

Quick Start Guide to Installing Open Genera 2.0

This guide's purpose is to help the reader start using Open Genera 2.0 to the point of being able to read the much more extensive on-line documentation database contained on the CD-ROM.

Introduction

Open Genera is an application program that runs under Digital Unix. It implements a Virtual Lisp Machine (VLM for short) which looks to the rest of the world as if it were a separate freestanding computer with its own name and network address. The first step is to install this application program (called *genera*) from the CD-ROM.

Once the *genera* program is installed, the second step is to check some Unix configuration options to see that they are compatible with the *genera* application and, if they are not, fix them.

The third step is to run the *genera* application; this will start up a virtual workstation running the Genera operating system.

At this point it is possible to read the on-line documentation contained on the CD-ROM. But for development use, the fourth step is to customize Genera to know about the local environment (a process called 'site configuration').

Finally, the fifth step is to use the Document Examiner (the Genera application which displays the documentation database) to consult the extensive on-line documentation provided with Open Genera. This guide explains the essential commands for doing this.

Installing Open Genera

Prerequisites

- Open Genera 2.0 CD-ROM
- DEC Alpha system running Digital Unix 3.2C or later with an available CD-ROM drive. This system must already be running in the local network environment, i.e. have its network addresses and names configured. (Unix configuration is beyond the scope of this document, but the appendix "Sample Digital Unix 4.0D Setup" lists the steps that were used to configure a particular out of the box Digital Unix 4.0 system for informational purposes.)
- Minimal familiarity with running Unix shell commands.
- Ability to login as (or su to) root (which means you need to know the root password).
- Ability to edit Unix configuration files (using whichever editor you prefer).

- Knowledge of the IP addresses for the DEC Alpha, the virtual lisp machine, any necessary gateways and any domain name or NIS servers.
- Knowledge of the device name and number for the CD-ROM drive (e.g. /dev/rz4c). See the section "Determining Which Device is the CD-ROM Under Digital Unix".
- A mount point for the CD-ROM (e.g. /cdrom); if none already exists, you can make one by typing


```
mkdir /cdrom
```

 while logged in as root.

Installation Steps

1. Install Genera Application
2. Configure Unix to support Genera
3. Run Genera in the Distribution Site

Detailed Instructions for Step 1: Install Genera Application

1. Insert the CD-ROM into the drive.
2. Logged in as (or su'ed to) root, type the following command (replacing the CD-ROM device name and mount point as appropriate):


```
/usr/sbin/mount -t cdfs -o noversion /dev/rz4c /cdrom
```

 It will just return to the shell without any messages if everything worked correctly.
3. Type the following command to run the installation script (which runs setld for you):


```
/cdrom/axp/osf/installgenera
```
4. When the setld programs asks which subsets to install, choose 1 (all subsets) since there is only one.
5. When the installation script asks for the name of the Virtual Lisp Machine, type in the host name which the virtual workstation running Genera will use. This name will normally also need to be put (along with its IP address) into your domain server's files or into the local /etc/hosts file.
6. If you plan to run more than one Virtual Lisp Machine on the Alpha host, you will have to add the names of the other hosts to the /sbin/init.d/SymbolicsVLM file by hand. Just duplicate the arp command line in that file for each additional VLM, replacing the original VLM's name with the additional VLM's name.

7. Unmount the CD-ROM by typing:

```
/usr/sbin/umount /cdrom
```

Detailed Instructions for Step 2: Configure Unix to Support Genera

There are three parts to configuring Unix to support genera:

1. Make sure that the various kernel configuration options are set to support genera.
2. Make sure that users who will be running Virtual Lisp Machines have read-write access to the packet filter devices.
3. Make sure all file systems needed by the Virtual Lisp Machines are exported to them.

Open Genera Kernel Options

1. Ensure that your kernel has the packet filter option turned on. If you are running a recent version of Digital Unix the chances are pretty good that this option is on by default.

One way to check is to run the `tcpdump` program. If that works, the packet filter option is on.

