TN 96

Interactive
Media
Systems

CD-I

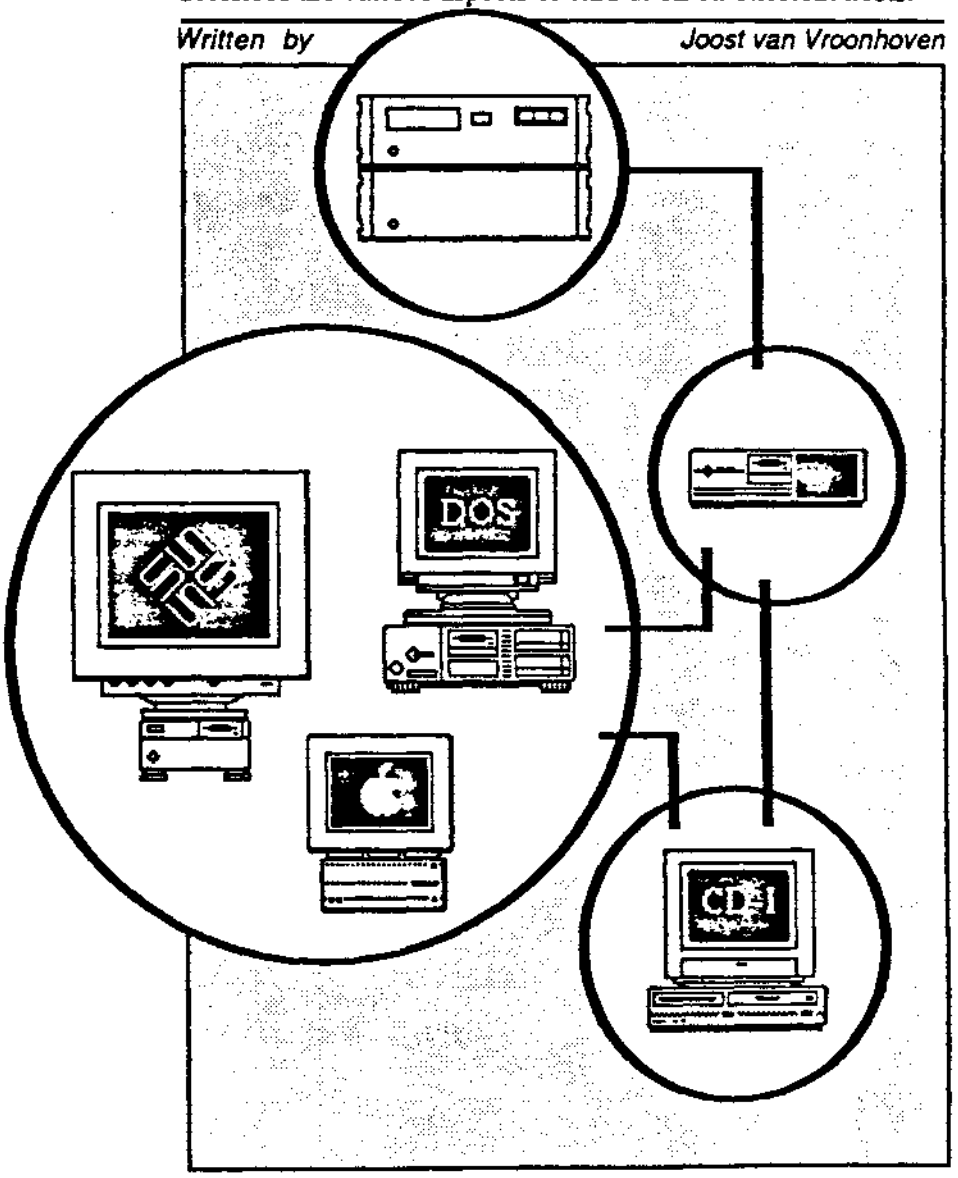



TSA APPLICATION NOTES NR. TSA-006 Oct 14-1992

Emulator & Emulation

Creating a CD-I application involves collecting and encoding of assets and the writing of application software to implement the interactivity. After pre-mastering the CD-I disc image can be tested on a CD-I player connected to an emulator. This note describes the various aspects of emulation on different hosts.

Written by *Joost van Vroonhoven*



Philips Consumer Electronics B.V. Interactive Media Systems Bld. SFH - 6 P.O. Box 80002 5600 JB EINDHOVEN Tel. +31 40 3540 ...

copyright 1992 IMS TSA nr. of pages: 70

PHILIPS

Introduction 4
General setup 5

Connecting emulator and player 6
Connecting emulator and Philips CDI 180 player . . . 6
Connecting emulator and Philips CDI 605 player . . . 8
Tools menu 8
Internal / external CD drive 9
Expansion memory 9
Video randomize and previous 10
Seek delay 10

Sun4/SPARC and Sun3 workstations 11
Hardware connections 11
SCSI connection 11
Changing SCSI-id's internally 14
SCSI id conflicts 17
Serial connection 18
System software installation 20
Software installation 25
Emulation 27
Burning WORMs 29
Creating CD-I image file tapes 30

386/486 PC systems 31
Hardware connections 31
SCSI connection 31
Serial connection 32
System software installation 34
Software installation 35
Emulation 36
Creating CD-I image file tapes 37

Apple Macintosh II or Quadra systems 38
Hardware connections 38
SCSI connection 38
Serial connection 39
System software installation 40
Software installation 41
Emulation 41
Creating CD-I image file tapes 42

Philips/OptImage OS9 emulator systems 44
Script-to-Disc 44
Hardware connection 44
SCSI connection 46
Terminal connection 46
System software installation 49
Discbuilding 49
Talk-to-Disc 56
Emulation 58
Creating CD-I image file tapes 60

Yamaha Programmable Disc System 61
Hardware installation 61
Burning WORM's 63
Error codes 66

Introduction

Development of applications for Compact Disc Interactive (CD-I) involves collecting and encoding audio and video assets and writing a program which implements the interactivity. The encoded assets and the executable program are then combined into a CD-I disc image file which contains all data as it will appear on the production disc.

After creating (pre-mastering) the CD-I disc image file on a development host with the "master" utility one can test this disc image on an Authoring player connected to the Philips / OptImage emulator. The emulator system can best be described as a harddisk based CD-drive imitator. It intercepts the read requests the CD-I player sends to its CD drive and sends the requested data from the disc image file stored on one of the development hosts harddisks. The CD-I player will receive the data at exactly the same speed as when this data is send from a normal CD drive. A trace window allows you to monitor the CD accesses, which could be useful for optimization of the disc layout.

The emulator allows you to test the disc image before committing it to mass-production. Additionally the emulator can be connected to a WORM (Write-Once-Read-Many) system to "burn" a sample disc playable on any CD-I player.

Currently supported development hosts are Sun3 and Sun4/SPARC workstations, Apple Macintosh II and Quadra series, 386 and 486 PC systems and the emulator itself. Supported CD-I players are the Philips Authoring players CDI 180 and CDI 605.

All versions of the Philips/OptImage emulator are available with or without a build-in harddisk drive and with or without a Yamaha PDS interface.
The harddisk normally is a CDC Wren VI (aka Seagate ST41766N) of 630 MB formatted capacity or a Hewlett-Packard HP97549S of 920 MB formatted capacity. It is possible to use most harddisks supported by the development host however.

The emulator is a computer system with a Motorola MC68020 CPU running at 16 MHz, 2 MB of memory, four serial ports, a parallel port, a floppy disk drive of 650 Kb capacity, all on the motherboard. Additionally it contains a fast SCSI controller on an expansion board and an interface to the CD-I player and the optional Yamaha PDS interface also on an expansion board. It is possible to add an Ethernet interface to the emulator as well, for special applications. The OS-9 operating system boots up from the floppy disk provided, loading the emulation software into memory.

The emulator's OS-9 harddisk device driver, as supplied with versions of the emulation software prior to version 2.0, cannot address data beyond the first 1 GB of the disks capacity. So please upgrade to the latest possible version, if you intend to use such a harddisk. Otherwise if one uses a harddisk with a formatted capacity of more than 1 GB, one should partition the disk into at least two partitions. The first partition may be up to 1 GB in size and may be used for emulation, while the second partition contains the rest of the drive's capacity but cannot be used for emulation.

General setup

What follows is a drawing of the connections used between the various devices connected with the development host when set up for emulation of CD-I disc images.

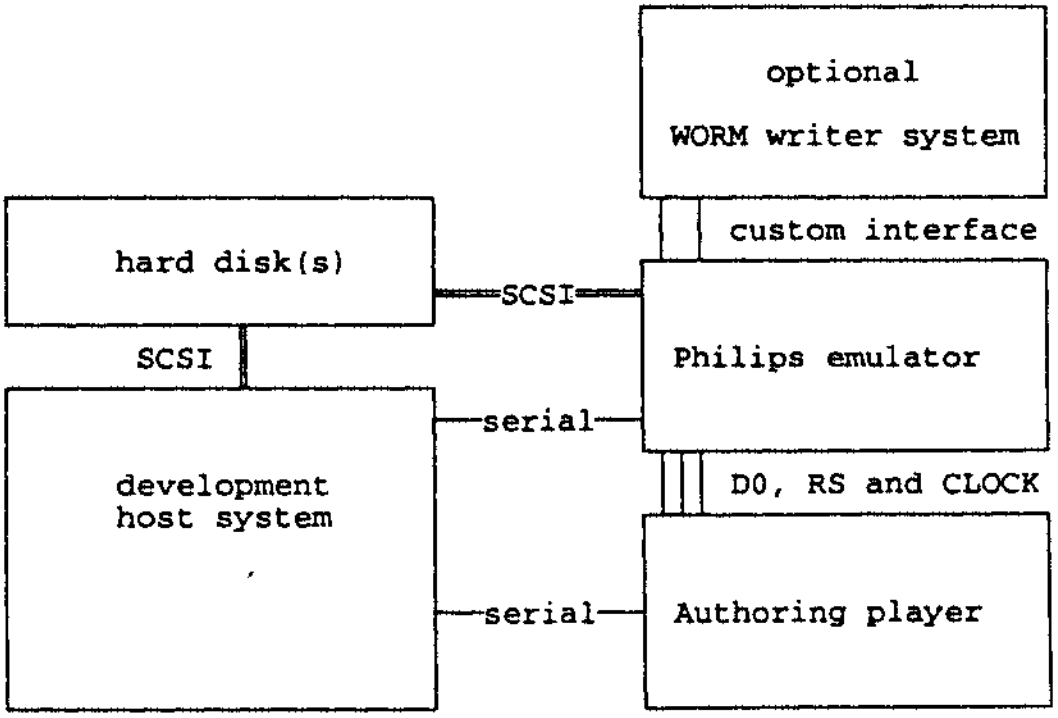


figure 1: General setup

The development host has at least one SCSI harddisk also connected to the emulator (the physical location of this disk is irrelevant). The emulator is also connected to the host via a serial connection, to the authoring player via the CD drive control lines D0, RS and CLOCK (180 only) and to the optional WORM system via a custom interface. The player also has a serial connection to the development host.

Connecting emulator and player

The emulator has to be connected to an authoring player which allows you to replace its CD drive. You may use either the Philips CDI 180 player or the Philips CDI 605 player.

Connecting emulator and Philips CDI 180 player

The CDI 180/181 player is the original CDI player and can be connected to the optional 182 expansion module which contains 1 MB of expansion (system) memory, SCSI controller, serial port, parallel port, optional thin-wire ethernet interface and two floppy disk drives.

The CD drive of the CDI 180 player (the 180 CDI module) is one of (up to) three separate modules connected to the rest of the system (the 181 MMC module and optional 182 EXPANSION module) by three cables called RS (the serial control connection named after the serial RS232C standard), D0 (the data out line) and CLOCK (the clock pulse line).
Instead of connecting the CD module directly to the MMC module we can put the emulator inbetween by using two triple cables: one from CD module to emulator and one from emulator to MMC module. Since the D0 connection will transfer all data it is recommended to use the thick, shielded cable with the white connectors for D0, the thin unshielded cable with the red connectors for CLOCK and the other thin unshielded cable with the yellow connectors for RS. See figure 2.

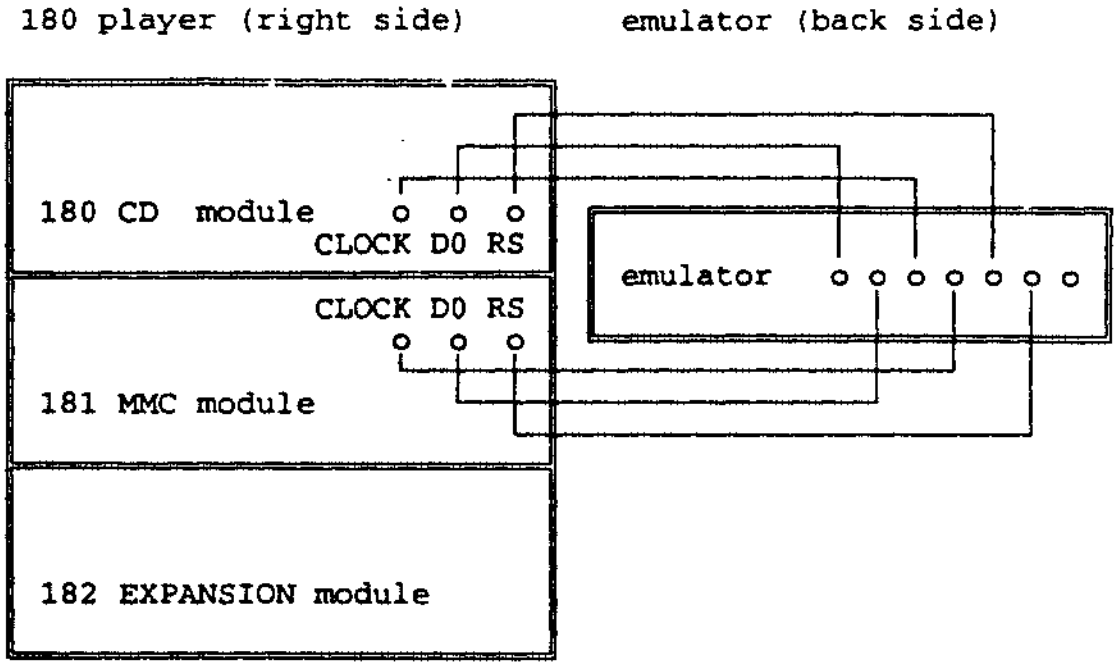


figure 2 : emulator and 180 player

To run an application from the emulator just start emulation from your host, reset the CDI 180 player and click on the CD-I button. The emulator will disconnect the CD module and connect itself in its place with a magnetic switch. To stop emulating, reset the player and stop the emulation process on your host.

Connecting emulator and Philips CDI 605 player

The Philips CDI 605 player, which is very similar to the consumer players (CDI 910 and CDI 205) contains a floppy disk drive, two serial ports, 4 MB of expansion memory, SCSI controller, parallel port and a thinwire ethernet interface.

The CD drive of the Philips CDI 605 player is built into the system enclosure, but the connectors for the emulator are on the back panel and special control hard- and software is built into the CDI 605 in order to enable emulation.

Only two cables are needed, the D0 (the data out line) and the RS (the serial control connection named after the serial RS232C standard), it is NOT necessary to connect the CLOCK (the clock pulse line), since the CLOCK signal is not connected inside the 605 player. Production 605 players have this connector relabelled to N.C. for Not Connected.
Since the D0 connection will transfer all data it is recommended to use the thicker, shielded cable with the white connectors for D0 and the thinner unshielded cable with the yellow connectors for RS. The cable with the red connectors is unused, as mentioned before. See figure 3.

