Open Genera Installation Guide

Open Genera 2.0 Description

Open Genera release 2.0 is the name for Genera 8.5 when running on a Virtual Lisp Machine. It is a proper subset of Genera 8.5, i.e. all the functionality that is in both is the same, but there are a few Genera 8.5 features missing from Open Genera 2.0. The most notable of these are:

- support for direct manipulation of SCSI devices
- support for direct manipulation of serial devices
- features which require either SCSI or serial support:
  - SLIP
  - Dialnet
  - Direct Genera access to CD-ROM
  - Direct Genera access to Erasable Optical devices
  - Direct Genera access to tape
- FEPFS (Front End Processor File System)
- LMFS (Lisp Machine File System, which requires FEPFS)
- FSMaint Utility Program (which requires LMFS)
- FEP manipulation and information commands

Open Genera runs as a user-mode application under Digital Unix. This application emulates an entire standalone workstation which has its own network address(es), multiple processes and application programs. Beginning with Open Genera 2.0, more than one copy of this Unix application can be running simultaneously (given sufficient memory and paging resources), thereby permitting developing and testing distributed applications on a single machine.

CD-ROM Distribution Media

The Open Genera 2.0 CD-ROM contains the Alpha-side software, the Open Genera 2.0 worlds, Genera sources (in two forms), fonts, documentation and the following loadable systems (some of which were previously layered products):

```
Color (also prebuilt into an incremental world)
Images (also prebuilt into an incremental world)
CLIM (also prebuilt into an incremental world)
Hacks
Conversion Tools
Server-Utilities
Mailer
Image-Substrate
Concordia
Statice
Joshua
Fortran
Pascal
C
CL-HTTP (contributed by John Mallery)
Macsyma (contributed by Macsyma, Inc. and needs license key to operate)
NS (a VLSI design system, also needs a license key to operate)
```

Disk Space Requirements for Open Genera

Worlds:

/usr/lib/symbolics/Genera-8-5.vlod
/usr/lib/symbolics/inc-color-from-Genera-8-5.vlod
/usr/lib/symbolics/inc-clim-from-Genera-8-5.vlod
/usr/lib/symbolics/inc-color-clim-from-Genera-8-5.vlod

The first world is exactly like the normal distribution world of previous releases, except that it includes both version-control (a previously in-house system for source control) and has a complete who-calls database.

The other three worlds are incremental worlds based on the distribution world, containing either color, CLIM or both.

Sources and loadable systems:

There are three ways to use the source hierarchy that is shipped with Open Genera 2.0:

1. Mount the Open Genera 2.0 CD-ROM on the Alpha and use it as a SYS host using the Rock Ridge extensions.

2. Mount the Open Genera 2.0 CD-ROM on another Symbolics Lisp Machine, using the bare ISO-9660 format.

3. Restore the distribution tape image contained on the CD-ROM to some file server, either a LMFS on a Symbolics non-VLM host or a Unix NFS server.

- Using the Open Genera 2.0 CD-ROM as a SYS host requires (for the SYS:SITE; directory) either approximately 260 LMFS records if you put it on another Symbolics Lisp machine, or about 275,000 bytes if you put it on a Unix system running NFS.

- Restoring the Open Genera 2.0 sources and binaries in their entirety requires approximately 178,250 KBytes for UNIX.

Installing Open Genera

This section provides instructions for installing Open Genera 2.0, as well as special instructions for Open Genera 1.0, Beta I and Beta II customers.

Overview of Open Genera Installation Process

The installation of Open Genera consists of several steps:

- Installing the Genera application and its support files. This is done using the Digital Unix "setld" program which installs software subsets. The support files include the X-window fonts, the genera application itself, and a modification to the system's init files to allow the Virtual Lisp Machine to talk to the network. You need root privileges to install or remove subsets.

- Ensuring that Unix is configured correctly so that the genera application can run. This is mostly done by checking and/or editing configuration files by hand. Editing the configuration files requires root privilege.

- Genera installation and site configuration. These are the steps that you would perform initially on any Genera platform, not just a Virtual Lisp Machine. Site configuration means customizing the distributed image for your particular environment.

- The final step in Open Genera installation is to select a way to make the source hierarchy shipped with Genera accessible to its users.

Prerequisites to Installing Open Genera

- If you have installed any previous version of Open Genera (namely Open Genera 1.0, Open Genera Beta I or Open Genera Beta II), you must remove the older version before installing Open Genera 2.0. In the case of Open Genera Beta I (which didn't use the SETLD utility), you must do so manually by executing the instructions in the section "Upgrading From Open Genera Beta I to Open Genera 1.0 ".

- If you have installed Open Genera Beta II or Open Genera 1.0 on your Alpha, you must remove it by performing the following steps:

  ```
  Login as root

  setld -d VLMBASE100
  ```

- Ensure that you are running Digital Unix V3.2C or later. Open Genera 2.0 is not supported on any earlier releases of Digital Unix. Note that Digital Unix is year 2000 compliant only as of release 4.0D.

- We recommend a minimum of 200 megabytes of swap space. The disk partitioning provided with your Alpha will probably be sufficient. If this partitioning is insufficient, consult the *"Installation Guide"* manual for instructions on how to repartition your system.

- Check that the host name you will use for Open Genera is known to your Alpha, either through an NIS or BIND server, or by adding it to `/etc/hosts`. Refer to the DEC manuals *"Configuring Your Network Software"* and *"Network Administration and Problem Solving"* for details.

- Select or create a mount point for the Open Genera 2.0 CD-ROM, and check that the mount point is exported. (Refer to the DEC manual *"Network Administration and Problem Solving"* for further information on mount.) In the following instructions, we use the mount point `/cdrom` and we assume that `/dev/rz4c` is the CD-ROM drive.

