



Creative Digital Research
Desktop CD-ROM Publishing Company

Publisher

Developer's Kit

Creative Digital Research

5300 Stevens Creek Blvd., Fourth Floor, San Jose, CA 95129

Tel: (408) 255-0999

Fax: (408) 255-1011

E-mail: info@HyCD.com

Web: www.HyCD.com

Copyright

© 1994-1997 Creative Digital Research

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owners.

This manual is published and printed by Creative Digital Research.

Any comments relating to the material contained herein may be submitted to:

Creative Digital Research
5300 Stevens Creek Blvd., Fourth Floor
San Jose, CA 95129

Trademarks

Sun Microsystems, SunPro and the Sun Logo are registered trademarks of Sun Microsystems, Inc.

XView is a registered trademark of Sun Microsystems, Inc.

Products bearing the SPARC trademark are based on an architecture developed by Sun Microsystems, Inc.

SPARCstation is a registered trademark of SPARC International.

UNIX is a registered trademark of UNIX System Laboratories, Inc.

X Window System is a registered trademark of the Massachusetts Institute of Technology.

Motif is a trademark of the Open Software Foundation.

Apple and Macintosh are registered trademarks and Power Macintosh is a trademark of Apple Computer, Inc.

All other products or services mentioned herein are identified by the trademarks, service marks, or product names as designated by the companies who market these products. Inquiries concerning such trademarks should be made directly to those companies. Creative Digital Research disclaims any responsibility for specifying which trademarks are owned by which companies or organizations.

HyCD Publisher Developer's Kit License Agreement

The Developer's Kit contains command line interfaces on how to communicate to Creative Digital Research's CD-ROM Mastering Engine. Developers can integrate the Engine into their existing programs in their work environment or into their product -- bringing advanced CD-ROM authoring capability directly to the developers. The Developer's Kit includes comprehensive documentation for you to integrate the HyCD Engine into your work environment. **YOU MAY NOT DISTRIBUTE OR INCLUDE THE HYCD ENGINE IN PRODUCTS OR APPLICATIONS THAT ARE FOR RESALE.** If you are interested in distributing the HyCD Engine integrated with your application you must enter into a Distribution License with CDR. Please call Creative Digital Research at **(408) 255-0999** for your copy of our Distribution License.

PLEASE READ THIS DOCUMENT CAREFULLY. THIS IS A LEGAL DOCUMENT BETWEEN YOU AND CREATIVE DIGITAL RESEARCH ("CDR"). BY PRESSING THE "I AGREE" ICON TO DOWNLOAD THIS DOCUMENT OR VIEWING THIS DOCUMENT (IN PAPER FORM OR ELECTRONICALLY) YOU ARE AGREEING TO BE BOUND BY THE TERMS AND CONDITIONS OF THIS AGREEMENT, INCLUDING THE SOFTWARE LICENSE AND DISCLAIMER OF SOFTWARE WARRANTY CONTAINED HEREIN. IF YOU ARE NOT WILLING TO BE BOUND BY THE TERMS OF THIS LICENSE AGREEMENT, YOU SHOULD NOT CLICK THE "I AGREE" ICON NOR READ FURTHER IN THIS DOCUMENTATION.

1. Definition

The Code is licensed (not sold) to you, and CDR owns all copyright (1995, CDR Software), trade secret, patent and other proprietary rights in the Code. CDR claims and reserves to itself all rights and benefits afforded under U.S. copyright law and all international copyright conventions in the Code, Maintenance Updates and Version Upgrades and any associated Documentation. The term "Code" includes only copies of the restricted use Developer's Kit interfaces and related documentation.

2. License

(a). Authorized Use. CDR grants you a nonexclusive license to use and integrate the Code into software products you have or are developing ("User Product"). This license is in addition to your current software license agreement with CDR. (b). Restrictions. You may not: (1) You may not distribute the Code to third parties as part of your end-user products; (2) modify or prepare derivative works of the Code; or (3) reverse engineer, decompile or disassemble the Code. (c) Your Obligations: You agree to keep confidential and to use your best efforts to prevent and protect the contents of the Code from unauthorized disclosure or use.

3. Disclaimer of Warranty

The Code is furnished "AS IS" and without warranty as to the performance or results you may obtain by using the Code. The entire risk as to the results and performance of the Code is assumed by you.

4. Limitation of Remedies

YOU UNDERSTAND AND AGREE AS FOLLOWS:

- A. THE DISCLAIMER OF WARRANTIES IN THIS AGREEMENT REPLACES ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. WE DISCLAIM AND EXCLUDE ALL OTHER WARRANTIES. IN NO EVENT WILL OUR LIABILITY OF ANY KIND INCLUDE ANY SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, EVEN IF WE HAVE KNOWLEDGE OF THE POTENTIAL LOSS OR DAMAGE.
- B. We will not be liable for any loss or damage caused by delay in furnishing the Code or any other performance under this Agreement.

5. Termination

This Agreement is effective until terminated. You may terminate it at any time by destroying the Code, including all computer programs and documentation, and erasing any copies residing on computer equipment. This Agreement also will terminate if you do not comply with any terms or conditions of this Agreement. Upon such termination you agree to destroy the Code and erase all copies residing on computer equipment.