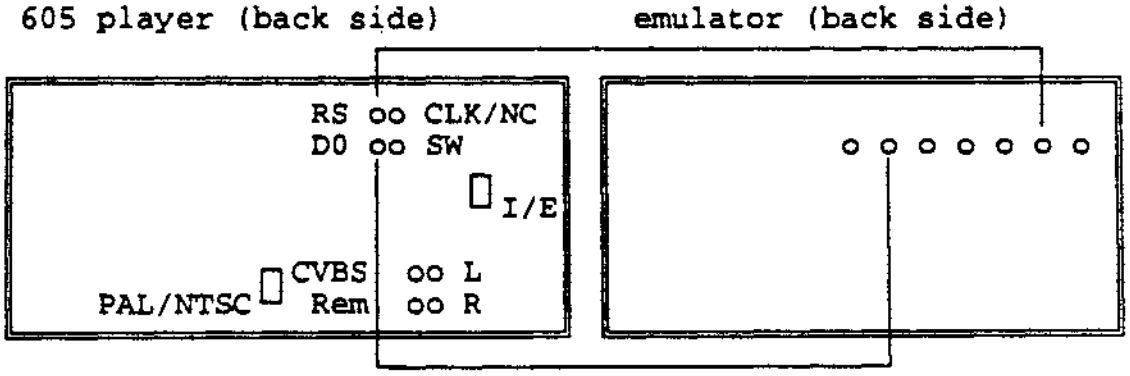


figure 3 : emulator and 605 player

Tools menu

The Philips CDI 605 is the authoring player and as such it offers special options for use in the authoring environment. Select the TOOLS button in the lower right corner of the player shell's main screen to enter the tools menu.

Internal / external CD drive

In order to support emulation the CDI 605 allows you to select its internal CD drive or an external drive, normally an emulator. There are three controls that allow you to switch between an internal and external source :

- the red SWITCH input jack next to the D0 and RS connectors on the back of the CDI 605 (marked SW in fig.3).
- the hardware internal/external switch next to the connectors D0, RS, CLOCK and SWITCH (marked I/E in fig.3).
- the software internal/external switch in the TOOLS menu of the player shell.

The SWITCH input jack (cinch type) always takes priority over the other settings, but is inoperative if unconnected. When the signal into the jack is low (GND) the source is internal and when high (+5V) the source is external.

The software switch determines the source while the player shell is active, while the hardware switch determines the source while the OS9 shell is active.

As soon as a CDI application is running the last setting from either the software or hardware switch is used, depending on how the application was started, either from the player shell or the OS9 shell.

So normally you would should leave the input jack unconnected and you may start your emulation from your host, select external in the player shell tools menu and click START on the main player shell screen. To run an application from a disc select internal, wait for the CD-I disc to be recognised and click START.

Expansion memory

Additionally the CDI 605 allows you to select the amount of expansion memory (aka. system memory) available to applications. The amount may vary between 0.0 and 4.0 MB in 0.5 MB steps. The amount currently available is displayed just above the tools menu button on the player shell's main screen. This concerns expansion memory only, the single MB of base case coloured video memory is always available to applications.

When the amount of expansion memory is set to 0.0, your application will run with a minimal amount of free memory, with both planes having 480 Kb of free video memory, as specified in the Green Book as worst-case base case.

Changing the setting for memory will result in a restart of the player. A restart will take longer than normal if the setting is less than 4.0, since the unused part of the expansion memory is treated as if it were ROM, so it must be searched for OS9 modules. You can use this feature to put your favourite utilities or system modules (for example the ethernet descriptor le0 and ethernet database inetdb) in memory so they are readily available after a player restart. To set it up:

- Set the amount of expansion memory to 4.0.
- Start up the OS9 shell, by clicking on SETTINGS and SYSTEM.
- Load the modules into memory:
$ load dir copy mdir
- Restart the player shell:
$ play >>>/nil
- Reduce the amount of expansion memory by at least as much as your loaded modules need.
- Restart the player.

These loaded modules will of course be lost if you power off the player.

Video randomize and previous

The player shell sets up the CDI player's video hardware to sensible values, since it requires the video itself. If your applications are only tested when started from this environment you may run into problems on other players. To help you test for incorrectly initialisation you can enable RANDOMIZE. This setting instructs the player shell to enter random values into the video, audio, pointer, cursor and timer settings of the player and to save those settings in non-volatile ram as /nvr/vid_regs.prf . Should you encounter a potential problem when running, an application in this environment you can then re-test with the same settings by enabling PREVIOUS (automatically disabling RANDOMIZE), which instructs the player to load the values from the file in NVR.

Seek delay

The Green Book specifies a maximum seek time for CD seek operations. Recent CD drives are much faster, but you may enable SEEK DELAY to force your application to wait the maximum of time, which is 1 second for seeks within 20 MB of the current position and up to 3 seconds for a full seek across the CD disc. See also the Green Book chapter VIII-24 and the "Technical Documentation for CDI 605 users" supplied with your 605 player.

Sun4/SPARC and Sun3 workstations

The Sun version of the Philips/OptImage emulator comes with pre-mastering software for both Sun SPARC and Sun-3 systems running SunOS 4.1.1 (Solaris 1.0.1) and higher and a small example application on 1/4" streamer tape (this is the same material used for the sample Balboa application described in IMS TSA Technical Note number 1). It can be connected to a Sun SPARCstation such as the SS1, SS1+, SS2, SLC, ELC, IPC, IPX and the new SS10 or a Sun-3 workstation such as the 3/50, 3/60 and 3/80. It could also be connected to a Sun3 or Sun4 server, (3/280, 4/330, 4/630 etc.) although some of the details such as device names may differ and we do not recommend connecting an emulator to a server machine.

Hardware connections

The emulator has to be connected to your Sun workstation through both the SCSI bus and a serial port.

SCSI connection

The emulator has a so-called Centronics SCSI connector located in the top right corner (while looking at the back of the emulator) and is supplied with a cable with a Centronics SCSI and a mini-SCSI connector as used by Sun4 systems or a 50-pin D-shell connector as used by Sun3 systems.

To connect the emulator to your Sun SPARCstation (aka Sun4):
If you have a diskless workstation you should locate the mini-SCSI connector on the back of your machine and connect the Emulator with the supplied SCSI cable.
If you only have SCSI peripherals connected which use the mini-SCSI connector you need to remove the SCSI terminator from the last SCSI device in the chain and connect the Emulator in its place. The Emulator must be the last device in the SCSI chain since it has a build in SCSI terminator and only a single external SCSI connector.
If you have SCSI peripherals connected which use the 50 pin D-shell connector (like the Sun "shoebox") or the Centronics connector (like many third-party harddisks) you should obtain the shortest possible cables which allow you to connect all these while positioning both the Sun and the Emulator at either end of the SCSI chain.

If your SCSI chain already is, or will be, nearly 6 metres in length (the official limit for single-ended SCSI chains) or if

you already have 7 or 8 devices attached (including the Sun itself), you should remove some devices, since the emulator counts as two SCSI devices of the maximum of eight. You could also consider obtaining an additional Sbus SCSI host adapter (aka controller) board in order to connect the Emulator, provided you have a free Sbus slot.

On any SCSI chain (or bus) all connected peripherals must have a unique number, ranging from 0 to 7, called the SCSI-id, short for SCSI bus identification number.
The Sun workstation (or more exactly the SCSI host adapter on the Sun motherboard) always has SCSI-id number 7. Sun uses SCSI-id's 0 through 3 for harddisks, 4 and 5 for tape drives and 6 for the CD-ROM drive by default. The Emulator normally uses SCSI-id number 6, while the standard setting for the Emulator harddisk, build into the Emulator enclosure, is SCSI-id 2. Since SCSI-id's must be unique you should make sure you assign unique SCSI-id's to all connected peripherals.

If you already have devices on your SCSI bus you should ask your system integrator how he configured your system. Or you can check the SCSI-id allocations yourself by halting the Sun and entering the command probe-scsi in the boot-prom program, note down what it reports and then reboot the Sun.

```
# halt
....
>n
ok probe-scsi
....
ok old-mode
> b
...
login:
```

As an example we look at "frodo", a Sun IPC client used by Philips IMS TSA in Eindhoven. It had a single 100 M harddisk with both root and swap partitions and mounts all other disks over NFS from the server. The SCSI-id of it's external harddisk is 0 and this disk has two Centronics connectors, one of which held the SCSI-terminator. To connect the emulator we removed the terminator and connected the emulator with a short (80 cm) Centronics to Centronics SCSI cable. The emulator was left set at it's default SCSI-id's: 2 for the harddisk and 6 for the emulator itself.

Another example is our server "imladris", which has 3 650 M harddisks (at SCSI-id's 0, 1 and 3), a streamer tape (at 4), an Exabyte tape (at 5) and a CD-ROM drive (at 6). The streamer tape and a harddisk are inside a so-called "shoebox" enclosure, which uses the 50-pin D-shell connector. The Sun IPC and the CD-ROM use the mini-SCSI connector and the 2 external harddisks use Centronics. We had to locate a 50-pin D-shell to Centronics cable to connect all these at the same time, while

keeping the length of the SCSI bus as small as possible. Additionally we had to disconnect the CD-ROM drive since it would create a conflict over SCSI-id number 6. Also this installation demonstrates why we recommend against installing an emulator to a server machine since heavy disk usage from the client or reading data from one of the tape drives prevents emulation from playing the disc image flawlessly or even prevents it from starting, since the emulator SCSI controller cannot access the data fast enough and aborts with error 211.

To change the SCSI-id on the emulator or it's build-in harddisk you can use the dipswitches, located at the back of the emulator, just above the round hole. There are eight switches whose default setting is:

	X			X	X			ON
X		X	X			X	X	OFF
1	2	3	4	5	6	7	8	

Switches 1 through 3 set the SCSI-id of the internal harddisk (2 by default), switch 1 the most significant bit, 2 the middle bit and 3 the least significant. Switches 4 and 8 enable SCSI bus parity which is NOT to be used. Switches 5 through 7 set the SCSI-id of the emulator's own SCSI controller (6 by default), again switch 5 the most significant bit, 6 the middle bit and 7 the least significant.

Changing SCSI-id's internally

Since these dipswitches are a fairly recent addition to the emulator, it is possible your emulator does not have them installed and the SCSI-id settings can then only be changed internally. Below is a (quite undetailed) drawing of the inside of the emulator, see figure 4.

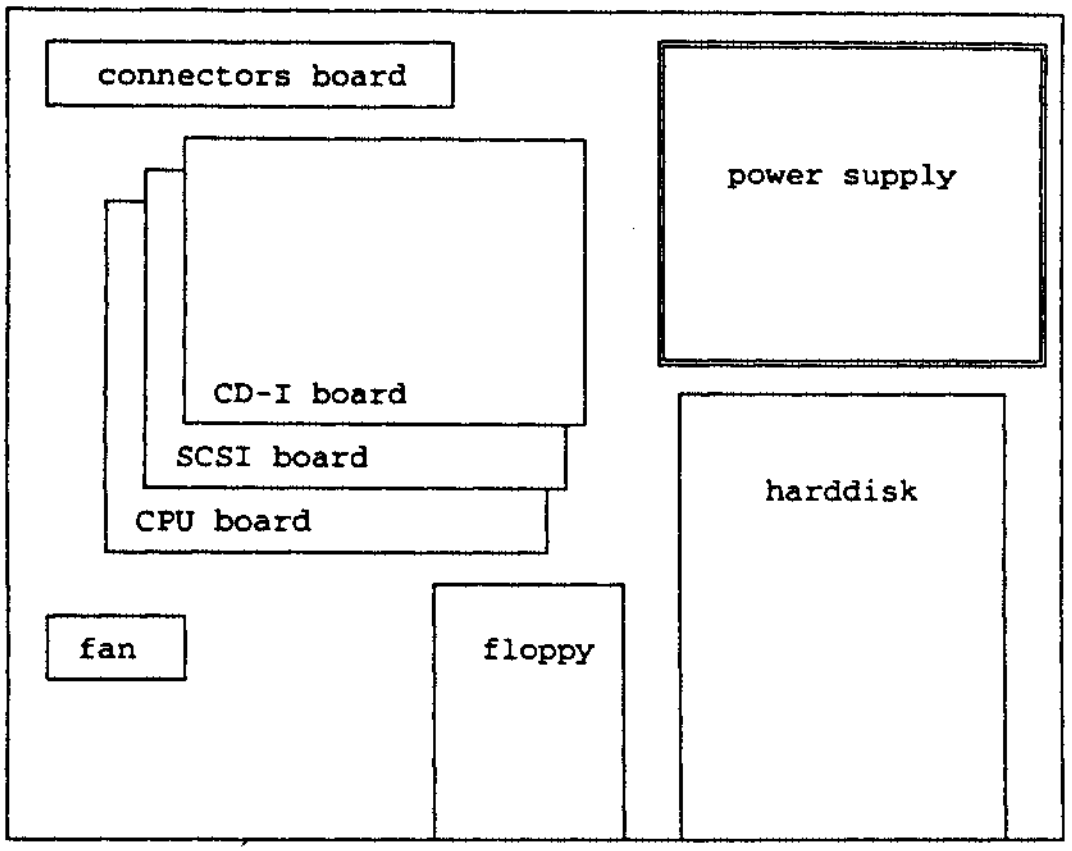


figure 4: emulator inside

In that case, to change the SCSI-id of the internal harddisk you must open the enclosure, by removing the single screw at the top centre at the back of the emulator and then moving the top part of the enclosure towards the front of the emulator and up. To reach the jumpers at the bottom of the harddrive you can either remove the floppy and harddrive or remove the power supply.

On the back of the CDC Wren VI / Seagate ST41766N harddrive you can see the following:

pin-headers	50-pin SCSI connector	4-pin power

1 2 3 4 5 6 7 8

Pin-header pair 1 should be jumpered if the drive is terminated and should supply terminator power. Pin 6 enables spin up on power-on instead of when the host sends a "start unit" command, pin 7 enables parity checking and pin pair 8 is reserved.
Pin-header pairs 3, 4 and 5 set the SCSI-id for this harddrive, with pin pair 3 the most significant bit, 4 the middle bit and 5 the least significant bit. So the setting for SCSI-id 2 would be (: meaning unconnected and | meaning connected) :

	: :	: : : :	50-pin SCSI connector	4-pin power

1 2 3 4 5 6 7 8

On the back of the Hewlett-Packard HP97549S harddrive you can see the following :

4-pin power	pin-headers	50-pin SCSI connector

0 1 2 3 4 5 6 7 8 9

Pin-header pair 0 are the synchronised spindle pins. Pin-header pairs 1 and 2 set terminator power. Pin pair 3 disables unit attention, 4 enables initiation at power-on and reset, 5 enables parity checking and 6 enables spin up on power-on instead of when the host sends a "start unit" command.
Pin-header pairs 7, 8 and 9 set the SCSI-id for this harddrive, with pin pair 7 the most significant bit, 8 the middle bit and 9 the least significant bit. So the setting for SCSI-id 2 would be (: meaning unconnected and | meaning connected) :

4-pin power	: : : : : :	:	:	50-pin SCSI connector

0 1 2 3 4 5 6 7 8 9

Other harddisk may have different connections and settings. Check with your harddrive's documentation or ask your vendor.

To change the SCSI-id of the SCSI controller of the emulator power off, remove the power cable, open the enclosure and located the stack of boards on the left side of the emulator.

The emulator's SCSI controller is the middle printed circuit board in the stack of three. The bottom board is the CPU board, above that is the SCSI controller board, the optional Ethernet board and on top is the CD-I interface board.

It is quite possible you cannot see or move the DIP switches on the SCSI board without removing the top board(s). Be extremely careful when disassembling the stack. You will need to remove all cables attached to the top board and unscrew the four screws holding the board(s) in place. Then lift the board slowly upward to disconnect the pin-header connector from the board below it. Now make the required change to the DIP-switches.

