```
==========================================
   T H E   " U N - O F F I C I A L"
     PLAYSTATION DEVELOPMENT FAQ

              CD
          CONFERENCE
==========================================

           Release v1.3
      Last Updated: February 29, 1996
```

[1.] CD

[1.1.]: CD Hardware

[1.1.1.]: What is the arm movement track to track stepping time for the CD?

A seeking action is not repeat of single track seek. The specification of a CD-ROM drive is described by "minimal access time".
A minimal access time is as follows:

minimal access time = 1/3 seek + settling time = 300msec.

  short seek (5min) + settling time = 300 msec.
  long seek (60min) + settling time = 1000 - 1300msec.

If the seeking distance is very short (less than 30 sec), the stepping
      action will be very fast. Because it goes without a pick-up unit
      moving. (only moving the lens).

Also, by using the equation above you can determine how many megabytes
      you can seek per second on a cd.

[1.1.2.]: Are there any ways to detect the end of CD-ROM XA data?

There are no ways to detect the end of CD-ROM XA playback automatically.
Determine the end of the playback by polling the sector ID periodically.

[1.1.3.]: Does SYSTEM.CNF have to be written on a CD-ROM if every system setting is default?

Yes. Since the boot file name must be designated on the master disc,
SYSTEM.CNF is indispensable even if every setting is default.However,
SYSTEM.CNF is unnecessary for the test disc.

[1.1.4.]: Before the end of a seek, can the new seek command be issued?

Seek requests are carried out sequentially, and storing the requests is
not in the specifications. Thus, issue the new seek request after the
preceding seek.

[1.2.]: The CD-ROM DTL-H2010?

[1.2.1.]: What is the program to use to boot from a CD?

The following sequence of commands will execute the CD program placed in
the DTL-H2010. Example:

```
    resetps 1
    pause
```

```
run c:\<psx path>\bin\snpatch
pause
run c:\<psx path>\cdexec\selcd
pause
run c:\<psx path>\cdexec\cdexec
```

[1.2.2.]: Programs cannot be started up with the debugging station, but can be booted with DTL-H2000 and a CD-ROM. What is the cause?

There are 3 types of tests on H2000 as described below.

1) Executing the following commands on DTL-H2000 and then booting from the CD.
DOS> run patchx
DOS> run cdexec (See 'Note')

  2) Setting the disc on H2010 and then executing the first boot CPE file. In other words, the boot file is downloaded, and the data file is read from the CD.

  3) Executing without discs and the CD functions as follows.
* Running after executing 'pqbload' for the data
* Using the PCread() function for data file reading

Based on the results from the above 3 ways, some causes may be listed up.

Note: 'cdexec.cpe' is a module for booting from the CD with patchx executed, and the source is attached to the library CD.

Case1: Works properly under the condition '3', but not '1' or '2'.
* Is the number of files or directories under the limitation?   With PlayStation, the number of directories on a CD-ROM must be 40 or less, and the number of files in a directory must be 30 or less. If they are over the limitation, the files may not be detected by the CdSerachFile() function, etc.

* Are capital letters used for the path names necessary for the CD functions?
Refer to the following example.
     CdSearchFile(&file,"\\DATA\\TITLE.TIM;1");
     LoadExec("cdrom:\\MODULE\\TITLE.EXE;1",0x801ffff0,0);

* Are the CD functions used with the proper procedure?
Check again referring to the sample if the next CdRead() is not called before the end of the preceding CdRead()'.

Case2: Works properly under the condition '2' and '3', but not '1'.
* Is the boot file on a CD-R in 'PS-X EXE format'?
Use CPE2X in order to create a file which is read from a CD to be executed.

* Is the boot file on CD-R named 'PSX.EXE'?

With PlayStation, as far as the name is not specified in SYSTEM.CNF, it is named 'PSX.EXE'.

* Is the number of files or directories within the limitation?   Refer to the case 1.

Case3: Works properly under the condition '1', '2', and '3'.
* Is the license data written on CD-R?
If the screen halt (PS mark doesn't turn up) when H1000 boots up,it means that the license data isn't recognized.
Designate 'licensej.dat' with the full path in [System Area File] of [Additional Information] on the volume screen of CD-ROM Generator.

* Isn't the pollhost() function executed?
The pollhost() execution will cause H1000's hang-up.

  * Is it a module which can operate on the PlayStation memory capacity (2 Mbytes)?
Doesn't it access the address, 2 Mbytes or more?
Check it referring to a map file.
If the work area is allocated with the absolute address designation, check the area also.
Moreover, whether the address more than 2 Mbytes is accessed or not can be checked with setmem(2) easily.

* Isn't the PC file system such as PCopen(), PCread(), PCclose() or 'open("sim:xxxx")' used?
The use of the PC file system on H1000 will cause the hang-up.

* Isn't the system area accessed irregularly?
Check if the area from 0x80000000 to 0x80010000 are not destroyed.

* Isn't the last file on CD-R read more than its actual length?
In this case, it is reported that the processing continues without problem on H2000, but that the CD-ROM system may be out of control on H1000. Handle this case with such procedures as writing dummy data onto the end of CD-R.

[1.3.]: CD Emulator

[1.3.1.]: Does the emulator uses any of my RAM (i.e 8 megs installed)?
The emulator is completely independent of PSX memory configuration and it doesn't make use of any PC high memory so either way the answer should be yes.