6. Special Remedies

In recognition of the fact that the unauthorized disclosure, copying, or use of the Code or any materials containing Proprietary Information would cause irreparable harm and significant injury to CDR, which may be difficult to measure with certainty or to compensate through damages, Licensee therefore agrees that, in the event of such unauthorized disclosure, that CDR shall be entitled to obtain injunctive relief against such breach in any court of competent jurisdiction, without the necessity of posting bond, even if otherwise normally required. Such injunctive relief will in no way limit the aggrieved party's right to obtain other remedies available under applicable law.

7. General

Interpretation: The validity, construction and performance of this Agreement shall be governed by and interpreted in accordance with the laws of the State of California. In construing this Agreement, none of the parties hereto shall have any term or provision construed against such party solely by reason of such party having drafted the same.

Assignment: The rights and obligations under this Agreement shall not be assignable without the prior written consent of the non assigning party. Except that a successor in interest to either business is assignable.

Notices: Any notice or other communication required or permitted to be given hereunder shall be in writing and shall be deemed to be properly given when deposited in the United States mail, postage prepaid or when deposited with a public telegraph company for transmittal, charges prepaid. Notice to shall be addressed:

Either of the parties may change their address by notice given to the other of the parties hereto in the manner set forth above. Notices given as herein provided shall be considered to have been given ten (10) days after the mailing thereof.

Entire Agreement: This instrument contains all of the agreements, understandings, representations, conditions, warranties, and covenants made between the parties hereto. Unless set forth herein, neither party shall be liable for any representations made, and all modifications and amendments made hereto must be in writing.

Severability: In the event that any provision hereof is determined to be illegal or such determination shall not affect the validity or enforceability of the remaining provisions hereof, all of which shall remain in full force and effect.

Effect of Headings: The headings used in this Agreement are included for convenience only and are not to be used in construing or interpreting this Agreement.

Modification, Etc. of Agreement: No oral information by either of the parties hereto shall alter the meaning or interpretation of this Agreement. No amendment or change hereof or addition hereto shall be effective or binding on either of the parties hereto unless reduced to writing and executed by the parties or their respective duly authorized representatives.

Attorneys' Fees: In the event of any dispute with respect to this Agreement, the prevailing party shall be entitled to reasonable attorney's fees and other costs and expenses incurred in resolving such dispute.

Taxes: Neither party shall be liable for the other parties federal, state, municipal or other governmental taxes, duties, charges or costs, however designated, now or hereinafter imposed, including, but not limited to, any sales or use tax, privilege or excise tax.

Delays: Neither party shall be liable for delays in the performance of its obligations hereunder due to causes beyond its reasonable control, including but not limited to acts of God, strikes or inability to obtain labor or materials on time.

Waiver: Either party's failure to exercise any right under this Agreement shall not constitute a waiver of any other terms of conditions of this Agreement with respect to any other or subsequent breach, or a waiver by the parties of their right at any time thereafter to require exact and strict compliance with the terms of this Agreement. The rights or remedies set forth in this Agreement are in addition to any other rights or remedies which may be granted by law.

Governing Law: This Agreement shall be construed and enforced in accordance with the laws of the State of California. The parties agree to Venue in Santa Clara County, California.

TABLE OF CONTENT

Copyright	2
Trademarks	2
CDR Publisher Developer's Kit License Agreement	4
TABLE OF CONTENT	i
Introduction	1
HyCD Engine	3
CDR Publisher and HyCD Engine	3
What You See is What You Get	4
It Works!	4
Overview	5
Before You Begin	5
Terminology	6
Display Conventions	8
Using HyCD Engine	9
System Overview	11
Location	11
The Environment Variable	11
HyCD Engine Commands	12
Program Communication	13
Create An ISO Image From File List	16
cdrp1.exe	16
cdrp2.exe	16
Cancelling the process	17
In The Event of An Error	17
Copy An ISO Image	18
cdrcp.exe	18
Canceling the process	18
In The Event of An Error	19
Validate An ISO Image	20

Profile System Speed	21
cdrprf.exe.....	21
Cancelling the process.....	21
In The Event of An Error	22
Scan SCSI Devices	23
cdrscan.exe	23
In The Event of An Error	23
Scan Tracks from CD	24
cdrscan.exe	24
In The Event of An Error	24
Open Recorder Tray or Eject Caddy.....	25
cdrscan.exe	25
Obtain First Writable Address	26
cdrscan2.exe	26
In The Event of An Error	26
Working with the Kodak Disc Transporter.....	27
Multiple CDs -- Unattended Operation.....	29
Create An ISO Image From File List with Disc Transporter	31
Cancelling the process.....	31
In The Event of An Error	32
Copy An ISO Image with Disc Transporter	33
Cancelling the process.....	33
In The Event of An Error	34
Optimizing Data Transfer	35
Multi-Buffer I/O	37
Balancing Time and Resources	38
Fine Tuning	39
Bar Code Reading Support	41
Bar Code Reading Support.....	43

Reference.....	47
frmtopt.cdr	49
project.vol	55
cancel.cdr	56
stop.cdr	57
progress.cdr.....	58
cpopt.cdr	61
prfopt.cdr.....	67
scanopt.cdr	69
scan2opt.cdr	73
Index.....	75