You should be able to see a quadruple DIP switch, named SW1, in the front-right corner of the SCSI controller board. These switches set the emulator's SCSI controller SCSI-id, figure 5 shows the position of the switches and the settings for SCSI-id 6. Switch 1 sets the least significant bit, 2 the middle bit and 3 the most significant. Switch 4 should always be set to ON.

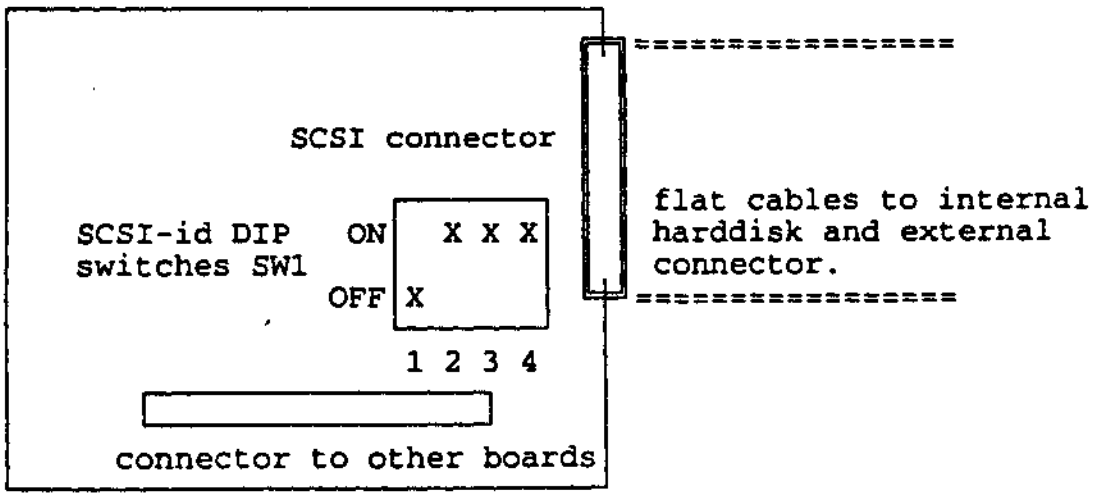


figure 5: location of the SCSI-id DIP switch

To re-assemble align the pins EXACTLY, since an incorrect connection will almost certainly destroy the machine, then apply enough pressure near the pin-headers to insert the board into the connector below. All four screw holes should align exactly before you insert the screws again and connect the flat-cables to their previous positions.

Finally close the enclosure again.

SCSI id conflicts

The most frequently encountered problem with installing an emulator are SCSI id number conflicts, ie. there are two devices on the SCSI bus who respond to the same SCSI address. The effect of such a conflict is a reset or complete blockade of the entire SCSI bus, causing numerous system messages or even a system crash, which could easily damage the data on all connected harddisks.

As mentioned before the default for the emulator harddisk is SCSI-id 2 and the emulator's SCSI controller at SCSI-id 6. Sun has assigned SCSI-id 6 to its CD-ROM drive by default and you should either remove the CD-ROM from your emulator equipped system or move it to another SCSI-id, which forces you to re-assign SCSI-id's, which is done by building a custom SunOS kernel.

If the system console reports errors as soon as you start emulation you either have a cabling problem or a SCSI-id conflict.

Messages like:

esp0: scsi bus reset

indicate you have two disk or tape devices at the same SCSI-id. Hit control-C in the window running emulate, halt the system, change the SCSI-id on the harddisk you emulated from and reboot.

Messages like:

esp0: unexpected selection

indicate you have two host adapters at the same address, which means both the Sun and the emulator are at the same id. Hit control-C in the window running emulate, halt the system, change the SCSI-id on the emulator and reboot.

If the system hangs up completely, ie. no response from keyboard or mouse, and the right red LED on the emulator from panel lights up continuously you have a disk device and the emulator on the same SCSI-id.
You will need to reset the Sun by pressing the Stop-key (L1) and the A-key at the same time, aborting the kernel and returning you to the PROM prompt ">" or "ok". Now change the SCSI-id on the emulator and reboot. It is quite possible your system will now report inconsistencies on your disks when it is rebooting.

There have been reports the connection between the DIP-switches at the back of the emulator and emulator's SCSI controller is sometimes incorrect. With the Sun at SCSI-id 7 and your emulator SCSI controller set to 0, this would report the "unexpected selection" problem, since the controller is really set to 7, although the settings indicate 0. If you encounter this you should interpret the switch settings as follows:

```
 X     X     X     X   ON   X     X     X     X   OFF
 1     2     3     4   OFF  5     6     7     8   ON
```

You then should check the connection to the Wren VI harddisk as well. The connector from the DIP switches should be on pins 3, 4 and 5 in the drawing below:

```
[ : : cable : : : ]  [ 50-pin SCSI connector ]  [ 4-pin power ]
  1 2 3 4 5 6 7 8
```

Similarly on any other type of harddisk, the cable connected to the DIPswitches should covers all headers used for SCSI-id selection.

Serial connection

In order to control the emulator from the host you need to establish a serial connection from the host to the console serial port of the emulator. A suitable cable is delivered with the emulator to connect to serial port A or B of the SPARCstation 1, 1+ and 2. This cable has the following internal connections:

DB 9 female connector in emulator serial port		DB 25 male connector in Sun serial port A or B
pin 2	———	pin 3
pin 3	———	pin 2
pin 4	———	pin 20
pin 5	———	pin 7
pin 8	———	pin 5

If you use a Sun SLC or ELC you will need Sun's serial splitter cable (Sun order number X985Z) to be able to connect both emulator and player to your Sun. If you are using a Sun IPC or IPX you will need two of Sun's DIN8 to DB25 conversion cables (Sun order number X988Y) for both port A and B.

Normally the emulator is connected to port B of the Sun and the serial port of the player is connected to port A. This enables you to see the console output messages from your CD-I application running on the player via port A, while controlling the emulator over port B.

To make sure the serial connection to the emulator and CD-I player will work by modifying the files /etc/remote and /-etc/ttytab.

Add the following to /etc/remote:

```
player:/dev/ttya:br#9600
emulator:/dev/ttyb:br#19200
```

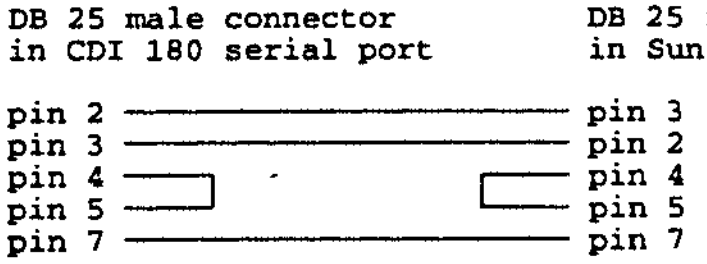
And make sure the lines mentioning ttya and ttyb in your /etc/ttytab have the word "off" in them. This disables logins on these serial ports so the other settings in these lines will be ignored. The lines should look like:

```
ttya    "/usr/etc/getty std.9600"    unknown   off local
ttyb    "/usr/etc/getty std.9600"    unknown   off local
```

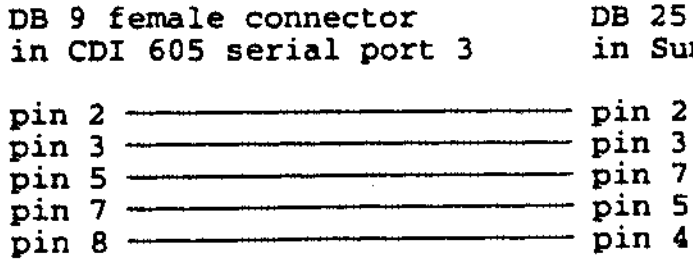
If you needed to modify the /etc/ttytab file you must inform the system of this by sending a HANGUP signal to the "init" process by:

```
# kill -HUP 1
```

The cable between the Sun and a CDI 180 player should be made as follows :



The cable between the Sun and a CDI 605 player should be made as follows :



You can now test the connection to the emulator with the "tip"

terminal emulation program. Open a shelltool (not a cmdtool, since this will not work as expected with tip) and type:

```
% tip emulator
connected
Emu:
```

After hitting the RETURN key a few times you should be able to see the emulator's prompt "Emu:" . Try to issue a "dir" command to the emulator, you should see something like :

```
Emu: dir
          Directory of .
     CMDS OS9Boot   SYS
Emu:
```

Now type ~. (tilde followed by a dot) to exit tip.

Now test the connection to the CD-I player also with "tip". Open a shelltool (not a cmdtool, since this will not work as expected with tip) and type:

```
% tip player
connected
$
```

After hitting the RETURN key a few times reset the player with the RESET button at the front. Then if you have a CDI 180 player, insert the CDRTOS system floppy disk into the left drive, select SYSTEM, then CD-RTOS and finally Chain. If you have a CDI 605 player select SETTINGS and then SYSTEM. You should be able to see the player's prompt "$" .

Now type ~. (tilde followed by a dot) to exit tip.

System software installation

The Emulator comes formatted for SunOS version 4.1.1 and loaded with the contents of the streamer tape.

If it is connected correctly to a diskless SPARC workstation running the default SunOS 4.1.1 GENERIC kernel, you would then be able to access the emulator's harddisk (at SCSI-id 2) as /dev/sd2 and mount the single "a" partition with the following commands issued as the super (root) user:

```
# mkdir /emulator
# fsck /dev/rsd2a
# mount -t 4.2 /dev/sd2a /emulator
```

To have the Sun mount the emulator's harddisk at every reboot add the following line to the file /etc/fstab :

```
/emulator       /dev/sd2a       4.2 rw 1 2
```

If you are using another kernel make sure your modified kernel has UFS support and can handle SCSI devices. The following lines should be present in your kernel configuration file, located in /sys/sun4c/conf/KERNELNAME :

```
#
...
options  UFS       # filesystem code for local disks
options  NFSCLIENT # NFS client side code
...
device-driver esp  # Emulex SCSI interface
...
#
# The following section describes
# SCSI device unit assignments.
#
scsibus0 at esp                        # declare first scsi bus
disk sd0 at scsibus0 target 3 lun 0 # first hard SCSI disk
disk sd1 at scsibus0 target 1 lun 0 # second hard SCSI  disk
disk sd2 at scsibus0 target 2 lun 0 # third hard SCSI disk
disk sd3 at scsibus0 target 0 lun 0 # fourth hard SCSI disk
tape st0 at scsibus0 target 4 lun 0 # first SCSI tape
tape st1 at scsibus0 target 5 lun 0 # second SCSI tape
disk sr0 at scsibus0 target 6 lun 0 # CD-ROM device
...
```

If you needed to edit your kernel configuration files you must recreate the kernel and reboot with it:

```
# vi KERNEL
# config KERNEL
# cd ../KERNEL
# make
# mv /vmunix /vmunix.old
# cp vmunix /vmunix
# reboot
```

See the Sun System and Network Administration Manual for details on the creating a custom kernel.

The SCSI bus identification number allocations can be changed by building a new SunOS kernel but this is not recommended, since you would not be able to access the non-standard connected peripherals if you are ever forced to boot with the GENERIC or MINIROOT kernel.

It is recommended to allow public write access to your emulator's harddisk so all users may store their disc image files there. This is done by the following command while the emulator disk is mounted:

```
# chmod 777 /emulator
```

If you want the users to be unable to remove files which are not their own you should use :

```
# chmod 1777 /emulator
```

You should only store disc image files on it so you are able to use the full capacity of the disc for them. To maximize the amount of space on the disc you can run tunefs on the unmounted filesystem to reduce the amount of reserved space (default is 10%) or recreate the filesystem on the disc (destroying all data currently stored there !) with less reserved space (-m, default is 10%) and optimized for space (-o, default is for time). See the SunOS manual pages for newfs, mkfs and tunefs for more details.

```
# umount /emulator

# tunefs -m 2 /dev/rsd2a
OR preferably
# newfs -m 2 -o space /dev/rsd2a

# fsck /dev/rsd2a
# mount /emulator
```

If you are running the latest version of the SunOS, version 4.1.2 (aka Solaris 1.0.1) you should recreate the emulator's harddisk filesystem with newfs, otherwise emulation may crash the system. This is probably caused by the so called "stable" flag, introduced in SunOS 4.1.2, which informs SunOS the contents of the filesystem on the disk are consistent. Since your new disk has been formatted for 4.1.1, the filesystem stored on your disk does not have this flag. Creating a new filesystem fixes that, but it will erase all files currently stored on the disk ! You should copy all files you intend to keep to another disk or to tape first.

```
# umount /emulator
# newfs -m 2 -o space /dev/rsd2a
# fsck /dev/rsd2a
# mount /emulator
```

If you are running an older version of the SunOS, before 4.1.1, such as SunOS 4.0.3 or 3.5 you should seriously consider upgrading to 4.1.x or reformat the harddisk to better ensure compatibility.

To reformat the emulator's harddisk with any version of SunOS you must add the following entries to the /etc/format.dat file, which describes the disks to be formatted for the format program :

```
#
```

```
# Emulator disk , JWJ 16-4-1991
#
disk_type = "CDC Wren VI 94191-766" \
     : ctlr = MD21 : fmt_time = 5 \
     : cache = 0x11 : trks_zone = 15 : asect = 1 \
     : ncyl = 1488 : acyl = 2 : pcyl = 1490 : nhead = 15 \
     : nsect = 59 : rpm = 3600 : bpt = 31410
#
# The OptImage 1200 MB disk
#
disk_type = "HP 97549S" \
     : ctlr = MD21 : fmt_time = 4 \
     : cache = 0x11 : trks_zone = 15 : asect = 5 : atrks = 30\
     : ncyl = 1908 : acyl = 0 : pcyl = 1908 : nhead = 16 \
     : nsect = 64 : rpm = 3600 : bpt = 30688 :
#
# Emulator disk , JWJ 16-4-1991
#
partition = "OptImage Emulator" \
     : disk = "CDC Wren VI 94191-766" : ctlr = MD21 \
     : a = 0, 1316880 : c = 0, 1316880
#
# The OptImage 1200 MB disk
#
partition = "HP 97549S" \
     : disk = "HP 97549S" : ctlr = MD21 \
     : a = 0, 1953792 : c = 0, 1953792
#
```

Then run format, preferably in single user mode. You should get the manufacturers defects list, commit it, format, partition and then label the disk. Then create a new filesystem on your freshly formatted disk by running newfs on it. A sample session follows, which involved adding a HP97549 harddisk of to a Sun SS1 with emulator with SunOS 4.1.1, assuming the above addition to /etc/format.dat and with some system messages removed.

```
# format
Searching for disks...done.

AVAILABLE DISK SELECTIONS:
 0. sd1 at esp0 slave 8
    sd1: <drive type unknown>
 1. sd2 at esp0 slave 16
    sd2: <CDC Wren VI 94191-766 cyl 1658 alt 2 hd 15 sec 52>
Specify disk (enter its number): 0

AVAILABLE DISK TYPES:
 0. Micropolis 1355
 ...
 14. HP 97549S
 ...
Specify disk type (enter its number): 14
```

```
selecting sd1: <HP 97549S>
[disk formatted, no defect list found]
...
format> defect
defect> extract
Extracting defect list...Extraction failed
defect> original
Extracting manufacturer's defect list...Extraction complete
defect> commit
Ready to update Current Defect List, continue? y
Current Defect List updated, total of 35 defects.
Disk must be reformatted for changes to take effect.
defect> quit
format> format
Ready to format. Formatting cannot be interrupted
and takes 35 minutes (estimated). Continue? y
Beginning format.
...
Formatting done.
Verifying media...
...
Total of 0 defective blocks repaired.
format> partition
format> select
 0. HP 97549S
Specify table (enter its number) [0]:0
partition> label
Ready to label disk, continue? y
partition> quit
format> quit
#
# /usr/etc/newfs -m 2 -o space /dev/rsd1a
/dev/rsd1a: 1953792 sectors in 1908 cylinders of 16 tracks, 64
sectors 1000.3 MB in 120 cylinder groups (16 c/g, 8.39MB/g,
3840 i/g)
super-block backups (for fsck -b #) at:
...
# fsck /dev/rsd1a
# mkdir /new_disk
# mount /new_disk
# chmod 777 /new_disk
```

See the Sun System and Network Administration Manual for details on the SunOS format, newfs, tunefs, fsck and mount commands.