How to Install Open Genera

Insert the Open Genera 2.0 CD-ROM into your Alpha's CD-ROM drive and perform the following steps:

1. Login as root.

2. Restore the Open Genera software from the CD-ROM as follows. (Note: User input text is in **bold**.)

   ```
   # mount -t cdfs -o noversion /dev/rz4c /cdrom
   # /cdrom/axp/osf/installgenera
   chmod: VLMBASE200.inv: Read-only file system
   chmod: VLMBASE200.scp: Read-only file system
   chmod: VLMBASE200.ctrl: Read-only file system
   chmod: VLMBASE200.scp: Read-only file system

   *** Enter subset selections ***

   The following subsets are mandatory and will be installed automatically
   unless you choose to exit without installing any subsets:
   ```

```
          * Open Genera 2.0 Base Software

    You may choose one of the following options:

          1) ALL of the above
          2) CANCEL selections and redisplay menus
          3) EXIT without installing any subsets

    Enter your choices or press RETURN to redisplay menus.

    Choices (for example, 1 2 4-6): 1

    You are installing the following mandatory subsets:

            Open Genera 2.0 Base Software

    You are installing the following optional subsets:

    Is this correct? (y/n): y

    Open Genera Base Software
       Copying from /tmp/VLM.980709 (disk)
            Working....Thu Jul 09 15:15:29 EDT 1993
       Verifying
            Working....Thu Jul 09 15:17:03 EDT 1993

        An Internet hostname is required for Genera.
        This name must be distinct from that of the OSF host which runs Genera.
      Please enter a hostname for Genera:
    YOUR-HOST-NAME

                You selected "YOUR-HOST-NAME" for the Genera hostname.
                Is this correct? (y/n)
    y

        If you need to change the hostname for Genera,
        edit the /sbin/init.d/SymbolicsVLM script.
```

3. **Optional:** If you will be running more than one VLM on the alpha host, you should edit the SymbolicsVLM script file to add an arp command for each additional VLM command. Each arp command should contain the same ethernet address (that of the host alpha) but should mention a different VLM. You might also want to add the VLMs' host names and IP addresses to the /etc/hosts file.

4. **Optional:** In order for the Open Genera system to run with a less pessimistic algorithm for reserving swap space, execute the following shell command:

   ```
   # rm /sbin/swapdefault
   ```

   (Digital Unix deals with swapspace in two modes. The default mode reserves all the space that a program might want to use when the program requests it. When reserved, as opposed to used, space exceeds 90% of swap space, warnings are issued. In the less pessimistic mode (called swap overcommit mode), only the space a program is actually using is reserved.

   If there is plenty of swap space, i.e. no chance that you will run out, there are performance advantages to running with the less pessimistic algorithm. However, if you do run out, i.e. all of a sudden enough programs increase their use to exceed the swapping available, Digital Unix just arbitrarily kills any process which needs more swap space than is available, which can cause system hangs or crashes if the process is crucial to the operation of the system, e.g. the X window system.)

5. Under certain situations you may have to rebuild the Digital Unix kernel. If you are missing packet filter support, you will have to rebuild the kernel. If you are running in a Digital Unix release before 4.0 and you have some virtual-memory related parameters set too low, you will have to rebuild the kernel. (After Digital Unix 4.0, you can set these parameters in the sysconfigtab file without having to rebuild the kernel.)

   If you are running an older version of Digital Unix, you might have to add packet filter support to the kernel. More recent versions (since about 3.2C) generally already have this support built in. (One quick way to see If you have packet filter support in the kernel is to attempt to run the `tcpdump` program. If it works it means that packet filter support is built into the kernel.)

   Edit `/sys/conf/HOSTNAME` (where HOSTNAME is the name of your Alpha workstation in uppercase) as follows:

   a. Add the lines

   ```
   options PACKETFILTER
   pseudo-device packetfilter
   ```

   to the file if they are not already present to enable DEC's packetfilter mechanism.

   b. If you are installing Open Genera for the first time into a version of Digital Unix before 4.0, add the lines

   ```
   vpagemax        524288
   mapentries      16384
   maxvas          4294967296
   ```

If you are installing into Digital Unix 4.0 or later and don't want to rebuild your kernel, you can get the same effect by adding the following to the end of the /etc/sysconfigtab file:

```
vm:

  vm-vpagemax = 524288
  vm-mapentries = 16384
  vm-maxvas = 4294967296
```

c. If you are upgrading from Beta I or Beta II and you intend to save incremental worlds, be sure that the value on the mapentries line is at least 1024.

d. If you did not make any changes to the file, skip to step 7.

e. Save the file.

6. Rebuild the Unix kernel:

```
# doconfig -c HOSTNAME
```

The following message will be displayed:

```
Do you wish to edit the configuration file [y/n]
```

Type n, which is the default.

Copy the new kernel to the root partition, after saving the old kernel.

```
# cp -p /vmunix /vmunix.save
# mv /sys/HOSTNAME/vmunix /vmunix
```

7. Reboot the system (in order to use the new kernel), and remount the Open Genera 2.0 CD-ROM (login as root after reboot):

```
# reboot
# mount -t cdfs -o noversion /dev/rz4c /cdrom
```

8. If the directory /dev/pf does not exist, you must create the packetfilter's special files and set their access appropriately.

```
# cd /dev
# ./MAKEDEV pfilt
# chmod a+w /dev/pf/*
```

9. If you are adding Virtual Lisp Machines to an existing site, use another Lisp machine at that site to create the namespace objects for both the Alpha host and the embedded Open Genera host, if you have not already done so.