Introduction

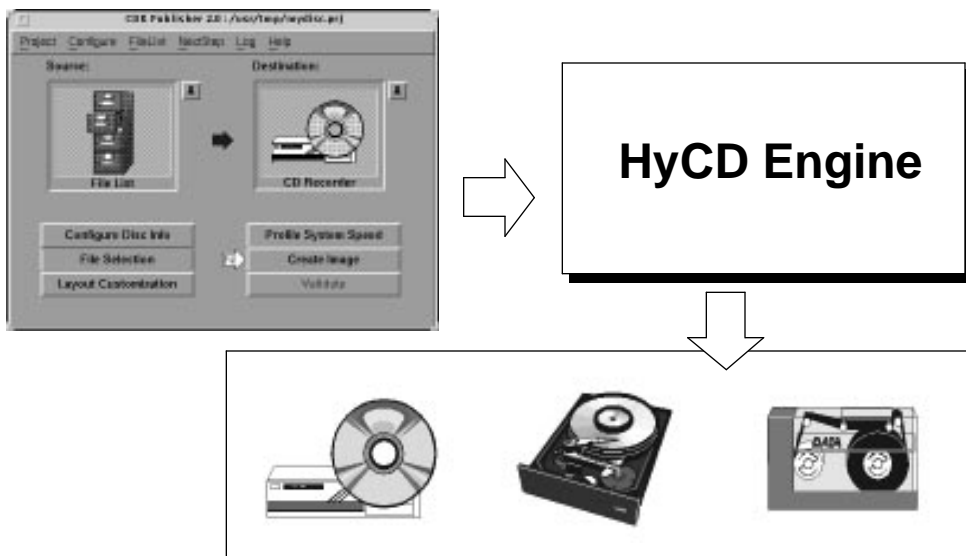
HyCD Engine

Creative Digital Research's HyCD Engine is a set of programs that enable the user to create and manipulate CD-ROMs. This set of programs can be easily adapted into an existing work environment or incorporated into existing products for creating Compact Discs.

For example: Backup large amounts of data directly to CD-ROM; automate internal and external distribution of corporate data and software; enable CD-ROM premastering capability in existing desktop publishing, imaging, database, multimedia authoring and presentation software.

HyCD Publisher and HyCD Engine

HyCD Publisher is a desktop CD-ROM premastering software system developed by Creative Digital Research. It consists of two parts: A Graphical User Interface, (GUI), and a set of programs that perform the CD-ROM premastering process. The set of programs that perform the premastering process is called the HyCD Engine.



What You See is What You Get

HyCD Publisher uses the same HyCD Engine that comes with this package. All technology and features that are supported by HyCD Publisher are available in the HyCD Engine.

It Works!

The best way to examine how the HyCD Engine works is by using HyCD Publisher. Based on the operation configured by the user, when the `Create Image` button is pressed, HyCD Publisher will invoke HyCD Engine with a set of parameters, and display the progress made by HyCD Engine.

At the completion of an operation, the Graphical User Interface (GUI) will display the final success or failure status window. Once the `OK` button is pressed, all communication files from the GUI to the HyCD Engine will be removed.

The user is encouraged to take a look at these communication files in order to understand how the HyCD Engine works.

Overview

This documentation is meant to be an easy guide to using the HyCD Engine. It assumes you already have HyCD Publisher and have experience using it. Please review the HyCD Publisher documentation if you have not yet done so.

A Reference Section has been provided at the end of this manual to provide you with all of the technical instructions you will require.

Before You Begin

You will need to take the following steps prior to using the HyCD Engine:

1. Determine your operating system requirements.
2. Familiarize yourself with the “Terminology” and “Display Conventions” Sections of this manual, as these are convenient references.
3. Install and configure your hardware.
4. Install and configure HyCD Publisher software.
5. Familiarize yourself with HyCD Publisher software.

If you encounter any difficulty in your set-up, consult your system administrator.

Terminology

ANSI	American National Standards Institute.
Device Driver	A computer program to be installed on your system in order for HyCD Publisher to communicate with the selected hardware devices.
High Sierra	The first proposed structure designed for CD-ROM, later amended to become ISO 9660.
ISO 9660	Industry standard file structure designed for the CD-ROM and accepted by the International Standards Organization (ISO).
Image File	A single file on your source hard drive which contains all the files selected for the image you are creating.
Image on-the-Fly	When a CD-ROM is created from a Source directly to a CD-ROM recorder.
License	Permission received from an application developer to use their application.
Mastering	Recording of encoded input data on a glass substrate.
Premastering	Preparing and encoding of input data to send to recording unit prior to mastering and replication.
RRIP	Rock Ridge Interchange Protocol (See Rock Ridge Extension).
Rock Ridge Extension	A File structure designed for users of POSIX and UNIX systems.
SCSI	Small Computer Systems Interface. Used as an interface to connect peripherals to computers.
Virtual Disk	An external hard drive to which you record data in the image of a CD-ROM. This can then be mounted back and viewed as though it were an actual CD-ROM to correct any mistakes or optimize the disc layout prior to actually recording to the CD-ROM.

Virtual Write

The Virtual Write process is used to send data to the recorder without physically writing to a recordable disc. This process is dependent on the CD-ROM recorder supporting such a feature.

