS  
MICROCODES

All sites who want up-to-date online Genera sources should restore the following systems: (Note that most of the non-loaded systems also include sources.)

SYSTEM  
CLOS  
DEVELOPMENT-UTILITIES  
OLD-TV  
UTILITIES  
SERVER-UTILITIES  
ZMAIL  
LMFS  
LMFS-DEFSTORAGE  
SCSI  
TAPE  
EXTENDED-HELP  
CL-DEVELOPER  
RPC  
EMBEDDING-SUPPORT  
MACIVORY-SUPPORT

UX-SUPPORT  
RPC-DEVELOPMENT  
MACIVORY-DEVELOPMENT  
UX-DEVELOPMENT  
ZWEI  
SERIAL  
HARDCOPY  
NSAGE  
IP-TCP  
X-REMOTE-SCREEN  
X-CLIENT-FONTS

Sites with MacIvory machines who want up-to-date online MacIvory C sources should restore the following systems:

MACIVORY-MACINTOSH-SOFTWARE (C sources for the Macintosh side of Symbolics MacIvory software)

All sites who want up-to-date online Genera documentation should restore the following systems:

DOC (Genera Documentation Database)  
IP-TCP-DOC  
NFS-DOCUMENTATION  
X-DOCUMENTATION

All sites might need to restore the following non-loaded systems:

PRINT (Print Spooler)  
LGP2-SUPPORT (LGP2/3 Printer support)  
DMP1-SUPPORT  
MAILER  
DOMAIN-NAME-SERVER

All sites may choose to restore the following non-loaded systems:

CLIM (Common Lisp Interface Manager)  
GENERA-CLIM  
CLX-CLIM  
POSTSCRIPT-CLIM  
CLIM-DEMO  
CLIM-COMPATIBILITY  
CLIM-DOC  
CLIM-TEST-SUITE  
CLIM-TUTORIALS  
METERING  
CONVERSION-TOOLS (Zetalisp to Common Lisp; Flavors to CLOS)  
HACKS (demos)

EXAMPLES

NFS-SERVER (includes NFS-CLIENT)

X-SERVER

X-SERVER-FONTS

STATIC-RUNTIME (CLOS-based Static-Runtime)

FLAVORS-STATIC-RUNTIME (Flavors-based Static-Runtime)

DBFS-UTILITIES (Static backup utilities)

MAC-DEX (a sample MacIvory application)

HYPERCARD-MACIVORY (MacIvory interface to HyperCard)

HELPFUL-TOOLS (utilities for upgrading namespace and dialnet to Genera 8.0)

8-1-NETBOOT-COMPATIBILITY-PATCH (allows 8.1 netboot servers to serve 8.3 clients)

UNSUPPORTED (systems that are no longer maintained)

Sites interested in experimenting with RPC can choose to restore the following systems:

MACINTOSH-EXAMPLES (MacIvory example files)

ADDITIONAL-RPC-EXAMPLES (RPC example files)

UX-RPC-AND-FRIENDS (UX RPC code)

Sites with 3600 Family machines can choose to restore the .bin files for the following systems, which can be loaded into 3600 Family Genera worlds:

DEPENDENT-MODULES

X-REMOTE-SCREEN

X-DOCUMENTATION

MACIVORY-SUPPORT

UX-SUPPORT

RPC-DEVELOPMENT

MACIVORY-DEVELOPMENT

UX-DEVELOPMENT

Sites with MacIvory Delivery machines can choose to restore the .ibin files for the following systems, which can be loaded into MacIvory Delivery worlds: (Note that for the Dex-Doc system, .sab files as well as .ibin files must be restored.)

DEPENDENT-MODULES

IP-TCP

IP-TCP-DOC

NFS-CLIENT

NFS-DOCUMENTATION

X-REMOTE-SCREEN

X-DOCUMENTATION

ZWEI

SERIAL

HARDCOPY  
NSAGE  
DELIVERY-DOCEX (Delivery Document Examiner)  
DEX-DOC (documentation for Delivery Docex)  
UX-SUPPORT  
RPC-DEVELOPMENT  
MACIVORY-DEVELOPMENT  
UX-DEVELOPMENT

Sites with UX Delivery machines can choose to restore the .ibin files for the following systems, which can be loaded into UX Delivery worlds:

ZWEI  
SERIAL  
HARDCOPY  
MACIVORY-SUPPORT  
RPC-DEVELOPMENT  
MACIVORY-DEVELOPMENT  
UX-DEVELOPMENT

Sites with 3600 Family machines can choose to restore the following system:

MICROCODE-SYMBOL-TABLES (for debugging purposes)

### **Copying New Flod Files**

New flod files are available for Ivory machines upgrading from previous releases of Genera to Genera 8.3.

This step should be skipped when installing a new UX machine, because the installation program performs it automatically.

Copy the newest FEP software to the FEP file system on each of your Symbolics machines. For Ivory-based machines this includes FEP kernel and flod files.

Give the following command at a Lisp Listener:

```
Copy Flod Files :version I328
```

Answer "Yes" to the query about updating the hello.boot file. By doing this, your hello.boot file will automatically be updated to scan the new flod files.

For information about the Copy Flod Files command, see the section "Copy Flod Files Command".

To use the new flods, you must cold boot the machine's FEP. Follow the instructions for your machine type:

After you have restored the documentation, sources, examples, fonts, and non-loaded systems, you are ready to make customized Genera 8.3 user and server worlds. (For further information on this topic, see the section "Customizing and Saving Worlds".)

## Making Customized General User and Server Worlds

Follow these instructions to make customized worlds containing the software you will be using at your site. For information about netbooting, See the section "Netbooting". For more information about Loadable Systems, see the section "Loadable Systems for All Machine Types".

## Making Worlds for User Machines

We recommend the use of Incremental Disk Save (IDS) for customized worlds. (This makes it easier to distribute newer worlds to machines with small disk configurations.)

1. Boot the site-configured world with IDS enabled. (On Ivory-based machines, check to see that your `hello.boot` file contains a Set Boot Options command line with the keyword and value `:IDS Enable`. On 3600-series machines, check that your `boot.boot` file contains an Enable IDS command line). If you have multiple machines at your site, *do not* use your namespace server machine to boot this world.

2. Disable Services, using the Disable Services command, as shown:

```
Disable Services
```

3. Log in as the default system user. Type to the command prompt:

```
Login Lisp-Machine
```

This command line logs you in as the default user and is useful during installation procedures, when you must perform certain operations after cold booting.

4. Load any software you want to include in the world such as non-loaded systems or layered products. For example, to load the non-loaded system called Metering, type:

```
Load System Metering
```

For more information on loadable systems, see the section "Loadable Systems for All Machine Types".

5. Optionally, use the Optimize World command:

```
Optimize World
```

The Optimize World command reorganizes the world to improve paging performance. It increases the size of an IDS world, however. More information is available about the Optimize World command. See the section "Optimize World Command".

6. Save the new world, using either the Save World Incremental command or the Save World Complete command. This world will serve as the base world

for all of the non-server machines at your site. For example, to save an incremental Metering world using Genera 8.1 software, and based on the site-configured world:

```
Save World Incremental FEP0:>Inc-Metering-8-1-from-Inc-Site.ilod
```

Information is available about which of the two Save World commands to use. See the section "Save World Command".

7. Once you have created these worlds, you can move them out to the appropriate machines, using the Copy World command.

**Note:** The `.ilod` file extension indicates world-load files for Ivory-based machines, just as the `.load` file extension indicates world-load files for Symbolics 3600-family machines. Files with the `.ilod` extension can be copied only between Ivory-based machines. Files with the `.load` extension can be copied only between Symbolics 3600-series machines.

## Loadable Systems for All Machine Types

### CLIM

CLIM stands for Common Lisp Interface Manager. It is a portable, powerful, high-level user interface management system toolkit intended for Common Lisp software developers.

For more information about CLIM, see the section "Common Lisp Interface Manager (CLIM): Release 1.0" and see the document *Genera 8.3 Release Notes*.

If you want online access to the CLIM tutorials, make sure that you have restored the CLIM-TUTORIAL files from the Genera source tape image.

No documentation available for section Installing CLIM 2.0.

The CLIM Lisp Listener is bound to `SELECT-λ` (`SELECT-symbol-shift-L`).

### CLIM-Demo

The CLIM Demos provide examples of small and medium-sized applications using CLIM.

For more information about the CLIM Demos, see the section "Using the CLOE CLIM Demos".

Before loading the CLIM Demo system, CLIM must be loaded.

To load the CLIM Demos, type

```
Load System CLIM-Demo
```

To run the CLIM Demos in Genera, type

```
(clim-demo::start-demo)
```

## **CLIM-Doc**

CLIM Doc contains the online documentation for Symbolics CLIM.

CLIM Doc depends on packages defined in the CLIM system.

To load CLIM Doc, type

```
Load System CLIM-Doc
```

## **Print**

The Print system defines the Genera Print Spooler. The Print Spooler manages multiple print requests sent to a printer, from any number of users. You must load the Print system on Symbolics hosts designated as print spoolers.

For more information about Printers and Print Spoolers, see the sections "Installing a Printer" and "Defining the Print Spooler" in *Site Operations*.

If you are using Symbolics LGP2 or LGP3 printers, make sure that you have restored the LGP2-SUPPORT files from the Genera source tapes. If you are using Symbolics DMP1 printers, make sure that you have restored the DMP1-SUPPORT files from the Genera source tapes. Note that the LGP2-SUPPORT and DMP1-SUPPORT systems are not loadable systems, they just contain fonts and metric files which the Print software expects to find on your disk.

To load the Print system, type

```
Load System Print
```

The Print Spooler activity is bound to SELECT-S.

## **Mailer**

The Mailer system defines the Genera Store-and-Forward Mailer. The Mailer is a program that provides mail forwarding and delivery services to users at a site. The Mailer is distinct from Zmail, which is a user program for reading and composing mail. You must load the Mailer system on Symbolics hosts designated as mail servers.

For more information about the Mailer, see the sections "Symbolics Store-and-Forward Mailer" and "Installing and Configuring the Mailer" in Document Examiner (see the section "Reading Online Documentation").

The mailer does not depend on any other loadable systems. However, if you wish to send and receive mail from other sites on the ARPA Internet, you may need to set up a Name server at your site. See the section "Loadable Systems on the CD-ROM - Domain Name Server".

To load the Mailer system, type

```
Load System Mailer
```

The Mailer Activity is bound to SELECT-O.

## Domain Name Server

The Domain Name Server software enables a Symbolics computer to act as a Name server host. A Name server provides other hosts on the network with information about the hosts in your domain. Hosts inside and (optionally) outside your site consult your Name server about objects in your domain. You must load the Domain Name Server system on Symbolics hosts designated as Name servers. Note that it is not necessary to load the Domain Name Server system on hosts which serve only as central name resolvers for your site.

For more information about Domain Names, Name servers and central name resolvers, see the section "Internet Domain Names" and see the section "Installing the Internet Domain Names System".

The Domain Name Server system does not depend on any other loadable systems. However, a Symbolics host that is designated as a Name server for the ARPA Internet must run IP/TCP, because it must be capable of communicating with other hosts on the Internet using IP/TCP. See the "IP-TCP" description below.

To load the Domain Name Server system, type

```
Load System Domain-Name-Server
```

The Domain Name Server activity is on `Select-@`.

## Metering

The Metering system defines a uniform interface to the various Genera metering tools. This interface makes it convenient to meter different aspects of program performance, such as time, paging, and consing.

For more information about Metering, see the section "Metering a Program's Performance".

The Metering system does not depend on any other loadable systems.

To load the Metering system, type

```
Load System Metering
```

The Metering Interface activity is bound to `SELECT-2`.

## Conversion-Tools

The Conversion Tools system defines special-purpose Zmacs commands which partially automate the code conversion process. Tools are included for the following types of conversions: CLIM 1.0 to CLIM 2.0, DW to CLIM, Flavors to CLOS, Zetalisp to Common Lisp, Symbolics Common Lisp to portable Common Lisp, and Package Conversion.

For more information about Conversion Tools, see the section "Conversion Tools". For more information about converting from Dynamic Windows to CLIM, see the section "Converting an Application From DW to CLIM".

The Conversion Tools system does not depend on any other loadable systems.

To load the Conversion Tools system, type

Load System Conversion-Tools

## **Hacks**

The Hacks system contains Genera Demos.

For more information about the Genera Demonstration facility, see the section "The Demonstrations Facility".

The Hacks system does not depend on any other loadable systems.