The following example illustrates an entry for a typical Alpha host:

```
Name: ETA
Site: SCRC
Machine Type: DEC-3000-500-AXP
System Type: UNIX42
Address: INTERNET 128.81.41.49
Pretty Name: Eta
Console Location: SCRC 2 W223 x1038 OmniLab
Service: EXEC TCP EXEC
Service: EXPAND-MAIL-RECIPIENT TCP SMTP
Service: FILE TCP TCP-FTP
Service: FILE UDP NFS
Service: LOGIN TCP TELNET
Service: MAIL-TO-USER TCP SMTP
Service: RPC TCP TCP-RPC
Service: RPC UDP UDP-RPC
Service: SEND TCP TALK
Service: SHOW-USERS TCP ASCII-NAME
Service: TAPE TCP UNIX-REXEC
Service: TIME UDP TIME-SIMPLE-MSB
Service: UNIX-REXEC TCP UNIX-REXEC
Service: X-WINDOW-SYSTEM TCP X-WINDOW-SYSTEM
File Control Lifetime: 30 minutes
```

The following example illustrates an entry for its embedded Open Genera host:

```
Name: ETA-FISH
Site: SCRC
Nickname: EF
Machine Type: VLM
System Type: LISPM
Address: INTERNET 128.81.41.50
Address: CHAOS 24462
Pretty Name: Ate A Fish
Console Location: SCRC 2 W223 x1028 OmniLab
Printer: COMIC-NEWS
Bitmap Printer: COMIC-NEWS
Service: CONFIGURATION TCP CONFIGURATION
Service: LISPM-FINGER UDP LISPM-FINGER
Service: LOGIN TCP TELNET
Service: LOGIN TCP 3600-LOGIN
Service: NOTIFY UDP NOTIFY
Service: SEND TCP SMTP
Service: SEND TCP CONVERSE
Service: SHOW-USERS TCP ASCII-NAME
Service: TIME CHAOS-SIMPLE TIME-SIMPLE
Service: UPTIME UDP UPTIME-SIMPLE
File Control Lifetime: 15 minutes
User Property: EMBEDDED-IN ETA
```

Note: The correct `Machine Type` for an Open Genera host is `VLM`. The value of `User Property EMBEDDED-IN` is the name of the Alpha host that is running Open Genera. The Machine Type for the Alpha host must contain both the string "DEC" and either "AXP" or "ALPHA" in order for the remote tape facility to correctly identify that host as a Digital Unix Alpha host.

10. On another Lisp machine at your site, edit the SYS:SITE;SYS.TRANSLATIONS file. The following example assumes that you intend to run multiple releases of Genera at your site.

```
;;; -*- 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, 8.3, 8.4 Beta and 8.5
  (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>**>*.*.*"))))
       (("4")
       (fs:set-logical-pathname-host "SYS" :translations
        '(("sys:**;*.*.*" "ACME-SMBX:>rel-8-4>sys>**>*.*.*"))))
       (("4 Beta II")
       (fs:set-logical-pathname-host "SYS" :translations
        '(("sys:**;*.*.*" "ACME-SMBX:>rel-8-4-Beta-II>sys>**>*.*.*"))))
       (("5")
       (fs:set-logical-pathname-host "SYS" :translations
        '(("sys:**;*.*.*" "ACME-SMBX:>rel-8-5>sys>**>*.*.*"))))
       (otherwise (error "~A unknown minor version of Release 8.~2T
                            Edit SYS:SITE;SYS.TRANSLATIONS to add this system."
                           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))))
```

11. Now start up Open Genera by using the genera command. Read the section "The Genera Digital Unix Application" before proceeding with the following instructions.

    Type the genera command with the following options if you plan to save an incremental world (**Note:** We recommend that you plan to save an incremental world.):

    ```
    # genera -network OPEN-GENERA-HOSTNAME -ids yes
    ```

    If you plan to save a full site-configured world instead of an incremental, type

    ```
    # genera -network OPEN-GENERA-HOSTNAME
    ```

12. Set site.

13. Execute the Restore Distribution command as follows:

    ```
    Restore Distribution (keywords): Use Disk (for input [default Yes]) Yes
    ```

    At the pathname prompt, type:

    ```
    ALPHA-HOST-NAME:/cdrom/distribution/8_4_beta.source_dist.tape
    ```

    This will restore all systems. The source contents of the Open Genera 2.0 CD-ROM are listed in the section"Contents of the Open Genera 2.0 CD-ROM".

    If you want to selectively restore systems, type:

    ```
    Restore Distribution (keywords): Menu Yes
    ```

    Click on `Disk` and then click on `Pathname` and type:

    ```
    ALPHA-HOST-NAME:/cdrom/distribution/og2_source_dist.tape
    ```

    Select systems to be restored from the menu.

14. Shutdown Open Genera using the `Halt Genera` command:

    ```
    Halt Genera (keywords) :Query No
    ```

15. Start Open Genera again using the `genera` command, as in step 10.

16. Set site.

17. Load any systems you intend to include in your site world. A list of available loadable systems is located in the section "Contents of the Open Genera 2.0 CD-ROM".

18. Save a site-configured Open Genera 2.0 world. After you save your world, the message

    ```
    You must update your ivory config file to use this world
    ```

    will appear. , Create a .VLM file as described in the section "Genera Command". This file will be read the next time you start Open Genera.