Display Conventions

This manual uses standard display conventions.

bold text	Keystrokes that you should type.
<code>courier typeface</code>	System messages, programming fragments, commands, files and directories.
bold courier	Critical parts of program listings, keystrokes that you must type.
<code><...></code>	Item that must be filled in by you.
<code>%</code>	C shell prompt.

Note: Windows and buttons (keys) which appear on your screen will appear (in this manual) in helvetica typeface.

Using HyCD Engine

System Overview

HyCD Engine consists of several programs. The following describes them in detail.

Location

Since HyCD Engine is multi-platform, you must locate the version that is supported on your computer system. Use the following table to locate the programs:

Platform	Location
SUN SunOS	<top level>/bin/SUNOS
SUN Solaris 2.3	<top level>/bin/SOLARIS23
SUN Solaris 2.4	<top level>/bin/SOLARIS24
SUN Solaris 2.5	<top level>/bin/SOLARIS25
SGI IRIX 5.2, 5.3, 6.0, 6.2, 6.3	<top level>/bin/SGI
PC Windows	<top level>/bin/MSWIN
Windows NT	<top level>/bin/WINNT
Windows 95	<top level>/bin/WIN95

The Environment Variable

The programs in HyCD Engine use the environment variable `CDRENGHOME` to communicate with each other and with the Graphical User Interface. It is very important to configure the environment variable to the location where all of the HyCD Engine programs are located before using the HyCD Engine.

For example:

You may set the environment variable using C shell:

```
% setenv CDRENGHOME <path>
```

HyCD Engine Commands

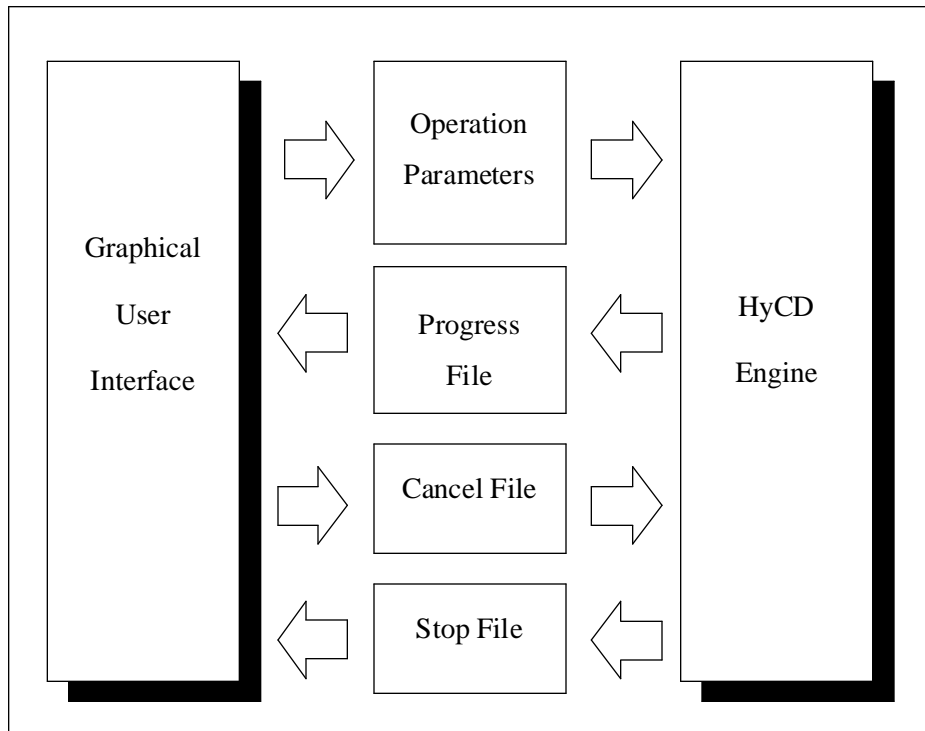
Program	Function
cdrp1.exe	Source = File List Destination = All Recursively traverse to subdirectories and gather all the file names to be incorporated into the CD-ROM image. A file containing the list of file names is generated.
cdrp2.exe	Source = File List Destination = All Using the file list generated by cdrp1.exe, create the CD-ROM image and write it out to the destination specified.
cdrcp.exe	Source = All but File List Destination = All Copy and translate from source devices to destination devices. When tape is used for source and/or destination, translation is done to read/write ANSI 8 K Byte per block format.
cdrprf.exe	Source = All but tape Read from the Source and profile the data accessing speed.
cdrlldr.exe	Source = All but tape Destination = CD-ROM Recorder Kodak Disc Transporter program that will invoke the transporter to move CD-ROMs.
cdrscan.exe	1. Scan SCSI devices 2. Scan tracks from CD 3. Eject CD Recorder caddy/tray
cdrscan2.exe	Obtain first writable address

Program Communication

Each program in the HyCD Engine communicates to the user with a set of communication files. These files serve as parameters for setting how the HyCD Engine is going to run. During program execution, HyCD Engine will use a set of communication files to communicate back to the user so that the user may display status.

During program execution, the communication files are written in the current working directory. If the current working directory is not writable, i.e. you are running the program from CD-ROM, you must ask the user to change to a writable directory before starting the programs. On the Windows 3.1 platform, the communication files will be written to the directory where the `temp` environment variable is configured.