To load the Hacks system, type

Load System Hacks

## **IP-TCP**

The IP/TCP software package enables a Symbolics computer to communicate with other systems supporting the ARPA Internet protocol family. Load IP/TCP on machines which will use IP/TCP protocols for remote login, file transfer, and/or electronic mail and messages.

For more information about IP-TCP, see the document *Symbolics IP/TCP Software Package*.

The IP-TCP system does not depend on any other loadable systems. Note that all Genera 8.3 worlds, except the Genera 8.3 MacIvory delivery world, contain IP-TCP.

To load the IP-TCP system, type

Load System IP-TCP

## **IP-TCP-Doc**

IP-TCP Doc contains the online documentation for Symbolics IP-TCP.

IP-TCP Doc depends on packages defined in the IP-TCP system. Note that all Genera 8.3 worlds, except Genera 8.3 MacIvory Delivery worlds, contain IP-TCP Doc.

To load IP-TCP Doc, type

Load System IP-TCP-Doc

## **NFS-Server and NFS-Client**

Symbolics NFS is a fully symmetrical implementation of the Sun Network File System protocol, as specified in the Sun Network File System Protocol Specification, Revision B of 17 February 1986. Symbolics NFS allows your Symbolics machine to act as an NFS client or server or both. Load the NFS Server system on Symbolics machines which will act as NFS servers (NFS-Client is a component sys-

tem of NFS-Server, and will be loaded automatically when you load NFS-Server). Load the NFS Client system on Symbolics machines which will act as NFS clients.

For more information about Symbolics NFS, see the document *Symbolics Network File System (NFS) User's Guide*.

The NFS Server and NFS Client systems depend on RPC and IP-TCP.

Note that all Genera 8.3 worlds, except the Genera 8.3 MacIvory Delivery world, contain RPC, IP-TCP and NFS Client.

To load the NFS-Server system, type

```
Load System NFS-Server
```

### **NFS-Documentation**

NFS Documentation contains the online documentation for Symbolics NFS.

NFS Documentation depends on packages defined in the NFS system. Note that all Genera 8.3 worlds, except the Genera 8.3 MacIvory Delivery world, contain NFS Documentation.

To load NFS Documentation, type

```
Load System NFS-Documentation
```

### **X-Server**

The Symbolics X Server program handles X protocol output and input requests from other systems on the network and performs the requested operations on the local system's screen. It allows you to operate X client applications (such as "xterm") running on other systems from your Symbolics Computer's console. The Symbolics X Server only works on non-embedded machines. On embedded machines (MacIvory or UX), run an X Server on the embed host, such as MacX for Macintosh or OpenWindows for Sun.

For More information about the Symbolics X Server, see the document *Symbolics X Window System User's Guide*.

The X-Server system does not depend on any other loadable systems. The C Runtime system, which includes software to support the C language implementations of the X Server, is a component system of X-Server. As long as you have restored C-Runtime from the Genera Source tape image, C-Runtime will automatically be loaded when you load the X-Server system. The standard set of X fonts are included in the X Server distribution, in the Server Native Format used by the X Server. To use these, make sure that you have restored the X-SERVER-FONTS files from the Genera Source tape image.

To load the X-Server system, type

```
Load System X-Server
```

The X11 Server activity is bound to SELECT-SQUARE.

### **X-Remote-Screen**

X Remote Screen is the client side of Symbolics X Windows. In X Windows, the client is the application seeking to present a user interface, and the server is a virtual console capable of doing so. The Genera X client supports multiple user interfaces simultaneously, possibly displayed on different consoles.

For More information about Symbolics X Remote Screen, see the document *Symbolics X Window System User's Guide*.

The X-Remote-Screen system depends on RPC, Embedding-Support, and IP-TCP.

If you wish the Genera X Client to use Genera X fonts instead of the standard 75dpi X fonts, make sure that you have restored the X-CLIENT-FONTS files from the Genera Source tapes, and that you have done any necessary server-dependent work to get the Genera X Client to use the Genera fonts. Note all Genera 8.3 worlds, except the Genera 8.3 MacIvory Delivery world and the Genera 8.3 3600-family distribution world, contain X-Remote-Screen.

To load the X-Remote-Screen system, type

```
Load System X-Remote-Screen
```

### **X-Documentation**

X-Documentation contains the online documentation for Symbolics X Windows (both the Symbolics X Server and X Remote Screen).

X-Documentation does not depend on any other loadable systems. Note all Genera 8.3 worlds, except the Genera 8.3 MacIvory Delivery world and the Genera 8.3 3600-family distribution world, contain X-Documentation.

To load X Documentation, type

```
Load System X-Documentation
```

### **Stalice-Runtime**

Stalice Runtime is a subset of Stalice, an object-oriented database system for the Genera programming environment. Stalice-Runtime enables Genera users to use, access, and maintain a Stalice database. Stalice Runtime does not include tools for developing Stalice applications -- those tools are available separately, in the Stalice Developer layered product.

As of release Genera 8.3, Stalice Runtime is a CLOS-based implementation. For compatability purposes, Flavors-based Runtime is also included. For more information about CLOS Stalice and Flavors Stalice, see the section "Changes to Stalice and CLOS".

For more information about Stalice Runtime, see the section "Stalice Runtime".

The Stalice-Runtime system does not depend on any other loadable systems.

To load the Stalice Runtime CLOS-based system, type

Load System Statice-Runtime

To load the Statice Runtime Flavors-based system, type

Load System Flavors-Statice-Runtime

### DBFS-Utilities

The DBFS-Utilities system defines utilities for making backup copies of databases stored in a Statice File System. If a disk containing a Statice File System is damaged or destroyed, a backup copy of the database can be restored onto a fresh disk.

For more information about Statice backup utilities, see the section "Statice File System Operations Program".

DBFS-Utilities depends on Statice-Runtime. Before loading DBFS-Utilities, make sure that the Statice-Runtime system is loaded on your machine.

To load the DBFS-Utilities system, type

Load System DBFS-Utilities

The Statice File System Operations program is bound to `SELECT-δ` (`SELECT-symbol-shift-D`).

### Mac-Dex

Mac-Dex is the Macintosh Document Examiner, a sample MacIvory application. Mac-Dex is provided as an example of how to write an application which presents a Macintosh style interface using Symbolics' standard Macintosh application and **zl-user:define-remote-program-framework**. The Mac-Dex system is the Lisp side of the Mac Dex application, which is our standard Macintosh application configured to run Mac-Dex. Loading Mac-Dex on a 3600-family machine (or other non-MacIvory Symbolics machines) would allow you to use Mac-Dex from a non-NuBus Macintosh by talking to your 3600.

For more information about using Symbolics' standard Macintosh application, see the section "Setting the Size of Application Icons That Use Ivory".

The Mac-Dex system depends on RPC, Embedding-Support, and MacIvory-Support. On 3600-family machines, load the RPC, Embedding-Support, and MacIvory-Support systems before loading Mac-Dex. Note that all Ivory distribution worlds already contain RPC, Embedding-Support, and MacIvory-Support.

To load the Mac-Dex system, type

Load System Mac-Dex

To run Mac Dex, do the following:

1. Load the system Mac-Dex into Genera.
2. Use [File / Quit] to get out of Genera.

3. Double-click on the Mac Dex icon, located in the MacIvory Applications folder.

You do not need to run the Genera icon when you run the Mac Dex application. The Mac Dex application will boot your Ivory if it is not booted. To access Genera documentation via Mac Dex, use the menu titled **Lookup** in the Apple menu bar.

### **HyperCard-MacIvory**

The HyperCard-MacIvory system contains the MacIvory interface to HyperCard. It also includes a collection of demos that illustrates the close communication and integration possible between Ivory and a host HyperCard application.

For more information about the HyperCard-MacIvory system, see the section "MacIvory Interface to HyperCard".

The HyperCard-MacIvory system depends on Mac-Dex. Note that the HyperCard interface always uses an embedded channel, so loading HyperCard-MacIvory is only useful on a MacIvory.

To load the HyperCard-MacIvory system, type

```
Load System HyperCard-MacIvory
```

### **Making Worlds for Server Machines**

#### **Server Machines:**

If you have server machines that run additional software (such as Print, Domain Name Server, or Mailer), follow the procedure used in "Making Worlds for User Machines" and build an IDS world, with the appropriate server systems loaded, for use on these machines.

Make non-namespace server worlds on non-namespace server machines. This will enable non-namespace servers to boot more quickly, because their worlds will be smaller.

#### **Namespace Server Machines:**

Since namespace server machines load all the current namespace information into virtual memory, worlds saved on them are larger and boot more slowly than those saved on non-namespace server machines. Follow the procedure used in "Making Worlds for User Machines" and build an IDS world, with the appropriate server systems loaded, on a namespace server machine.

### **Backing Up Worlds to Tape**

Back up the world(s) you just created (and the appropriate microcode) to tape. For complete information, see the section "FEP-Tape System" in *Site Operations*. Here is the procedure in brief:

1. Type the command in the Lisp Listener:  
     Select Activity FEP-Tape
2. Use the command Add File to add a single file to the list of files to be written to tape. For example, type to the FEP-Tape command prompt:  
     Add File *the pathname of the base site world*  
     Add File *other world loads and microcodes you want to back up*

Or, you can click on [Add File]. If you back up both the world load and the microcode files, these files can both be restored if needed.

3. Use the command Write Tape to write the file to tape; type this to the FEP-Tape command prompt:  
     Write Tape

We recommend that you type the Write Tape command, rather than clicking on the [Write Tape] menu item, because typing the command gives you more control over the tape specification.

### Special Installation Situations

During site configuration and your software installation, you may encounter special situations. Such situations include:

- Adding non-Symbolics computers to your site.
- Installing software from a CD-ROM drive that is on a non-Symbolics machine (a Sun workstation).
- Storing documentation, sources, examples, fonts, and non-loaded systems on a non-Symbolics computer.

This chapter contains information about these special situations.

### Adding Non-Symbolics Computers to Your Site

Once the non-Symbolics hardware and software have been correctly installed (and verified) at your site, use the Namespace Editor (specifically, the Edit Namespace Object command) to add non-Symbolics machines to your namespace database. More information is available about the Namespace Editor.

See the section "Using the Namespace Editor".

- If another host at your site is already running the same operating system as the new host, use the [Copy Object] menu item in the Namespace Editor window. Be sure to change the network address(es) and name(s), though.

- If no other host at your site is already running the same operating system as the new host, use the [Create Object] menu item in the Namespace Editor window. For information about the attributes of a new host, see the section "Attributes for Objects of Class "Host"".

### Using a CD-ROM as a SYS Host

In order to use a CD-ROM as a SYS host, the CD-ROM drive must be connected to one of:

- an NXP1000
- an XL Family machine running Genera 8.1.1 or later
- a MacIvory running Genera 8.1.1 or later
- a Sun-4

**Note:** Genera 8.1 is not sufficient for a Symbolics machine to serve as a CD-ROM SYS host.

The procedure for setting up your site to use a CD-ROM SYS host is as follows:

1. Copy everything from the >SYS>SITE> directory on the CD-ROM to the SYS:SITE; directory for your site. The SYS:SITE; directory for a site is specified in the Site namespace object.

```
Copy File HOST|CDROMn:>SYS>SITE>*.;* * SYS:SITE;*. *.*
```

**Note:** If your CD-ROM drive is on a Sun, this step must be executed in a Genera 8.3 world as follows:

```
Copy File sun-host:/cdrom/sys.sct/site/* sys:site;*. *.*
```

2. Edit your SYS:SITE;SYS.TRANSLATIONS file to translate logical pathnames to physical pathnames of files on the CD-ROM. After making the necessary changes, save and load the new sys.translations file, or build it into your site-configured Genera 8.3 worlds. Once the updated translations are loaded, you should be able to access files on CD-ROM as easily as you would access files on a LMFS.

In the following examples, *HOST* is the name of the NXP, XL, or MacIvory CD-ROM host, and *n* in CDROM*n* is the SCSI address of the CD-ROM drive.