Upgrading From Open Genera Beta I to Open Genera 1.0

If you installed Open Genera Beta I on your AXP, you must manually remove it before installing Open Genera 1.0. Execute the following steps to remove Beta I:

1. Login to the Alpha as root.

2. Remove the VLM command:

    ```
    # rm /usr/local/bin/VLM
    ```

3. If you created the directory /usr/local/bin/ during the installation of Beta I, remove it as well:

   ```
   # rmdir /usr/local/bin
   ```

4. Remove the Beta I world and related files:

   ```
    # rm -r /usr/local/ivory
   ```

5. Remove the Genera X fonts:

   ```
   # rm -r /usr/lib/X11/fonts/genera
   ```

6. Remove the startup file

   ```
   # rm /sbin/init.d/SymbolicsVLM
   ```

7. Remove the Beta I installation scripts and packet filter binaries

   ```
   # rm -r /usr/local/install
   # rm -r /usr/local/pfkit7
   ```

Contents of the Open Genera 2.0 CD-ROM

Open Genera Sources

The Open Genera 2.0 CD-ROM contains sources for the following systems:

- System
- CLOS
- RPC
- Embedding Support
- UX Support
- Development Utilities
- Old TV
- Zwei
- Utilities
- RPC Development
- UX Development

- Hardcopy
- Zmail
- Tape
- NSage
- Extended Help
- CL Developer
- Documentation Database
- IP-TCP
- IP-TCP Documentation
- CLX
- X Remote Screen
- X Documentation
- NFS-Client
- NFS Documentation

Open Genera Loadable Systems

The Open Genera 2.0 CD-ROM contains the binary files for the following loadable systems:

- MacIvory-Support
- CLIM
- Genera-CLIM
- CLX-CLIM
- Postscript-CLIM
- CLIM-Demo
- CLIM-Doc
- Conversion-Tools

- Hacks
- Examples
- Server-Utilities
- Mailer
- Domain-Name-Server
- Statice-Runtime
- Statice
- Statice-Server
- Statice-Documentation
- Concordia
- Metering
- Joshua
- Joshua-Metering
- Joshua-Doc
- Jericho
- C-Runtime
- C
- Fortran-Runtime
- Fortran
- Pascal-Runtime
- Pascal

In addition, the following pieces of contributed software are also directly loadable from the CD-ROM:

- From MIT, a pre-release development version of Common Lisp Hypermedia server which includes

  - ° CL-HTTP -- basic server
  - ° HTTP-BASE-CLIENT -- minimalist client
  - ° W4 -- web walker
  - ° HTTP-PROXY -- proxy support
  - ° LAMBDA-IR -- full text retrieval software

  In the SYS:CONTRIBUTED;CL-HTTP;DISTRIBUTIONS; directory are distributions of the last released version and the sources for other Lisp implementations.

- From Macsyma, Inc. version 421 of Macsyma, a symbolic math manipulation system. This requires a license to use. For details contact either Macsyma, Inc. (www.macsyma.com) or Symbolics, Inc.

- NS, a schematic, VLSI and PC board design system originally developed by Symbolics. This also requires a license to use, for details contact Symbolics, Inc.

Special Installation Situations for Open Genera

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

- Adding non-Symbolics computers to your site.

- Using a CD-ROM as a SYS host.

- Storing documentation, sources, examples, fonts, and non-loaded systems on a non-Symbolics computer.

- Upgrading from Open Genera Beta I or Beta II or Open Genera 1.0 to Open Genera 2.0.

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:

- a DEC Alpha workstation running Digital Unix 3.2 or later.
- a Sun-4 running SunOS 4.1.2 or later.
- a Symblics NXP1000
- a Symbolics Family machine running Genera 8.1.1 or later
- a Symbolics MacIvory running Genera 8.1.1 or later

Note: In order to serve as a SYS host, a non-Lisp machine CD-ROM host must support ISO9660 Rock Ridge format, which allows for Unix-style extended length names instead of just the ISO9660 names.

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 and not from the translations of the SYS logical host.

   If you CD-ROM drive is connected to a Lisp machine, you can do this by:

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

   If your CD-ROM drive is connected to a Sun or Alpha, this step must be executed in a Genera 8.3 or later world as follows (after mounting the CD-ROM properly to enable Rock Ridge):

   ```
   Copy File sun-or-alpha-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. (See below for examples.)

   After making the necessary changes, save the new sys.translations file. To use it, load it into your world. Once the updated translations are loaded, you should be able to access files on CD-ROM as easily as you would access files on any other file systems, including all the on-line documentation files.

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 Open Genera 2.0, 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 or Alpha, use the following example translations as a guide:

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

If some Symbolics machines at your site are running a release older than Genera 8.5, 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
   (8
   ;; Translations for Genera 8.3
     (selector minor-version equalp
        (("3")
         (fs:set-logical-pathname-host "SYS" :translations
           '(("sys:**;*.*.*" "ACME-SMBX:>rel-8-3>sys>**>*.*.*"))))
        (("4")
         (fs:set-logical-pathname-host "SYS" :translations
           '(("sys:**;*.*.*" "ACME-SMBX:>rel-8-4>sys>**>*.*.*"))))
        (("5")
         (fs:set-logical-pathname-host "SYS" :translations
               '(("sys:**;*.*.*" "HOST|CDROMn:>sys>**>*.*;*"))))
        (otherwise (error "~A unknown minor version of Genera 8.~2T
                              Edit SYS:SITE;SYS.TRANSLATIONS to add this system."
                            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 8.5 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 8.5):