[1.3.2.]: Is interrupt used for anything other than cdmon?
Currently only CDMON will make use of this interrupt.
When you are using boot image 2.3 (the one that can drive CDMON)the
        emulator will regularly disconnect from the HD, connect to the PC,
        send CDMON info to the PC and then disconnect from the PC). For
        this version of the boot code to work at all you must have a fast
        interrupt response - no response or a slow response (e.g. if

running it under windows) will not be able to turn the SCSI bus
around fast enough.

[1.3.3.]: What hard disk drives can be used with the CD emulator?

The only models which SONY certifies to work with the CD Emulator are
the MICROPOLIS 3243 AV Ext SR2000 and the MICROPOLIS 4221 AV Ext
SR2000.

Also, the MICROPOLIS 1936 AV 3 gig will also work, but is not officially
supported.

Other developers have used many different types of AV SCSI hard drives
and there has been no problems reported.

[1.3.4.]: What does the message "SCSI error reading disk sectors per track, Error 0x0300"?

This message will appear if your IRQs and DMAs are conflicting. You may
have to determine what IRQs and DMAs you are using for each of
your board so they do not conflict.

You may run the Microsoft Dos program "msd" which can give you a listing
of what is being used.

It is suggested also that you reset your IRQs and DMAs
for the dexbios and cdbios. If they exist in your AUTOEXEC.BAT file,
you should take them out until the conflicts are resolved.

Also, SCSI ids of 3 or greater and less than 7 seem to work. SCSI ids 0
- 2 and 7 seem not to work.

[1.3.5.]: "My CD emulator runs 2 to 3 times slower than the debugging station."

Use CDBOOT16.BIN/CDBOOT26.BIN or later version.
CDBOOT18.BIN/CDBOOT28.BIN for best result.

[1.3.6.]: Installing Emulator Hardware and Software.

Make sure that the CD emulator card is set at the default SCSI
address of 7.Set the CD emulator hard drive to address 3,4, or 5. People
have reported problems with addresses 1 and 2 Addresses 0, 6 and 7 are
used by the board.
Hook the internal cable up to the DTL2000 boards as shown in
manual(don't laugh, it has been forgotten). Choose a base address,
interrupt, and DMA channel that does not conflict with other cards.
Choosing the wrong DMA channel will occasionally take your BIOS out. It
is a great idea to back it up first if you are unsure which DMA channels
are available.
If you are unsure what address, interrupt, and DMA setting to use on a
first time installation, take all of your other boards besides DTL-H2000
and emulator board out (network, sound cards, etc)

Installing software:
Do not use the /s option when installing CDBIOS

When calling CDDISK, remember that the number passed as a parameter is
the SCSI ID of your CD emulator hard drive. For the example of
initializing a hard disk with SCSI ID 3 the call would be

        CDDISK -n 3

where the -n option specifies initialization or re-initialization. You
only need the -n option once. When you load the system area, load

CDBOOT16.BIN or CDBOOT26.BIN

(available on the BBS). These are the current versions of the boot
programs. Do not use the old versions. CDBOOT26.BIN allows you to
monitor the emulator for debugging. Some people have trouble with it.
Try

        CDBOOT18.BIN

first to make sure your system works, and then use

        CDBOOT28.BIN.

When calling BUILDCD, the -s option specifies the SCSI ID and partition
of the emulator hard drive you wish to access.

For example:
        BUILDCD -s3:1 CD.CTI

will access SCSI address 3, partition 1.

When running RCUBE or any other program:

RCUBE is ancient, and the RESETPS 0 call that brings it to life does not
work for every program. Here is a list of commands that will launch
anything:

        RESETPS 1   (Version 1.03 is the newest)
        Pause
        RUN SNPATCH (Currently better than PATCHX)
        Pause
        RUN SELEMU (You only need to call this once after power-on or
after a SELCD command)
        Pause
        RUN CDEXEC (instead of RESETPS 0)

The pause commands indicate that you should wait for the commands to
complete, or you will get "Cannot connect to target" error messages. I
use the DOS batch file command PAUSE and hit keys between each command.
Needless to say, you will have to insert path names where appropriate to
access RESETPS, SNPATCH, and SELEMU. Under normal circumstances, RCUBE
takes a max of 20 seconds to begin execution.

"Cannot connect to target" is a message that means the DTL-H2000 needs
to be reset. It is not an error message from the emulator.

Finally, if you have just installed your emulator card, changed SCSI IDs, reset the DTL2000 a zillion times, re-installed your BIOS after the emulator lunched it, kicked the CPU a few times while launching multiple DOS sessions with complicated communications packages, loaded RCUBE from different SCSI IDs, and had 106 "Cannot connect to target" messages all in one power-on session, it is advised that you power everything down, power it all up again, and start by rerunning CDDISK -n ## fresh. A cold boot does wonders for this process.

[1.3.7.]: CDBOOT26.BIN does not work on some systems?

The configuration for CDBOOT 2.6 is more critical than 1.6. The following points are the things to check if you find that 1.6 works ok but 2.6 does not:

1. The PC side of the system must be set to scsi id 7 (i.e. all scsi id jumpers on - the default configuration).

2. An interrupt jumper must be fitted on the cd emulator card and it must not clash with another interrupt in the system.