If all machines at your site are running Genera 8.3, and if your CD-ROM drive is connected to an NXP, XL, or MacIvory, use the following example translations as a guide:

```
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: USER -*-
(fs:set-logical-pathname-host "SYS" :translations
 '(("sys:**;*. *.*" "HOST|CDROMn:>SYS>**>*. *.*;*")))
```

If your CD-ROM drive is connected to a Sun, use the following example translations as a guide:

```
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: USER -*-
(fs:set-logical-pathname-host "SYS" :translations
 '(("sys:**;*. *.*" "sun-host:/cdrom/sys.sct/**/*")))
```

If some machines at your site are running a release older than Genera 8.3, use the following example translations as a guide:

```
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: USER -*-
(multiple-value-bind (major-version minor-version)
  (sct:get-release-version)
  (select major-version
    ;; Translations for Genera 8.1, 8.1.1
    (8 (selector minor-version equalp
      ("1.1")
      (fs:set-logical-pathname-host "SYS" :translations
        '(("sys:**;*. *.*" "ACME-SMBX:>rel-8-1>sys>***>*. *.*")))))
      ("2 NXP")
      (fs:set-logical-pathname-host "SYS" :translations
        '(("sys:**;*. *.*" "HOST|CDROMn:>sys>***>*. *.*;*"))))
      (3)
      (fs:set-logical-pathname-host "SYS" :translations
        '(("sys:**;*. *.*" "HOST|CDROMn:>sys>***>*. *.*;*"))))
      (otherwise (error "~A ~A unknown release or system version.~2T
        Edit SYS:SITE;SYS.TRANSLATIONS to add this system."
          major-version minor-version))))
    (otherwise (error "~A ~A unknown release or system version.~2T
      Edit SYS:SITE;SYS.TRANSLATIONS to add this system."
        major-version minor-version))))))
```

If you wish to use the CD-ROM as the SYS host for only specific parts of the Genera 8.3 documentation, sources, examples, fonts, and non-loaded systems, use the following example translations as a guide (This example assumes that all Symbolics machines at your site are running Genera 8.3):

```
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: USER -*-
(fs:set-logical-pathname-host "SYS" :translations
 ;; Access documenatation and fonts from the CD
 '(("sys:doc;**;*. *.*" "HOST|CDROMn:>SYS>DOC>***>*. *.*;*")
  ("sys:fonts;**;*. *.*" "HOST|CDROMn:>SYS>FONTS>***>*. *.*;*")
 ;; Access everything else from a LMFS on a file server
  ("sys:**;*. *.*" "SMBX:>rel-8-3>sys>***>*. *.*")))
```

### Known limitations and problems with using a CD-ROM SYS host

1. The ISO9660 filesystem does not understand unusual characters such as +, \$, and & in pathnames. There are a few pathnames in the Genera distribution which contain such characters. If you run into one, simply take the retry option to specify another pathname, and respecify the same pathname, but drop the offending character. For example,

```
SYS:EMBEDDING;MACIVORY;TOOLBOX;LISP+C-SCT-SUPPORT.BIN appears as
SYS:EMBEDDING;MACIVORY;TOOLBOX;LISPC-SCT-SUPPORT.BIN
```

in the CD-ROM filesystem.

2. The ISO9660 filesystem limits the length of file and directory names to 31 characters, including periods and file types. File names longer than this are truncated. There are many truncated pathnames in the Genera distribution, however only three files produced ambiguous pathnames when truncated. On the Genera CD-ROM, we renamed these three files to avoid truncation problems:

```
SYS:EXAMPLES;CONSTRAINT-FRAME-LANGUAGE-1.LISP
SYS:EXAMPLES;CONSTRAINT-FRAME-LANGUAGE-2.LISP
SYS:EXAMPLES;CONSTRAINT-FRAME-LANGUAGE-3.LISP
```

are named

```
SYS:EXAMPLES;CONSTRAINT_FRAME_LANGUAGE_.LISP
SYS:EXAMPLES;CONSTRAINT_FRAME_LANGUAG_2.LISP
SYS:EXAMPLES;CONSTRAINT_FRAME_LANGUAG_3.LISP
```

Note that the ISO9660 translations rules automatically do the correct thing for all other truncated pathnames on the CD-ROM. For example, Genera automatically translates

```
SYS:METERING;INTERFACE;STATISTICAL-CALL-TREE-METERING-RUN.IBIN
```

to the correct CD-ROM pathname:

```
CDROMn:>sys>metering>interfaces>statistical_call_tree_mete.ibin
```

3. When using a CD-ROM as a SYS host, you must reset access paths whenever you change CD disks.

### Storing Genera Sources on a Non-Symbolics Machine

You can use a UNIX machine running NFS as a file server to store your Symbolics sources.

Use this example for your SYS.TRANSLATIONS file:

```
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: USER -*-
(fs:set-logical-pathname-host "SYS" :translations
 '(("sys:**;" "ACME-UNIX:/usr/share/symbolics/rel-8-4/sys.sct/**/"))))
```

If you are running multiple releases (for example, some combination of Genera 8.3 and Genera 8.4), use the following example:

```
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: USER -*-
(multiple-value-bind (major-version minor-version)
  (sct:get-release-version)
  (select major-version
    ;; Translations for Genera 8.1.1, 8.2, and 8.3
    (8 (selector minor-version equalp
      (("3") (fs:set-logical-pathname-host "SYS" :translations
        '(("sys:**;"
          "ACME-UNIX:/usr/share/symbolics/rel-8-3/sys.sct/**/")))))
      ((4)
        (fs:set-logical-pathname-host "SYS" :translations
          '(("sys:**;"
            "ACME-UNIX:/usr/share/symbolics/rel-8-4/sys.sct/**/")))))
      (otherwise (error "~A ~A unknown release or system version.~2T
        Edit SYS:SITE;SYS.TRANSLATIONS to add this system."
          major-version minor-version))))))
  (otherwise (error "~A ~A unknown release or system version.~2T
    Edit SYS:SITE;SYS.TRANSLATIONS to add this system."
      major-version minor-version)))
```

### Using SCT with a UNIX File System

The System Construction Tool (SCT) presents a special case in using UNIX backup files. SCT requires a version number for the files it uses. For directories maintained by SCT, files are maintained so that the newest versions of the files have an explicit version number. (See the section "System Construction Tool" for background information on SCT).

Use the name of a UNIX directory to specify that files contained in it are to be maintained by SCT and should all end with explicit version numbers. If the name of a UNIX directory ends with the .sct file extension, Symbolics NFS creates all new files under that directory with explicit version numbers.

Here's an example. Suppose a sys.translations file looked like this:

```
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: USER -*-

(fs:set-logical-pathname-host "SYS" :translations
 '(("**;" "ACME-UNIX:/usr/share/symbolics/rel-8-0/sys.sct/**/")))
```

In this case, all new files created with Symbolics NFS under /usr/share/symbolics/rel-8-0/sys.sct/ are created with explicit version numbers

in the newest versions of files, meeting SCT requirements. This is true for all sub-directories of a directory whose name ends in .sct.

The double-asterisk (\*\*) wildcard can be used only in UNIX pathnames that are accessed by means of NFS. Other UNIX file protocols, such as FTP, do not recognize this wildcard. For further information on the use of the double-asterisk wildcard, see the section "LMFS Pathnames".

Genera considers UNIX pathnames with any directory components ending in the string ".sct" to be for SCT-maintained files, and hence to contain version numbers. The syntax that Genera uses for backup file version numbers in UNIX pathnames is a number surrounded by tildes separated by a dot from the file type.

This pathname is considered to have a version number:

```
ENIAC:/u0/mwra/visi-brain.sct/foo.lisp.~23~
```

So merging with it as a default behaves like this:

<i>Default</i>	<i>Input</i>	<i>Merged</i>
ENIAC:/u0/dpw.sct/code/foo.lisp.~23~	bar.⌘	ENIAC:/u0/dpw.sct/code/bar.lisp
ENIAC:/u0/dpw.sct/code/foo.lisp.~23~	⌘.bin	ENIAC:/u0/dpw.sct/code/foo.bin

Note that backup file version numbers are not defaulted as part of the merging operation.

You can set the number of backup versions that are retained by Genera with the "NFS-GENERATION-RETENTION-COUNT User Property" for host namespace objects.

## Miscellaneous CD-ROM Information

### Recommended Care and Handling of a 3M CD-ROM

The CD-ROM itself is a piece of plastic worth several dollars. The value of the CD-ROM is in the information which it stores. The CD-ROM is resistant to damage from scratches, fingerprints, and dust. It is not, however, indestructible. To assure the information's integrity, 3M recommends observing the following:

1. Handle the disc by the edges to prevent fingerprints on either surface of the disc. When removing the disc from the jewel box, be sure to press down on the middle hub of the jewel box to release the disc. Do not pull up hard enough on the disc to bend it more than a half (.5) inch before it releases. Place the disc carefully into the CD-ROM drive, making sure not to scratch the disc against the drive or any other obstruction.
2. When the disc is not in use, place it in its jewel box. This prevents scratching of the disc's surfaces. The information is covered by protective layers, but it is possible to scratch through the label and damage the information. Placing an unprotected disc flat on a desk or table is inviting trouble.

3. Dust is attracted to the surface of a CD-ROM. This dust can affect the readability of the disc. Remove dust from the unlabeled side of the CD-ROM before loading it into the drive. Dust carried into a drive can also be deposited on the lens assembly, which can affect readability of subsequent discs on the system.
4. If the disc becomes soiled with fingerprints or spots, wipe it with a soft dry cloth. If the cloth alone does not remove the spot, attempt to fog the disc with your breath and rewipe. For severe spots resisting removal by this method, a small amount of ethyl alcohol can be placed on the cloth to dampen it. Using the dampened cloth, wipe the area. This method should only be used if the disc is unusable in its present state.
5. No abrasive or solvent cleaners should be used on the disc. Do not use conventional vinyl record cleaning solutions because they contain chemicals that can damage a CD-ROM.

### **Ejecting the CD From the Drive**

- On an NXP or an XL, press the eject button on the front of the CD-ROM drive.
- On a MacIvory, use the Eject menu item in the File menu of the Macintosh Finder.
- On a Sun, use the *umount* command to unmount the CD-ROM drive, then press the eject button on the front of the CD-ROM drive. For more information on the UNIX *umount* command, see the SunOS Reference Manual entry on *umount*.

### **Reading Online Documentation**

#### **Introduction to Document Examiner**

*Document Examiner* is a utility for finding and reading documentation. The entire Symbolics document set can be read using Document Examiner.

You can look up documentation in Document Examiner's own window, in a Lisp Listener, or from the editor. No matter where you request to view documentation, Document Examiner manages all online lookup queries.

Using Document Examiner is similar to using the printed documentation. For example, just as books in the document set are available on a shelf or on your desk, Document Examiner, when first selected, displays the names of all the books in the document set. See Figure 48.

You can open to a book's table of contents and find a topic to peruse. You can also open a book to the topic and read through to the end of that topic. Similarly, Doc-

ument Examiner lets you scan a document's table of contents online and select the desired topic, or, if you know what you are looking for, you can "open" the documentation set directly to any topic and read to the end of that topic.

Just as you can browse through the index of a book to find the primary topic (and secondary topics) you want to read about, you can ask Document Examiner to show you all the topics that are indexed under a particular entry.

The big difference between the Document Examiner and a book or set of books is that Document Examiner provides powerful tools for moving from one kind of information to another.

Document Examiner allows quick, direct access to all levels of information:

- General, conceptual discussions of a topic
- Reference material on a topic
- Specific programming language information (for example, a Lisp function)

### How Documentation Is Stored

The online documentation is kept in a *documentation database*, which consists of documentation binary files. When you boot your machine, the world load contains location information about the documentation topics, not the actual files themselves; thus, world size is increased by only a few hundred pages by having the document set available. The documentation database is stored on a file server; when you request to see a documentation topic, the appropriate part of the appropriate file is read in from the server.

Each documentation topic is stored as a *record* in the documentation database. Each record contains information on a particular topic and is uniquely identified by a record name. For example, the section you are currently reading is a record whose name is "How Documentation is Stored".

Records fall into two categories:

- *Object records* documenting Lisp code objects, such as **zl:setf** or **tv:menu**.
- *Concept records* documenting abstract ideas that are not tied directly to code, such as "Introduction to Document Examiner" or "Show Documentation Command".

Records have types. Examples of object record types are Lisp function, flavor, and variable. Concept records are called sections. The documentation lookup commands use the type indicator to distinguish between like-named records. For example, if you request to see the documentation for **error**, Document Examiner lets you know that documentation exists for **error** as both a flavor and as a function.

Records have *topics*. The Show Documentation command looks up records by their topics. The topic uniquely identifies a record.

Records also have *keywords* with them. A keyword is comparable to a word in an index entry. This record contains the keywords "documentation record". The Show Candidates command compares your lookup request to all the keywords and documentation topic titles in the database. So, for example, if you want to find out what information is available on the general topic "record", Document Examiner includes the record you are now reading in its list of possible choices because "record" is one of its keywords. You can see the keywords for any record in the overview of the record.