Software installation

The 1/4" tape you received with the emulator contains the discbuilding and emulation software for both Sun3 and Sun4 systems. Load the tape into your tape drive and use the tar command to copy the tape to a suitable, preferable new, directory.

```
# md emulation
# cd emulation
# tar xvf /dev/rst0
```

This should result in the following directory tree for version 2.0 of the pre-mastering and emulation software for Sun:

```
#ls -Rl
Demo:
-rw-r--r--  1 joost    1944 Jun 17 13:37 demo.cdi.scr

Demo/AUDIO:
-rw-r--r--  1 joost 2116356 Jun 10 14:13 message.long
-rw-r--r--  1 joost  571908 Jun 10 14:13 message.short
-rw-r--r--  1 joost  653110 Jun 10 14:13 pop60.acm

Demo/CMDS:
-rwxr-xr-x  1 joost   17452 Jun 10 14:13 play_demo*

Demo/SRC:
-rw-r--r--  1 joost    1369 Jun 10 14:13 default.mk
-rw-r--r--  1 joost     512 Jun 10 14:13 makefile
-rw-r--r--  1 joost   12298 Jun 10 14:13 play_demo.c

Demo/SRC/RELS:
-rw-r--r--  1 joost    4200 Jun 10 14:13 play_demo.r

Demo/TXT:
-rw-r--r--  1 joost       2 Jun 10 14:13 abstract.txt
-rw-r--r--  1 joost       2 Jun 10 14:13 bibliographic.txt
-rw-r--r--  1 joost       2 Jun 10 14:13 copyright.txt

Demo/VIDEO:
-rw-r--r--  1 joost   92244 Jun 10 14:13 bob.dyu
-rw-r--r--  1 joost   92244 Jun 10 14:13 group.dyu
-rw-r--r--  1 joost   92244 Jun 10 14:13 john.dyu
-rw-r--r--  1 joost   92244 Jun 10 14:13 ken.dyu
-rw-r--r--  1 joost   92244 Jun 10 14:13 pam.dyu
-rw-r--r--  1 joost   92244 Jun 10 14:13 randy.dyu
-rw-r--r--  1 joost   92244 Jun 10 14:13 sean.dyu
-rw-r--r--  1 joost   92244 Jun 10 14:13 steve.dyu
-rw-r--r--  1 joost   92244 Jun 10 14:13 tom.dyu
-rw-r--r--  1 joost   92244 Jun 10 14:13 vicki.dyu
```

```
Sample_Scripts:
-rw-r--r--  1 joost    2129 Jun 10 14:13 demo.cdi.scr
-rw-r--r--  1 joost    1012 Jun 10 14:13 demo.grn.scr
-rw-r--r--  1 joost    1356 Jun 10 14:13 demo.pre.scr
-rw-r--r--  1 joost    2490 Jun 10 14:13 demo.rdy.scr
-rw-r--r--  1 joost     778 Jun 10 14:13 demo.red.scr
-rw-r--r--  1 joost   17799 Jun 10 14:13 demo.xa.scr

Sun3:
-rwxr-xr-x  1 joost   16384 Jun 10 14:13 burncd*
-rwxr-xr-x  1 joost   49152 Jun 10 14:13 cdedit*
-rwxr-xr-x  1 joost   16546 Jun 10 14:13 discmap*
-rwxr-xr-x  1 joost   16384 Jun 10 14:13 emulate*
-rwxr-xr-x  1 joost  106496 Jun 10 14:13 green*
-rwxr-xr-x  1 joost  122880 Jun 10 14:13 master*

Sun4:
-rwxr-xr-x  1 joost   16384 Jun 10 14:13 burncd*
-rwxr-xr-x  1 joost   57344 Jun 10 14:13 cdedit*
-rwxr-xr-x  1 joost   16384 Jun 10 14:13 discmap*
-rwxr-xr-x  1 joost   16384 Jun 10 14:13 emulate*
-rwxr-xr-x  1 joost   49152 Jun 10 14:13 emutool*
-rwxr-xr-x  1 joost  122880 Jun 10 14:13 green*
-rwxr-xr-x  1 joost  147456 Jun 10 14:14 master*
#
```

Now copy all binaries for your system, either sun3 or sun4, to a directory which is (or will be) in every user's PATH. I recommend you copy all binaries to /usr/local/bin. Then make the binaries emulate, emutool and burncd SUID root. This is necessary since these programs create a list of the absolute block addresses on the disk which contain your CD-I disc image file. To determine these, the program must read from the raw disk device, which is only allowed for the super (root) user. With the SUID root option, the program runs as if it was run by the super user so it will have the needed permissions.

```
# cp Sun4/* /usr/local/bin
# cd /usr/local/bin
# chown 0.0 emulate emutool burncd discmap master green
# chmod 755 discmap master green
# chmod 4755 emulate emutool burncd
# ls -l burncd cdedit discmap emulate emutool green master
-rwsr-xr-x  1 root    16384 Jun 10 14:13 burncd*
-rwxr-xr-x  1 joost   57344 Jun 10 14:13 cdedit*
-rwxr-xr-x  1 joost   16384 Jun 10 14:13 discmap*
-rwsr-xr-x  1 root    16384 Jun 10 14:13 emulate*
-rwsr-xr-x  1 root    49152 Jun 10 14:13 emutool*
-rwxr-xr-x  1 joost  122880 Jun 10 14:13 green*
-rwxr-xr-x  1 joost  147456 Jun 10 14:14 master*
#
```

If you cannot find the "green" program you may have an older version, where that program was named "ready".

Emulation

To start emulation you should first create a CD-I disc image file. To do that copy the demo directory you read from the tape to your emulator harddisk, preferably as a normal user. This user should have added the /usr/local/bin directory in his PATH, by adding it to his .login or .cshrc file. If you change the PATH while logged in (and while using csh or tcsh) you must issue a rehash command. With /usr/local/bin in your PATH the system can find the new master and emulate commands.

```
% setenv PATH (/usr/local/bin .....)
% rehash
% cp -rp demo /emulator
```

Now you have copied the demonstration files we can generate the CD-I disc image file with master:

```
% cd /emulator/demo
% master demo.cdi.scr
...
% ls demo.cdi
```

It is recommended to instruct master to generate a table-of-contents (TOC) file with the -m option as in:

```
master -m=demo.cdi.toc demo.cdi.scr
```

When a disc image file exists we can start emulation. You should run the emulate (and burncd) programs in a SunView or OpenWindows shelltool, not a commandtool (cmdtool) since the commandtool does not work correctly when used with tip which is used by emulate (and burncd).

```
% emulate demo.cdi
```

When the message "stop at sector 0" appears you can select the CDI or PLAY button on your connected player to start the application. To stop emulating hit control-C and you will be returned to your shell prompt.

The latest version of emulate for Sun offers the following command line options:

```
% emulate -\?
CD Emulation Utilities for Sun 4 - Version 2.0
Copyright 1992 by OptImage Interactive Services Company, L.P.

Syntax:   emulate [<opts>] (<disc_image_file>) [<opts>]
Function: Emulate CD Disc Image on GMX Emulator
Options:
  -?              Print this help message
```

```
-a              CD-Rom XA Image
-c=<TOC_file>   Use <TOC_file> from 'Master'
-d=/<dev>       Emulator Device Containing Disc Image
                (default: /h2@)
-f              Force Creation of New Image Map
```

The -c=<TOC_file> enables you to emulate (and burn) a CD-I disc image file which contains CD-DA tracks besides the CD-I track or a CD-ROM XA track. A TOC file will be required when the Philips CDD 521 WORM writer becomes available for CD-I use. For more details on how to create such disc image files please refer to the documentation you received with the pre-mastering tools.

When running SunView or OpenWindows you can also use the emutool program to run emulation in a SunView window. This program uses two environment variables, EMUPORT and EMUDIR, with which you can select the default directory emutool looks in for disc image files (EMUDIR) and the serial port the emulator is connected by (EMUPORT).

```
% setenv EMUPORT /dev/ttyb
% setenv EMUDIR /emulator
```

When these environment variables have not been defined the emutool program uses /dev/ttyb and the current directory respectively.

From emutool you select the disc image file with the "Select image file" button and the optional table-of-contents (TOC) file with the "Select TOC file" button. Then you can click the "START" button to start emulating. When you click on the "Options" button a window appears in which you can select whether you want to emulate or burn a WORM, if you want the regenerate the map file always or only when needed, if you disc image file is type CD-I / CD-I Ready / CD-I with CD-DA or type CD-ROM XA and what wildcards you prefer for image and TOC files.

Should you see messages appear in your console window such as:

```
ld.so: warning: /usr/lib/libsuntool.so.0.53 has older revision
than unexpected 54
ld.so: warning: /usr/lib/libsunwindow.so.0.53 has older revi-
sion than unexpected 55
```

you have an older version of SunOS as the version used by the developer of the emulation tools. This should not prevent them from working correctly, but you should obtain the most recent version.

Burning WORMs

If you have a Yamaha PDS WORM system connected to you emulator, switched on and loaded with a new empty WORM disc, you may burn a CD-I disc image file into the WORM disc with the burncd command :

```
% burncd cdimage
```

It is recommended you do not use your Sun for any other purpose while burncd is running. Using any program will at least cause some disk usage, which will use your Sun's SCSI bus and since the emulator is reading data from your emulator harddisk over the same SCSI bus at that time and since this data MUST be delivered via the emulator to the Yamaha system within a small timeframe, you could cause a write error on your WORM disc, which would make the WORM unusable.

You should also make sure the Yamaha system does not move or is exposed to shocks while burning a WORM since this may also cause a write error on your WORM disc, which would make the WORM unusable.

When burncd reports it is done you may exit burncd with control-C and you will be returned to your shell prompt.

The latest version of burncd for Sun offers the following command line options:

```
% burncd -\?
CD Emulation Utilities for Sun 4 - Version 2.0
Copyright 1992 by OptImage Interactive Services Company, L.P.

Syntax:   burncd [<opts>] {<disc_image_file>} [<opts>]
Function: Burn a Yamaha WORM Disc
Options:
  -?              Print this help message
  -a              CD-ROM XA Image
  -c=<TOC_file>   Use <TOC_file> from 'Master'
  -d=/<dev>       Emulator Device Containing Disc Image
                  (default: /h2@)
  -f              Force Creation of New Image Map
  -t              Test Mode (Don't Actually Burn Disc)
```

Creating CD-I image file tapes

To send a disc image file to another site or the disc pressing factory you can use an Exabyte tape drive and the SunOS tar program to write your CD-I disc image file to an Exabyte tape, which can directly be read on any UNIX system to be written to a WORM or you can use the ansitape program to create an ANSI labelled Exabyte tape used by the disc pressing factory.

To create a tar tape using 60 Kb buffering (default is 20, equal to 10 Kb) enter the following command line:

```
% tar cvbf 120 /dev/rst1 discimage.cdi discimage.cdi.toc
```

PDO however requests you use at most 32 Kb for buffering:

```
% tar cvbf 64 /dev/rst1 discimage.cdi discimage.cdi.toc
```

The correct commandline to create an ANSI labelled tape on your local Exabyte tape drive at SCSI-id 5 is:

```
% ansitape cv mt=/dev/rst1 bs=14112 rs=2352 cc=e rf=f cdimage
```

From the manual page of ansitape:

```
ansitape txrc[vqfaei3] [mt=device] [vo=volume-name]
     [rs=[ r | recordsize ]] [bs=blocksize] [rf=[ v | f ]]
     [cc=[ i | f | e ]] filename1 filename2 . . .
```

For CD-I disc image files we create an ANSI labelled tape (c) verbosely (v) on the second tape drive (mt=/dev/rst1) with a block size of 14112 bytes (bs=14112) equal to 6 records, a record size of a single CD-I sector (rs=2352), embedded carriage control (cc=e) and a fixed record format (rf=f).

386/486 PC systems

The PC version of the Philips/OptImage emulator comes with an Always Technologies IN-2000 SCSI controller and pre-mastering software for use with Microsoft Windows version 3.0 or higher running in 386 enhanced mode. It can be connected to any PC with a free 16-bit ISA (or EISA) slot, with an 80386 or 80486 processor and at least 8 MB of memory (conventional and extended).

Hardware connections

The emulator has to be connected to your PC system through both the SCSI bus, on the Always IN2000, and the COM1 or COM2 serial port.

SCSI connection

To install an emulator with your PC system you must first install the SCSI controller into your system. We supply the Always Technologies IN-2000 SCSI controller with the PC version of the emulator. This is a 16-bit ISA bus expansion board, which comes with an on-board BIOS, whose default address is 0xC800. This, as well as the default setting for I/O address and interrupt level, could conflict with your existing controller, so you should collect and examine the documentation you have on your installed expansion boards and select the correct settings for the IN-2000 with the help of the Installation Guide.

The IN-2000 BIOS version 3.33 supports 2 harddisk drives directly at the BIOS level, minus the number of previously installed harddisk drives (controlled with another controller board). So if you already have a drive C: connected via an AT/IDE or ESDI controller it will only support one drive (D:) from the BIOS.
A newer version of the IN-2000 BIOS, version VCN 1.02 offers support for up to seven drives, when you are using MS-DOS version 5.0 . See the Installation Guide for details.

The SCSI-id's of the harddisks connected to the IN-2000 (which itself is at SCSI-id 7) must be incremental, ie. you should use SCSI-id 0 for the first, 1 for the second and so on. So you should set the SCSI-id of the harddisk inside the emulator enclosure to 0 and the SCSI-id for the emulator's SCSI controller to 6, resulting in the following settings for the DIPswitches on the back of the emulator:

	1	2	3	4	5	6	7	8
ON					X	X		
OFF	X	X	X	X			X	X

You can then connect the emulator to the IN-2000 inside your PC with the supplied SCSI cable. Please check you have used the IN-2000's SCSI connector and not a parallel port, since the external connectors are exactly the same (DB 25 female) !

After a reboot the IN-2000 BIOS should report how many disks it detected and it will make your new disk available, if it was formatted correctly. See below if that is not the case.

Serial connection

In order to control the emulator from the PC you need to establish a serial connection from the PC's COM1 or COM2 port to the console serial port of the emulator. A suitable cable is delivered with the emulator. This cable has the following internal connections:

DB 9 female connector in emulator serial port		DB 25 male connector in PC serial port
pin 2	———	pin 2
pin 3	———	pin 3
pin 4	———	pin 4
pin 5	———	pin 5
pin 8	———	pin 8

Normally the emulator is connected to the COM2 port of the PC and the serial port of the player is connected to port COM1, except if your mouse is attached to COM1 in which case you should obtain a third serial port COM3 and connect the CD-I player to COM3. Be aware however of the IRQ number used by COM3, normally this is IRQ 5 which could conflict with your Ethernet board ! This serial connection to the player enables you to see the console output messages from your CD-I application running on the player via COM1 or COM3, while controlling the emulator over COM2.

You could run into problems however, since a PC and/or Windows appear to have problems with the 19200 baud speed of the emulator serial connection. In that case your best alternative would then be to use an old VT100 compatible terminal to see the serial output from the CDI player.