3. You must have the latest version of cdbios.

4. You must specify the interrupt number to cdbios with the /I switch.

5. You must use the latest version of resetps.

6. Do not run under windows. There will be a windows version of cdmon later..

7. The cause is cdboot26.bin and cdboot16.bin fixes the problem. The symptoms are either a rampant flashing of access light, or no light flashing of access light.

[1.3.8.]: Which SCSI addresses are taken by the PC?

SCSI address 6 is reserved for the communication channel on the disk side of the emulator. SCSI address 7 is used on the PC side of the emulator by default. This can be changed, but CDMON and CDBOOT26.BIN require the address to be set to 7.

[1.3.9.]: What is the difference between CDBOOT15.BIN and CDBOOT25.BIN?

CDBOOT26.BIN allows feedback to be sent using CDMON to see what the emulator is doing.

[1.3.10.]: What is the The /s switch?

The /s switch is *not* for telling cdbios what the scsi id of the drive is. It is for setting the scsi id of the pc side of the system. In general people don't need this option at all. It may be deleted from future versions of cdbios to avoid confusion.

[1.3.11.]: What does the message "Target does not respond?"

First check to make sure your ISA boards fits snuggly in respective slots. Then make sure the cdbios command line settings match the actial board jumper settings.

[1.3.12.]: Why doesn't program reading from CD-ROM Emulator with the functions such as Load()/Exec() succeed?

The _96_init() function must be executed in order to use the Load(), open(), or read() function (in other words, to use ISO-9660 file system). Accesses to files on a CD-ROM or CD-ROM Emulator must be designated in the format cm: PROG.EXE;1'.

[1.4.]: CD Generator

[1.4.1.] Is there a way, or can one be added, in CDGEN to specify MINLENGTHS for files?

This would allow the SCSI disk image and the CD-ROM image to look the same. It would also allow more options for reading data off of the CD.

Not currently, I passed the request along. For now, you might just pad your files with blanks if you wanna do this.

[1.4.2.]: What is the rule to create a CD-ROM with sub-directories?

If there are sub-directories, the files in a directory must be allocatedcontinuously on the CD-ROM. Check the location of each file in LAYOUT mode when creating it.

```
< Good example >
    \
    \AAA\
    \AAA\DATA1.DAT
    \AAA\DATA2.DAT
    \BBB\
    \BBB\DATA3.DAT
    \BBB\CCC\
    \BBB\CCC\DATA4.DAT

< Bad example >
    \
    \AAA\
    \BBB\
    \BBB\CCC\
    \AAA\DATA1.DAT
    \AAA\DATA2.DAT
    \BBB\DATA3.DAT
    \BBB\CCC\DATA4.DAT
```

- ----------------------------

[1.4.3.]: What is the rule to use a 71-min. type CD-R for creating a master disc?

The recording time of CD-ROM Generator for a 71-min. type CD-R must be set at 74 min.

[1.4.4.]: What is the rule to determine the data layout structure on a CD-ROM?

With the CD-ROM for PlayStation, every data must be stored in the first track. The track structure will be either of the following.

```
 1) No CD-DA
    Track 1 only (Data track)

 2) Has CD-DA
    Track 1:  data track
    Track 2 and later:  CD-DA track
```

Multiple CD-DA tracks are available.(up to Track 99) It doesn't matter whether the source files for CD-DA are in the directory structure or not.

[1.4.5.]: Why is the file number not incremented automatically when writing some XA files on a CD with CD-ROM Generator?

The file number is incremented automatically only when the "file interleave" is carried out for multiple files in an area. Thus, except for this case, the file numbers will be all 1 as default, (in the case of sub-header attachment). However, manual-designated numbers would take precedence.

Note that the file interleave cannot be designated with CD-ROM Emulator.

[1.4.6.]: Is 1350 the maximum number of files written on a CD (45 directories x 30 files)?

As described on the page in Library Reference Vol. 1 (DTL-D2140A), the list supports only the range stored in 1 sector (2048 bytes). Based on this, approximate number of directories and files will be 45 and 30. However, since the list was variable-length structure, more files can be handled if their names are short.

[1.4.7.]: How can a file time-sheet be obtained with CD-ROM Generator?

A mini check window will turn up by pushing REC button. At this time, a temporary file named 'C0000000.LOC' is under '\CDTMP'. In this file, the location information is stored in the text file format. Thus, convert the target file location into minute/second/sector. Note that the absolute sector 0 becomes 0 minute 2 seconds 0 sector.

[1.4.6.]: When multiple files are designated with the CD-ROM Generator, they are sorted in order of their names automatically. Are they allocated according to the sorted order on a CD-ROM?

The CD-ROM Generator doesn't adopt the sort by the names. In [Directory] mode the screen display is carried out in alphabetical order, but eh allocation on the CD-ROM is determined according to the order by the user's designation as seen in [Layout] mode.

[1.4.7.]: How can a part of RGB data be replaced when creating moving pictures? And how if the RGB file is stored separately in multiple drives?