### Help in Document Examiner

Press the HELP key in the Document Examiner window to display a summary of commands. Clicking Left on [Help] in the command pane produces the same abbreviated display.

Complete documentation of all features is available by command is equivalent to clicking Middle on [Help] in the command pane or by typing Document Examiner Documentation at the prompt.

### Document Examiner Window

You can select Document Examiner in one of two ways:

- Keyboard: Press SELECT D or type Select Activity Document Examiner at the Command Processor.
- Menu: Click on [Document Examiner] in the System menu.

The Document Examiner window contains the following panes, as displayed in Figure 64:

- *Viewer* — where the documentation is displayed
- *Current candidates* — list of current choices, usually resulting from a Show Candidates or Show Table of Contents command
- *Bookmarks* — list of topics you've looked at or indicated that you wanted to look at
- *Command* — where you enter commands, either by mouse or by typing a command

Each of these panes is discussed in detail in this section.

### Viewer Pane

The large area on the left of the window is called the *Viewer pane*. All the documentation you select to read is displayed in the viewer.

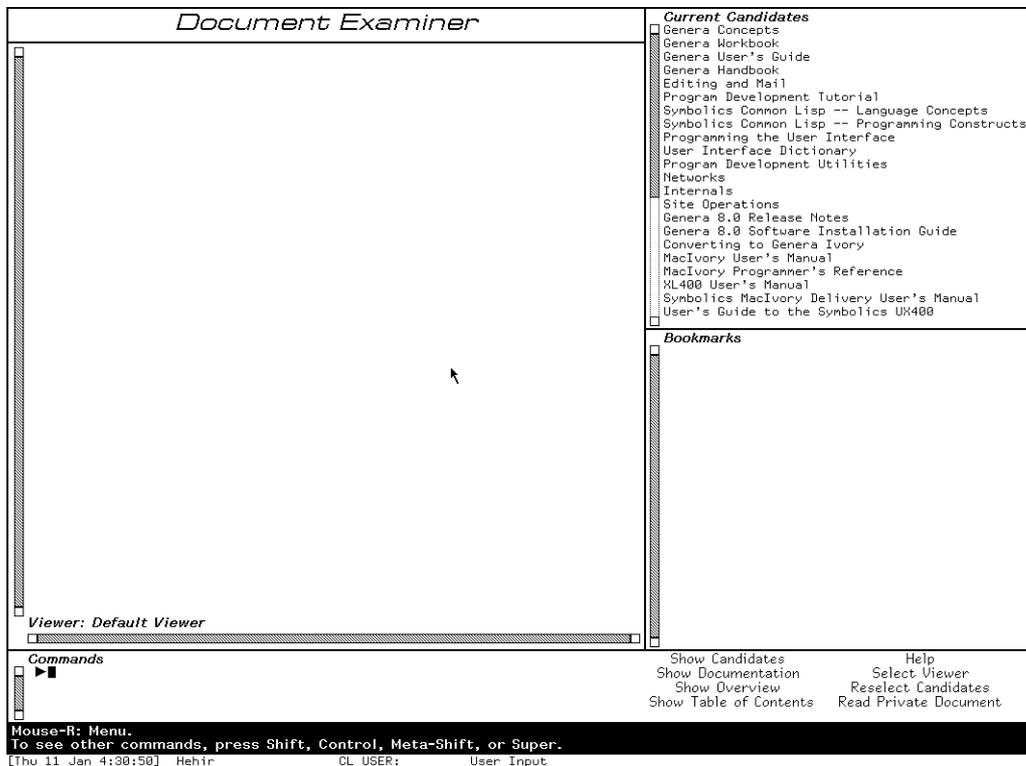


Figure 64. Upon selection, the Document Examiner displays all the books in the document set.

When you select a topic for viewing in Document Examiner, the topic is displayed at the end of the viewer and the topic's name is added to the list of bookmarks. Topics displayed in a viewer are separated by a light horizontal line.

Certain items displayed in the viewer are mouse sensitive; for example, clicking Left on cross-references or the names of documented Lisp objects displays their documentation in the viewer, clicking Middle displays an overview of the topic (a summary of where it is and what it is), and clicking Right presents a menu of operations. Check the mouse documentation line for details.

The viewer is a dynamic window, which gives you the ability to copy a region of text from the viewer into another window, such as Zmacs or Zmail. You can mark a region with the mouse by pressing the CONTROL key and push the marked text on the kill ring using  $\text{⌘-W}$ . For example, you can extract an interesting Lisp code example from a documentation topic and run it in an editor buffer or Lisp listener. Or you can include selections from the documentation in mail or text files. You can also search for a particular string using  $\text{⌘-R}$  to search backward and  $\text{⌘-F}$  to search forward. For an introduction to using dynamic windows, see the section "Using Your Output History".

When you first select Document Examiner, you see the *default viewer*. It is labeled as such in the lower left-hand corner of the pane. You can create or select another viewer, just as you can create or select another editor buffer. To create a new viewer click Left on [Select Viewer] or enter the Select Viewer command and name a nonexistent viewer. You can see a mouse-sensitive list of existing viewers by pressing `c-?`. To go between two viewers, press `c-m-L`. To select another viewer, use Select Viewer.

When you look up documentation in a Lisp Listener (using the Show Documentation command) or in the editor (using `m-Sh-D` or `m-X` Show Documentation), bookmarks for those topics are inserted in a special viewer called the *Background viewer*.

Examples of Lisp code whose lines are wider than the viewer display with those lines wrapped around. When you need to see such examples in a more readable layout, use the Show Documentation command in a wider window, such as a Lisp Listener.

### Current Candidates Pane

The upper right-hand pane of Document Examiner's window displays the names of those topics that are the *current candidates*, that is, a menu of choices from which you can select a topic for viewing. These candidate lists are produced by the Show Candidates and Show Table of Contents commands.

When you first enter Document Examiner, the current candidates pane displays the list of documents registered in the documentation database, as displayed in Figure 64. Each time you use Show Candidates or Show Table of Contents, a new list of current candidates is generated and displayed. Note that lines that are wider than the current candidates pane are truncated.

You can redisplay a list of candidates not currently showing by clicking on [Reselect Candidates] or using the Reselect Candidates command. You'll see a menu of all the documentation searches conducted during the current session. This list is helpful when, for instance, you need to cycle through several sets of candidates. Instead of having to remember exactly what you asked for each time you want to look at it, use the Reselect Candidates command and click on the candidates pane that you want to see.

### Bookmarks Pane

The lower right-hand pane of the Document Examiner window contains a list of *bookmarks*. Bookmarks are the names of topics that you have displayed in the viewer or the names of topics that you want to display later on. Thus, a Document Examiner bookmark is similar to a bookmark you might place in a printed book — as a reminder of something you have just read and might want to reread or as a pointer to something you should read in the future.

Document Examiner automatically inserts a bookmark for a topic when that topic is first displayed in the viewer. The list represents a history of your selections for

a particular viewer. You can also explicitly add a bookmark to the list of bookmarks without having the topic displayed in the viewer. See the section "Putting Topics Aside for Future Reading".

The list of bookmarks distinguishes between bookmarks whose topics have been displayed and those that have not. Topics that are displayed in the viewer are listed on a white background in the order in which you looked them up. Topics not displayed in the viewer follow and are listed on a gray background in the order in which you created the bookmarks. See Figure 49.

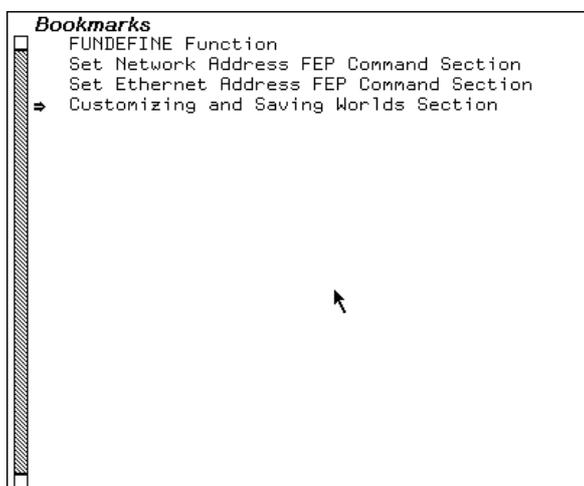


Figure 65. The bookmarks pane shows which topics have been displayed in the viewer.

Names that are wider than the bookmarks pane are truncated.

You also use a set of bookmarks in creating a private document. See the section "Creating a Private Document".

### Command Pane

The bottom portion of the Document Examiner window is called the command pane. It accepts hand-typed commands at the triangular Command Processor prompt to the left and displays a menu of Document Examiner command menu items to the right.

The Command Processor offers completion on command names as well as topic names. The HELP key (after you type at least one character) displays a mouse-sensitive list of possible completions. Pressing HELP before you start to type a command name displays the available commands. After you type at least one character, `c-?` displays a mouse-sensitive list of commands *starting with* the character or characters you have typed and `c-/` displays a mouse-sensitive list of commands *including* the character or characters you have typed.

The mouse documentation line, situated below the command pane, tells you how to use the mouse to activate a particular command from the menu.

The following list describes the commands. All these commands are also available as menu items.

- Help — offers a brief command summary. Or click Right on [Help].
- Document Examiner Documentation — displays the full Document Examiner documentation. Or click Middle on [Help].
- Show Candidates — searches the entire database for documentation topics with titles or keywords that resemble or match what you're looking for and displays a list for you to choose from.
- Show Documentation — displays the documentation for a specific topic and is useful when you know exactly what you're looking for.
- Show Overview — displays basic information about a topic, including a tree graph that shows how the topic fits in with other topics and what keywords are specified for it.
- Show Table of Contents — displays the table of contents of a topic, that is, what other topics that topic includes. It is most useful for getting information about an entire chapter, section, or book.
- Reselect Candidates — displays a menu of all the candidates lists created for a particular viewer during the current work session. Candidates lists are created whenever you invoke the Show Candidates or Show Table of Contents commands. The display is a mouse-sensitive menu from which you can redisplay the results of a previously executed command.
- Select Viewer — changes from one viewer to another or creates a new one. You can create a new viewer by naming a nonexistent viewer in the Select Viewer command. You can move between two viewers by pressing `C-M-L`.
- Remove Viewer — removes a viewer from the list of viewers. Also available from the menu presented when you click Right on [Select Viewer].
- Hardcopy Viewer — hardcopies the contents of a viewer. Also available from the menu presented when you click Right on [Select Viewer].
- Save Private Document — saves the list of bookmarks in the current viewer as a private document. This means you can create your own private document by creating bookmarks for one or more documentation records and then issuing this command to write the collection of bookmarks to a file. Also available from the menu presented when you click Right on [Read Private Document]. You can read or load a private document when you need it. It is not part of the docu-

mentation database. See the section "Creating a Private Document". Read Private Document — allows you to view private document collections created from the documentation database. This command loads all the text associated with the private document into a viewer.

- Load Private Document — loads the bookmarks associated with a private document, but does not load the text. Also available from the menu presented when you click Right on [Read Private Document].
- Hardcopy Private Document — sends all the text in a private document to a printer. Also available from the menu presented when you click Right on [Read Private Document].

### Looking Up Documentation

When you look through a printed book for information, you probably scan the table of contents to see how the book is organized and maybe skim a few chapters to see their major sections, perhaps inserting a bookmark or two when things look interesting. You might then look over the index to see the number and kinds of main and related entries on a particular subject. Sometimes you know exactly what you are looking for and can go directly to the section you want to read.

Document Examiner provides a variety of commands and tools for online lookup that are analogous to the methods you use to examine a printed book, but are much more flexible. They are not limited to a single book, however, and they allow you to browse in the entire documentation database with just a few keystrokes and to take advantage of some "smart" aids to readers, such as the Show Candidates command.

Requests for online lookup are always made with respect to a *documentation topic name*, which can be anything from the name of a document, chapter, or other book subdivision, to the name of a Lisp code object, like a function or flavor. Every Lisp object is documented in its own documentation record; the documentation topic name of the record is the name of the Lisp object. (You may need a package prefix. See the documentation for languages other than Lisp in case there are special procedures.)

#### Show Documentation

Displays the documentation for a specific topic name in the viewer and inserts a bookmark for the topic. You can do the same thing by clicking Left on the name of any documentation topic. You could look up the section you are reading now by using Show Documentation and specifying "Looking Up Documentation".