The following illustrates the communication process between the GUI and the HyCD Engine in HyCD Publisher:



The following table lists the communication files required before executing the programs:

File Name	Purpose
frmtopt.cdr	Options file. It contains all the parameters for cdrp1.exe, cdrp2.exe.
cpopt.cdr	Options file. It contains all the parameters for cdrpc.exe.
cancel.cdr	Cancel File. Modify the content of the Cancel File to stop the various programs in progress. The content of this file must contain a zero (0) before executing any programs.
prfopt.cdr	Options file. It contains all the parameters for cdrprf.exe.

Additional parameter files related to each project:

File Name	Purpose
<project>.prj	Project File. This contains all the information about a specific project. It is generated by the GUI. This is an optional file.
<project>.vol	Volume Information File. It contains all the parameters required on the ISO 9660 Volume Table. This file is required for cdrp2.exe. Initially the content of this file may be empty. However, this file must exist.

<project>.sfl	<p>Project Source List.</p> <p>It contains the list of directories and files to be recorded into the ISO image.</p> <p>This file is required for cdrp1.exe.</p>
<project>.cfl	<p>Expanded Source List.</p> <p>It contains the list of expanded directories and files to be recorded into the ISO image.</p> <p>This file is generated by cdrp1.exe and is required by cdrp2.exe.</p>

During program execution, the programs update the following files to indicate their status:

File Name	Purpose
progress.cdr	<p>Progress file.</p> <p>Progress report of the current execution for cdrp1.exe, cdrp2.exe, cdrcp.exe and cdrprf.exe.</p>
stop.cdr	<p>Stop File.</p> <p>It contains the current execution status for cdrp1.exe, cdrp2.exe, cdrcp.exe and cdrprf.exe.</p>

Create An ISO Image From File List

The process of creating an ISO image from a file list requires two programs: `cdrp1.exe` and `cdrp2.exe`.

cdrp1.exe

`cdrp1.exe` will base on a small list of source files and directories and expand the entire source file list. Follow these steps to run `cdrp1.exe`:

1. Configure the parameter file `frmtopt.cdr`.
2. Configure the `cancel.cdr` file to contain a 0.
3. List the source files and directories to be expanded by `cdrp1.exe` in `<project>.sfl`.
4. Execute `cdrp1.exe`.

For UNIX Systems execute `cdrp1.exe` with the path name of the writable working directory:

% `cdrp1.exe`

For PC Systems execute `cdrp1.exe` by issuing the RUN command in the File Manager with the path name of the writable working directory as argument.

5. Wait for `cdrp1.exe` to stop by reading `stop.cdr` file. When 0 turns into 1 in the `stop.cdr` file, `cdrp1.exe` is completed.
6. The user may also read `progress.cdr` file to find out the on-going progress of `cdrp1.exe`.
7. After `cdrp1.exe` is completed, it will create the `<project>.cfl` file in the working directory.

cdrp2.exe

`cdrp2.exe` will base on the expanded list of source files and directories, create the ISO image, then output to the destination selected. Follow these steps to run `cdrp2.exe`:

1. Configure the parameter file `frmtopt.cdr`.
2. Configure the `cancel.cdr` file to contain a 0.

3. Execute `cdrp2.exe`.

For UNIX Systems execute `cdrp2.exe` with the path name of the writable working directory:

% **`cdrp2.exe`**

For PC Systems execute `cdrp2.exe` by issuing the RUN command in the File Manager with the path name of the writable working directory as argument.

4. Wait for `cdrp2.exe` to stop by reading `stop.cdr` file. When 1 turns into 2 in the `stop.cdr` file, `cdrp2.exe` is completed.
5. The user may also read `progress.cdr` file to find out the on-going progress of `cdrp2.exe`.

Canceling the process

The user may cancel `cdrp1.exe` or `cdrp2.exe` during execution. Follow these steps to cancel the process:

1. Modify the parameter file `cancel.cdr` by changing 0 to a 1.
2. Modify the parameter file `stop.cdr` by changing it to a 9.
3. Wait for `cdrp1.exe` or `cdrp2.exe` to stop by reading `stop.cdr` file. When it shows -2, in the `stop.cdr` file, the `cdrp1.exe` or `cdrp2.exe` program is stopped.

In The Event of An Error

When waiting for either program to stop by reading the `stop.cdr` file, in the event of an error, the `stop.cdr` file will contain a value of -1. The user can read the `<project>.log` file to identify the cause of the error.

Copy An ISO Image

Once an ISO image is created, the user can copy the image to any destination device.

cdrpc.exe

Follow these steps to run `cdrpc.exe`:

1. Configure the parameter file `cpopt.cdr`.
2. Configure the `cancel.cdr` file to contain a 0.
3. Execute `cdrpc.exe`.

For UNIX Systems execute `cdrp1.exe` with the path name of the writable working directory:

```
% cdrpc.exe
```

For PC Systems execute `cdrpc.exe` by issuing the RUN command in the File Manager with the path name of the writable working directory as argument.

4. Wait for `cdrpc.exe` to stop by reading `stop.cdr` file. When 0 turns into 2 in the `stop.cdr` file, `cdrpc.exe` is completed.
5. The user may also read `progress.cdr` file to find out the on-going progress of `cdrpc.exe`.