Save the compressed result in 'bs' format not in 'str'. 'bs' is 1-frame 1-file format, so it is possible to interchange the data partly by interchanging the files. If the original data is stored in multiple separate drives, compress it in 'bs' format drive by drive. In this case, set the appropriate value for the frame number of a output file name. For example, when the rgb file is separated as follows,

    Drive C: f0000.rgb _` f0999.rgb
    Drive D: f1000.rgb _` f1999.rgb

the appropriate output file name will be obtained by the following designation.

    Output file name when compressed on C drive: f0000.bs
    Output file name when compressed on D drive: f1000.bs

After compressed in 'bs', these are transferred to 1 drive, and converted to 'str'. The conversion from 'bs' to 'str' will be carried out with the interleave functionality of MovConv.

This is a simple example to interchange the data. Since the edits of more general 'str' or 'bs' data are not supported, perform the complicate edits before compression.

[1.5.]: CD-ROM Burner

[1.5.1.]: What card is compatable with the 900E CD-ROM Burner?

The CDW900E presently only works with the following family of Adaptec cards 154x.

[1.5.2.]: What is the difference between the 900E's vs. 900N model?

The CDW 900N functions as a read-only player for Playstation disks. It may well write CD-ROM disks, but it cannot write Playstation master disks. To write Playstation master disks a CDW 900E is required.

CDW-900N is a specially modified CDW-900E. In testing site, the testing team will receive around 8 master discs (By the way the number can be decided at your end considering developers and licensees' situation locally.) from developers or licensees. Then the testing team will keep 3 discs for sending to factory for the last stamping after testing and use 5 discs for the testing. Sometimes testing team need more discs. For example, when they find more testors are available or necessary to check the title etc., the team need more discs. Then they have to create more discs by duplication. Of course the testing team efinitely have CDW-900E. Howver, CDW-900E is basically just a recorder. record (or write) data on discs but cannot be player. So, we asked factory to design CDW-900N that has an ability of player. By the way after the process our testing team select one best disc out of 3 discs kept separately and send the best disc to factory after completion of testing.

[1.5.3.]: Steps for making a debug station CDs?

The following are steps taken to make debug station CDs:

1) Remove PCread, polehost and etal.
2) Link with 2mbyte.obj
3) Perform a cpe2x /C[area] on the .cpe file.
4) Add files to the CDGEN.
   a) Choose the correct file type for XA items.
        - Use Mode 2 Form 1 for game data.
        - Use Mode 2 Form 2 for XA files.
        - Select both Form 1 and Form 2 for a combined   Audio and
Video file.
        - Use the file type button to set it for each file.
        - Standard file is Mode 2 Form 1.

5) Using the additional information dialog button in the volume panel of
CDGEN, set System Area File to the path of your company's
   license.dat file.
   Example:
        c:\cdgen\licenseA.dat

6) Using the master dialog button in the layout panel of CDGEN, set
   License Area to:
        - J if you have a Japanese debug station.
        - A if you have an American debug station.
        - E if you have an European debug station.

7) Set the minutes to 74 minutes. However, you should use 71 minute
media. A 74 minute media might work but we do not support it. You must
use the CD-R71PS for the mastering process for submissions to SONY.

Please use it at your own risk.
   a) Press the record (REC) button. The double speed record should
work ok.

8) Always verify the baked disc after recording.

[1.5.4.]: Where did my licenseX.dat file go?

First you will need to get a licenseA.dat file from SONY.
The license file has been posted on the BBS along with the CD Mastering
Process document.

After burning your disk you will not see the licenseA.dat file on the
final discs because it is stored in the system area. The system area is
in the lead-in area which is unreadable.

[1.5.5.]: Why do we have to use the 71 min. media?

The density of the 74 minute media is different from the 71 minute
media. The SONY PlayStation has been tuned specifically for the 71 min.
media which make is more reliable. Using a 74 min. media can increase
your error rate.

[1.5.6.]: Can we use the CDW950E with the CDGEN software?

The CD-ROM generator doesn't support CDW950E because it is not
compatible with the SCSI protocol nor does it have the capability to
create a master disc.

[1.5.7.]: Where is the location of the factory we will use to burn black disks?

For US/Canadian market, DADC is the factory for burning black discs

[1.5.8.]: Are we able to burn black CD's with either the CDW 900E or CDW 900N? If so, how?

No, we cannot. Black disc can only be stamped at factory in mass
manufacturing.

[1.5.9.]: Basic problems going from emulation (dev system) to debugging station (blue playstation):

Two meg problems
Make your program fit into two meg minus 64K used by the Rom kernel.
Generate and check your map file to be sure executables fit if you are
unsure. Check your malloc() calls to be sure they succeed. Remove
references to memory outside the 2Meg memory map. Your 2Meg memory map
is between 0x80010000 (the 0x10000 is the 64K used by the kernel) and
0x801FFFFF. Either link with 2MBYTE.OBJ or use _ramsize and
_stacksize(specify as externs). Just because it compiles with 2MBYTE.OBJ
does not mean it will fit in 2Meg. That is up to you.

Burning with CDGEN.EXE
Put LICENSEA.DAT into the system file area using the "additional
information" button on the "volume" screen of CDGEN. You do not need to

include LICENSEA.DAT in your list of files on the "directory" screen.
Put a "A" in the System area box under the "master" button on the bottom
of the "layout" screen of CDGEN. Be forewarned that this "J" is not
saved with the rest of the information when you save your .CCS file in
CDGEN. You must reenter it every time.
If you get the license information entered correctly, the debugging
station (not a development system) will display a black screen with the
"PS" logo and licensing text after the white screen with the gold logo.
If the system hangs on the black screen, the problem is in the program,
not the licensing information.