Show Candidates Displays, after searching the database, all topic names that satisfy your lookup request. This is often the best way to search the documentation database. By default, Show Candidates uses

a heuristic, or "smart", searching strategy that allows you to see all the candidates that are close to what you asked for. For instance, if you specify "local" to Show Candidates, the candidates list includes "locals" "locative", "location", and "localize", as well as "local". Show Candidates lets you specify several search strategies. For information on these alternative search strategies, see the section "Show Candidates Command".

**Show Overview** Displays the local context of a topic name, showing which topic includes it in the document hierarchy, which topics are on the same level, and which topics are included by it in the hierarchy. This is good for examining the organization of topics and tracing topics to more general and more specific levels of abstraction. The display also provides related information, such as, which documents a topic appears in and what keywords are associated with it.

**Show Table of Contents** Displays the table of contents for a given topic name, showing the supplied topic name and all of its subordinate topics. This command is good for examining the organization of topics within a document.

Genera's online documentation facility is flexible. You can look up information from several contexts — in a Lisp Listener, in an editor buffer, or directly in Document Examiner. Where you look up documentation probably depends on the kind of work you are doing and your own particular workstyle.

- If you are programming in an editor buffer, you can use `M-X Show Documentation` or its keyboard accelerator, `M-SH-D`.
- In a Lisp Listener, you can issue Show Documentation at the Command Processor, `M-SH-A` to see documentation for a Lisp function, `M-SH-F` to see documentation for a flavor, or `M-SH-V` to see documentation for a Lisp variable.
- If you plan to poke around or to read a great deal of documentation, you can take advantage of the extra features offered by Document Examiner, such as bookmarks and the Show Candidates command. Shifting to Document Examiner in such cases is easy. Bookmarks for all the topics you've looked at outside Document Examiner are inserted in the Background viewer.

In addition, Document Examiner allows you to:

- Insert a bookmark in a documentation topic, see the section "Putting Topics Aside for Future Reading".
- Display the documentation for topics that are explicitly cross-referenced, see the section "Looking Up a Specific Topic".

- Display the documentation for implicit cross-references, such as the following mouse-sensitive reference to **zl:string-capitalize-words**.

Any topic name appearing in any pane of the Document Examiner window is *mouse-sensitive*. That is, when you move the mouse over a topic name, the mouse highlights the name in an outline box; clicking one of the mouse buttons on a highlighted topic name causes some action to occur. Clicking Left displays the documentation and clicking Middle displays the overview.

Clicking Right brings up a menu of operations that can be performed on that topic name:

- View the documentation
- Hardcopy documentation
- Display a table of contents
- Insert a bookmark
- Display an overview
- Discard a topic

The lookup commands distinguish among like-named topics. For example, if you use Show Documentation for **evalhook**, Document Examiner asks you to select **evalhook** as a function or as a variable. See Figure 50. Similarly, the lookup commands distinguish between Zetalisp objects and those that are Common Lisp or Symbolics Common Lisp objects; Zetalisp objects are displayed with the **zl:** package prefix.

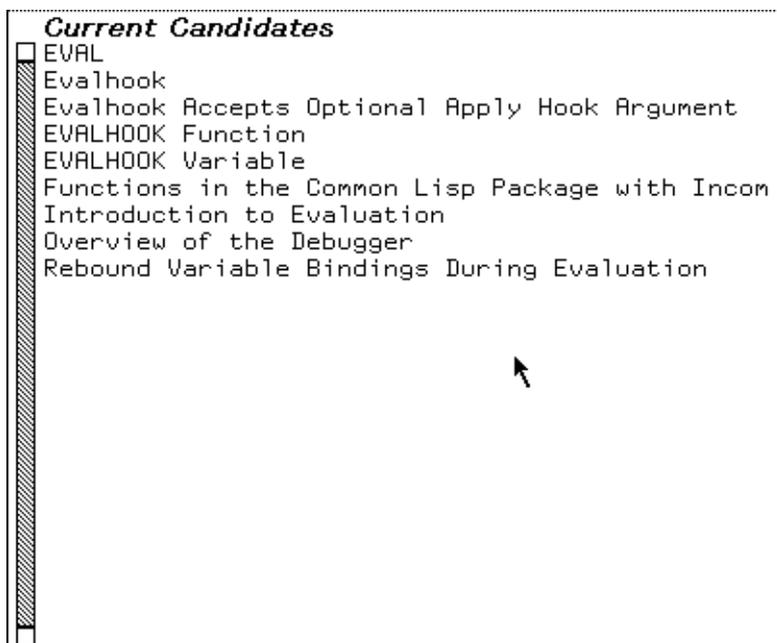


Figure 66. Document Examiner lets you choose among like-named topics.

## Looking Up a Specific Topic

Document Examiner provides several ways to look up documentation for a specific topic, once you know the right topic name:

- Use the Show Documentation command.
- Use the mouse to display the documentation for cross-references and the names of Lisp objects. All topic names are mouse-sensitive; clicking Left on a topic name displays the associated documentation. The names of Lisp objects mentioned in the documentation appear in a boldface character style.

Example: the topic **\*load-set-default-pathname\***, displayed in the viewer, mentions the variable **\*load-pathname-defaults\***, as shown in Figure 51. Clicking Left on the variable name displays the documentation for the topic **"\*load-pathname-defaults\*"**. See Figure 51.

- Use the mouse to display the documentation associated with current candidates, bookmarks, or topic names displayed in an overview or a table of contents.

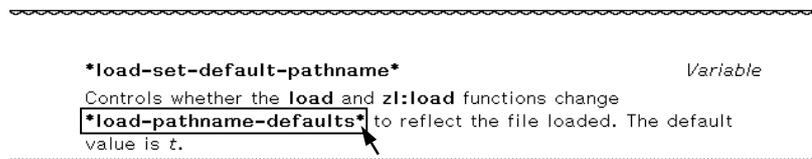


Figure 67. The names of Lisp objects displayed in the viewer are mouse-sensitive.

In addition, you can look at documentation topics in the editor and in a Lisp Listener.

## Show Documentation Document Examiner Command

Show Documentation is useful when you know exactly what you are looking for. Show Documentation looks up a topic, displays it in the viewer, and, if it is a new topic, inserts a bookmark for the topic in the bookmarks pane. To invoke the command you can either click on [Show Documentation] in the command menu or type Show Documentation in the command pane. The command prompts for a topic name.

Clicking Left on a topic is the equivalent of Show Documentation.

If you aren't quite sure what you're looking for, see the section "Show Candidates Command".

Topic names for methods are of the form (flavor:method *generic-function-name flavor-name*). Thus, to use Show Documentation for a method, give the topic name of the method in this form, for example:

```
Show Documentation (flavor:method :set-edges tv:menu)
```

Show Documentation is also an editor command and a Command Processor command. In an editor, `M-X Show Documentation` (or its keyboard equivalent `M-SH-D`) prompts you for a topic name, with completion.

When you are typing in a Lisp Listener or a Lisp mode buffer in Zmacs, the following commands give you documentation for the current Lisp object (the one nearest the cursor):

<code>M-SH-A</code>	Looks up the documentation for the current function.
<code>M-SH-V</code>	Looks up the documentation for the current variable.
<code>M-SH-F</code>	Looks up the documentation for the current flavor.

You can use `C-/` and `C-?` to find topic names containing or starting with the the sequence of characters you have typed so far. For example,

Show Documentation search `C-/`

displays a mouse-sensitive list of all documentation topics *containing* "search" in their names. And,

Show Documentation search `C-?`

displays a mouse-sensitive list of all documentation topics whose names *start with* "search".

See the section "`C-?` and `C-/`".

## Searching Through Documentation

In addition to looking up specific topics online, you can search through the entire documentation database. Document Examiner includes many tools to make this searching easy.

The Show Candidates command, for instance, can search for a broad range of possible documentation topics using heuristic, or "smart", techniques. This means that the command locates a word and variations on it instead of searching for a strict match. The command defines the different variations for the word by checking for common suffixes using a technique called *stemming*. Thus, if you specify "local" to Show Candidates, the result includes "locals", "locative", "location", and "localize", as well as "local".

Show Candidates isn't limited to documentation titles in doing the search. It searches through an index that contains all the words in topic titles and a list of *keywords* (somewhat like an index entry) for each documentation topic.

If, in the course of your searching, you see topics that look interesting, you can either read them as you go or add bookmarks so you can look at them later.

Document Examiner also helps you search by enabling you to jump from topic to topic.

- You can click on implicit cross-references to language objects, such as **setq**, **cdaddr** and **zl:string-pluralize**.
- You can click on explicit cross-references, such as, see the section "String Conversion".

### Searching with the Show Candidates Command

The Show Candidates command is your first choice for browsing through the documentation database. While Show Candidates is fundamentally an index-search command, it has several capacities beyond what is normally found in index-searching.

- In the first place, Show Candidates searches through all the topic names in the database, and also searches through all the keywords (which are similar to index entries) as well.
- Secondly, by default, Show Candidates performs "smart" (heuristic) searching. This means that you no longer have to think about the form of the word to supply as a lookup request; the heuristic approach finds singulars and plurals, gerunds, negations, and so on. This is because the search is based on the *stem* of your lookup request.

For example: The stem of "move" is "mov". Show Candidates `move` will match any topic names and keywords containing "move", "moves", "moved", "moving", `*move-mumble*`, `:"move"`, and so on. (This is a hypothetical example. In fact `*move-mumble*` is not documented.)

If you supply two or more words to Show Candidates, it searches for candidates that match *all* stems in any order. Thus, if you supply both "pathname" and "completion" to Show Candidates, your candidates list includes both **fs:complete-pathname** and "Pathname Completion is Supported". This is the default behavior.

You can specify adjacent search and also string searching by exact match, initial string, and substring by typing in your search term or terms and then pressing `m-COMplete` to display a menu from which you can choose those options. Press `END` to accept the choices displayed in boldface or `ABORT` to abort.

If you select adjacent search and supply "pathname" and "completion" in that order, your candidates list will include only "Pathname Completion is Supported".

Use the same menu to limit your search to exact matching, initial substring matching, and substring matching.

### Putting Topics Aside for Future Reading

Occasionally in browsing through the database you notice topic names that look interesting but that you cannot or do not want to read immediately. Suppose, for example, that you are scrolling through several hundred current candidates generated by the Show Candidates command. Rather than read the documentation for each

interesting topic as you come across it, you prefer to scan all the candidates first, mark all those topics that look like they might be helpful, and read them all at once later. Remembering their topic names would be extremely tedious.

Document Examiner lets you put documentation aside for future reading by allowing you to explicitly insert a *bookmark*, or pointer, for a documentation topic. The topic name is added to the bookmarks pane. To insert a bookmark click  $\mathfrak{H}$ -Middle on a mouse-sensitive topic name.

Bookmarks are automatically inserted in two cases: whenever you read the documentation for a topic into a viewer and when you look up documentation in a Lisp Listener or in the editor. In the latter case, the bookmarked is inserted in a special viewer called the *Background viewer*, but the topic is not displayed there.

Bookmarks are used to create your own document for online viewing, see the section "Creating a Private Document".

Particular bookmarks are associated with a particular viewer, so that when you select another viewer, the list of bookmarks associated with it is also selected.

### Getting Information About a Documentation Topic

Two commands — Show Overview and Show Table of Contents — are useful for examining the organization of topics within a document and for learning the position of a particular topic in the book or books in which it appears.

#### Show Overview

Sometimes you notice an intriguing-looking topic name among a list of current candidates but are not quite sure that the topic is the one you need. Before you commit yourself to reading it into the viewer, you might want to see how the topic fits into the structure of the book or books in which it appears or which other topics, related to it, are covered in the same book. Other times you might read a topic into the viewer and find that it is not quite at the right level of abstraction: it provides too much reference information, whereas you need more general coverage.

Document Examiner provides a command — Show Overview — that displays the position of a topic within the document or documents in which it appears. Given a topic, the command displays a two-part overview of the topic in a dynamic typeout window:

The top part includes:

- The type (is it a section of text or does it document a Lisp object, like a function, for instance) and the name of the topic
- Possibly a short summary of the topic
- The names of any other topics in the documentation that include this one

- The names of any printed books that contain the topic
- The topic's keywords (similar to index entries)
- The names of related topics (cross-references)

The names of topics and books in this display are mouse-sensitive.