Another way to check is to look in the kernel configuration file. This is a file in the `/sys/conf` directory whose name is the usually the Unix host name all in capital letters (signified in the examples by "`<HOST>`"). If

```
grep -i packet /sys/conf/<HOST>
```

returns a line that says

```
options PACKETFILTER
```

then the kernel built from that configuration file has the `packetfilter` option turned on.

If the kernel doesn't have the `packetfilter` option enabled, you will have to add this line to the configuration file and rebuild the kernel. See the section "Rebuilding a Digital Unix Kernel".

2. Ensure that the limits for virtual memory of applications are adequate for running Open Genera. The parameters in question are

```
vpgemax
mapentries
maxvas
```

Before Digital Unix 4.0, you had to build these into the kernel using the

same configuration file as the packet filter option and any changes required a kernel rebuild.

Starting with Digital Unix 4.0 they can be included in the file `/etc/sysconfigtab` which is read in at boot time.

To set these parameters to their recommended values, you can edit them into the `/etc/sysconfigtab` file. If there is no section in that file headed "vm:", then you can just add the following to the end of the file:

```
vm:

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

If there is an already-existing section for virtual memory parameters, if any of the above parameters already exist, edit their values to be at least as large as the recommended values. If a parameter doesn't already exist, add it to the vm section.

Ensuring Access to the Packet Filter Devices

Firstly, you must make sure that the packet filter devices exist. Check this by typing

```
ls -l /dev/pf
```

If none are listed, you will have to make them while logged in as root.

```
cd /dev
./MAKEDEV pfilt
```

Secondly, you must make sure that the users who will be running Virtual Lisp Machines have write access to the packet filter devices. The easiest way to ensure this is to change the packet filters' group id to one that all VLM users will share and then give group write permission, e.g.

```
cd /dev/pf
chgrp users *
chmod 664 *
```

(In the above example, we used the existing system group called users. If there are many sorts of users of your Digital Unix it might make more sense to create a special group for VLM users and use that as the group for the packet filter devices).

Exporting File Systems to the VLMs

The Unix configuration file which controls exporting of file systems over NFS is `/etc/exports`. It is normally only editable while you are logged in as root.

You can check which file systems are exported to which hosts with the command

```
showmount -e
```

which produces a listing like

```
Exports list on localhost:
/cdrom                Everyone
/usr                  kilauea-vlm kilauea-2
/                     kilauea-vlm kilauea-2
```

Running this command has the additional effect that it signals the mount daemon to reread the `/etc/exports` file if it has changed.

The contents of `/etc/exports` which produces the above output is:

```
/ -access:kilauea-vlm:kilauea-2
/usr -access:kilauea-vlm:kilauea-2
/cdrom
```

For each file system to which you want to give VLM access, you'll have to

1. find the line starting out with that file system or add a new line for it if it doesn't already exist
2. add an `-access` entry to the line naming that file system, for example:

```
/myfs -access:foo:bar:vlm
```

which allows three hosts to access the `/myfs` file system, namely hosts `foo`, `bar` and `vlm`.

Detailed Instructions for Step 3: Run Genera in Distribution Site

Make sure that the Open Genera 2.0 CD-ROM is mounted in Rock Ridge format (which usually needs to be done as root), *viz.*

```
/usr/sbin/mount -t cdfs -o rrip /dev/rz?c /cdrom
```

This makes the source and documentation hierarchy accessible to Genera,

At this point, the simplest command to run `genera` is

```
genera -network <vlm-name>
```

or, if your network uses a non-default subnet mask (see "Non-default Subnet Masks" for more information)

```
genera -network "<vlm-name>;mask=xxx.yyy.zzz.aaa;gateway=bbb.ccc.ddd.eee"
```

A small iconified window will appear, then a large window. This window is the screen of the virtual lisp machine. It will display a fair amount of information on the screen, then provide a "command:" at the bottom of the initial window. You can type either commands or lisp expressions at this prompt.

Symbolics machines use a keyboard with many specially-labelled keys. There is a plastic overlay which identifies the mapping between those keys and the keys on the DEC Alpha PCXAL keyboard. If you don't have the overlay, the command