You can test the connection to the emulator with a terminal emulation program like Procomm or Windows' Terminal. Set the

terminal emulation to 19200 baud, 8 data bits, no parity and 1 stop bit. After hitting the RETURN key a few times you should be able to see the emulator's prompt "Emu:" .
If you are unable to get this prompt you may have received an incorrect cable. Try to insert a null-modem between the PC and the cable, using the following connections:

DB25 male connector to DB25 female connector

```
pin  2 ------------------------- pin 3
pin  3 ------------------------- pin 2
pin  4 ------------------------- pin 5
pin  5 ------------------------- pin 4
pin  7 ------------------------- pin 7
pin  6 --┐
pin  8 ---┴----------------------- pin 20
pin 20 ------------------------┐-- pin 6
                               └-- pin 8
```

This should fix the problem. Try to issue a "dir" command to the emulator, you should see something like :

```
Emu: dir
          Directory of .
     CMDS OS9Boot   SYS
Emu:
```

The cable between the PC and a CDI 180 player should be made as follows :

```
DB 25 male connector          DB 25 female connector
in CDI 180 serial port        in PC serial port

pin 2 ------------------------- pin 3
pin 3 ------------------------- pin 2
pin 4 --┐                  ┌-- pin 4
pin 5 ---┴------------------┴--- pin 5
pin 7 ------------------------- pin 7
```

The cable between the PC and a CDI 605 player should be made as follows, if you use a 25 pin connector on your PC :

```
DB 9 female connector       DB 25 female connector
in CDI 605 serial port 3    in PC serial port

pin 2 ------------------------- pin 2
pin 3 ------------------------- pin 3
pin 5 ------------------------- pin 7
pin 7 ------------------------- pin 5
pin 8 ------------------------- pin 4
```

Or if you use a 9 pin serial connector on your PC :

DB 9 female connector DB 9 female connector

in CDI 605 serial port 3	in PC serial port
pin 2	pin 3
pin 3	pin 2
pin 5	pin 5
pin 7	pin 8
pin 8	pin 7

If you only have an DB 9 connector on your selected serial port you could also obtain a 25 pin to 9 pin serial cable converter or make one yourself with the connections as follows:

DB 25 male connector in cable to CDI	DB 9 female connector in PC serial port
pin 2	pin 2
pin 3	pin 3
pin 7	pin 5
pin 4	pin 7
pin 5	pin 8

You should next test the connection to the player. Make sure your terminal emulator supports COM3 if you use that serial port ! Set the terminal emulation to 9600 baud, 8 data bits, no parity and 1 stop bit. After hitting the RETURN key a few times reset the player with the RESET button at the front. Then if you have a CDI 180 player, insert the CDRTOS system floppy disk into the left drive, select SYSTEM, then CD-RTOS and finally Chain. If you have a CDI 605 player select SETTINGS and then SYSTEM. You should be able to see the player's prompt "$" .

System software installation

The harddisk inside your emulator is normally already formatted. Should you wish (or need) to reformat it you can do so by first partitioning it into a single DOS partition with FDISK, rebooting and then running FORMAT on it. This can take up half an hour so please be patient.

When installed in some PC systems the IN-2000 with the version 3.33 BIOS assumes it is the primary controller, ie. it assumes you intend to boot from the SCSI drive at id 0, even if you already have an otherwise controlled C: drive. During boot however the BIOS software cannot locate the system files so it aborts the boot. The net result of this situation is you need to boot from a floppy disk. This can be a nuisance but it also offers the use of different setups to boot up from. This problem does not occur on IN-2000 boards with the latest (VCN 1.02) BIOS ROM and MS-DOS version 5.0 .

Software installation

The software provided with the emulator runs only when you are using Microsoft Windows version 3.0 or 3.1, preferably in 386 enhanced mode. It consists of a number of 3.5" floppy disks.

On each is an INSTALL batch file which starts up Microsoft Windows and then runs the WINSTALL program, so you will need to install Windows before installing the CD-I premastering software. To run the installer type A:INSTALL at the DOS prompt (while Windows is NOT running) or from within Windows select "File, Run" and enter the name a:winstall.

The installer program copies the programs to your harddisk and modifies files in your root directory (AUTOEXEC.BAT) and in your Windows directory (WIN.INI). In AUTOEXEC.BAT it appends c:\CDI_PAK (or the directory you installed the software in) to your path and sets the environment variable NUMSTDHD to 0. NUMSTDHD should be set to the number of harddisks NOT mounted by the IN-2000 SCSI controller board. To WIN.INI it adds the following :

```
[Optimage]
emulate_opts=c:\CDI_Pak\opts\
burncd_opts=c:\CDI_Pak\opts\
```

Subsequently the installer exits itself AND Windows. The installer does its job quite incorrectly however since:

(1) WINSTALL program replaces ALL lines containing the word PATH to a line setting the new PATH environment variable.

(2) WINSTALL changes everthing in your autoexec.bat file to upper case. So make a backup copy of your autoexec.bat and make the changes to the PATH and the NUMSTDHD environment variable yourself.

(3) The Program Manager icons you will find will all just mention the name of the executable program, NOT the full path of the executable. Therefore Windows is unable to locate the programs so it display the default icon (a monitor) instead of the intended icon (a CD-I disc). You should add the path (normally C:\CDI_PAK\) in front of the program name by selecting the icon, selecting "File, Properties", add the path, select Change Icon, click OK and finally click OK again.

When you upgrade to Microsoft Windows 3.1 you should re-install all CD-I Authoring software.

Emulation

We recommend you put all four CD-I Windows applications, GREEN, MASTER, EMULATE and BURNCD in a separate group. You may set the default directory to the emulator harddisk (Windows 3.1 only) via the File,Properties menu of the Program Manager.

To try your emulation system you can create the demonstration disc image, distributed as "CD-I Disc Building Tutorial" (this is the same material used for the sample Balboa application described in IMS TSA Technical Note number 1). To create it double click on Master and select the file "TUTORIAL/BUILD.CD" as input. Master will generate the CD-I disc image in the file "CDIMAGE" of some 16 MB in the current directory, so make sure you are writing to the emulator harddrive (normally D:). Then double click on Emulate, select the file "CDIMAGE", click OK, set "Use COM port 2" if your emulator is connected to port 2 (otherwise it must be connected to port 1), click OK to start emulating.

When the message "stop at sector 0" appears you can select the CDI or PLAY button on your connected player to start the application. To stop emulating hit control-C and the emulation program will exit.

There also exists a command line version of emulate called EMULATE_.EXE, runnable without Windows, which offers the following options:

```
C:\>emulate_
OptImage emulation software version 1.0, revision 1 02/26/91
Syntax: emulate_ [<opts>] <file>
where
 <opts>:
     /b   image map is binary
     /p   serial port is COMM2, (default is COMM1)
     /?   print this help message
 <file>:  path of CD-I disc image file
```

To burn a WORM on a Yamaha PDS system attached to your emulator and loaded with a new empty WORM disc, double click on BurnCD, select the file "CDIMAGE", click OK, set "Use COM port 2" if your emulator is connected to port 2 (otherwise it must be connected to port 1), click OK to start the burning.

There also exists a command line version of burncd called BURN_CD.EXE, runnable without Windows, which offers the following options:

```
C:\> burn_cd
OptImage burncd software version 1.0, revision 1 02/26/91
```

```
Syntax: burn_cd [<opts>] <file>
where
 <opts>:
     /p   serial port is COMM2, (default is COMM1)
     /t   test mode (don't actually burn a disc)
     /?   print this help message
 <file>:  path of CD-I disc image file
```

Creating CD-I image file tapes

To send a disc image file to another site or the disc pressing factory you can use an Exabyte tape drive and SCSI Express (sold as PC Tar) to write your CD-I disc image file to an Exabyte tape in the so called tar format, which can directly be read on any UNIX system to be written to a WORM or an ANSI labelled Exabyte tape used by the disc pressing factory.

If you have connected the tape drive to the Always IN2000 controller at SCSI-id 5 and installed SCSI Express (aka PCTar) with the following line in your config.sys :

```
DEVICE = c:\scsi\express.sys /hin2
```

you should be able to copy the CDI disc image file to tape with the command:

```
c:\> xtar cbf 120 5 cdimage
```

Apple Macintosh II or Quadra systems

The Mac version of the Philips/OptImage emulator comes with pre-mastering and emulation software for Apple Macintosh II and Quadra computers with at least 4 MB of memory, running Finder version 6.0.2 and higher.

Hardware connections

The emulator has to be connected to your Apple Macintosh II or Quadra system through both the SCSI bus and the modem or printer serial port.

SCSI connection

The emulator has to be connected to your Mac's SCSI port with the supplied SCSI cable. The harddisk is normally delivered formatted and loaded with the CD-I discbuilding demonstration.

The SCSI-id's of the harddisks connected to the Mac (which itself is at SCSI-id 7) must be different. So you should set the SCSI-id of the harddisk inside the emulator enclosure to a SCSI-id not used yet. You can use an Apple Control Panel like SCSI-Probe or SCSI-Info to determine the SCSI-id's already in use. Apple normally installs the internal harddisk of a Macintosh at SCSI-id 0. If you have no additional disks or tape drives connected you should put the emulator's internal harddisk at SCSI-id 2 and set the SCSI-id for the emulator's SCSI controller to 6 with the DIP switches located at the back of the emulator as follows :

	1	2	3	4	5	6	7	8
ON		X			X	X		
OFF	X		X	X			X	X

The Mac and the emulator should be located at either end of the bus. Since the emulator is internally terminated you should not connect any extra terminators to the bus. Except when you have a Macintosh Iifx, since this version requires special termination circuitry, which is build into the black terminator block which comes with your Iifx. You should put this block between the emulator's external SCSI connector and the SCSI cable running to the next device on the SCSI bus.

Serial connection

In order to control the emulator from the Mac you need to establish a serial connection from the Mac's modem or printer port to the console serial port of the emulator. A suitable cable is delivered with the emulator. This cable has the following internal connections:

DB 9 female connector in emulator serial port		mini DIN8 male connector in Mac printer port
pin 2	————————	pin 3
pin 3	————————	pin 5
pin 5	————————	pin 4

Normally the emulator is connected to the printer port of the Mac and the serial port of the player is connected to the modem port. This enables you to control the emulator from one window, while you can use the second window as the console of your CD-I player during emulation. This window is a full terminal, both in- and output are possible.

You will need to disconnect your Mac from any LocalTalk network however in order to use the two ports at the same time.

You can test the connection to the emulator with a terminal emulation program like Red Ryder or White Knight. Set the terminal emulation to the printer port, VT100 emulation, (in Red Ryder: "Customize, Terminal Emulation Preferences...") 19200 baud, 8 data bits, no parity and 1 stop bit (in Red Ryder: "Local, Change Serial port settings..."). After hitting the RETURN key a few time you should be able to see the emulator's prompt "Emu:" .

Some versions of Red Ryder and White Knight do not correctly set the port and communication settings if you change them while the program is running. You should try the following: save your new settings, quit the terminal emulator and restart it by double clicking your new settings file.

Try to issue a "dir" command to the emulator, you should see something like :

```
Emu: dir
          Directory of .
     CMDS OS9Boot   SYS
Emu:
```

To connect a CDI 180 serial port to a Macintosh running suitable terminal emulation software create or obtain the a cable with the following connections:

DIN 8 male connector in Mac modem port	DB 25 female connector in CDI 180 serial port
pin 3	pin 3
pin 5	pin 2
pin 4	pin 7
	pin 4
	pin 5
	pin 6
	pin 8
	pin 20

To connect a CDI 605 serial port 3 to a Macintosh running suitable terminal emulation software create or obtain the a cable with the following connections:

DIN 8 male connector in Mac modem port	DB 9 female connector in CDI 605 serial port
pin 3	pin 3
pin 5	pin 2
pin 4	pin 5
	pin 7
	pin 8

You should next test the connection to the player. Set the terminal emulation to the modem port, VT100 emulation, 9600 baud, 8 data bits, no parity and 1 stop bit. After hitting the RETURN key a few times reset the player with the RESET button at the front. Then if you have a CDI 180 player, insert the CDRTOS system floppy disk into the left drive, select SYSTEM, then CD-RTOS and finally Chain. If you have a CDI 605 player select SETTINGS and then SYSTEM. You should be able to see the player's prompt "$" .

System software installation

Should you wish (or need) to reformat the harddisk inside the emulator's enclosure your must have a copy of the formatting software from MacLand called SmartFormat which can recognise the CDC Wren VI disk.

Double click on the SmartFormat icon and the formatter will search the SCSI bus for disks it recognises. These will appear at the top of the window, with their SCSI-id's not faded. Moving the mouse pointer over the dot will make SmartFormat display information on that disk in the bottom of the window. Select the emulator harddisk (normally at SCSI-id 2 with manufacturer CDC type Wren VI 94191-766-15). Then click Format, then Yes. Formatting will take up to 30 minutes, after which you can quit SmartFormat. Your freshly formatted disk should already be mounted, otherwise restart the Mac to make it available.

Software installation

To install the emulation and burn software follow the instructions from the accompanying manuals. The current Macintosh emulation software comes with an installer program. The burn software is a single program which you should copy to the same folder the emulation software is in.

Emulation

To try your emulation system you can create the demonstration disc image, distributed with the Mac Emulation package (this is the same material used for the sample Balboa application described in IMS TSA Technical Note number 1).
To create it double click on Master and select the file "Tutorial/tutorial.script" or double click on that file from the Finder. Master will generate the CD-I disc image in the file "tutorial.cdi" of some 16 MB in the current directory, so make sure you are writing to a harddrive with sufficient free space.

To emulate just double click on the Emulate application icon, select "File, Emulate Disk" and select a CD-I disc image file located on any harddisk or double click on the disc image file icon from the Finder. The emulation software will detect the SCSI-id of the harddisk you file resides on and will inform the emulator. You can see the results on the top window, where the command line appears:

```
Emu: cdem -d=/h2@ -s=1234560
waiting to receive map
reading map
+6
stop at sector 0
```

When the message "stop at sector 0" appears you can select the CDI or PLAY button on your connected player to start the application. To stop emulation select "File, Stop Emulation".

To burn a WORM on a Yamaha PDS system attached to your emulator and loaded with a new empty WORM disc, double click on BurnCD_Mac which then also opens two windows, one controlling the emulator and a status window. To start select "File, Burn Disc", select a disc image file and click on either TestDIsc or BurnDisc. When you click TestDisc everything is done exactly as if you would burn a WORM except the Yamaha recorder's laser is not switched on. BurnDisc will really burn a WORM disc.

Creating CD-I image file tapes

To send a disc image file to another site or the disc pressing factory you can use an Exabyte tape drive and MacTar to write you disc image file to a "tar" formatted tape readable on most UNIX systems or QuTape version 1.64 to write your disc image file to an ANSI labelled Exabyte tape which can directly be used by most disc pressing factories.

Once you have inserted the tape into the tape drive and the tape drive is ready (signalled by the green LED on the drive door), you should check if the Exabyte tape drive accepts commands from QuTape by selecting "Control, Rewind Tape". Should QuTape then report "Tape error # -36" your SCSI setting may be incorrect. Select "Tape, Tape Test..." and set the SCSI id (normally 4), set No.of loops to 1 and click TEST. When this test completes successfully you should be able to rewind the tape with "Control, Rewind Tape", proving QuTape now controls the Exabyte tape drive.

Before you write the CD-I disc image file you must first initialize the tape. We strongly recommend you use a brand new tape. Then select the "Tape, Label Tape...". You will again be presented with a dialogue box. You may choose your own volume label (something indicative of the tape's contents), and then select the "New Tape" button and confirm your choice.

Now that you have an initialized tape, you need to set the actual tape writing parameters. You will have to create these by selecting "Tape, Set Tape Block Parameters...". You will be presented with a dialogue box containing several settings. You should fill in the box so it agrees completely with the following:

- Bytes per Tape block: 14112
- Records per Tape block: 6
- Fixed record format (not Variable)
- Field descriptors "2352," (setting the record size)
- End of field character is None
- Short length tape blocks is Allow
- Image Mode transfer is NOT checked

Once you have entered the above options, you should use the "File, Save Settings..." choice to save your settings so you will not need to re-enter them.