The bottom part is a graph of the document hierarchy around the topic you choose. (If the topic appears in more than one document — multiple graphs are displayed.) This graph includes:

- The topic that includes the original topic
- Other topics included by the topic that includes the original topic
- The topics the original topic includes

Figure 52 shows the display produced by doing Show Overview of the topic "Disk Error Handling".

All the items in the overview graph are mouse-sensitive.

For example, when you get an overview of "Disk Error Handling", you can see that the topic is included in the topic "Disk System User Interface" and that it includes four other topics. If this overview does not give you enough information, try an overview on the parent topic (in this case, "Disk System User Interface").

While the Show Candidates command searches through the entire documentation database, Show Overview helps you find out where you are in a document and how to navigate your way through it.

Show Overview differs significantly from "Show Table Of Contents" . Compare the display of the two commands. See Figure 53.

You can also find the printed book in which the topic appears by using the `m-x` What Document command in the editor. It displays the name of the printed book that contains the given documentation topic. If the topic is included in more than one book, the titles of all the books containing the given topic are listed.

### Show Table Of Contents

Much as with printed documentation, you might want to become familiar with the overall structure of an online document before reading about any one topic. When you examine documentation online, the more general the topic you look up, the larger the amount of documentation you see for it. The most general topic names, obviously, are the names of the books in the printed documentation set. For example, if you asked to see the documentation for the topic name "General User's Guide", Document Examiner would display the contents of the entire book. This might not be what you want!

The Show Table of Contents command is useful on those occasions:

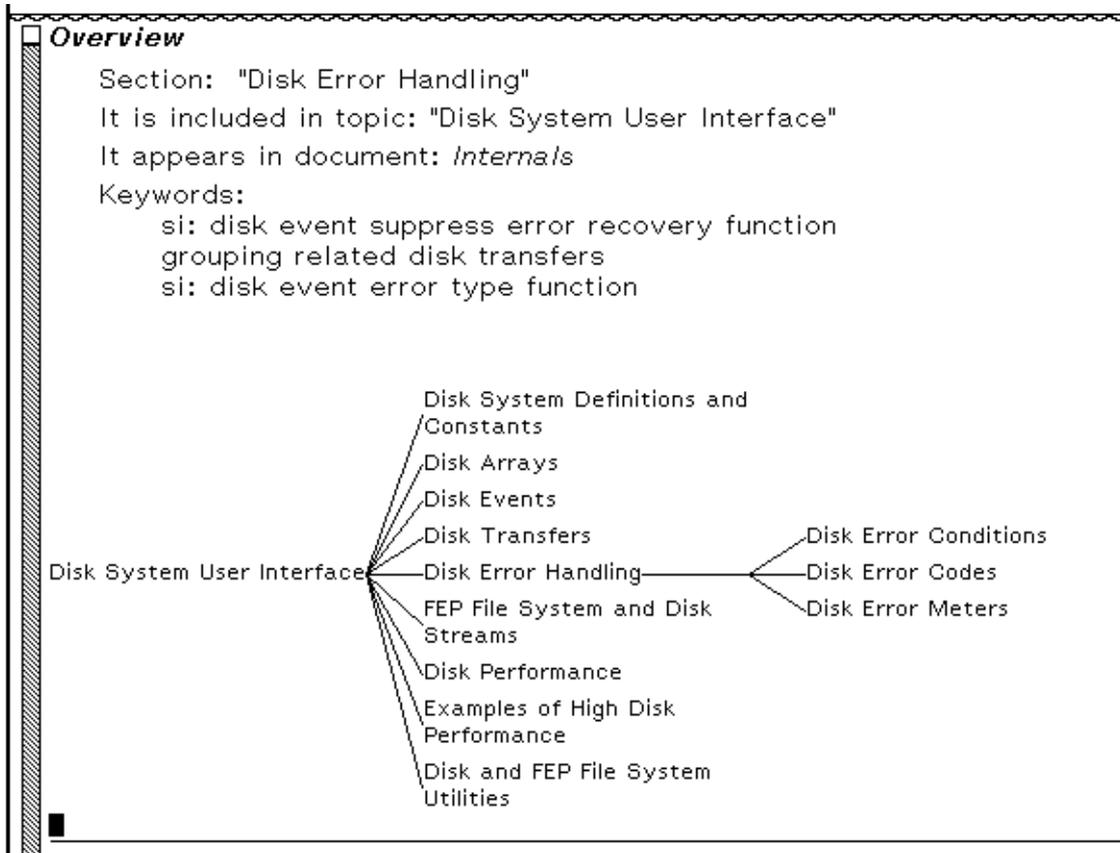


Figure 68. Document Examiner display of Show Overview of topic "Disk Error Handling".

**Document Examiner**

**Overview**

Section: "Lists"  
 It is included in topic: "Data Types"  
 It appears in document: *Symbolics Common Lisp: Language Concepts*  
 Keywords:  
 Lists

Data Types

- Data Types and Type Specifiers
- Numbers
- Symbols, Keywords, and Variables
- Lists
  - Introduction to Lists
  - Type Specifiers and Type Hierarchy for Lists
  - Printed Representation of Lists
  - How the Reader Recognizes Lists
  - Special Types of Lists
  - Operations with Lists
  - Obj-Coding
  - List Functions and Macros in the Common Lisp P
- Arrays
- Sequences
- Characters
- Strings

**Current Candidates**

- Lists
  - Introduction to Lists
  - Type Specifiers and Type Hierarchy for Lists
  - Printed Representation of Lists
  - How the Reader Recognizes Lists
  - Special Types of Lists
    - Property Lists
      - Creating and Modifying Property Lists
      - Functions That Operate on Property Lists
    - Dotted Lists
    - Association Lists
      - Creating and Modifying Association Lists
      - Functions that Operate on Association L
  - Trees
  - Operations with Lists
    - Controlling List Operations with Keyword Ar
    - Predicates that Operate on Lists
    - Functions for Finding Information About Lis
    - Functions for Constructing Lists and Conseqs
    - Functions for Copying Lists
    - Functions for Extracting from Lists
    - Functions for Modifying Lists

**Bookmarks**

Figure 69. The candidates pane shows the table of contents for a topic; the type-out window shows the overview of the same topic.

- When you are unsure what level in the documentation you need (chapter level or section level, for example)
- When you would like to become familiar with the organization of a document

Given a documentation topic name, the command displays the topic name and all the topic names subordinate to it in the candidates pane. For example, Figure 54 shows the first pane of the table of contents of the document "Genera User's Guide".

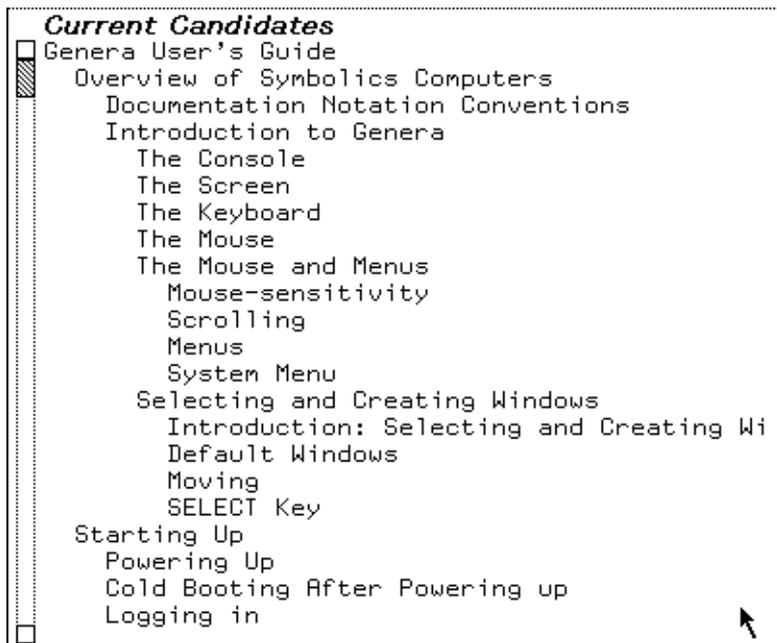


Figure 70. The candidates pane displays the table of contents for an entire document.

The indentation of the candidates indicates the hierarchy of topics.

You can ask to see a table of contents for any topic, not just a book. Of course, if a topic doesn't contain any other topics, then it has no table of contents.

This command is also available in the editor as `m-X Show Table of Contents`. `c-U m-X Show Table of Contents` prints the table of contents on a printer.

For more general searching, see the section "Searching with the Show Candidates Command".

### Site Configuration: A Conceptual Overview

*Site configuration* enables your Symbolics computers to describe and access the resources available to them. Once your site has been configured, all of its Symbolics computers can find out about the other hosts, printers, and users there.

Namespace service is at the heart of site configuration. In order for one machine (a local host) to use any of the resources provided by other machines (remote hosts), namespace service — managed and provided by a namespace server — is required.

A local host depends on the namespace server for answers to these questions:

- How is the remote host connected to the local host (what is the remote host's network address)?
- What network protocol must be used to obtain the desired service?

By configuring your site, you give each Symbolics machine sufficient information to know where and how to obtain namespace service. You also give your namespace server sufficient information to provide that service.

More information is available about the terms we use when discussing site configuration. See the section "Glossary of Terms for Site Configuration".

### **Site Configuration and Namespace Service**

The namespace server uses namespace files to describe each resource at a particular site. The namespace server might not store the namespace files locally, but it knows where to locate them.

If your site is large, with many (over ten) user machines to make demands on the namespace server, Symbolics recommends that you create a dedicated namespace server; use one Symbolics machine that's unavailable for user applications. If your site is small (under ten user machines), designate one of the user machines as the namespace server.

The namespace server's purpose is to collect and maintain information for a site. All of the information known about a site's network(s) and each host, printer, and user is stored in the namespace database. More information is available about namespace objects and the namespace system. See the section "Setting Up and Maintaining the Namespace Database".

There can be more than one namespace server at a site. One server is the primary namespace server; the others are secondary namespace servers. More information is available about the differences between server machines. See the section "Machines and Worlds".

A typical Symbolics site uses a namespace server to store the namespace database, and a file server to store system sources and online documentation. It is possible for one machine to perform both these services, provided it has enough disk space.

There are some restrictions pertaining to servers. The namespace server must be a Symbolics computer. The system sources and online documentation must reside on one of the following:

- A Symbolics computer.
- Any UNIX host running NFS, in a .sct directory (see the section "Using SCT with a UNIX File System").

More information is available about system sources and online documentation. See the section "System Sources and Online Documentation".

### Choosing a Site Name

As part of the site configuration process, you'll select a site name that designates the physical locale of your machines. (This is different from a host name. Host names designate particular machines at a given location.)

When you are ready to configure your site, choose a site name that — in all probability — will not be duplicated by another site. For example, if you are setting up a site at ACE Computers in Nutley, New Jersey, and your department works on robotics, you might call your site ACE-Robotics or ACE-Nutley rather than just ACE.

**Note:** You cannot use a dot [.] as part of the site name that you choose.

### Choosing Machine Names and Network Addresses

It is easier to refer to a computer by its name than by its Chaosnet (the Symbolics network) or Internet address. It is possible to refer to a host by its Chaosnet address, but this can become tedious.

Most customers choose a theme by which to name their machines. For example, each Symbolics computer at your site might be named after different mountains, or famous scientists.

Most sites have all of their machines on the same Ethernet cable. When you create a site, you select a subnet number for this cable. You also select a unique number for each machine. If you have machines on more than one Ethernet cable, you select a unique number for each cable.

If you use networks such as NFS, DNA or IP-TCP along with Chaosnet, we suggest that when you configure your site, you employ an addressing scheme that will encompass all of these. More information is available about network addressing schemes, see the section "Network Addressing".

**Note:** XL-family, 3600-family, UX-family and MacIvory machines come from Symbolics with Chaosnet address 52525. This is almost never what you want. In fact, if a site already has one machine using that address, the new one will be confused with the old one. NXP1000 machines come pre-configured with the Internet address that you specify at the time of purchase. If you did not specify one, you must boot your machine by hand when you install it, see the section "Caveats About NXP Installation".

### Format of Chaosnet Addresses

A Chaos address is a 16-bit quantity, in which the high-order 8 bits represent the subnet number, and the low-order 8 bits represent the host number on that sub-

net. Neither the subnet number nor the host number can be zero. Chaos addresses are expressed in octal.

### Example: Chaos Address 401

```

    15 14 13 12 11 10 9  8  7  6  5  4  3  2  1  0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|<-----Subnet number----->|<-----Host number----->|

```

The subnet number is 1.

The host number is 1.