```
Show X Keyboard Mapping
```

will list all the key bindings.

When Genera starts up, you will be in a site called DISTRIBUTION which doesn't know about your local user, printer, and host names. So to login you should type

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

To tell the system to use the CD-ROM as the source for documentation, you should type

```
(fs::set-logical-pathname-host "SYS"
 :translations '(("**;*.*" "dis-emb-host:/cdrom/sys.sct/**/*.*")))
```

Finally, to check that this is working, type

```
(cons
```

followed by m-sh-A (this is read as "meta shift A" and means to hold down the meta and shift keys while typing the letter A). (On a PC-style keyboard, the meta key maps to the alt key.) The documentation for the lisp function CONS should appear in line. Type <clear-input> to get rid of the partial input. (On a PC-style keyboard, the <clear-input> key maps to F9.)

(The <help> key (which maps to the insert key on the PC-style keyboard) and which is available in all contexts where there is keyboard input, gives a lot of useful information for first-time users. In particular, while typing any command or command argument, the <help> key will show what is being asked for in that particular context. If the output from the <help> key includes choices, those are mouse sensitive and can be chosen merely by clicking on them.)

Congratulations, you have now reached the point of being able to run the Genera distribution world. Normally the next step is to make a customized world which contains site-specific information, but if this is your first time using Genera you might want to skip ahead to "Quick Guide to Reading On-line Documentation" in order to browse the overview documentation on Genera.

Site Configuration

Genera runs in an environment called a world. This contains a prebuilt image of many objects in the Lisp environment. You can load additional objects into the world or modify existing objects, and can then save out a new world which contains the modified set of objects.

One subset of the objects contained in the world relate to things specific to the site in which Genera is running, e.g. the name of the Genera host, the name of the Digital Unix host, the names of users, printers, print spooler hosts, etc.

When the distributed Genera starts up it is in a site called "DISTRIBUTION". This site knows only about certain hosts required to actually load the software when the world was built at Symbolics. The first task after a Genera is booted for the first time at a site is to create a world which is site-configured, i.e. which contains the appropriate information about the user's, as opposed to the DISTRIBUTION, site. Normally Genera users will boot this new, site-configured world instead of the distribution world. (You may also choose to load other software into this world, so that this other software is available immediately upon boot rather

than having to be reloaded each time. This quick guide doesn't cover doing that, but the full documentation does.)

The information about a site is contained in a namespace which has a name, e.g. SYMBOLICS and which is maintained by a namespace server host running Genera. Inside a namespace, there can be one or more named sites. If there is only one site, the convention is that it has the same name as the namespace. A site can be a standalone site, in which the Genera is both the namespace client and the namespace server, or it can have multiple hosts sharing a namespace, in which case one Genera host is the primary namespace server and all the others are clients of that primary namespace server.

Defining a new site

To create a new site, you run the Define Site command.

```
:define site <new-site-name>
```

When you hit <return> after this command, it expands in place for you to enter more parameters (in the following example the site name is "testing").

```
Command: :Define Site (site name) testing
Defining site TESTING with the local host as the Primary Namespace Server
Namespace Server Name: the name of the primary namespace server
aborts, uses these values
```

Click left on the italicized text ("the name of the primary namespace server") to be able to enter text into that field. Type in the name of the server (in this case "testing-server") and then type <return> to finish editing that field. The template expands once again for more parameters.