```
;;; -*- 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-8-5>sys>**>*.*.*")))
```

You can also do the reverse, namely have some directories reside on disk (so that they can be modified) and have the rest come from the CD-ROM.

Known limitations and problems with using a CD-ROM mounted on a Lispm as SYS host

These comments don't apply if you are using the CD-ROM mounted on a Unix system which supports the Rock Ridge format.

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 subdirectories 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:

Default	*Input*	*Merged*
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 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 resistent 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*.

- On a DEC AXP, use the `umount` command to dismount ("unmount") the CD-ROM drive, then press the eject button on the front of the CD-ROM drive. Refer to your DEC OSF/1 Reference Manual for more information on the UNIX `umount` command.

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 80.

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, Document 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:

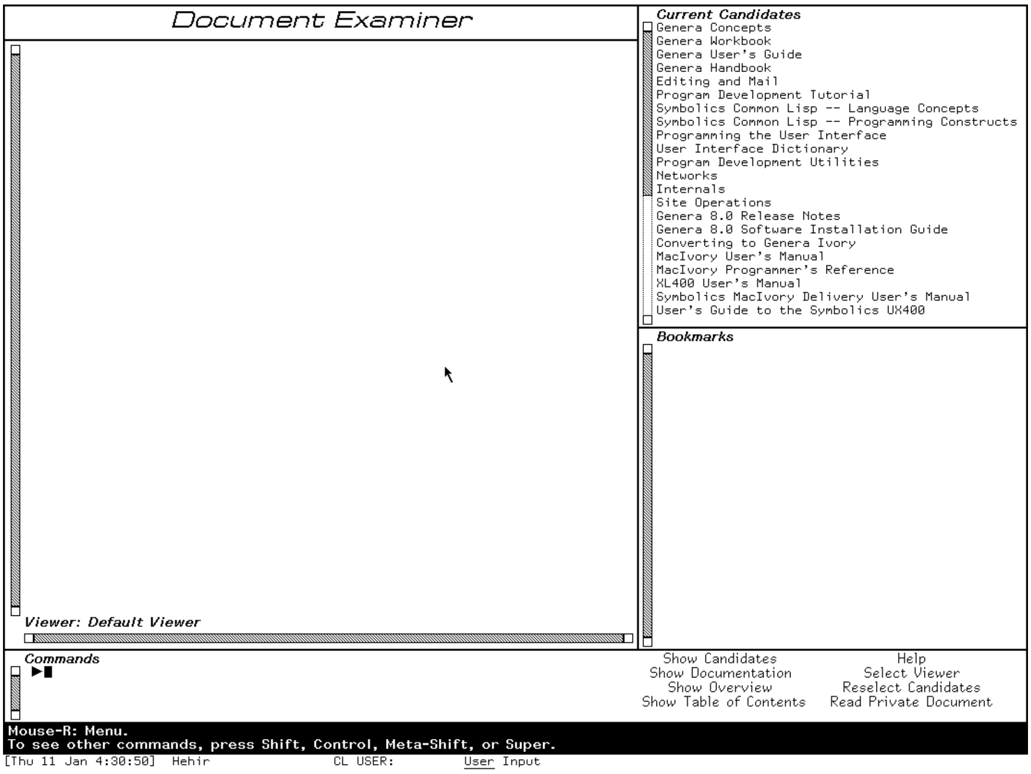
- Keyboard: Press SELECT D or type Select Activity Document Examiner at the Command Processor.

- Menu: Click on [Document Examiner] in the System menu.

The Document Examiner window contains the following panes, as displayed in Figure 114:

- *Viewer* — where the documentation is displayed

- *Current candidates* — list of current choices, usually resulting from a Show Candidates or Show Table of Contents command



Current Candidates

Genera Concepts
Genera Workbook
Genera User's Guide
Genera Handbook
Editing and Mail
Program Development Tutorial
Symbolics Common Lisp -- Language Concepts
Symbolics Common Lisp -- Programming Constructs
Programming the User Interface
User Interface Dictionary
Program Development Utilities
Networks
Internals
Site Operations
Genera 8.0 Release Notes
Genera 8.0 Software Installation Guide
Converting to Genera Ivory
MacIvory User's Manual
MacIvory Programmer's Reference
XL400 User's Manual
Symbolics MacIvory Delivery User's Manual
User's Guide to the Symbolics UX400

```
Mouse-R: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Thu 11 Jan 4:30:50]  Hehir              CL USER:        User Input
```

Figure 114. Upon selection, the Document Examiner displays all the books in the document set.

- *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.

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 s-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 s-R to to search backward and s-S 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 114. 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.

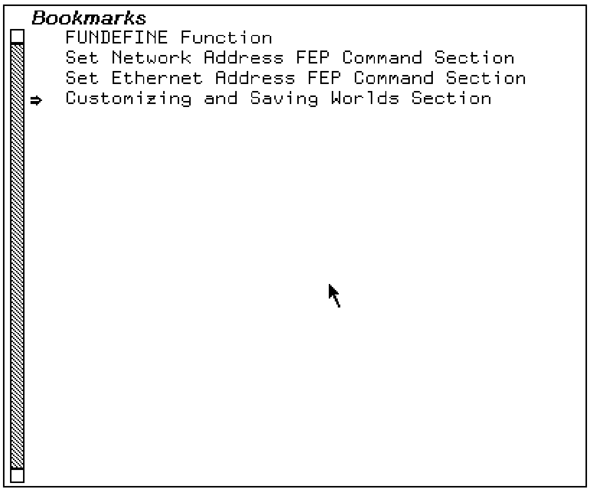
You can redisplay a list of candidates not currently showing by clicking on [Reselect Candidates] or using the Reselect Candidates command. You'll see a menu of all the documentation searches conducted during the current session. This list is helpful when, for instance, you need to cycle through several 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 81.



Bookmarks