Canceling the process

The user may cancel `cdrpc.exe` during its execution. Follow these steps to cancel the process:

1. Modify the parameter file `cancel.cdr` by changing 0 to a 1.
2. Modify the parameter file `stop.cdr` by changing it to a 9.
3. Wait for `cdrpc.exe` to stop by reading `stop.cdr` file. When it shows -2, in the `stop.cdr` file, the `cdrp1.exe` or `cdrp2.exe` program is stopped.

In The Event of An Error

When waiting for the `cdrcp.exe` to stop by reading the `stop.cdr` file, in the event of an error, the `stop.cdr` file will contain a value of -1. The user can read the `<project>.log` file to identify the cause of the error.

Validate An ISO Image

The user can compare the source and destination image by issuing the validate option. Both option files, `frmtopt.cdr` and `cpopt.cdr`, for the commands `cdrp1.exe`, `cdrp2.exe` and `cdrcp.exe` contain a validation flag.

With the flag turned on, source will be compared with destination instead of being output to destination.

Please reference `frmtopt.cdr` field 28 and `cpopt.cdr` field 5 for details.

Profile System Speed

The `cdrprf.exe` program is used to test the accessing speed of the selected source device.

cdrprf.exe

Follow these steps to run `cdrprf.exe`:

1. Configure the parameter file `prfopt.cdr`.
2. Configure the `cancel.cdr` file to contain a 0.
3. Execute `cdrprf.exe`.

For UNIX Systems execute `cdrprf.exe` with the path name of the writable working directory:

```
% cdrprf.exe
```

For PC Systems execute `cdrprf.exe` by issuing the RUN command in the File Manager with the path name of the writable working directory as argument.

4. Wait for `cdrprf.exe` to stop by reading `stop.cdr` file. When 0 turns into 2 in the `stop.cdr` file, `cdrprf.exe` is completed.
5. After `cdrprf.exe` is completed, it will record the results in `progress.cdr` file in the working directory.
6. The user may read `progress.cdr` file to find out the on-going progress of `cdrprf.exe`.
7. Since the default of `cdrprf.exe` is to read the entire source, the user may cancel the process in the middle and read the cumulative results thus far.

Cancelling the process

The user may cancel `cdrprf.exe` during its execution. Follow these steps to cancel the process:

1. Modify the parameter file `cancel.cdr` by changing 0 to a 1.
2. Modify the parameter file `stop.cdr` by changing it to a 9.
3. Wait for `cdrprf.exe` to stop by reading `stop.cdr` file. When it shows -2, in the `stop.cdr` file, the `cdrprf.exe` program is stopped.

In The Event of An Error

When waiting for `cdrprf.exe` to stop by reading the `stop.cdr` file, in the event of an error, `stop.cdr` file will contain a value of -1. The user can read the `<project>.log` file to identify the cause of the error.

Scan SCSI Devices

The `cdrscan.exe` program is used to scan all SCSI devices.

cdrscan.exe

Follow these steps to run `cdrscan.exe`:

1. Configure the parameter file `scanopt.cdr`.
2. Configure the `cancel.cdr` file to contain a 0.
3. Execute `cdrscan.exe`.

For UNIX Systems execute `cdrscan.exe` with the path name of the writable working directory:

```
% cdrscan.exe
```

For PC Systems execute `cdrscan.exe` by issuing the RUN command in the File Manager with the path name of the writable working directory as argument.

4. Wait for `cdrscan.exe` to stop by reading `stop.cdr` file. When 0 turns into 2 in the `stop.cdr` file, `cdrscan.exe` is completed.
5. After `cdrscan.exe` is completed, it will output the results in the `scsidev.cdr` file in the working directory.

In The Event of An Error

When waiting for `cdrscan.exe` to stop by reading the `stop.cdr` file, in the event of an error, `stop.cdr` file will contain a value of -1. The user can read the `<project>.log` file to identify the cause of the error.

Scan Tracks from CD

The `cdrscan.exe` program is also used to scan tracks from CD.

cdrscan.exe

Follow these steps to run `cdrscan.exe`:

1. Configure the parameter file `scanopt.cdr`.
2. Configure the `cancel.cdr` file to contain a 0.
3. Execute `cdrscan.exe`.

For UNIX Systems execute `cdrscan.exe` with the path name of the writable working directory:

```
% cdrscan.exe
```

For PC Systems execute `cdrscan.exe` by issuing the `RUN` command in the `File Manager` with the path name of the writable working directory as argument.

4. Wait for `cdrscan.exe` to stop by reading `stop.cdr` file. When 0 turns into 2 in the `stop.cdr` file, `cdrscan.exe` is completed.
5. After `cdrscan.exe` is completed, it will output the results in `toc.cdr` file in the working directory.

In The Event of An Error

When waiting for `cdrscan.exe` to stop by reading the `stop.cdr` file, in the event of an error, `stop.cdr` file will contain a value of -1. The user can read the `<project>.log` file to identify the cause of the error.

Open Recorder Tray or Eject Caddy

The `cdrscan.exe` program is also used to open the recorder tray or eject a caddy.

cdrscan.exe

Follow these steps to run `cdrscan.exe`:

1. Configure the parameter file `scanopt.cdr`.
2. Configure the `cancel.cdr` file to contain a 0.
3. Execute `cdrscan.exe`.