```
Command: :Define Site (site name) testing
Defining site TESTING with the local host as the Primary Namespace Server
Namespace Server Name: testing-server
System File Directory: HOST:/var/lib/symbolics/sys.sct/site/
Namespace Descriptor File: HOST:/var/lib/symbolics/sys.sct/site/testing-namespace.text
Unix Host Name: the name of the DEC-AXP host on which Open Genera is running
Default Login: Lisp-Machine
Host for Bug Reports: testing-server
Local Timezone: EDT
Standalone Site: Yes No
aborts, uses these values
```

If you want to change the value in a field starting with what's already there, click middle on the field. If you want to change the value but start with a blank field, click left. Once you are editing a field, simple Emacs commands work to move around in and change the field. When you are done editing the field, you finish by typing either <return> to accept your changes or <abort> to revert to the original value.

All the fields and values are described in the main documentation, I list only the ones you need might need to change here.

- **System File Directory:** this is where the site-specific configuration files go. It must be a directory that the user of the VLM has write access to. It can be any directory at all, the newly defined site will remember where it is.
- **Namespace Descriptor File:** this is the name of the file which holds the site information. It is usually in the system file directory, but need not be.
- **Unix Host Name:** this is the network name of the Alpha host machine in the new site.
- **Local Timezone:** the timezone in which this site resides. Defaults to the same timezone as the Alpha host.
- **Standalone Site:** hosts in a standalone site never send out a broadcast network message to verify their identity, hosts in a non-standalone site do.

Once all the fields have been filled in, type `<end>` to finish the define site command. This will write the site information files and switch you to that site.

You should then copy all the files in the site directory on the CD-ROM to the `SYS:SITE;` directory, e.g.

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

These files serve as a starting point for the newly-defined site. They contain translations for various logical hosts (e.g. `SYS`, pointing to the CD-ROM) as well as system definition and demo definition files.

After copying these files, you should boot the distribution world and then switch to the site you have just defined. This will make use of the new translation files. To do this, use the `SET SITE` Genera command instead of `DEFINE SITE`, specifying the same site name, host name, and location for the site files.

Using an existing site

To configure Open Genera to an existing site, you need to know:

1. The name of the site.
2. The host name of the namespace server for that site.
3. The network address of the namespace server for that site.
4. Whether or not your site uses non-default subnetting and, if so, what the subnet mask is and what the default gateway host address is. See the section "Non-default Subnet Masks".
5. If you are setting to a site for which the Virtual Lisp Machine is the namespace server and the site configuration files have non-default names and locations, you'll need to know those. (You will have set those using the "define site" Genera command.)

Armed with this knowledge, do the following:

1. Boot an Open Genera distribution world, using the subnet mask and gateway if you need to:

- If you are using default subnet mask:

```
genera -network <vlm-host-name>
```

- If you are using a non-default subnet mask:

```
genera -network "<vlm-host-name>;mask=x.y.z.a;gateway=b.c.d.e"
```

2. When Genera comes up type:

```
:set site <site-name>
```

to execute the command called "SET SITE". This command requires more parameters, so after you type the return to execute the command more output will appear in place, *viz*:

```
Command: :Set Site (site name [default Get from network]) <site-name>
```

```
Parameters for site <SITE-NAME>
```

```
Namespace Name: <SITE-NAME>
```

```
Namespace Server Name: the name of the primary namespace server
```

```
aborts, uses these values
```

If you move the mouse over the italicized text reading "the name of the primary namespace server", you'll notice that a highlight box appears around the text.

If you click left on the italicized text which reads "the name of the primary namespace server", you will then be editing that field. Type in the real name of the namespace server and type <return> to finish editing that field.

This will cause another field to appear in the output, *viz*:

```
Command: :Set Site (site name [default Get from network]) <site-name>
```

```
Parameters for site <SITE-NAME>
```

```
Namespace Name: <SITE-NAME>
```

```
Namespace Server Name: <site-namespace-host>
```

```
Namespace Server Address: a pair of a network and an address
```

```
aborts, uses these values
```

At this point we have two situations:

- a. The namespace server is another lisp machine somewhere on the net
- b. The namespace server is the Virtual Lisp Machine we are currently using

Namespace Server Elsewhere on the Network

Click left on the italicized text that reads "a pair of a network and an address" and type the word "INTERNET" (without the quotes) followed by a single space followed by the IP address of the namespace server host in the decimal dotted form, e.g. 128.81.41.1, followed by <return> to finish editing the field.

Finally, type <end> to execute the :SET SITE command.

Namespace Server is VLM

Click left on the italicized text that reads "a pair of a network and an address" and type the word "INTERNET" (without the quotes) followed by a single space followed by the IP address of this VLM in the decimal dotted form, e.g. 128.81.41.1, followed by <return> to finish editing the field.

This will cause another field to appear:

```
Command: :Set Site (site name [default Get from network]) test4
Parameters for site <SITE-NAME>
Namespace Name: <SITE-NAME>
Namespace Server Name: <site-namespace-host>
Namespace Server Address: INTERNET 128.81.41.1
Namespace Descriptor File: HOST:/var/lib/symbolics/sys.sct/site/<site-name>-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: *The real name of the Unix host in the site*
 aborts, uses these values

Fill in the host name for the Alpha host, and make sure that the namespace descriptor file agrees with what was used when the site was defined.

Finally, type <end> to execute the :SET SITE command

Saving a Site-Configured World

Once you have booted into a distribution world and issued the :SET SITE command to change to the local site, you can save that world into a new file by typing the command

```
Save World<space><space>
```

Immediately after typing the second <space> the default name will appear on the command line, looking something like

```
HOST:/usr/opt/VLM200/lib/symbolics/System-452-17.vlod.~↔~
```

You can either accept the default and finish the command by typing <return> (which maps to <enter> on PC keyboards) or edit the name using Emacs commands and then type <return>.

Once the world is saved, you can use it either by specifying it on the genera command line using the `-world` option,

```
genera -network foo-vlm -world /usr/lib/symbolics/foo.vlod
```

or by adding a line like the following to the `.VLM` file in your home directory

```
genera.world: /usr/lib/symbolics/foo.vlod
```

Reading On-line Documentation

The Genera application for searching and viewing documentation is called the Document Examiner.

You select the Document examiner by typing the two-key sequence `<select>D`. Once the document examiner starts up you will see four major areas on the screen (called panes):

1. A large pane on the left taking up most of the height of the screen where both the overview graphs of the documentation structure and the documentation text itself will appear.
2. A short pane underneath it for typing commands; this pane also has some menu items to the right.
3. A small pane at the top right labelled "Current Candidates".
4. A small pane at the bottom right labelled "Bookmarks".

The documentation database consists of a large number of documentation records, each of which is identified by a name and various keywords.

The Show Candidates command prompts for entering some words, and displays all the records which match those words (either in their names or their keywords) in the candidates pane. These candidate records are all mouse sensitive.

If there are more records than will fit into the pane, moving the mouse onto the shaded part of the scroll bar to the left of the record names will turn the mouse cursor into a double-headed arrow with an attached dotted horizontal line. Clicking left will move the candidate name nearest the dotted line to the top of the pane (i.e. scrolling down) and double clicking left (or holding down Shift while clicking left once) will move the marked line to the bottom of the pane (i.e. scrolling up).

The Show Overview command (also available by clicking middle on a record name) prints a graph of the immediate parents and children of the record. Each of the records in the graph is also mouse-sensitive, i.e. you can click middle on any of them to explore the hierarchy further. This is useful for determining if the record found by Show Candidates is in the right context or at the right level of detail.

The Show Documentation command (also available by clicking left on a record) begins displaying the contents of the documentation record. If scrolling is necessary, you can use the Scroll key to scroll forward, and m-Scroll (that is, holding down the meta key while typing the Scroll key) to scroll backward. For example, you

could get more help with using the Document examiner by typing

Show Documentation Document Examiner

Finally, one of the menu items to the right of the command line is "Reselect Candidates". Clicking right on this menu item pops up a menu of the previous sets of candidates including the initial set, which is the Table of Contents for the entire documentation database.