Development system to debugging station transition
      Remove Pollhost() calls.
      Remove PCIO calls.
      In library 3.0, CdSearhFile(), has some bugs that show up on the
debug station. These bugs are fixed in library 3.1 and later. With lib
3.0, CdSearchFile() may work under emulation and the black development
CD-ROM drive, but fail on the debugging station. A solution is to hard
code your files and their locations into your program. Although this is
less flexible, the location of the files will be fixed on the final CD-
ROM anyhow. You can use CDGEN to do a mock layout with your file
structures built in, then go back and fill them in with the correct
locations. If you need it, the formula for converting between time and
absolute sector is as follows:
   Absolute sector = (minutes * 60 + seconds) * 75 + sector -150

[1.5.10.]: How to specify the Volume Descriptor and Master Information

1.Volume descriptor

 The fields in the Volume-mode.

NECESSARY
 [Application]
  Always "PLAYSTATION"

 [Creation date]
  Automaticaly filled, but can be modified.

NOT NECESSARY
 [Volume],[Volume set],[Publisher],[Data Preparer],
 [Copyright],[Abstract],[Bibliography],[Modification date],
 [Expiration date],[Effective date]

2.Additional Information

 The fields in the dialogs that appears when you press
"Additional information" button in Volume-mode screen.

NECESSARY
 [System Identifier]
  Always "PLAYSTATION"

 [System Area File]
  The full path for license data file.
  (licencej.dat/licencea.dat/licencee.dat)

```
NOT NECESSARY
 [Volume Descriptor File]
```

[1.6.]: CD Library

[1.6.1.]: What does CdInit do?

```
It tries this several time
      init some hardware/library variables
      ResetCallback
      Hook CD system callback
      reset SPU volume
      reset CD-ROM volume
      reset SPU
      CdlReset
      CdSync

if successful above, it calls these functions, hooking default callbacks
      CdSyncCallback
      CdReadyCallback
      CdReadCallback
```

[1.6.2.]: What does CdInitFilesystem do?

It is very similar to CdInit. It calls the same initialization routine
that CdInit does, but CdInit tries 4X before bailing out.

It does not work reliably. Do not use it.

[1.6.3.]: What is the maximum number of directories on PS-X ?

```
      directory:      30
      files:          45 (including subdirectories)
      total files:    30*45
```

[1.6.4.]: Is a directory listed as file in the parent directory?

 Yes.

[1.6.5.]: Is a subdirectory considered a file in the parent directory?

 Yes

[4-6-6]: Is there any error detection/correction of XA-ADPCM or CD-DA
data?
 Yes

[1.6.6.]: Does CD-DA use the CD_ROM subsystem local memory?

Yes

[1.6.7.]: Will the function CdReady() be in the final library?

Yes:

CdReady
Wait until CD-ROM sector data is setup in local buffer

```
        Synopsis:
                int CdReady(mode, result)
                int mode;
                u_char result[8];

        Argment:
                mode    0: block until one sector data is setup.
                        1: return current status

                result  the latest status. Its contents  depends on
                        the latest primitive command.

        Return:
                CdlDataReady    Sector data ready
                CdlDiskError    Error detected
                CdlNoIntr       Data not ready
```


[1.6.8.]: Will the function CdGetStatus() be in the final library?

No. But CdGetStatus can be re-written using CdControl() as follows:

```
        int CdGetStatus(u_char *status)
        {
                u_char  result[8];
                int     ret;

                ret = CdControlB(CdlNop, 0, result);
                *status = result[0];
                return(ret);
        }
```


[1.6.9.]: What is the possible status of CdGetStatus()?

```
#define CdlStatPlay     0x80    /* playing CD-DA */
        #define CdlStatSeek     0x40    /* seeking */
        #define CdlStatRead     0x20    /* reading data sectors */
        #define CdlStatStandby  0x01    /* spindle motor rotating */
```


[1.6.10.]: Why does CdGetToc2 () exist?

There is no means. It is only for compatibility to very old version.

[1.6.11.]: Should you enter a critical section when you execute CdSyncCallback() etc ?

All callbacks are executed in a critical section.

[1.6.12.]: Does calling CdSearchFile() cause the disk to start spinning then stop again ?

After calling CdSearchFile(), the disk is in pause status, not stop status.

[1.6.13.]: What does the function CdSync() wait for?

It waits for the end of CdControl(). Since CdControl() a is non-block function, CdSync is used to detect the end of CdControl.

[1.6.14.]: What does the function CdReady() wait for?

It waits for the sector data ready. CdReady returns CdlDataReady when sector data is setup.
It is used both for data reading and report in CD-DA report mode.

[1.6.15.]: What does the function CdReadReady() wait for?

CdReadReady() waits for the end of CdRead(). It is a higher level function, and it uses CdReady() and CdSync() internal.

[1.6.16.]: What does the setting mode CdlModeSize1/CdlSize0/CdlSubHeaderFilter

Here is the mode of CD-ROM.