For UNIX Systems execute `cdrscan.exe` with the path name of the writable working directory:

```
% cdrscan.exe
```

For PC Systems execute `cdrscan.exe` by issuing the `RUN` command in the `File Manager` with the path name of the writable working directory as argument.

4. Wait for `cdrscan.exe` to stop by reading `stop.cdr` file. When 0 turns into 2 in the `stop.cdr` file, `cdrscan.exe` is completed.

Obtain First Writable Address

The `cdrscan2.exe` program is used to obtain First Writable Address.

cdrscan2.exe

Follow these steps to run `cdrscan2.exe`:

1. Configure the parameter file `scan2opt.cdr`.
2. Configure the `cancel.cdr` file to contain a 0.
3. Execute `cdrscan2.exe`.

For UNIX Systems execute `cdrscan2.exe` with the path name of the writable working directory:

```
% cdrscan2.exe
```

For PC Systems execute `cdrscan2.exe` by issuing the `RUN` command in the `File Manager` with the path name of the writable working directory as argument.

4. Wait for `cdrscan2.exe` to stop by reading `stop.cdr` file. When 0 turns into 2 in the `stop.cdr` file, `cdrscan2.exe` is completed.
5. After `cdrscan2.exe` is completed, it will output the results in `fstwradd.cdr` file in the working directory.

In The Event of An Error

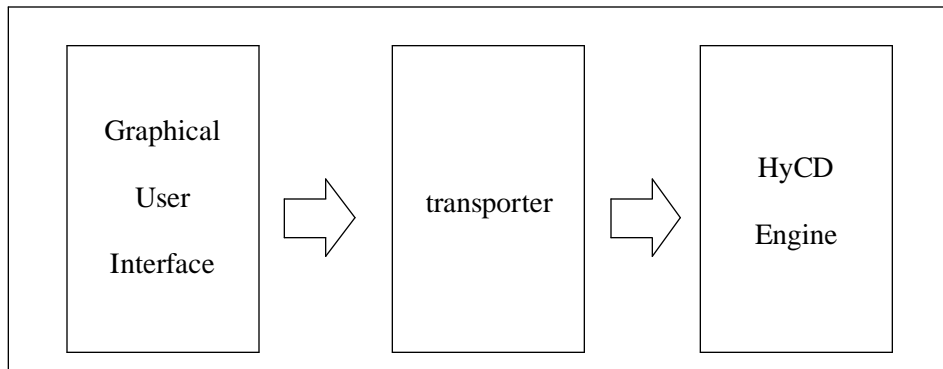
When waiting for `cdrscan2.exe` to stop by reading the `stop.cdr` file, in the event of an error, `stop.cdr` file will contain a value of -1. The user can read the `<project>.log` file to identify the cause of the error.

Working with the Kodak Disc Trans- porter

Multiple CDs -- Unattended Operation

Using the Kodak Disc Transporter which automatically inserts and removes writable CD media, the user is able to record up to 75 individually customized writable CDs, or multiple copies of the same disc, in an unattended, overnight operation.

CDR provides the user with an automatic loader program that communicates with the disc transporter to move writable discs. It operates between the GUI and the previously mentioned programs, `cdrp2.exe` and `cdrcp.exe`, to provide repetitive commands for changing writable CDs.



The command is:

```
cdrlldr.exe #_of_copies transporter_card_id transporter_scsi_id  
p2_or_cp working_dir_path non_stop_when_error
```

where:

#_of_copies:	number of copies for loader
transporter_card_id:	loader's card id
transporter_scsi_id:	loader's scsi id
p2_or_cp:	execute <code>cdrp2.exe</code> or <code>cdrcp.exe</code> (<code>cdrp</code> = 1; <code>cdrp2</code> = 2)
working_dir_path:	working directory (where communication files are)
non_stop_when_error:	when error occurs, if it's 1, eject bad CD into basket and continue with next CD; if it's 0, stop the whole process

For example,

```
% cdrlldr.exe 3 0 5 1 /tmp 1
```

The following section describes the process of creating CD-ROMs with the disc transporter.

Create An ISO Image From File List with the Kodak Disc Transporter

The process of creating an ISO image from a file list requires three programs: `cdrp1.exe`, `cdrlldr.exe` and `cdrp2.exe`. The process is very similar to the process described in “Creating An ISO Image From File List”. The only difference is that the `stop.cdr` file is being used for communicating each stage of the recording process.

The steps are as follows:

1. Run `cdrp1.exe` normally.
2. Run `cdrlldr.exe` with arguments to run `cdrp2.exe`.
 - 2.1. `cdrlldr.exe` reads a 1 in `stop.cdr` then starts the `cdrp2.exe` program.
 - 2.2. When `cdrp2.exe` stops, it writes a 2 in `stop.cdr` file.
 - 2.3. `cdrlldr.exe` checks the `stop.cdr` file and changes the content from 2 to 3.
 - 2.4. When GUI sees a 3 in `stop.cdr` file, it informs the user that a disc is done and resets it back to 1. In the mean time, `cdrlldr.exe` continues to run but in a waiting state. When `cdrlldr.exe` sees the change in the `stop.cdr` file, it continues.
 - 2.5. `cdrlldr.exe` repeats the process from 2.1 to 2.4 until it finishes all the required number of discs specified in the command parameter and outputs a 4 into `stop.cdr` file.
 - 2.6. GUI sees a 4 in the `stop.cdr` file and informs the user that all copies are done.