Finally we should set the FileType so we are able to save CD-I disc image files. Select "File, Set FileType..." and enter (case sensitive!) type "DcIm" and creator "DcMp".

Once you have entered all the options, select the "File, Write Tape...". You will see the dialogue box again to confirm the settings. Check to make sure they agree with the above settings; if so, confirm the choice. You will then see a dialogue box asking for the amount data to write and what fork to use. The number 10 (or any value that appears) is irrelevant, because you should select the "to End-of-File" radio button, the Data button and then confirm your choice.

Finally, you will be presented with a standard file selection dialogue box. Simply select your CD-I disc image, and the program will begin writing the tape. This may take quite a while, so you should start it only when you do not need to use your Mac for some time.

Philips/OptImage OS9 emulator systems

The OS9 version of the Philips/OptImage emulator comes with Script-to-Disc pre-mastering software for use with applications created by the MediaMogul CD-I authoring tool.

The development host is the player itself if you are using MediaMogul or the forthcoming MediaShowcase CD-I authoring tools. Titles created with MediaMogul (MM) must be converted into a CD-I disc image file with the Script-to-Disc (S2D) software which runs on an emulator system with (at least) one large harddisk. Titles created with MediaShowcase (MSC) will also need conversion into a CD-I disc image file with the forthcoming Talk-to-Disc (T2D) software.

Script-to-Disc

Once all video and audio assets have been acquired and converted into file formats suitable for MediaMogul, they are stored on the MM harddisk. After all script and menu files have been created, and after all script files have been analyzed for disc building constraints, the application is completely defined. The MM harddisk is then connected to the S2D emulator, and the ScriptToDisc system automatically creates a CD-I disc image on the emulators required harddisk. This disc image can then be emulated for verification, and finally pressed into a CD-I disc.

The ScriptToDisc system includes the following programs: cdbuild, which reads script, menu, and asset files to create discbuilding scripts for rtrb and master; rtrb, which creates a separate interleaved realtime file for each script; and master, which assembles the realtime files and other data files into a disc image.

Hardware connection

The emulator running S2D needs at least one harddisk to store the S2D software, generated realtime files and your disc image file.

If the size of your disc image file will me greater than half the size of the single required harddisk you should obtain a second harddisk and connect it, otherwise your will not have sufficient disk storage space, since S2D requires twice the amount of space the disc image file needs.

Since S2D needs to read your MM assets and scripts you must also connect your MM harddisk to the emulator's SCSI bus.

And if you wish to send your completed CD-I disc image file to a CD manufacturing plant you will also need a tape drive capable of recording the entire file on a single tape such as a 1/4" streamer tape drive (up to 250 MB), a DAT tape drive (up to 1200 MB) or an Exabyte tape drive (up to 2300 MB) and the appropriate software to control the tape drive from the emulator or CD-I player.

Additionally you can connect the optional WORM system to the emulator, so you can "burn" your own WORM discs.

Finally you will need an authoring player (CDI 180 or CDI 605) to connect to the emulator, in order to run the disc image files created on by the S2D system, for which you normally would use your MM player. Although MM runs fine on the CDI 602 with additional memory and a SCSI interface this player does not have the connections required for emulation.

This leads to the following setup, see figure 6:

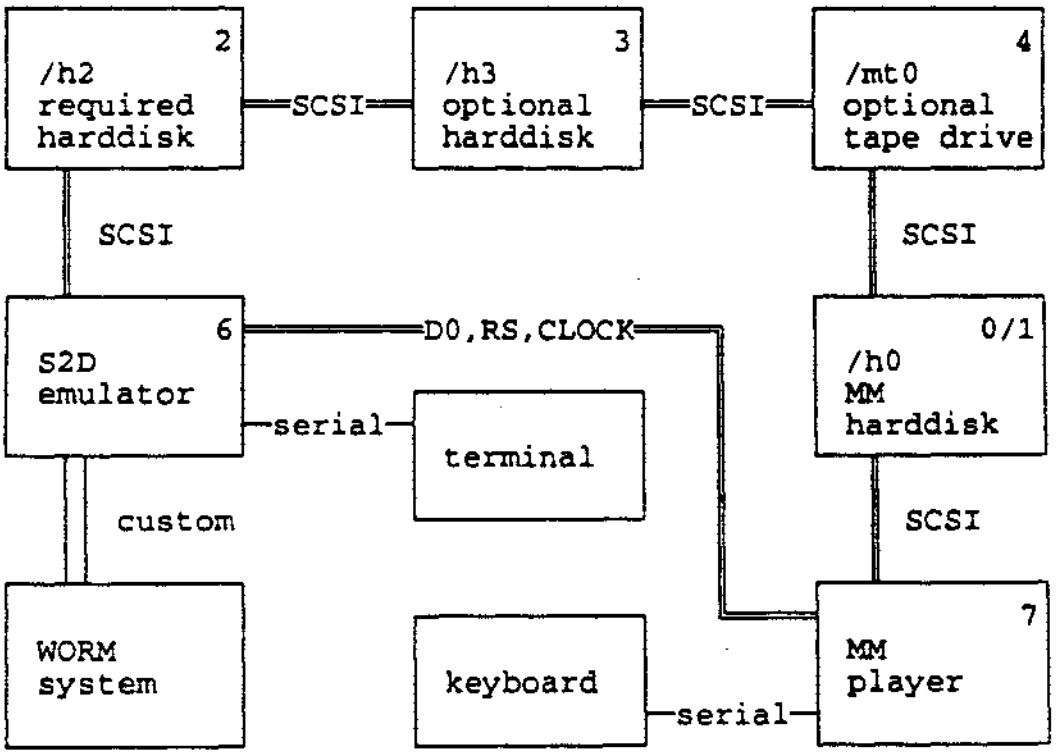


figure 6: S2D emulator setup

SCSI connection

Since all devices connected to the same SCSI bus must have different addresses we allocate the following SCSI-id's :

MM harddisk (605/602)	: id 0
MM harddisk (180)	: id 1
required harddisk	: id 2
optional harddisk	: id 3
optional tape drive	: id 4
S2D emulator	: id 6
MM player	: id 7

This ensures we see no SCSI address conflicts. There could appear problems however if both the S2D emulator and the MM player would use the SCSI bus at the same time or even after each other, resulting in a so called "hangup" of the SCSI bus, blocking both machines. So we highly recommend you do not use or even switch off your MM player when you are using the S2D emulator. You may also choose to disconnect it completely from the SCSI bus when using the emulator, in which case you should connect a SCSI bus terminator, normally supplied with your harddisk or tape drive, to the last device on the SCSI bus chain of devices.

On the CDI 605 players it is even required to disconnect the MM player, since the 605 expects the MM harddisk at another SCSI id as the emulator (0 and 1 respectively).

Terminal connection

The emulator needs a console terminal or some host running a terminal emulation program. The connection is running at 19200 baud, 8 data bits, no parity and one stop bit. A cable suitable for connecting a VT100 compatible terminal is provided and should be inserted into the leftmost of three 9-pin D-shell connectors on the back of the emulator (leftmost while looking at the back).

A little used option, and **unsupported** (!), is to use the CDI 180 player's keyboard as the emulator terminal as well as the MediaMogul terminal at the same time. This is done by connecting the MM keyboard to the emulator's third serial port /t3 with a special adapter cable and then connecting a second cable from emulator serial port /t1 to the CDI 180 player's serial port. A software program, named "emu_plyr", running in the background on the emulator allows to to direct your keystrokes to either the emulator console or the MM system.

This solution involves creation of two non-standard cables as follows:

cable A

DB 9 female connector in emulator serial port 3	DB 25 female connector in CDI 180 serial port
pin 1	pin 8
pin 2	pin 3
pin 3	pin 2
pin 4	pin 20
pin 5	pin 7
pin 6	pin 6
pin 7	pin 4
pin 8	pin 5
pin 9	pin 22

cable B

DB 9 female connector in emulator serial port 1	DB 25 male connector in MM keyboard connector
pin 1	pin 8
pin 3	pin 3
pin 4	pin 20
pin 5	pin 7
pin 6	pin 6
pin 7	pin 4
pin 8	pin 5

The connections can then be made as in figure 7:

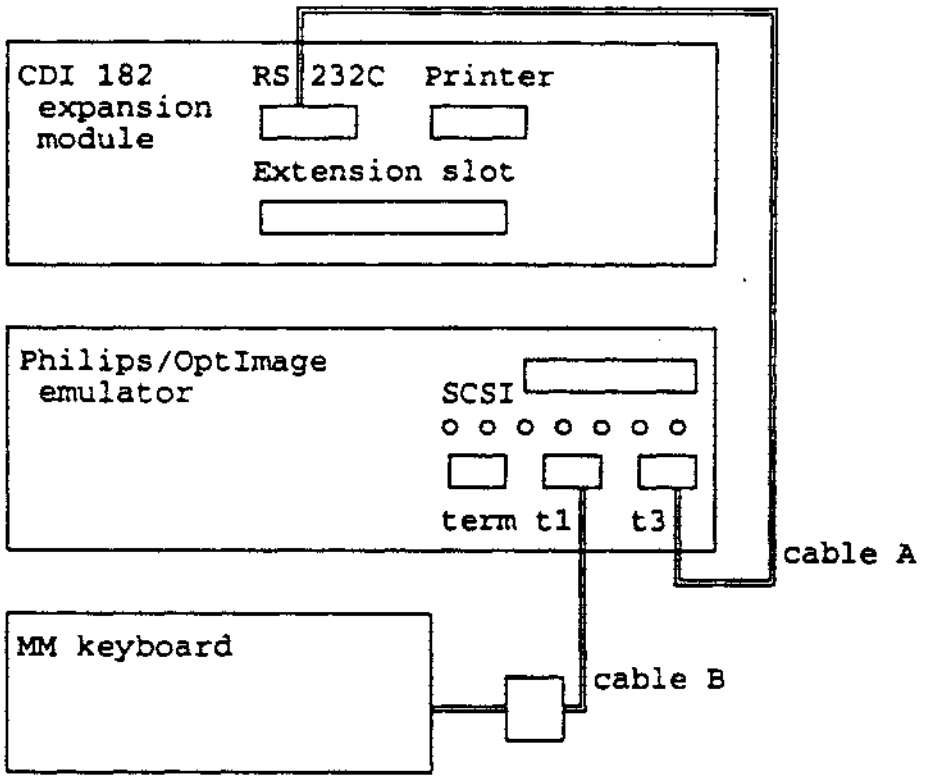


figure 7: MM keyboard as emulator terminal

You may now use the MM keyboard to control both emulator and player. To switch to "emulator mode" enter control-D, to switch to "player mode" enter ESC twice. As mentioned before however this method is not completely reliable, unsupported on the CDI 180 player and untested on the CDI 605.

System software installation

To install the software from the distribution floppies to the harddisk you should use the method described int the Script-to-Disc manual. If your harddisk is not yet formatted you should enter :

```
Emu: chd /d0/cmds/bootobjs
Emu: load -d h2fmt
```

Now start format and instruct it to format the disk at SCSI-id 2 (/h2fmt), to name the disk "ScripttoDisc", to use a cluster-size of 32 blocks minimizing fragmentation and to use an interleave of 1 which speeds up disk accesses:

```
Emu: format /h2fmt -rv=ScripttoDisc -c=32 -i=1
```

Answer "y" to the four questions which follow and patiently wait for format to format, verify and create the filesystem on your disk. Then you can copy the contents of the floppies to the harddisk by entering :

```
Emu: chd /d0
Emu: dsave -resnib200 /h2
```

You may use the same command to copy each of the floppies.

To format the optional, but recommended, second harddisk you may use:

```
Emu: format /h3fmt -rv=BigBoy -c=32 -i=1
```

Discbuilding

First a remark on Scrip-to-Disc's speed. We ran S2D for a medium sized application whose MM harddisk contained some 40 MB of data and the final disc image file was 228 MB in size. Running S2D for this application required about 3 hours ! For another the statistics were 80 MB data, disc image 300 MB and time 6 hours. So please take enough time for the S2D part of CD-I application production, and take into account you will need to run S2D a number of times.

When you are ready to run S2D on a new project you should create a new subdirectory for it and copy the contents of the CDIMAGES directory to your new directory:

```
Emu: makdir /h2/PROJECT
Emu: chd /h2/PROJECT
```

```
Emu: copy /h2/CDIMAGES/* -w=/h2/PROJECT -b=400
```

If you want to re-run S2D on a project again you should clean up the emulator's harddisk(s) by removing any old temporary files (which all have filenames starting with the underscore character) and/or disc image files (which always have a .cd extension).

```
Emu: chd /h2/PROJECT
Emu: del _*
Emu: del *.cd
```

Your directory /h2/PROJECT should now contain:

```
Emu: dir -e
                          Directory of . 09:25:30
 Owner  Last modified Attributes Bytecnt Name
------- ------------- ---------- ------- ----
  0.0   92/02/21 1139  ----r-wr        2 abstract.txt
  0.0   92/02/21 1139  ----r-wr        2 bibliographic.txt
  0.0   92/06/10 1042  --e-rewr     2092 cdi_appl
  0.0   92/06/10 1042  --e-rewr    38350 cdi_audiopro
  0.0   92/06/10 1042  --e-rewr    14262 cdi_inputpro
  0.0   92/06/10 1042  --e-rewr    14290 cdi_inputpro.unmouse
  0.0   92/06/10 1042  --e-rewr     4920 cdi_statepro
  0.0   92/06/10 1042  --e-rewr    32476 cdi_videopro
  0.0   92/02/21 1139  ----r-wr     2390 click.cmono
  0.0   92/02/21 1139  ----r-wr        2 copyright.txt
  0.0   92/02/21 1152  ----r-wr      772 master_script
  0.0   92/04/24 1526  ----r-wr   571908 message.cda
Emu:
```

Running S2D produces a large number of message on your terminal. To make sure S2D continues even if you have not read all of the messages, you should disable the "pause" feature of the OS-9 shell program. If enabled "pause" will stop terminal output, and thus the running cdbuild program, after 24 lines of text have been produced and will continue only after the any key is pressed.

```
Emu: tmode nopause
```

Now locate the startup script of your MM application and run cdbuild with it as the argument:

```
Emu: cdbuild /h0/SCRIPTS/main
```

Now cdbuild will first analyze all your scripts and menu files to determine which audio and video files it needs, check their availability and sizes, verify rtrb can indeed generate a realtime file from them and generate the rtrb-scripts (which have filenames ending with _r). If this first phase reports errors you should check the problem by running MM analyze on the script and eventually modify the script in MM.

The default value used for the so called "safety factor" used by version 1.1 of Script-to-Disc is 156. You must inform the MediaMogul analysis function of this by inserting (or modifying) the line :

#ANALYSIS 156

in the file /h0/Mogul.config on your MM harddisk.

You could check specific scripts by running analysis on the emulator, as follows:

```
Put analysis update floppy in the emulator's flopy drive.
Enter :

Emu: copy /d0/analysis -w=/h2/cmds
Emu: analysis </h0/scripts/some_script
Standard real-time file: 1234567 bytes
Emu:
```

Or you can force S2D to use the "safety factor" you used when running the MM analysis function with the command line option -s to cdbuild as in:

```
Emu: cdbuild -s=75 /h0/SCRIPTS/main
```

If the first phase is completed cdbuild calls rtrb for each of the generated rtrb-scripts to create the realtime files by means of an OS9 shell script it has generated, called "_shell_script" which contains lines like:

```
rtrb -c <_script1_r
rtrb -c <_script2_r
rtrb -c <_script3_r
```

After all realtime files have been created, cdbuild calls master to make the disc image file. The master script is called master_script and it includes three generated files _master_script_include0, _master_script_include1 and _master_script_include2 which contain application specific information.