```
   FUNDEFINE Function
   Set Network Address FEP Command Section
   Set Ethernet Address FEP Command Section
⇒  Customizing and Saving Worlds Section
```

Figure 115. 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 documentation 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 highlighed 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

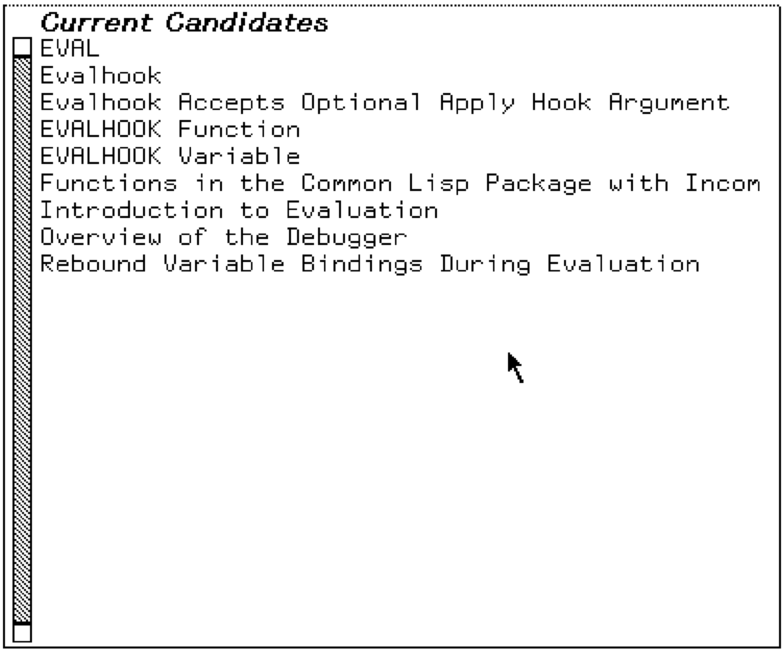
The lookup commands distinguish among like-named topics. For example, if you use Show Documentation for **evalhook**, Document Examiner asks you to select **evalhook** as a function or as a variable. See Figure 82. Similarly, the lookup commands distinguish between Zetalisp objects and those that are Common Lisp or Symbolics Common Lisp objects; Zetalisp objects are displayed with the **zl:** package prefix.

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.

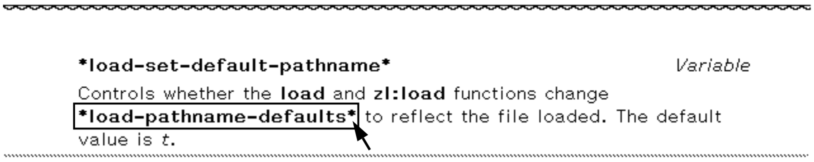


```
Current Candidates
EVAL
Evalhook
Evalhook Accepts Optional Apply Hook Argument
EVALHOOK Function
EVALHOOK Variable
Functions in the Common Lisp Package with Incom
Introduction to Evaluation
Overview of the Debugger
Rebound Variable Bindings During Evaluation
```

Figure 116. Document Examiner lets you choose among like-named topics.

Example: the topic ***load-set-default-pathname***, displayed in the viewer, mentions the variable ***load-pathname-defaults***, as shown in Figure 83. Clicking Left on the variable name displays the documentation for the topic "*load-pathname-defaults*". See Figure 83.

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



load-set-default-pathname *Variable*

Controls whether the **load** and **zl:load** functions change ***load-pathname-defaults*** to reflect the file loaded. The default value is *t*.

Figure 117. 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):

`m-sh-A`	Looks up the documentation for the current function.
`m-sh-V`	Looks up the documentation for the current variable.
`m-sh-F`	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 sh-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 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

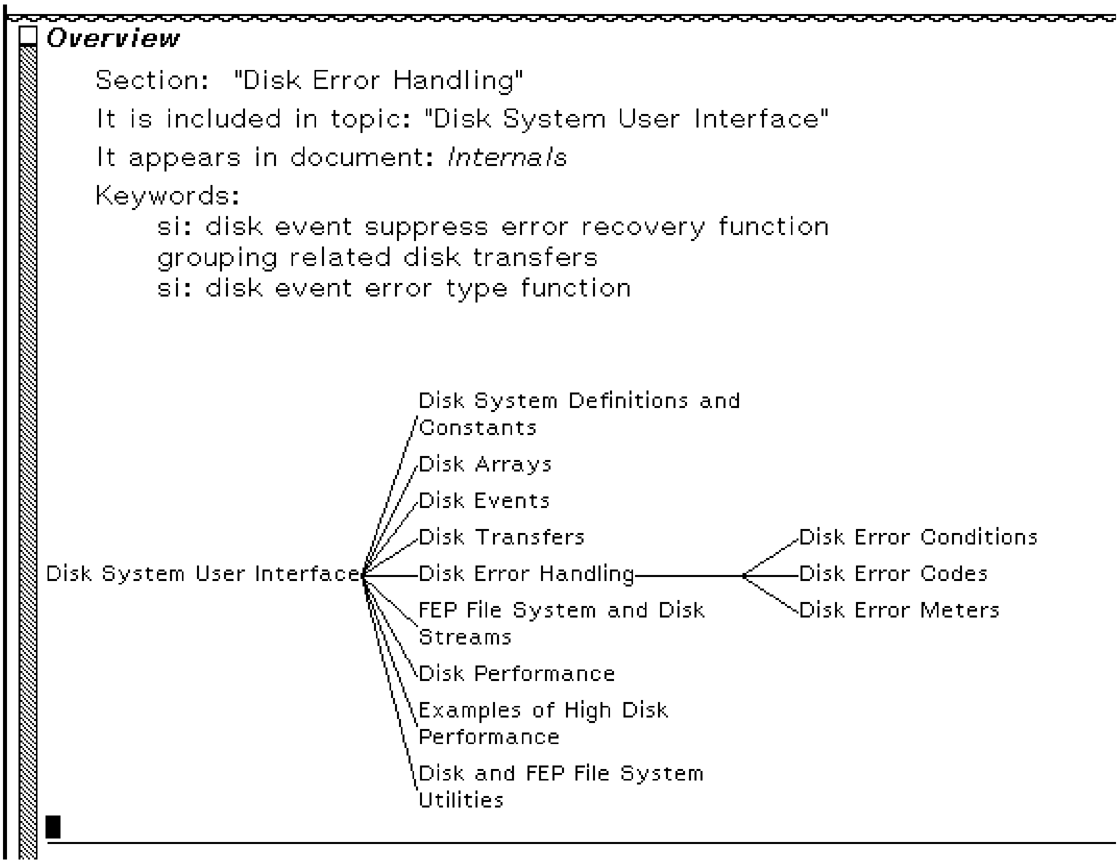
Figure 84shows the display produced by doing Show Overview of the topic "Disk Error Handling".
All the items in the overview graph are mouse-sensitive.

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 85.

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.



Overview

Section: "Disk Error Handling"

It is included in topic: "Disk System User Interface"

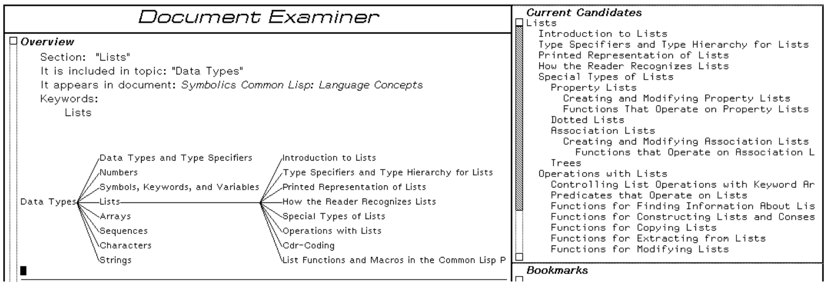
It appears in document: *Internals*

Keywords:

si: disk event suppress error recovery function
grouping related disk transfers
si: disk event error type function

Disk System User Interface
Disk System Definitions and Constants
Disk Arrays
Disk Events
Disk Transfers
Disk Error Handling
FEP File System and Disk Streams
Disk Performance
Examples of High Disk Performance
Disk and FEP File System Utilities
Disk Error Conditions
Disk Error Codes
Disk Error Meters

Figure 118. Document Examiner display of Show Overview of topic "Disk Error Handling".



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
Arrays
Sequences
Characters
Strings
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
Cdr-Coding
List Functions and Macros in the Common Lisp P

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 Conses
    Functions for Copying Lists
    Functions for Extracting from Lists
    Functions for Modifying Lists
```