Some other documentation sections which might be useful starting points are

- *Genera Concepts*
- *Genera Workbook*
- *Symbolics Common Lisp Language Concepts*

Appendix 1: Procedure used to configure a Digital Unix system

This is a listing of the steps involved in configuring a particular Alpha to fit into an existing NIS networked environment. There is no explanation for the novice, i.e. this is just a checklist of what needed to be done in this instance. It is included here for informational purposes only. The customization of Unix for Open Genera is included in "Sample Open Genera 2.0 Installation"

This host was called Kilauea and the two Virtual Lisp Machines were called Kilauea-VLM and Kilauea-2.

1. At the console >>> prompt. Make sure auto_action is BOOT using
 - show | more
 - If it is not, set it
 - set auto_action boot
 - You might also set the ethernet link default mode to auto,
 - set ewa_0_mode auto
 - if appropriate. Login as root
2. Close the help and file viewer CDE windows which automatically appear.
3. Click above notepad with pencil icon, select terminal
4. Type
 - checklist
5. Click on network configuration application
6. Click on ethernet interface tu0
7. Click on configure
8. Fill in host name (kilauea)
 - Type in ip address (128.81.37.16)
 - Type in network mask (255.255.255.0)
 - Click on commit
 - Say yes to dialog about starting network services
 - Click on close
9. Click on Static Routes file
 - Click on configure
 - Select default for destination type
 - Type in IP for gateway (128.81.37.10)
 - Click on Add
 - Click on commit
 - Answer yes to query about applying routing changes now
 - Click on close

10. Click on host file (in configured section)
Click modify
Add host name for first vlm (kilauea-vlm)
Add host address (128.81.37.18)
Click add
Do same for second VLM (kilauea-2, 128.81.37.77)
Click commit
11. Click close
12. Select Exit from the file menu
13. Click on Bind configuration from the checklist
Click on client
Click on configure
Type in local domain (ai.mit.edu)
Click on the Name server's host name
Type in life
Type in name server's address (128.81.37.21)
Click on Add before
Click yes to adding info to /etc/hosts
Click on commit
Click OK to dialog about setting system name
If you get a dialog asking about adding localhost to X access control list, click
Click on close
Select Exit from the file menu
14. Click on NIS in checklist
Type c to continue
Press return to continue
Press return to continue
Type in NIS domain name (ai.mit.edu)
Confirm that it is correct (type y).
Choose 3 to be a client
Type c to continue
Type n to not use a secure server
Type y to use S
Type in names of NIS servers
Type c to continue
Choose 3 to diallow ypset requests
Type y to confirm
Type y to use all databases
Type y to start nis daemons now
Type return to exit
15. Click on NFS configuration Application in checklist
Select NFS Client setup
Click on configure

Select configure for automount button
Click on commit
Say yes to making the changes take effect now.
Click on close
Click on NFS server setup
Select "allow nonroot mounts"
Click on commit
Click yes to have changes take effect now
Click on close
Select exit from file menu

16. Select Graphical UI Selection Facility Type xdm to select xdm (instead of cde)
Type y for is this correct Type y for do you want to start xdm now Type y
again for notification that all sessions will be terminated

At this point the Digital Unix is minimally configured to allow users to login.

Appendix 2: Procedure used to install Open Genera

1. Login as root
2. `mkdir /cdrom`
3. `/usr/sbin/mount -t cdfs -o noversion /dev/rz?c /cdrom`
 ? is the device number of the CD-ROM, typically 4-6. See the section "Determining Which Device is the CD-ROM Under Digital Unix".
4. `/cdrom/axp/osf/installgenera`
5. 1 to choose all subsets
6. y for is this correct
7. type kilauea-vlm for hostname for Genera type y for is this correct
8. Add the following to the end of the `/etc/sysconfigtab` file

```
vm:

vm-vpagemax = 524288
vm-mapentries = 16384
vm-maxvas = 4294967296
```
9. Make packet filter devices if not already present:

```
cd /dev
./MAKEDEV pfilt
cd pf
chgrp lispn *
chmod 664 *
```