CdlSubHeaderFilter equals to CdlModeSF.
AutoReport equals to CdlModeRept.

CD-ROM has 3 sector modes (2048/2340/2328byte).
But considering error detection, we strongly recomend that you use MODE1.

You can use AutoReport only in CD-DA. Report data can be catched CdReady(), or callback in CdReadyCallback().

```
#define CdlModeSpeed    0x80    /* 0: normal speed        1: double speed
*/
#define CdlModeRT       0x40    /* 0: ADPCM off           1: ADPCM on
*/
#define CdlModeSize1    0x20    /* 0: 2048 byte           1: 2340byte
*/
#define CdlModeSize0    0x10    /* 0: -                     1: 2328byte
*/
#define CdlModeSF       0x08    /* 0: Channel off         1: Channel on
*/
#define CdlModeRept     0x04    /* 0: Report off          1: Report on
*/
```

```
#define CdlModeAP        0x02    /* 0: AutoPause off     1: AutoPause on
*/
#define CdlModeDA        0x01    /* 0: CD-DA off         1: CD-DA on
*/
```

[1.6.17.]: Should the callback function set take two parameters and what are these?

```
        result:   latest status. The contents depends on the latest
primitive command.

        intr:     Interrupt cause.

                  CdSyncCallback/CdReadCallback
                  CdlComplete Command complete
                  CdlDiskError    Command error

CdReadyCallback:
                  CdlDataReady    Data is setup.
                  CdlDiskError    Error detected
                  CdlDataEnd      End of Disk
```

[1.6.18.]: Will the final callback functions work this way and what will they be?
Yes. Callback interface is lower level one than Event handler.

[1.6.19.]: What are the meanings of the two parameters in CdSetFilter?
CdSetFilter can be expressed using CdControl() as follows:

```
int CdSetFilter(u_char file, u_char chan)
        {
                u_char  param[4];

                param[0] = file;
                param[1] = chan;
                return(CdControlB(CdlSetfilter, param, 0));
        }
```

[1.6.20.]: Can you filter selectively on several channels ?
No. you have to select one of several channels at the same time.

[1.6.21.]: How is channel filtering turned off? Is it by changing the mode?
 Yes. You can turn off clearing CdlModeSF to 0.

[1.6.22.]: Is CdDePause() being removed from the library?
Yes. CdDePause() is same as follows:

```
        CdDePause(void)
        {
                CdControl(CdlPlay, 0, 0);
        }
```

[1.6.23.]: What commands do the CdlModeSize1 and CdlModeSize2 conmmands effect?

CdlModeSize1/2 should be fixed 0 (mode0) because of error corrections.
We strongly recommend that all data should be stored by MODE0
(2048byte/sector)

[1.6.24.]: What is the file number in CdSetFilter

File # is used to select the playing channel of multi-channel XA-ADPCM
disk.

[1.6.25.]: Why do the debugging stations require licence data?

To avoid confusion and to keep compatibilty to real PlayStation.

[1.6.26.]: Will the developers get a copy of this file so they can burn gold disks for the debugging station?

Yes. LicenseA.dat file is located with the CD Mastering document on the
BBS.

[1.6.27.]: Is there any reason not to set modes such as CdlModeRT or CdlModeDA when not doing ADPCM playback, DA play etc?

Since default mode is CDlModeDA, DA-mode is taken if no mode is set.

[1.6.28.]: Does mode CdlModeRT allow or inhibit retries ?

No. CdReadN retries even when CdlModeRT mode. You can use CdlReadS to
inhibit retry.

[1.6.29.]: What happends to ADPCM play back if CdlModeRT and CdlModeReadN are set together?

CdReadN retries even when CdlModeRT mode.

[1.6.30.]: What are all the possible statuses returned by CD functions?

```
  /*
 * Status
 */
```

```
#define CdlStatPlay     0x80    /* playing CD-DA */
#define CdlStatSeek     0x40    /* seeking */
#define CdlStatRead     0x20    /* reading data sectors */
#define CdlStatStandby  0x02    /* spindle motor rotating */
```

[1.6.31.]: What status can you expect from which function?

It depends on the current status, but most status can be
predictable.(i.e. after CdlPlay, the status have to be CdlStatStandby
|CdlStatPlay etc)

[1.6.32.]: What are the differences between Setloc and SeekP?

CdlSetloc only set the 'target position'.
CdlSeekP set the 'target position' and start seeking toward this
position.

[1.6.33.]: Is XA-ADPCM decompressed in the CD-subsystem local memory?

Yes, and XA-ADPCM stored as XA-ADPCM in local memory.

[1.6.34.]: Should I pause a CD before seeking?

 No. It does not matter.

[1.6.35.]: Can files on a CD-ROM be read by the open() function within a critical section?

In the critical section the CD interrupt is also prohibited. Thus, the
function cannot be called. In this case, it is necessary to execute the
ExitCriticalSection() function to get out of the critical section.

[1.6.36.]: Where are the compressor functions in libpress?

  Unfortunately, the compressor functions are not released at this
     point.

[1.6.37.]: How can the synchronous playback of moving pictures (AVI files) and the sound be performed?

The data can be created by either converting the sound AVI file to the
STR file with MovConv, or encoding the moving pictures and the sound
separately before mixing them with MovConv. The playback should be
performed by the sample attached to the library.