Bookmarks

Figure 119. The candidates pane shows the table of contents for a topic; the type-out window shows the overview of the same topic.

Show Table Of Contents

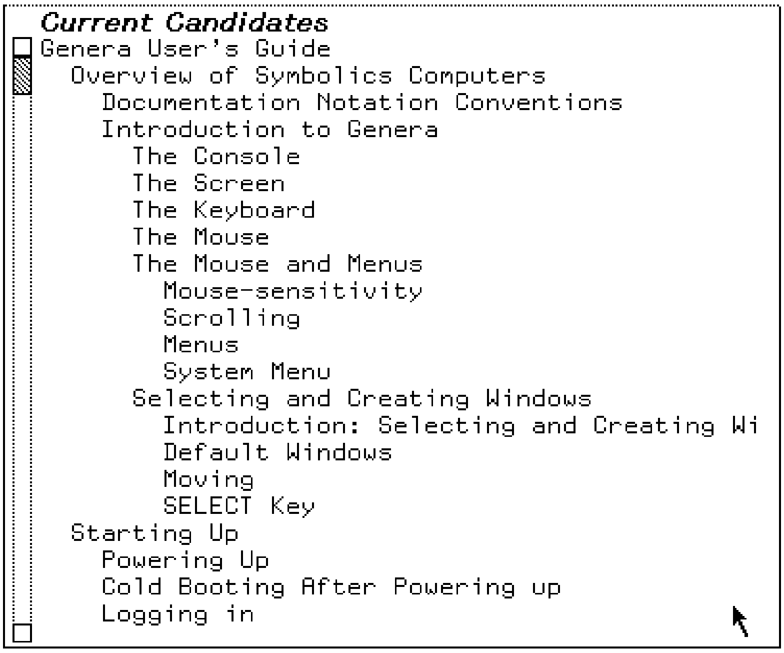
Much as with printed documentation, you might want to become familiar with the overall structure of an online document before reading about any one topic. When you examine documentation online, the more general the topic you look up, the larger the amount of documentation you see for it. The most general 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 "Genera 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:

- 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 86 shows the first pane of the table of contents of the document "Genera User's Guide".



```
Current Candidates
Genera User's Guide
  Overview of Symbolics Computers
    Documentation Notation Conventions
    Introduction to Genera
      The Console
      The Screen
      The Keyboard
      The Mouse
      The Mouse and Menus
        Mouse-sensitivity
        Scrolling
        Menus
        System Menu
      Selecting and Creating Windows
        Introduction: Selecting and Creating Wi
        Default Windows
        Moving
        SELECT Key
  Starting Up
    Powering Up
    Cold Booting After Powering up
    Logging in
```

Figure 120. 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 subnet. 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: *Genera Concepts* *Genera Workbook* *Genera User's Guide* *Genera Handbook*			
	cp;	hard;	lms;
	miscf;	miscu;	miscui
	nota;	sage;	user;

Book 5: *Editing and Mail*			
	conv;	fed;	zmacs;
	zmailc;	zmailt;	

Book 6: *Program Development Tutorial*			
	clyde;	tools;	workstyles;

Books 7, 8, and 9: *Symbolics Common Lisp -* *Language Concepts* *Programming Constructs* *Dictionary*			
	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: *Programming the User Interface* *User Interface Dictionary*			
	ined;	menus;	new-uims;
	uims;	scroll;	windoc;

Book 12: *Program Development Utilities*			
	comp;	debug;	maint;
	meter;	meter-int;	misct;

Book 13: *Networks*			
	netio;	nfile;	prot;

Book 14: *Internals*			
	audio;	fep;	instr;
	init;	int;	iprim;
	proc;	sched;	stor;
	tape;		

Site Administration			
	fep;	file;	fsed;
	pig;	site;	

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

Installation Guides:	
	sig;

Genera 8.0 Release Notes	
	rn8-0;

Setting and Defining Sites for Open Genera

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, or Open Genera:

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.