 This example assumes all users which need to run VLMs are members of the group called `lispn` which already existed.
10. `/usr/sbin/umount /cdrom`
11. `/usr/sbin/mount -t cdfs -o rrip /dev/rz?a /cdrom`
 This remounts the CD-ROM in Rock Ridge format, suitable for being accessed as SYS host by Open Genera.
12. Logout as root
13. Login as normal user

14. The minimal command to run `genera` at this point is:

```
genera -network <vlm-host-name>
```

which will load the default world (`Genera-8-5.vlod` in the `/usr/opt/VLM200/lib/symbolics/` directory).

If your network is more complicated, you might have to use the extended form of the `-network` option, or explicitly name the world, amount of virtual memory to allocate, window geometry, etc. Here's a typical example:

```
genera -network "internet|128.81.xx.yy;mask=255.255.255.0;gateway=128.81.xx.10"
      -world <name of world file including .vlod>
      -vm <number of megabytes, min and default 200>
      -geometry 1280x1024-0-0
```

but that is pretty inconvenient to type.

Usually, you create `.VLM` file to contain the defaults or you make a Unix script file to issue a command like the above with parameter substitution.

15. You could create a `.VLM` file in your home directory which looks approximately like this:

```
genera.virtualMemory: 500
genera.main.geometry: 1280x1022-0-0
genera.world: /usr/opt/VLM200/lib/symbolics/Genera-8-5.vlod
genera.network: internet|xxx.yyy.zzz.nnn;mask=255.255.255.0;gateway=xxx.yyy.zzz.mmm
```

This file provides defaults; you can always override them by specifying the command parameters with the `genera` command.

The geometry parameter specifies a window that is the full size of the screen (1280x1024). Because X-windows usually puts a border around the window, if you make a full size window you must manually move the top corner of the screen to make the border as small as possible.

There are many more parameters described in the full documentation.

16. You could also create a Unix command file. The following is a short example

```
#!/bin/ksh
WORLD=${1:-`echo /usr/opt/VLM200/lib/symbolics/Genera-8-5-local.vlod`}
VM=${2:-500}
genera -network "internet|128.81.37.81;mask=255.255.255.0;gateway=128.81.37.10" -world $WORLD -
geometry 1280x1022-0-0 -ids yes -vm $VM
```

This command file uses the Korn shell and accepts two parameters, the name of a world and the number of megabytes of virtual memory. The first defaults to a world called `Genera-8-5-local.vlod` and the second defaults to 500.

Appendix 3: Determine the device name of the CD-ROM drive

If you don't know which device is the CD-ROM drive, you can figure it out in one of two ways:

1. The first way is with the `file` command, e.g. logged in as (or `su`'ed to) root type:

```
file /dev/r*a | grep RRD
```

If that doesn't give the required information, try the command without piping it to `grep`. That will list out all the character devices, one might stand out as different from the disk devices and is therefore likely to be the CD-ROM drive. On my system:

```
# file /dev/r*a | grep RRD
/dev/rrz4a:    character special (8/4096) SCSI #0 RRD45 disk #32 (SCSI ID #4)
```

2. A less convenient second way is to type the following at the console (`>>>`) prompt (either before booting or after halting the Alpha):

```
show device
```

This will list all the SCSI devices; usually the CD-ROM drive is identifiable from the manufacturer's model string. Then, after booting Unix, you can use the `file` command to determine which device corresponds to that SCSI id on that particular SCSI bus.

Appendix 4: Rebuilding the kernel

The procedure for rebuilding and installing a Digital Unix kernel is

```
cd /sys/conf
```

Edit the configuration file (usually the same name as the system, but in all capital letters).

```
doconfig -c <HOST>
```

This will build a new kernel and tell you where it put it, usually a directory with the system name under /sys/.

Rename the old kernel

```
mv /vmunix /vmunix.saved
```

and copy the newly built kernel kernel to the default place

```
cp /sys/<HOST>/vmunix /vmunix
```