[1.6.38.]: It is said that the processing must be inserted between the _96_init() and the _96_remove() in order to use ISO-9660 file system when booting from a CD-ROM. How can the

screen be faded out by the interrupt while reading the data with the read() function?

There are 2 ways to read the file contents from the CD. One is the synchronous mode, and the open() or read() function is used. In this mode, the background operations are not permitted at all. (waits for the end of reading) Another is the asynchronous mode, and the background operations are available. However, the easy file-access with the open() or read() function cannot be used. Instead, asynchronous CD access functions such as the CdSearchFile() or CdRead() function are prepared.

Refer to the sample program (cdtuto\tuto2.c) attached to the library. In the asynchronous mode, only the CdInit() function is needed because ISO-9660 file system handling is abandoned, and any procedure such as the _96_init() function is not necessary.

[1.6.39.]: How can TIM files be read in from CD-ROM?

Refer to the following sample to read the data from CD-ROM.
See the manual or the sample attached to the library for details.

```
== main.c =============================================================

/*
 *
 */
#include <asm.h>
#include <kernel.h>
#include <sys/types.h>
#include <sys/file.h>
#include <libetc.h>
#include <libgte.h>
#include <libgpu.h>
#include <libcd.h>

u_short load_cd_t2v(char *name, int px, int py, long length);
int read_cd(char *name, u_long *buffer, int length);

main()
{
        static char     *files[] = {
                ""\\TAMA.BMP;1"",
                ""\\NAOYA2.BMP;1"",
                ""\\KISS.BMP;1"",
                ""\\MEG.BMP;1"",
                0,
        };

        int             i;
        int             px, py;

        ResetCallback();
        CdInit();
        CdSetDebug(0);
        PadInit(0);
        ResetGraph(0);
```

```c
            SetDispMask(1);

            while (PadRead(1) == 0) {
                    px = 64 + (rand() % (256-64));
                    py = 60 + (rand() % (240-60));
                    for (i = 0; files[i]; i++) {
                            DrawSync(0);
                            VSync(3);
                            load_cd_t2v(files[i], px, py, px*py*2);
    }
            }

            CdSync(0, 0);
            PadStop();
            StopCallback();
            exit();
}

u_short load_cd_t2v(char *name, int px, int py, long length)
{
            static u_long   *buf = (u_long *)0x801d0000;
            RECT rect;

            read_cd(name, buf, length);
            setRECT(&rect, 0, 0, 256, 240);
            ClearImage(&rect, 0, 0, 0);
            setRECT(&rect, (256-px)/2, (240-py)/2, px, py);
            LoadImage(&rect, buf);
            return(0);
}

int read_cd(char *name, u_long *buffer, int length)
{
            long    sector;
            CdlFILE fp;                 /* Information on file location and size
*/
            int     mode = CdlModeSpeed;
            int             i, cnt;

            if (CdSearchFile(&fp, name) == 0) {
                    exit();
            }
            sector = (fp.size+2048-1) / 2048;

            for (i = 0; i < 10; i++) {       /* Ten times retrying */
                    /* Setting of target position */
                    CdControl(CdlSetloc, (u_char *)&fp.pos, 0);
                    /* Start of reading */
                    CdRead(sector, buffer, mode);

                    while ((cnt = CdReadSync(1, 0)) > 0)
                            ;
                    if (cnt == 0) {
                            break;
                    }
            }
            return (0);
```

```
}

== makefile.mak =======================================================

all:
        ccpsx -g -Xo$80010000 main.c -omain.cpe,main.sym
        cpe2x main.cpe
        cp main.exe psx.exe

== test.cti ===========================================================

Disc CDROM CD.EMU

  CatalogNumber 0000000000000

  LeadIn MODE1

    Empty       1000
    PostGap     150

  EndTrack


  Track MODE1

     Pause      150

Volume          ISO9660         cd.ISO

  PrimaryVolume

    SystemIdentifier                ""PSX""
    VolumeIdentifier                ""PSXTEST""
    VolumeSetIdentifier             ""PSXTEST""
    PublisherIdentifier             ""SONY""
    DataPreparerIdentifier          ""SONY""
    ApplicationIdentifier           ""SONY""

    Lpath
    Mpath

    Hierarchy

        File            PSX.EXE
        Source          PSX.EXE
        EndFile

        File            TAMA.BMP
        Source          TAMA.BMP
        EndFile

        File            NAOYA2.BMP
        Source          NAOYA2.BMP
        EndFile

        File            KISS.BMP
        Source          KISS.BMP
        EndFile
```

```
        File            MEG.BMP
        Source          MEG.BMP
        EndFile

    EndHierarchy

  EndPrimaryVolume

EndVolume

    Empty        300
    PostGap      150

  EndTrack

  LeadOut MODE1

    Empty 500

  EndTrack


EndDisc
```

[1.7.]: Questions using the MOVCONV?

[1.7.1.]: When will MOVCONV allow for creation of str or bs data that is less than 5 sectors a frame?

MovConv 1.8e ignore limits you specify on the [MDEC Parameters] dialog box. This is a bug with MOVCONV 1.8e and will be improved upon in the next release.

[1.8.]: Questions about MDEC?

[1.8.1.]: Is information about the MDEC frame format being released to any tool development companies and licencees?