Command line options to cdbuild, rtrb and master are :

```
Emu: cdbuild -?
cdbuild: edition 123:
Copyright 1991 OptImage Interactive Services Co. L.P.
Syntax: cdbuild [opts] scriptfile
Creates _p files, _r and master include scripts,
runs rtrb to create _g files, and runs master to
create cdi_SCRIPTNAME.cd disc image.
Options:
  -?    print this help
  -r    produce _p, _r and master include files
```

```
        (do not run rtrb nor master)
  -m    produce _p and _r files and run rtrb
        (do not run master)
  -b    process one script only
        (do not process branching scripts)
  -f    do not filter filenames; pass other than
        alphanumerics and _
  -astring1=string2
        alias string2 for string1 in all audio, video, script,
        menu, subroutine, and font filenames
  -c=1  force standard (uncompacted) realtime files
  -c=2  force compacted realtime files
  -s=n  set safety parameter for compacted realtime files
        (default=156)
  -u    Touch_on device=unmouse (default=touchscreen)
  -p    omit redundant copies of preambles from disc image
Emu:
Emu: rtrb -?
rtrb: edition 123:
Copyright 1991 OptImage Interactive Services Co. L.P.
Syntax: rtrb [<opts>]
Function: build real time records using stdin script
Options:
  -v        verbose
  -f        write RTF instead of RTR
  -c        write CTI 'ready' format instead of RTR
  -b=<size> buffer size

Emu:
Emu: master -?

Disc Building Utilities for OS9  - Version 4.0 (alpha)
Copyright 1992 by OptImage Interactive Services Company, L.P.

     master [-qdgelprsctozmv] script-file [-qdgelprsctozmv]

Generates CD-I disc image files using named script

Parameters:
  script-file      Input mastering script file

Options:
  -q     Quiet mode (disable progress report messages)
  -d     Dry run (disable output CD file generation)
  -g     Non Green mode (relax checks about green book specs)
  -e     Generate embedded directories
  -l[=n] Generate n [1..600] seconds of leadout
         (defaults 210 seconds)
  -p     Obey padding requests
  -r     Build runtime maps
  -s     Disable sectors scrambling
  -c     Convert empty sectors with events to data
         sectors in channel 31
  -t[=<tocs-file>]
```

```
          Generate a tracks (or Table Of Content) report
          (<tocs-file> defaults to "tocs.lis")
  -m[=<maps-file>]
          Generate a map report
          (<maps-file> defaults to "maps.lis")
  -z      Generate embedded PMM_RT_MAPs
  -o      Generate PMM_RT_MAPs in "RT_file.pmm"
          Options:   r,o,z - INCOMPATIBLE
  -v      Generate RED image
  -u      Generate CD-READY image
  -xa     Generate CDROM-XA image
  -h/-?   Give this help
Emu:
```

If you find you need to change only a few MM scripts to make the application work correctly you can instruct cdbuild to recreate only those realtime files whose MM scripts have been modified with:

```
Emu: cdbuild -b -m /h0/SCRIPT/modified1
Emu: cdbuild -b -m /h0/SCRIPT/modified2
Emu: master master_script
```

You can do this ONLY if you only modified MM scripts. If you added any MM scripts, or if you removed any of the temporary files, this method will fail and you should start from the beginning.

If your disc image file will exceed the amount of free space on the default harddisk (normally /h2) cdbuild will continue until it cannot write any more data to the harddisk (either error number 217 : segment list full or 248 : media full). The harddisk will be full and your disc image file incomplete. You should remove the incomplete disc image file and edit the text file _master_script_include0 (with the edt editor) to have master write the disc image file on your second harddisk (normally /h3), provided you have connected one. Then you can run master again. The latest versions of S2D provide an option "-o=directoryname" to allow to assign a destination directory for the disc image file.

```
Emu: del *.cd
Emu: edt _master_script_include0
*0001     define album "cdi_SCRIPTNAME" publisher "OptImage"
preparer "David S. Lampert"
E:
```

hit the return key,

```
*0002   volume "cdi_SCRIPTNAME" in "cdi_SCRIPTNAME.cd"
E:
```

now type "d" to delete the current line,

```
E:d
*0002   application file cdi_SCRIPTNAME from "cdi_appl"
E:
```

now type a space (so we enter the line insert mode of edt) followed by the new version of the line we just deleted, in which we prefix "/h3/" to the filename after the word "in",

```
E: volume "cdi_SCRIPTNAME" in "/h3/cdi_SCRIPTNAME.cd"
*0003   application file cdi_SCRIPTNAME from "cdi_appl"
E:
```

now we can check if the correct change was made by listing the entire file, and type q to quit the editor, saving the new version,

```
E:l
*0001   define album "cdi_SCRIPTNAME" publisher "OptImage"
preparer "David S. Lampert"
*0002   volume "cdi_SCRIPTNAME" in "/h3/cdi_SCRIPTNAME.cd"
*0003   application file cdi_SCRIPTNAME from "cdi_appl"
E:q
Emu:
```

Now the file is OK, it will write the CD-I disc image file to the second harddisk named /h3, so we can run master:

```
Emu: master master_script
```

There is a (remote) possibility the amount of memory built into the emulator (2 MB) is not sufficient to complete a S2D run. This can occur if you have more than some 1500 script files. Should this indeed be the case you could try to run S2D on your CDI 605 player, which is equipped with 5 MB of memory although it may take substantially longer to complete the run. To do this you need to connect the harddisk(s) and a terminal to your 605. The correct serial connection of a VT100 or similar terminal is as follows:

```
DB 9 female connector      DB 25 female connector
in CDI 605 serial port 3   in terminal serial port

pin 2 ------------------------- pin 2
pin 3 ------------------------- pin 3
pin 5 ------------------------- pin 7
pin 7 ------------------------- pin 5
pin 8 ------------------------- pin 4
```

You may also connect the MM keyboard to the CDI 605 by switching lines 2 and 3 in the drawing above.

Or if you use a 9 pin serial connector on your PC running some terminal emulation program:

```
DB 9 female connector      DB 9 female connector
in CDI 605 serial port 3   in PC serial port

pin 2 ------------------------------ pin 3
pin 3 ------------------------------ pin 2
pin 5 ------------------------------ pin 5
pin 7 ------------------------------ pin 8
pin 8 ------------------------------ pin 7
```

Another option is to run Script-to-Disc on a Sun workstation, which normally has enough harddisk space and virtual memory, solving the limitations of both emulator and 605. But the SunOS filesystem uses case sensitive filenames, so you should (have) used the same case for all file- and directory names. To copy all your MM scripts and assets to the Sun you may copy all data to a streamer or Exabyte tape drive connected to your player (or emulator) running tar and reading this tape on the Sun workstation using SunOS tar. You will also need the Sun version of Script-to-Disc. An example session could look like:

```
% cd /proj
% tar xf /dev/rst0
% md /proj/cdbuild
% cd /proj/cdbuild
% cp /home/release/S2D/CDIMAGES/* .
% cdbuild -a/h0=/proj /proj/SCRIPTS/myproject
...
% ls *.cd
cdi_myproject.cd
%
```

Talk-to-Disc

Once all video and audio assets have been acquired and converted into file formats suitable for Mediashowcase/CDITalk, they are stored on the MSC harddisk. After all CDITalk scripts have been created and tested the application is completely defined. To create a CDI disc image file you may then run Talk2Disc (T2D) with the project directory as the argument as in:

```
MediaShowCase> talk2disc /h0/SCRIPT/myproject.prj
```

The resulting disc image is then stored in the file:

```
/h0/SCRIPT/myproject.prj/BUILDCD/MASTER/myproject.cd
```

The CDI disc image file should then be copied to an emulator, or you can just connect the entire harddisk to an emulator since both player and emulator run versions of OS9. Then you can emulate the disc image for verification, "burn" it on a WORM disc or copy it to tape to be pressed into a CD-I disc.

The Talk2Disc system includes the following programs: talk2-disc, which reads the CDITalk scripts and asset files to create discbuilding scripts for master and master itself, which assembles all files and other data files into a disc image.

The options to Talk2Disc are:

```
MediaShowCase> talk2disc -?

*** No Arguments given! ***

Talk2Disc (MediaShowcase utility)
(Development Version) 06 July 1992

Usage: talk2disc [<options>] <project dir> [<options>]

Project Directory must be an ABSOLUTE path,
i.e. start with a '/'

Options:

  -?, -h          Produce this help message
  -i[=]<abs path>[:<abs path>]
                  Absolute path to directory/file to Include
  -e[=]<abs path>[:<abs path>]
                  Absolute path to directory/file to Exclude
  -w[=]<abs path>
                  Absolute path to build output directory
  -r[=]<abs path>
                  Absolute path to Runtime directory
  -d              Dry Run.  Do not produce any output.
  -nm             Do NOT make actual CD-I Disc
  -nb             Do NOT make Runtime Modules (Pre-Exist)
  -nw             Suppress WARNING messages (not advisable)
  -nv             Suppress normal status messages
  -nc[=]</dev>[:</dev>]
                  Set device names for case insensitive NFS
  -s<d|m|o|u>     Set target system for disc builder scripts
                  d=dos, m=mac, o=os9, u=unix - default os9
  -c=<filepath>   Where to build the disc image
  -f              Only include Fonts (UCM & PRL) in data file
  -m[[=]<abs path>]
                  Produce Map File
  -x              Who's responsible for this?

e.g. talk2disc /h0/PROJECTS/Bigdemo.prj -fm
```

Emulation

When master has (finally) completed your CD-I disc image file will be in the current directory (or on /h3) with a name like cdi_MAIN.cd, if main was the name of your MM application's startup script. You may then start emulation of it with :

```
Emu: emulate cdi_MAIN.cd
```

or

```
Emu: emulate /h3/cdi_MAIN.cd
waiting to receive map
reading map
Scrambled CD-I Image
+6
stop at sector 0
```

When the message "stop at sector 0" appears you can select the CDI or PLAY button on your connected player to start the application. To end emulation hit control-C.

Options to emulate are :

```
Emu: emulate -?
'cdem9' - emulate a CD-I disc image, edition #33
Syntax: cdem9 [<opts>] <disc image> [<opts>]
Options:
     -c=<path> TOC file from "master" utility
     -g        print debug information
     -o=<num>  file offset byte count
     -r=/<dev> RS control port
     -t=<num>  tracks per groves
     -?        print this help message
Emu:
```

If your disc image is correct you can "burn" a WORM disc with it, provided you have a Yamaha PDS system attached, turned on and loaded with an empty, new WORM disc, with the following command :

```
Emu: burncd9 cdi_MAIN.cd
```

When burncd9 reports it is done you are returned to the "Emu:" prompt.

Options to burncd9 are:

```
Emu: burncd9 -?
CD-I Emulation Utilities for OS-9 Systems - Version 2.0
Copyright 1992 by OptImage Interactive Services Company, L.P.
```

```
Syntax: burncd9 [<opts>] <disc image> [<opts>]
Options:
     -a              CD-ROM XA Image
     -c=<toc-file>   TOC File from "MASTER" Utility
     -g              Print Debug Information
     -t              Test Mode (Don't Actually Burn a Disc)
     -?              Print This Help Message

Emu:
```

Creating CD-I image file tapes

To send a disc image file to another site or the disc pressing factory you can use an Exabyte tape drive and Emulator or Player tape package to write your CD-I disc image file to an Exabyte tape in the so called tar format, which can directly be read on any UNIX system to be written to a WORM or an ANSI labelled Exabyte tape used by the disc pressing factory.

If you want to copy the disc image file to an Exabyte tape, and you have received and installed the proper hard- and software you should first load the required system software with the "StartExaTape" script, rewind the tape with "tape -r" and then start copying with the "tar" command:

```
Emu: chd /h2
Emu: start_exa
Emu: chd /h3
Emu: tape -r
Emu: tar -c cdi_MAIN.cd
```

options to tar are :

```
Emu: tar -?
tar: you must specify exactly one of the c, t, or x options
tar: valid options:
-b N    blocking factor N (block size = Nx512 bytes)
-B      reblock as we read
-c      create an archive
-D      dump record number within archive with each message
-e      swap bytes within 16-bit hunks
-f F    read/write archive from file or device F
-i      ignore blocks of zeros in the archive,
        which normally mean EOF
-k      keep existing files,
        don't overwrite them from the archive
-m      don't extract file modified time
-o      write an old V7 format archive,
        rather than P1003 format
-p      extract protection information
-s      list of names to extract is
        sorted to match the archive
-t      list a table of contents of an archive
-T F    get names to extract or create from file F
-v      verbosely list what files we process
-x      extract files from an archive
Emu:
```

Yamaha Programmable Disc System

The Yamaha PDS system normally consists of two 19" units, the Yamaha YPE 101 Encoding unit and the Yamaha YPR 201 Recording unit. It is possible to connect up to fourteen Recording units to an Encoding unit however, enabling you to create up to 14 equal WORM discs at the same time. Normally the encoding unit comes with a PC expansion board and PC software, but these items are unused in our environment since the emulator controls the Yamaha PDS.

Hardware installation

The Yamaha PDS interface is a special expansion board to be build into the Yamaha YPE 101 (or 102) Encoding unit. To install this board please refer to the "CD-I Emulator for Sun Workstations" manual you received with your software. If you received an YPE 102 you must remove the AES/EBU connector interface board and the first internal board connected to it before you can install the custom interface to the Emulator.

The emulator is connected to this interface and to the Encoding unit. To connect the emulator to the interface use the supplied cable with the 25 pin D-shell connector on the emulator end and the 36-pin Centronics connector on the interface end. Then connect the cable with the large 50-pin Centronics connectors to both the Encoding unit's "PC Control Port" and the emulator. Do NOT connect the Encoding unit to the SCSI connector at the top right or to the parallel port at bottom right of the emulator !

Subsequently you can connect your Recording unit to the Encoding unit with the only matching cable from the Yamaha boxes, normally packaged in the double bottom in one of of the boxes. Use the "Data/Control out" port on the Encoding unit and the "Data control in" port on the Recording unit to connect the cable. If you have more than a single Recording unit you can daisy-chain them by connecting the second Recording unit's "Data control in" port to the previous Recording unit's "Data control out" port. Additional Recording units must be assigned a drive-id different from all others with the small rotating drive-id selector on the top back of the Recording unit. The last (or only) Recording unit must have the terminator (a connector without cable) connected to it's "Data control out" port. See figure 8 below for a drawing of the connections.