Sample Set Site Dialogue for Open Genera

Note: If you intend to execute :Set Site, do not log into your AXP as root. Log in with your usual userid instead.

```
Command: Login Lisp-Machine
Command: Set Site (site name [default Get from network]) test
Parameters for site TEST
Namespace Name: Test
Namespace Server Name: eta-fish  (the name of your Open Genera host)
Namespace Server Address: INTERNET A.B.C.D (the address of your Open Genera host)
Namespace Descriptor File: DIS-EMB-HOST:/var/lib/symbolics/sys.sct/site/test-namespace.text
Since the Namespace Descriptor File is on the Unix host (i.e., the DEC-AXP),
the name of the Unix host in the new site must be provided.
 Real Name of Unix Host: ETA  (the name of your local AXP)
 aborts,  uses these values
[16:20:21 Namespace on DIS-LOCAL-HOST: Reloading namespace TEST.
          Recent servers contacted are DIS-LOCAL-HOST]
The local host is now ETA-FISH.
[16:20:27 Namespace TEST has become unloaded:
          No longer server for this namespace.]
[16:20:43 Namespace on ETA-FISH: Reloading namespace TEST.
          Recent servers contacted are ETA-FISH]
```

Sample Define Site Dialogue for Open Genera

The following dialogue will set up a Symbolics namespace with the local Open Genera host as namespace server, with all namespace and system files stored in the local AXP filesystem. In the following example, the word "HOST" should be typed literally.

Note: If you intend to execute :Define Site, do not log into your AXP as root. Log in with your usual userid instead.

```
Command: Login Lisp-Machine
Command: Define Site (site name) test
Defining site TEST with the local host as the Primary Namespace Server
Namespace Server Name: ETA-FISH (the name of your Open Genera host)
System File Directory: HOST:/var/lib/symbolics/sys.sct/site/
Namespace Descriptor File: HOST:/var/lib/symbolics/sys.sct/site/test-namespace.text
Unix Host Name: ETA (the name of your local AXP)
Default Login: Lisp-Machine
Host for Bug Reports: ETA-FISH (the name of your Open Genera host)
Local Timezone: EDT
Standalone Site: Yes  No
 aborts,  uses these values
Created ETA:/var/lib/symbolics/sys.sct/site.
[13:46:25 Namespace on DIS-LOCAL-HOST: Reloading namespace TEST.
          Recent servers contacted are DIS-LOCAL-HOST]
The local host is now ETA-FISH.
[13:46:29 Namespace TEST has become unloaded:
          No longer server for this namespace.]
```

Note: Executing Define Site in Open Genera will create an Internet-only namespace. If you wish to use CHAOS to communicate with other Symbolics machines at your site, create a namespace object for the CHAOS network, and add a CHAOS address and CHAOS services to your Open Genera host object.

Sample Host Namespace Object for an Open Genera Host

Showing HOST ETA-FISH in namespace TEST:

```
Site: TEST
Machine Type: VLM
System Type: LISPM
Address: INTERNET A.B.C.D
Address: CHAOS 52525
Pretty Name: Eta-Fish
Service: BAND-TRANSFER TCP BAND-TRANSFER
Service: CHAOS-STATUS CHAOS-SIMPLE CHAOS-STATUS
Service: CONFIGURATION TCP CONFIGURATION
Service: CONFIGURATION CHAOS CONFIGURATION
Service: FILE CHAOS NFILE
Service: FILE CHAOS QFILE
Service: FILE TCP TCP-FTP
Service: FILE TCP NFILE
Service: LISPM-FINGER CHAOS-SIMPLE LISPM-FINGER
Service: LOGIN CHAOS TELNET
Service: LOGIN CHAOS SUPDUP
Service: LOGIN TCP 3600-LOGIN
Service: LOGIN CHAOS 3600-LOGIN
Service: LOGIN TCP TELNET
Service: NAMESPACE CHAOS NAMESPACE
Service: NAMESPACE TCP NAMESPACE
Service: NAMESPACE-TIMESTAMP UDP NAMESPACE-TIMESTAMP
Service: NAMESPACE-TIMESTAMP CHAOS-SIMPLE NAMESPACE-TIMESTAMP
Service: NOTIFY UDP NOTIFY
Service: NOTIFY CHAOS-SIMPLE NOTIFY
Service: SEND CHAOS SEND
Service: SEND CHAOS CONVERSE
Service: SEND TCP SMTP
Service: SHOW-USERS TCP ASCII-NAME
Service: SHOW-USERS CHAOS NAME
Service: TIME UDP TIME-SIMPLE-MSB
Service: TIME TCP TIME-MSB
Service: TIME CHAOS-SIMPLE TIME-SIMPLE
Service: UPTIME CHAOS-SIMPLE UPTIME-SIMPLE
Server Machine: Yes
File Control Lifetime: 30 minutes
User Property: EMBEDDED-IN ETA
```

Sample Host Namespace Object for a DEC AXP Host

Showing HOST ETA in namespace TEST:

```
Site: TEST
Machine Type: DEC-3000-500-AXP
System Type: UNIX42
Address: INTERNET A.B.C.D
Pretty Name: Eta
Service: EXEC TCP EXEC
Service: EXPAND-MAIL-RECIPIENT TCP SMTP
Service: FILE TCP TCP-FTP
Service: FILE UDP NFS
Service: LOGIN TCP TELNET
Service: MAIL-TO-USER TCP SMTP
Service: RPC TCP TCP-RPC
Service: RPC UDP UDP-RPC
Service: SEND TCP TALK
Service: SHOW-USERS TCP ASCII-NAME
Service: TAPE TCP UNIX-REXEC
Service: TIME UDP TIME-SIMPLE-MSB
Service: UNIX-REXEC TCP UNIX-REXEC
Service: X-WINDOW-SYSTEM TCP X-WINDOW-SYSTEM
File Control Lifetime: 30 minutes
```

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:

Time Zone	*Abbreviation*
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

Time Zone	*Abbreviation for Daylight Savings*
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.