The Chaos address is 401 octal.

For technical details on how the Chaosnet address is used: See the section "Chaosnet Addresses and Indices".

Chaosnet addresses are displayed in octal, and each Chaosnet address at a site must be unique. For example, if you have a machine whose cable (subnet) number is 1 and whose machine (host) number is 2, the resulting Chaosnet address is 402.

The lowest Chaosnet address that you can have is 401. Symbolics suggests that you use this number (401) as the Chaosnet address for your first machine, and that you number subsequent machines in ascending order.

### Format of Internet Addresses

Internet addresses are expressed in decimal, in four octets separated by periods. Each octet is 8 bits long. There are three kinds of Internet addresses: Class A, Class B, and Class C.

#### Examples of Internet addresses:

- 10.2.0.7 is host 2.0.7 on Class A network 10.
- 139.41.0.3 is host 0.3 on Class B network 139.41.
- 192.10.0.200 is host 200 on Class C network 192.10.0.

Note that the host number cannot be zero or 255, because those are considered broadcast addresses.

## Interpreting Internet Addresses

Internet addresses consist of network and host fields. The network field identifies the network, and the host field identifies the host on that network. This size of the Internet address depends on the address and the configuration of the network.

You can use a subnet field for networks containing subnets. Using a subnet field divides the address into three fields. A subnet mask determines the bits used for selecting a subnet. Note that the rules for determining a subnet field vary for each network.

**Note:** You cannot fill a field (network, subnet, or host field) with all zeros or ones for representing a network, subnet, or host.

### Class A Addresses

A Class A Internet address is a 32-bit number, in which the high-order octet (8-bits) represents the network number and the following three octets represent the host number. The first octet is less than 128.

#### Example of Class A Internet Address: 10.2.0.7

```

+-----+-----+-----+-----+
|00001010|00000010|00000000|00000111|
+-----+-----+-----+-----+

|<-net-->|<-----host----->|

```

### Class B Addresses

A Class B Internet address is a 32-bit number, in which the two high-order octets represent the network number and the following two octets represent the host number. The first octet of a Class B network is greater than or equal to 128 and less than 192.

#### Example of Class B Internet Address: 139.41.0.3

```

+-----+-----+-----+-----+
|10001011|00101001|00000000|00000011|
+-----+-----+-----+-----+

|<---network----->|<-----host----->|

```

### Class C Addresses

A Class C Internet address is a 32-bit number, in which the three high-order octets represent the network number and the low-order octet represents the host number. The first octet of a Class C network is greater than or equal to 192, and less than 224.

**Example of Class C Internet Address: 192.10.0.200**

```

+-----+-----+-----+-----+
|11000000|00001010|00000000|11001000|
+-----+-----+-----+-----+

|<-----network----->|<-host->|

```

**Internet Subnet Number**

The Internet subnet number is the Internet address resulting from replacing the host field of an Internet address with zeros. You can determine the Internet subnet number of a network by determining the class of a host address and replacing the host portion of the address with zeros. For example, the class B address 128.81.38.232 corresponds to the Internet Subnet number 128.81.0.0.

**Subnet Masks**

A subnet mask determines the field of the Internet address specifying the subnet on the network. A subnet mask is a 32-bit quantity containing one in every bit corresponding to the official Internet subnet number. Additionally, the subnet mask contains a zero in every bit selecting a host on a subnet. For example, A class B network (128.81.0.0) is broken into many subnets, using the third octet of the address for selecting a subnet. The Class B default mask is 255.255.0.0; since the third octet determines a subnet, you have to fill it with ones. The resulting subnet mask is 255.255.255.0.

**Mapping an Internet Address into a Chaos Address**

Once you have an Internet address for a host, you can map that address into a Chaos address. You can then assign sequential Chaos addresses for all Chaos hosts on the network. If you are on the Internet, you can use each host's Internet address to derive a Chaos address.

The mapping process is best explained by example. The following two examples show the mapping of a Class B and Class C Internet address into a Chaosnet address:

**Class C Internet address: 192.10.41.48 decimal.**

Step 1: Get the Chaos subnet number and host number.

```

192.10.41.48 is the Internet address.
192          is unused in the mapping.
  10         is unused in the mapping.
    41       is the Chaos subnet number.
      48     is the Chaos host number.

```

Step 2: Convert the decimal subnet and host numbers to octal.

The subnet number (41 decimal is 51 octal.)

The host number (48 decimal is 60 octal.)

Step 3: Insert subnet and host numbers into two eight-bit bytes.

```

    15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |<-----Subnet number----->|<-----Host number----->|

```

Step 4: Express the quantity in octal notation; this is the Chaos address.

```

    0 010 100 100 110 000    (binary representation)
      2  4  4  6  0    (octal representation)

```

The resulting Chaos address is 24460 octal.

### **Class B Internet address: 139.41.9.3 decimal.**

The subnet number is 9 decimal.

The host number is 3 decimal.

```

    15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |<-----Subnet number----->|<-----Host number----->|

```

The resulting Chaos address is 4403 octal.

### **Logical Pathnames and the SYS Host**

A logical pathname is one that does not correspond to any particular physical file system on a host. Logical pathnames make it easy to keep software on more than one type of file system.

For example, the set of files containing the Symbolics system sources and online documentation system is stored at each site. Some sites store these files on a Lisp Machine File System (LMFS), others store them on a VAX/Berkeley UNIX host with the Chaosnet package, and still others use a UNIX host running NFS in a .sct directory (see the section "Using SCT with a UNIX File System"). It is also possible to use a VAX/VMS file server running DNA. More information is available about using VAX hosts. See the section "Site Configuration and Namespace Service".

Symbolics software uses logical pathnames. All sites create a logical host (called *SYS*). Logical pathnames and the logical *SYS* host allow software to work correctly (and the same way) at every site. All pathnames for system software files are logical, and all begin with the logical host *SYS*. Only the translation of each logical pathname to a physical pathname differs at each site.

The translation of logical to physical pathnames depends on the translations files loaded into the current world. For more information about the translations files, see the section "Pathname Translation".

A site that stores the system software on a UNIX system translates logical pathnames into UNIX pathnames. A site that stores the system software on a LMFS translates logical pathnames into LMFS pathnames.

The flexibility of logical pathnames enables sites to split their logical *SYS* host across several physical hosts. A given physical host might contain some of the system software, but the logical entity called a *SYS* host contains all of it.

## **Machines and Worlds**

When you configure your site, you set up different machines to perform specific functions. Each type of machine (user, server, or namespace server) requires a different kind of "world".

### **User machines**

User machines may or may not have special software loaded into their worlds. The distinguishing characteristic of user machines is that they rely on other machines for specific services, such as file service (including access to the online documentation), printer service, and mail delivery.

You can build special worlds for user machines if you want them to run special software (software that is not part of the distribution world). For example, you may load Symbolics non-loaded systems such as Conversion-Tools or Metering into the worlds that you run on the user machines.

### **Server Machines**

Server machines are those machines designated to provide network services to many — or all — of the machines at a site. Typically, server machines provide the following services:

- **File storage:** They have a large file system (the LMFS) where many users store files.
- **Print spooling:** They receive requests to hardcopy information, queue the information, and print it on one or more printers.
- **Store-and-Forward Mail Service:** They receive mail messages and store them locally in mail files for users, or forward them over the network.

Any machine can offer these services. However, Symbolics recommends that a machine upon which many (more than ten) user machines depend for services be used only as a server, and not also as a user machine.

Often, servers run software that is not part of the distribution world. In particular, the Print and Mailer systems have to be loaded in order to provide printer and store and forward mail services, respectively.

To avoid loading these systems every time a server is booted — and to improve the server's performance — you will want to save separate server Incremental Disk Save (IDS) Worlds. You can then Optimize and Save the server worlds that contain whatever non-standard software systems your servers run.

For information about IDS Worlds, the Optimize World command, and the Save World command, see the section "Making, Distributing, and Using Worlds".

### **Namespace Servers**

Primary namespace servers read and write the permanent files of the namespace database. These files can be stored on any file server on the network. However, namespace update operations will run much faster if you store the namespace files on the primary namespace server's local LMFS. (This is only possible if the machine has enough disk space to store the files.)

Secondary servers are not namespace authorities, but they can serve as backups in case the primary server is temporarily unavailable. These servers attempt to keep a copy of the namespace information current by querying the primary server more often than a non-server machine would.

Since namespace servers load namespace changes into virtual memory at boot time, booting on a namespace server will take progressively longer as your namespace grows.

To save time, Symbolics recommends that you periodically create updated namespace server worlds. If you boot your most current site-configured world on a namespace server, all of the newest namespace information will (automatically) be loaded. When you save the newly booted world, you'll get a permanent copy of the up-to-date namespace information.

Remember that each namespace server loads the namespace information only about (and for) those machines that it serves. If you have multiple namespace servers — and they serve different namespaces — you'll need to boot and save a different, updated world on each.

## The Distribution World

Core software, containing the Lisp language and all of the programming tools including the Zmacs editor, the Lisp Machine File System, the Zmail electronic mail facility, Document Examiner and others, is contained within the distribution world. This world is formally known as some version of *Genera*.

When you receive a distribution world from Symbolics, it is without any customizations. Once you have loaded the distribution world onto a machine, you can make modifications to it (such as loading special systems or software) to customize it for your site. More information is available about customizing worlds for user, server, and namespace server machines. See the section "Machines and Worlds".

## Non-Loaded Systems

Systems are collections of files that — when loaded together — affect the Lisp environment in useful ways. For example, if you load the Print system into the Lisp environment of your machine, your machine will then have the capabilities of a print spooler. It will take hardcopy requests from other hosts and print them, one at a time, on a local printer.

Some systems come already loaded into your world by Symbolics; these loaded systems make up what is called the distribution (or base) world. Document Examiner is an example of a system that comes to you as part of the distribution world.

Other systems aren't pre-loaded into the distribution world. These non-loaded systems provide additional facilities that might only be needed by some Symbolics computers. An example of a non-loaded system is the Store-and-Forward Mailer.

Most systems that Symbolics ships contain several files, and optionally include some patches. The patches are modifications that are too small to warrant compiling the entire system.

Each system has a major and minor version number. The major version number indicates how many revisions and recompilations the system has had. The minor version number indicates how many patches were made to the system since the last revision or recompilation was made to it.

More information is available about loading systems. See the section "Load System Command".

## System Sources and Online Documentation

Symbolics distributes a portion of the sources from which the Genera environment was developed, including all of the documentation database.

If you set up one or more machine(s) as server(s) for these files, then you have access to a portion of the system sources. This greatly enhances your development environment and facilitates your development of applications.

You can access the Genera sources from anywhere on the network, without using any disk space other than that on the server machine. For more information: See the section "Logical Pathnames and the SYS Host".

In the Genera environment, all of the printed documentation is available online. You need only to select the Symbolics online documentation system, called Document Examiner, or to execute a few simple commands, in order to view any part of the documentation for the system.

To use Document Examiner, press `SELECT D`. You can get a list of Document Examiner commands by clicking Left on Help. You can get a list of topics to look up by using the Show Candidates command with several words of the general subject you want to find. You can display documentation by using the Show Documentation command on a known topic name or by clicking Left on a topic name in the candidates pane.

### **Pruning the Documentation Database**

The documentation database consists of many files in a large directory on a file server. At some sites, file server space is limited and not all of the documentation files can be kept online. If this is the case at your site, you can selectively remove some of the documentation while still leaving users with online access to commonly used documentation.

Pruning affects only the file space used on the documentation server; it does not affect the part of the document database that resides in the worlds on individual machines. Document Examiner commands (such as Show Candidates and Show Overview) still find the names of sections that are in documents whose files have been deleted.

When a user tries to look up a topic from a removed book, Document Examiner displays a notice that the topic appears in a book that has been removed from the online set. (The notice shows the book name and the name of the file that contains that particular section.)

If users request it, you can restore files from tape to the documentation directory. The material in those files will be available for lookup as soon as they have been restored.

### **Documentation Database Directories**

The directory `SYS:DOC:INSTALLED-N`; contains the set of directories that hold the documentation files. Because of the hypertext nature of our documentation, there is not a complete correspondance between a given set of files and a book (some files may contain records for more than one book, or some records may be used by several books). However, as a rough cut to help prune the documentation files, here is a list of the directories as they relate, for the most part, to particular books. This should allow you to remove the directories for those books that you do not use. Simply use `Dired` in `Zmacs` or the `Command Processor` command `Delete Directory` to delete the directories for those books you want to remove. **Note:** When using the `Delete Directory` command be sure that the pathname you specify is a directory, not a file in a directory. In the latter case, the directory one level up is the one that gets deleted. See the section "Delete Directory Command".