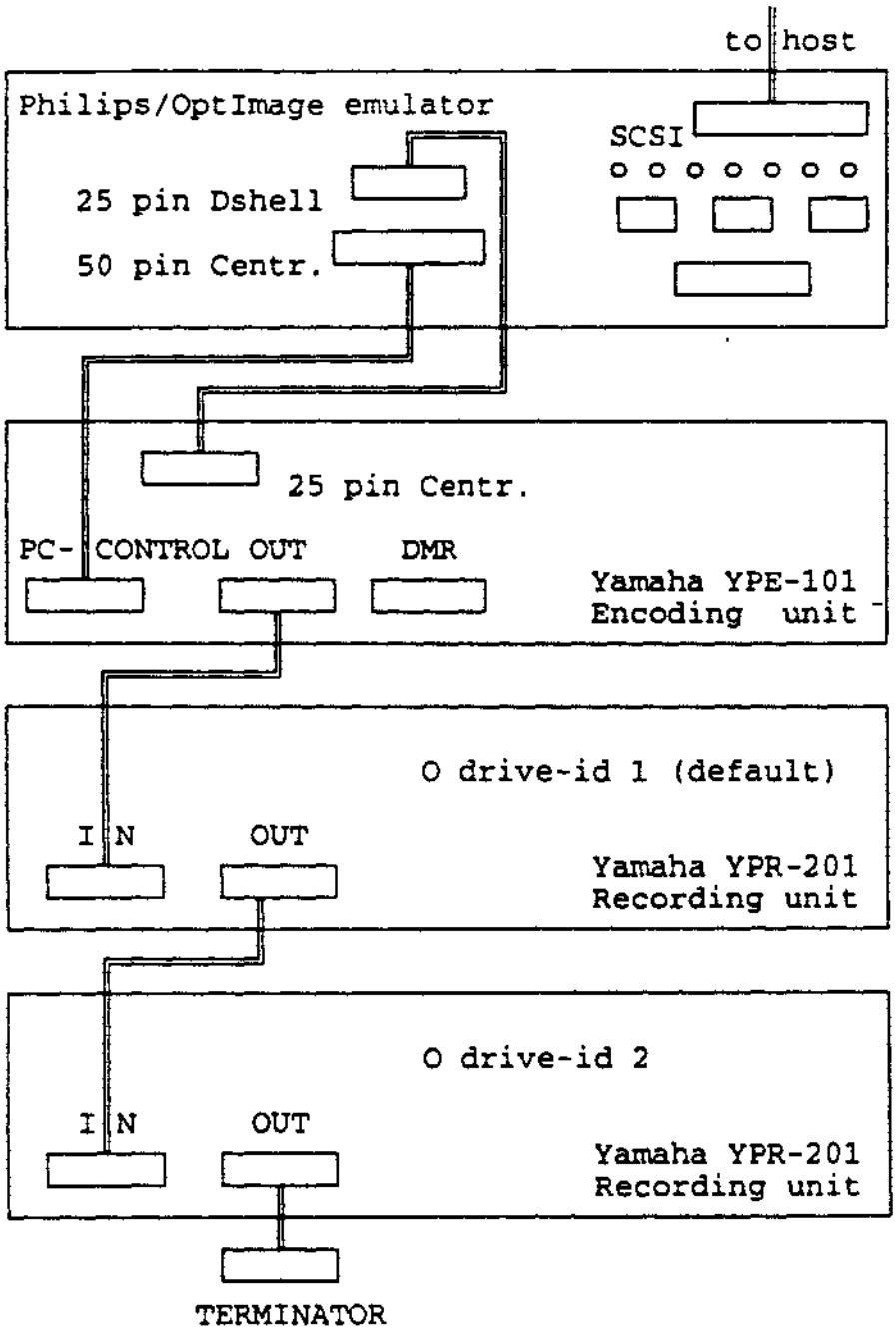


figure 8: connecting emulator to Yamaha PDS

If you have an old version of the emulator (pre-1991) you should check jumper JA2 on the top printed circuit board inside the emulator. If the jumper is not set correctly the Yamaha system will only write lead-in data to your WORM disc, until the disc is full.

To gain access to the jumper you must remove the emulator's cover by removing the single screw at the top-centre of the back of the emulator and then sliding the cover forward and up. You may then identify the stack of three boards, the topmost being named "CD-I Emulator/PDS controller". Jumper JA2 is located near the front of the board, directly to the right of the expansion connector to the other boards. See figure 8.

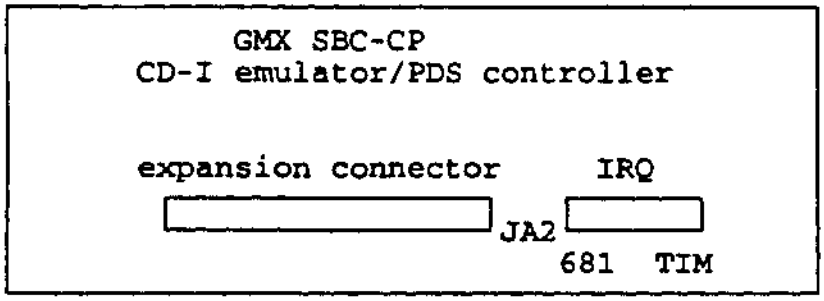


figure 8: jumper JA2 location

The JA2 jumper has three pins resulting in three possible settings: no jumper at all, jumper connecting the centre pin with the leftmost pin called 681 or jumper connecting the centre pin with the rightmost pin called TIM. The correct setting is TIM.

Burning WORM's

The YOD-201 WORM discs used to record your CD-I disc image files are very delicate. You should handle them according to the instructions included with each package of five discs. Especially you should consider the maximum storage (and use) conditions for Yamaha WORM discs, which are 5 to 25 degrees Centigrade temperature and 8 to 60% relative humidity.

Before you start burning a WORM disc it is recommended to warm-up the Yamaha equipment for at least 15 minutes. Subsequently open the tray by pressing the OPEN/CLOSE button or the Recording unit and place the WORM disc inside the Recording unit's tray with the golden side upward. Do not touch either the top or bottom of the disc. Now press OPEN/CLOSE again to close the tray.

Now you can start the burning process by running the burncd software on your host. The burning process will start, closing the tray if it is still open. You should see a display such as:

```
host: burncd cdi.cd

  Disc image size is 09:52.00
  Creating disk map ..
  connected
  Emu: burncd -s=104428800   -c
  waiting to receive map
  reading map
  Scrambled CD-I Image
  +6
  Initializing Yamaha PDS System
  1 drive System
  10:56:20 Starting to Burn CD
  Disc Image Contains 44400 Sectors
  Finished at approx. 11:11:33
  10:56:21 Starting Leadin (approx. 02:20.00)
  10:59:21 Starting Program Area ( 09:52.00 )
  11:00:35 12% Complete
  11:01:49 25% Complete
  11:03:03 37% Complete
  11:04:17 50% Complete
  11:05:31 62% Complete
  11:06:45 75% Complete
  11:07:59 87% Complete
  11:08:53 Starting Track 2
  11:09:13 100% Complete
  11:09:13 Starting Leadout (2:00.00)
  11:11:13 Leadout Complete
  11:11:17 Done.
  Emu:
```

The system establishes a connection between host and emulator, sends a so called map file, which describes the position of the data of the disc image file on the host's harddisk, checks the type of disc image and initialises the Yamaha system. Here is detects how many recording units are connected. Should you have more than one, keep in mind that if one of the recorders generates a fatal error, writing will be aborted and ALL WORM discs will be incomplete and useless. If all is well however, the system starts writing a lead-in area, followed by the data area and finally lead-out.

Also you should not move or shock the table(s) the equipment is placed on and it is recommended not to use the host controlling the burning process for any other purpose while the WORM is written to.

The burncd program running on the emulator is intended to be called from the development host, but for completeness it offers the following options:

```
Emu: burncd -?
CD-I Emulation Utilities for Non OS-9 Systems - Version 2.0
Copyright 1992 by OptImage Interactive Services Company, L.P.

Syntax: burncd [<opts>]
Options:
     -a              CD-ROM XA Image
     -c=<toc-file>   TOC File from "MASTER" Utility
     -d=/<dev>       Specifies the Hard Disk Device Name
     -g              Print Debug Information
     -m=/<dev>       Segment Map Device Name
     -s=<num>        Disc Image Size, in Bytes
     -t              Test Mode (Don't Actually Burn a Disc)
     -?              Print This Help Message
Emu:
```

Error codes

With a complex system such as the development host, emulator and Yamaha PDS some things may go wrong. Possible error messages from the Yamaha PDS via the burncd program are lines with "Err 97" between the normal output of burncd. We have seen errors such as:

```
Err ff
Err 92
Err 97
burncd: Error hardware reset 0
burncd: Error Preparation 2b
burncd: Error Preparation 37
burncd: Error test disc 34
```

The messages you are seeing are the hexadecimal codes that have been obtained from the Yahama from an Intel 8251 status register. Error code ff indicates a cabling problem. The emulator is unable to communicate with the Yamaha system. This error repeats itself so may times, it is capable of crashing the Emulator. Error codes between 90 and 97 are due to an overrun error on the serial line. The bits in the 8251 status register are:

```
Bit 0 - TxRDY
Bit 1 - RxRDY
Bit 2 - TxEMPTY
Bit 3 - Parity Error
Bit 4 - Overrun Error
Bit 5 - Framing Error
Bit 6 - SYNDET/BRKDET
Bit 7 - DATA SET READY
```

These codes should be considered warnings if they occur between the "Initializing Yamaha PDS System" message and the "Starting to Burn CD" or "Test Mode - Starting Dry Run" messages. During this period, the Yahama is resetting and is not servicing the 8251 data register. Status request characters received from the emulator during the reset cause an overrun condition. Only if these codes appear after the "Start" message, do they indicate an actual error condition.

Here is a copy of the error codes we have for the Yahama that are not related to communications:

Hex	Command	Status
24	disc test	command complete
25	start write	command received
26	preparation for write	top of program area
27	all servo off	end of program area
28		end of lead_out area
29		YPR set chilt servo off
2b	LD read power on	illegal command
2c	LD write power on	illegal parameter
2d	LD off (focus off)	
2e	focus on	
2f	tracking on (kick off)	
30	tracking off	
31	kick on	
32	seek	
34	feed on	focus servo not on (not virgin disc or no disc)
35	feed off	focus servo error
36		during kick hold
37	spindle on	tracking servo error
38		maybe track jump error occur
39	spindle off	tilt servo error
3a	tilt servo on	disc check (angular deviation too much)
3b	tilt servo off	disc check (deflection too much)
3c	tray open/close	tracking offset level is not ordinary
3d	tray lock/unlock	seek command illegal parameter
3e		seek command executed error
3f		spindle servo error
40		feed servo error
41	send used time	during write
42		tray can't move
43	send servo status	EEPROM write error
44	send drive status	
46	send current position	LD current over
47	drive reset	
54	command break	command break received
48	char '0'	char '0'
49	char '1'	char '1'
4a	char '2'	char '2'
4b	char '3'	char '3'
4c	char '4'	char '4'
4d	char '5'	char '5'
4e	char '6'	char '6'
4f	char '7'	char '7'
50	char '8'	char '8'
51	char '9'	char '9'
90		serial overrun error
..		(may be ignored)
97		

Philips CDD 521

The Philips CDD 521 Compact Disc Recorder is the latest development for creating WORM discs. The 521 is capable of writing CD-ROM, CD-ROM XA, CD-Audio, CD-I and CD-I Bridge discs (NOT CD-I Ready !) in half the time required by the Yamaha system. But since the 521 was designed to conform the the so called Orange Book, which defines the CD-Recordable standard, the resulting disc will not be fully compliant with the Green Book (so called run-out blocks are added after each track taking two extra sectors and the pause between the CD-I track and any additional tracks is set to 3 seconds, not the default of 2). Also the 521 is quite intelligent, but it assumes a standard track numbering and this prohibits us from creating CD-I Ready discs which put the CD-I track in the so called pre-gap.

And yes, it is possible to connect both the Yamaha PDS and the Philips CDD 521 to the same emulator, although it is not possible to write WORMs with both devices at the same time.

Hardware installation

To use the CDD 521 with an emulator you need a CDD 521 with ROM version 0.77 or higher, special cabling and software for your host and emulator. The 521 is connected to the emulator's SASI interface which is on the emulator motherboard.

Connect the flat cable's pin-header connector to the row of pins named P5 on the motherboard (see pages xi, 17-19 and b-1 of the GMX Micro-20 Hardware Technical Manual and mind the position of pin number 1) and let the flat cable exit the enclosure. Alternatively you may disconnect the 50-pin flat cable from the left of the topmost PCB and move it to the rightmost (P5) connector on the bottom (CPU) board.

Then connect the SCSI cable between the flat cable and one of the 521's mini-SCSI connectors. We use the same SCSI cable for this as the one used to connect the emulator to a Sun workstation.

On the back of the 521 you may see four DIP switches. These are used to set the SCSI-id of the device and they should all be set in the down position to set SCSI-id to 0.

See figure 9 for a drawing of the connections.

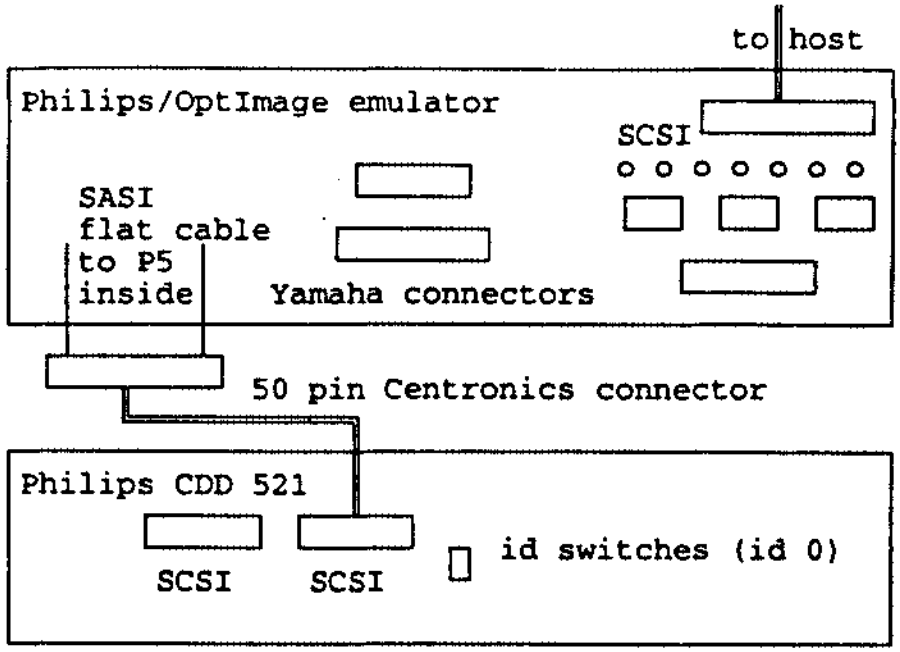


figure 9: connecting emulator to Philips CDD 521

The 521 should be turned on after the emulator has started up,since the 521 does not handle SCSI/SASI bus resets as expected.

Of course you will also need suitable WORM discs such as the Philips Professional CD Recordable 63 minutes or the Yamaha YOD 201 (if they have the required wobble).

Software installation

The emulator boot floppy for release 2 for Sun is required with the addition of the following files (in the alpha release) :

CMDS/burn521	Executable.	Main Application
CMDS/cdd521	Executable.	521 Device Descriptor
CMDS/cdd521dvr	Executable.	521 Device Driver
CMDS/cdd521man	Executable.	521 File Manager
CMDS/diskread	Executable.	Secondary Application
CMDS/seg	Executable.	Segmented File Device
CMDS/segdvr	Executable.	Segmented Device Descriptor
CMDS/segman	Executable.	Segmented File Manager
CMDS/BOOTOBJS/t3	Executable.	SCF Device Descriptor
SYS/startup	Script.	Bootup File for Boot Floppy

And your Sun 3 or Sun 4 system should have the following executables in your path:

```
burn521          Executable. Command Line Interface for 521
emutool          Executable. SunView Application, Sun 4 ONLY
discmap          Executable. Secondary Application
```

To load BURN521 utility, the startup file of the emulator needs to be replaced with the new one provided.

Burning WORMs

To burn WORMs on the 521 you will need a special version of your development host's burncd software. Currently we only have such a version under development for the Sun4/SPARC. This preliminary version consists of the programs burn521, discmap and emutool. The burn521 utility offers the following options:

```
% burn521 -\?

CD Emulation Utilities for Sun 4 - Version A.2.1
Copyright 1992 by OptImage Interactive Services Company, L.P.

Syntax:   burn521 [<opts>] {<disc_image_file>} [<opts>]
Function: Burn a Philips CDD 521 WORM Disc
Options:
  -?              Print this help message
  -d=/<dev>       Emulator Device Containing Disc Image
                  (default: /h2@)
  -f              Force Creation of New Image Map
  -t=<TOC_file>   Use <TOC_file> from 'Master'
  -u              Image is Unscrambled and Byte Swapped
  -x              Test Mode - Do NOT write to Disc

NOTE:  The '-t' parameter is required.
```

Notice the TOC file MUST be used to create a WORM disc on the 521, so you MUST use the -t=toc-file option to master.

The new version of the emutool utility offers support for both the Yamaha and Philips WORM systems, selectable trough the Options button.