The information on the bs format (MDEC frame format) has not been released to any tool development companies and licensees.

[1.9.]: With MDEC, 4/8-bit compressed texture data cannot be decompressed.

Unfortunately, MDEC corresponds to 16/24-bit graphics data only now.

[1.10.]: When will a new version of the CD Generator software be released?

A version is being developed to support both Windows95 and Windows 3.1.
In this new version, any kind of SCSI host adapter that supports ASPI manager will be available.

The official date has not been specified, but is mentioned to be in the 2nd half of this year.

[1.11.]: How can one run the CDREC (part of CDGEN that does not work in WIN95) under WIN95?

But in another way, you can use the current version of CDREC.EXE on Windows95 by following steps.

1) Install EZ-SCSI Pro Ver3.5 or EZ-SCSI Ver4.0 into your PC.

2) Add the description about ASPI4DOS.SYS into CONFIG.SYS.

3) Execute an Adaptec's application of "SCSI-Switch" and disable
 protect mode driver and enable real mode driver(ASPI4DOS.SYS).

[1.12.]: Troubleshooting the CD Emulator

NOTE: There is no version of SNPATCHW.CPE or PATCHW.CPE that works with the emulator. Use SNPATCH.CPE that is attached to this message.

This is my troubleshooting guide to installing your emulator.
It is in psuedo-spaghetti code.

        InstallDevBoards(); // extern
        Remove your IRQ and DMA jumpers for now. Do not lose them.
        Check to make sure that the SCSI ID jumpers are set at the factory default of 7. Do
not   change it.

        // Pick a base address 10
        Select a base address with the base address jumper.
        Lovingly cram emulator board into an open slot in your PC.
        Firmly attach white-ish cable between dev boards and emulator.
        Attach emulator Hard drive to emulator board and turn it on.
        Edit autoexec.bat to contain CDBIOS /a###, where ### is the base address you
selected.
                Ex:
                        CDBIOS /a388
                will use base address 388 - which should match jumper on card.
        if (Computer does not boot || CDBIOS does not recognize emulator board) goto 10
and   select a new base address.

If (you try all base addresses and cannot get it to boot or find emu with CDBIOS) try
any/all of the following
{
      Press all the chips into their sockets harder.
      Be sure you are using CDBIOS version 1.02 or later.
      Maybe your computer is overheating. Check it out.
}
if (none of the above works) your emulator, dev boards, or computer may be
malfunctioning. Call BBS for help. Return(-1).

// communicate with the emulator drive through the PC bus
Set the SCSI ID on your harddrive to 3, 4, or 5. The manual says you can choose
others. Don't. Run CDDISK -n #, where # is the SCSI ID you have set on your harddrive
(above).
      Ex:
          CDDISK -n 4
      will activate and format the drive with SCSI ID 4 for use.

If (CDDISK does not run happily)
{
      Be sure your emulator hard drive is turned on and the cables are connected.
      If so, goto 10 and select a new base address.
      If that does not work, call BBS.
}

// install boot file
While in CDDISK (above section), install the boot file.
Use CDBOOT16.BIN or CDBOOT17.BIN (attached), DO NOT try any others yet. Try
them AFTER it works.

// build an emulation image
Detach BASIC.CTI and PSX.EXE from this memo on the BBS for use in the next
step.
Run BUILDCD -s#:1 BASIC.CTI, where # is the SCSI address of the emulator hard
drive   (1 is the first partition).
      Ex:
          BUILDCD -s4:1 BASIC.CTI
      will build an image to partition 1 of SCSI ID 4 hard drive.
Ignore warning 66 if it happens.
If an error occurs, be sure CDBIOS is installed and your emulator hard drive is turned
on   with cables attached.

// run the image - communicating through the white-ish cable to the DTL-H2000
boards 20   RESETPS 1
// version 1.03 or later, NOT 1.02. If you use 1.02, it may fail big time.
RUN SNPATCH
// NOT the version from the CD, NO version of SNPATCHW.CPE works with the
emulator
This will say "done" if it works, or "cannot connect to target" if it fails.
If it "cannot connect to to target" goto 20
If it says "cannot open source file" you need to fix your path to find the file.
RUN SELEMU
This will say "done" if it works, or "cannot connect to target" if it fails.
If it "cannot connect to to target" goto 20

If it says "cannot open source file" you need to fix your path to find the file.
RUN CDEXEC
This will say "done" if it works, or "cannot connect to target" if it fails.
If it "cannot connect to to target" goto 20
If it says "cannot open source file" you need to fix your path to find the file.

If your emulator still will not work:
0) reboot
1) try different emulator boards to see if your board is bad.
2) try different sets of dev boards to see if your dev boards are screwed up.
3) try different hard drives to see if your hard drive is bad
4) try different brand of computer to see if your computer is incompatible
5) leave a message in the CD newsgroup of the BBS detailing your traumatic
emulation   experience. Be specific.

BONUS STUFF:
Now you can advance to the following:

1) install interrupt jumper, make ammendment to CDBIOS line
2) install CDBOOT18.BIN
3) install CDBOOT2#.BIN, this will pave the way for step 4
4) install CDMON (a TSR that spews info back at you from the emulator).
5) install DMA jumper, make ammendment to CDBIOS line