Books 1, 2, 3, and 4: <i>Genera Concepts</i> <i>Genera Workbook</i> <i>Genera User's Guide</i> <i>Genera Handbook</i>	cp; hard; lms; miscf; miscu; miscui nota; sage; user;
Book 5: <i>Editing and Mail</i>	conv; fed; zmacs; zmailc; zmailt;
Book 6: <i>Program Development Tutorial</i>	clyde; tools; workstyles;
Books 7, 8, and 9: <i>Symbolics Common Lisp - Language Concepts Programming Constructs Dictionary</i>	ansi-cl; arr; char; clcp; clos; cond; conversion-tools; data-types; defs; dictionary; eval; flav; flow; func; io; mac; miscf; pkg; prim; scope; strings; str;
Books 10 and 11: <i>Programming the User Interface</i> <i>User Interface Dictionary</i>	ined; menus; new-uims; uims; scroll; windoc;
Book 12: <i>Program Development Utilities</i>	comp; debug; maint; meter; meter-int; misct;

Book 13: <i>Networks</i>	
	netio; nfile; prot;

Book 14: <i>Internals</i>	
	audio; fep; instr; init; int; iprim; proc; sched; stor; tape;

<i>Site Administration</i>	
	fep; file; fsed; pig; site;

Ivory Documentation:	
	ivory; macivory; rpc; ux400; xl400;

Installation Guides:	
	sig;

<i>Genera 8.0 Release Notes</i>	
	rn8-0;

### Setting and Defining Sites

This section describes the criteria for using the Set Site and Define Site Commands, and shows the Set and Define Site dialogues. See the section "Set Site Command".

See the section "Define Site Command".

Here are the criteria for determining whether to use the Set Site or Define Site command on a Symbolics 3600-family or Ivory-based machine:

Set Site	If the namespace files have already been created, use the Set Site comand. This gives your machine access to the already configured site's resources.
----------	---

Define Site	If the namespace files do not yet exist, use the Define Site command. This creates a new namespace that you can expand later to include other hosts, sites, users, printers, and networks.
-------------	--

## Set Site Dialogue

Issue the Set Site command. If you accept the default ("Get from network"), your machine broadcasts a query to other network machines to learn their site name and the identity of the local machine.

```
Command: Set Site (site name [default Get from network])RETURN
```

If you wish to change your site, or if information about your site is unavailable, enter a site name manually. In the examples that follow, *Downunder* is the name of a sample site.

1. When the local host is the namespace server for your site:

```
Command: Set Site (site name [default Get from network]) downunder
Parameters for site DOWNUNDER
Namespace Name: Downunder
Namespace Server Name: Local
Namespace Descriptor File: HATCH:>sys>site>downunder-namespace.text
<ABORT> aborts, <END> uses these values
```

Figure 71. Local host is the namespace server

If the name of the namespace server provided is "Local" or the name of the local host, the menu adjusts to ask for the name of the namespace descriptor file, since you are saying that the local host is the primary namespace server for the site, as in Figure 71 .

2. If the name of the namespace server is different from the local host's name, the menu adjusts to ask for the network address of the Primary Namespace Server, as in Figure 56 .

```
Command: Set Site (site name [default Get from network]) downunder
Parameters for site DOWNUNDER
Namespace Name: Downunder
Namespace Server Name: Wombat
Namespace Server Address: a pair of a network and an address
<ABORT> aborts, <END> uses these values
```

Figure 72. Namespace server is not the local host

3. The network must be a primary network for the local machine. The address must be a valid address for the network. Figure 57 shows the complete dialog for setting the site when the Primary Namespace Server is not the local host: These are all the parameters needed for setting site when the local host is not the primary namespace server. Of course, you could just say Set Site with no parameters, since the same server specified explicitly in this last example would presumably respond to the broadcast request from the local host.

```

Command: Set Site (site name [default Get from network]) downunder
Parameters for site DOWNUNDER
Namespace Name: Downunder
Namespace Server Name: Wombat
Namespace Server Address: CHAOS 12345
<ABORT> aborts, <END> uses these values

```

Figure 73. Complete menu for setting the site when the Primary Namespace Server is not the local host.

4. When the local host is the primary namespace server, some additional parameters may be necessary. Figure 58 shows the dialog when the local host is the Primary Namespace Server. The defaults are set up for the local LMFS.

```

Command: Set Site (site name [default Get from network]) downunder
Parameters for site DOWNUNDER
Namespace Name: Downunder
Namespace Server Name: Local
Namespace Descriptor File: HATCH:>sys>site>downunder-namespace.text
<ABORT> aborts, <END> uses these values

```

Figure 74. Additional parameters needed when the local host is the Primary Namespace Server

If the user provides a name for the local host that is not the current name or "Local". The address of the server will be requested. If the address is that of the local machine, the menu looks like this:

```

Command: Set Site (site name [default Get from network]) Downunder
Parameters for site DOWNUNDER
Namespace Name: Downunder
Namespace Server Name: Wombat
Namespace Server Address: CHAOS 24503
Namespace Descriptor File: HATCH:>sys>site>downunder-namespace.text
<ABORT> aborts, <END> uses these values

```

5. If the descriptor file is not on the local host, the minimal parameters necessary to access the host holding the descriptor file will be required. Note: Due to a bug, descriptor files cannot be stored on UNIX hosts.

See the section "Customizing and Saving Worlds".

## Define Site Dialogue

In a Lisp Listener, issue the Define Site command and give a site name. This might be the name of your company, or it can be related to the machine names you have chosen.

```
Command: Define Site (site name) Downunder
```

A menu comes up with with the necessary fields to be filled in, as in Figure 61 :

```
Command: Define Site (site name) Downunder
Defining site DOWNUNDER with the local host as the Primary Namespace Server
Namespace Server Name: the name of the primary namespace server
<ABORT> aborts, <END> uses these values
```

Figure 75. The basic Define Site menu

Enter the name of your namespace server. The Define Site dialogue expects an actual name here, so do not enter Local for the Namespace Server name.

After you have entered the name of your namespace server, additional menu slots are displayed, with the appropriate default information filled in. You can, of course, change them.

```
Command: Define Site (site name) downunder
Defining site DOWNUNDER with the local host as the Primary Namespace Server
Namespace Server Name: wombat
System File Directory: wombat:>sys>site>
Namespace Descriptor File: wombat:>sys>site>DOWNUNDER-namespace.text
Default Login: Lisp-Machine
Host for Bug Reports: wombat
Local Timezone: EDT
Standalone Site: Yes No
<ABORT> aborts, <END> uses these values
```

Figure 76. An expanded Define Site menu

#### Namespace Server Name

The name of the local host in the new site.

#### System File Directory

The physical directory that will correspond to the logical directory SYS:SITE; in the new site. For more information, see the section "Logical Pathnames and the SYS Host".

#### Namespace Descriptor File

The name of the file that will hold the namespace description

that is used to find all the files that make up the namespace database.

Default Login	The "Lispm name" of the user object "Lisp-Machine".
Host for Bug Reports	The name of the host that will accept bug-report mail. For information about the mailer, see the section "Installing and Configuring the Mailer".
Local Timezone	The abbreviation of the local timezone or an integer indicating the hour offset from GMT. For more information about time zones, see the section "Specifying a Time Zone for Your Site".
Standalone Site	Whether the local site consists of just the local machine or not. A broadcast is used to verify the identity of the local host and get the current time when booting at non-standalone sites. Saying "Yes" means that these broadcasts will not be performed.

Press END to define your site.

If a name other than the Namespace Server Name is used in the Namespace Descriptor File, System File Directory, or Host for Bug Reports fields, the dialog will ask for additional parameters to define these hosts in the new site. This information consists of system-types, addresses, and file protocols used to access files on the hosts.

You are now ready to log in and use the newly configured software. The local host is automatically registered at the newly defined site. You may want to save the new, site-specific version of your world. For more information, see the section "Customizing and Saving Worlds".

### **Specifying a Time Zone for Your Site**

All sites are in some time zone. The time zone translates user-specified time into universal time. Users specify their site's time zone by modifying the time zone attribute of their site object. This can be done during the Define Site dialogue and/or with the Namespace Editor.

When you boot a distribution world, if the machine is unable to get the time from the network and does not trust its calendar clock (typically because the FEP has been reset, or the board has been replaced, or the machine is new) you are first prompted for a local time zone and then for the local time.

It is possible to set your time zone with mnemonic symbols. Besides Eastern Standard Time (EST), Central Standard Time (CST), and Pacific Standard Time (PST), you can use other commonly accepted mnemonics for time zones for all over the world. You can specify a time zone as an offset from Greenwich Mean Time (GMT) as well as by mnemonic name.

Specify time zones west of GMT by their mnemonic if one is defined, or by a four-digit number preceded by a - (for west of GMT) or a + (for east of GMT). Four-digit numbers must be in the range -1200 to 1200 and can end either in 00 or 30.

For example, -0500 means five hours west of Greenwich, and is equivalent to EST. It is also possible to allow time zones on the half hour, like South Australian Standard time, which is 9.5 hours east of GMT. You specify this by either SAST or +0930. See the function **time:timezone-string**.

### Daylight Savings Time

Every area of the world has its own rules that control when daylight savings time is in effect. The system "knows about" daylight savings time for time zones in the United States (but not for European countries or other time zones).

If you are in a time zone outside of the United States, you must manually tell the system when daylight savings time is in effect. Do this by using the Namespace Editor to edit your site object.

Edit your site object with the Edit Namespace Object command, using a namespace class of **Site**. Use the mouse to replace the time zone with a new one of your choice.

**Note:** When you create a site, you are queried for a time zone (this happens during the Define Site dialogue). If your site is in the United States, enter the standard version of your local time zone (like EST), regardless of whether daylight savings is in effect. If you enter your time zone using the Namespace Editor, enter just the standard time zone as well.

Here is a complete list of the time zones that adjust automatically during daylight savings:

<i>Time Zone</i>	<i>Abbreviation</i>
Atlantic Standard Time	AST
Central Standard Time	CST
Eastern Standard Time	EST
Mountain Standard Time	MST
Pacific Standard Time	PST
Yukon Standard Time	YST
Alaska-Hawaii Standard Time	AHST
Hawaii Standard Time	HST

<i>Time Zone</i>	<i>Abbreviation for Daylight Savings</i>
Atlantic Standard Time	ADT
Eastern Standard Time	EDT
Central Standard Time	CDT
Mountain Standard Time	MDT
Pacific Standard Time	PDT
Yukon Standard Time	YDT
Alaska-Hawaii Standard Time	AHDT

New Zealand Daylight Time      NZD

### Glossary of Terms for Site Configuration

This section defines some terms related to site configuration.

*namespace*            Every site has a *namespace* associated with it. The namespace is a database in which names of users, hosts, and printers are resolved into their representative objects. (See *namespace database*, below.)

*namespace server*    A *namespace server* maintains the namespace database files (see *namespace*, above). A namespace server also processes queries and updates about the mappings and the objects to which those mappings refer. All hosts chosen as namespace servers must be Symbolics computers.

*primary namespace server*  
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.

*secondary namespace server*  
A namespace can have one or more *secondary namespace servers*. These servers answer queries when their information is up-to-date, and they can ask the primary server for up-to-date information. Secondary namespace servers are most useful for large sites (more than 25 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, click on [Namespace Editor] in the System menu or use the Edit Namespace Object command. The namespace database usually resides in files located in the SYS:SITE; logical directory.

The default location for the namespace database is on the primary namespace server. The default name for the file that contains the names of the other files making up the database (the namespace description file) is as follows (where 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-objects.text

```

The Edit Namespace Object command 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*. Objects can fall within different *classes*. The set of classes in the namespace database is as follows:

```

namespace
site
host
user
network
printer

```

For example, suppose you have a user, CHARLIE, logged in to Symbolics machine TUNA at site FISHERY. The user, host, and site are all registered in the namespace called FISHERY.

- FISHERY is an object of class namespace (because it's the name of the namespace that contains the site FISHERY, the user CHARLIE, and the host TUNA).
- FISHERY (a second object) is a object of class site (because it's also the name of the site).
- TUNA is a host object.
- CHARLIE is a user object.

*namespace object* A *namespace object* is a resource (a namespace, site, host, user, network, or printer) that's registered in a namespace database.

*standalone site* 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.