Cancelling the process

The user may cancel `cdrp1.exe` or `cdrp2.exe` during their execution. Follow these steps to cancel the process:

1. Modify the parameter file `cancel.cdr` by changing 0 to a 1.
2. Modify the parameter file `stop.cdr` by changing it to a 9.
3. Wait for `cdrp1.exe` or `cdrp2.exe` to stop by reading `stop.cdr` file. When it shows -2, in the `stop.cdr` file, `cdrp1.exe` or `cdrp2.exe` program is stopped.
4. `cdrlldr.exe` will change the -2 to -4 in the `stop.cdr` file.
5. GUI reports to user that the loader has stopped.

In The Event of An Error

When waiting for either program to stop by reading the `stop.cdr` file, in the event of an error, `stop.cdr` file will contain a value of -1.

The `cdrldr.exe` program will change it from -1 to -3.

The GUI will advise the user to read the `<project>.log` file to identify the cause of the error.

Copy An ISO Image with Disc Transporter

Copying an image using a transporter is also similar to the previous section. The following describes the steps.

1. Run `cdrlldr.exe` with arguments to run `cdrcp.exe`.
 - 1.1. `cdrlldr.exe` reads a 1 in `stop.cdr` then starts `cdrcp.exe` program.
 - 1.2. when `cdrcp.exe` stops, it writes a 2 in `stop.cdr` file.
 - 1.3. `cdrlldr.exe` checks the `stop.cdr` file and change the content from 2 to 3.
 - 1.4. When the GUI sees a 3 in `stop.cdr` file, it informs the user that a disc is done and resets it back to 1. In the mean time, `cdrlldr.exe` continues to run but in a waiting state. When `cdrlldr.exe` sees the change in the `stop.cdr` file, it continues.
 - 1.5. `cdrlldr.exe` repeats the process from 1.1 to 1.4 until it finishes all the required number of discs specified in the command parameter and outputs a 4 into the `stop.cdr` file.
 - 1.6. GUI sees a 4 in the `stop.cdr` file and informs the user that all copies are done.

Cancelling the process

The user may cancel `cdrep1.exe` or `cdrep2.exe` during execution. Follow these steps to cancel the process:

1. Modify the parameter file `cancel.cdr` by changing 0 to a 1.
2. Modify the parameter file `stop.cdr` by changing it to a 9.
3. Wait for `cdrcp.exe` to stop by reading `stop.cdr` file. When it shows -2, in the `stop.cdr` file, `cdrcp.exe` program is stopped.
4. `cdrlldr.exe` will change the -2 to -4 in the `stop.cdr` file.
5. GUI reports to user that the transporter has stopped.

In The Event of An Error

When waiting for either program to stop by reading the `stop.cdr` file, in the event of an error, `stop.cdr` file will contain a value of -1.

The `cdrldr.exe` program will change it from -1 to -3.

The GUI will advise the user to read the `<project>.log` file to identify the cause of the error.

Optimizing Data Transfer

Multi-Buffer I/O

Because the CD-ROM recorder requires a sustained data transfer rate for creating CD-ROMs, it is essential that the HyCD Engine can maintain the data gathering rate with the data consumption rate of the CD recorder. This is why we have developed a Multi-Buffer Input / Output System.

The HyCD Engine incorporates a multi-buffer I/O scheme for reading and writing all data. It has two parameters that the user can tune for optimal performance. The parameters are Buffer Number and Buffer Size. Buffer Number is the total number of individual buffers, and buffer size is the amount of data being held in each buffer.

Before we explain how Buffer Number and Buffer Size are being used, let us first understand the way the HyCD Engine works. There are two stages that the HyCD Engine goes through to output the data to the CD-ROM recorder:

Stage 1: Read the source data and fill the entire buffer that the CD-ROM recorder has before sending the write command to the recorder.

Send the write command to the recorder to begin the record process.

Stage 2: Perform repetitive read-data-from-source and write-data-to-recorder process.

In Stage 2, different computer platforms process data differently; some use non-block I/O, some use blocking I/O. Therefore, the user will get different performance on different computer platforms.

Buffer Number and Buffer Size are both used in Stage 1, whereas Buffer Size only is used in Stage 2.

Balancing Time and Resources

In general, the user must try to balance the Buffer Number and Buffer Size based on the following criteria:

1. Try to match the recorder memory size by multiplying the Buffer Number and the Buffer Size without running out of system's physical memory. If you must, you can use Virtual Memory (PC) or Swap Space (UNIX).

You must keep in mind that Virtual Memory or Swap Space is slower than Physical Memory.

2. Buffer Size must be a number that is supported by your recorder and your platform.

For example, the maximum buffer size for PC is 64K.

3. The time it takes to fill each buffer must be shorter than the time it takes for the recorder to write the data out to the CD.

For example, if your buffer size is too large, by the time the computer system fills it, your CD recorder will have already exhausted its internal buffer.

4. Different buffer sizes work well in different scenarios.

For example, if there are a few large size files in contiguous hard disk sectors, it will take a very short time to fill large buffers. On the other hand, if there are many small size files fragmented across different sectors, it will take a very long time to fill large buffers.

The image on-the-fly process requires reading many files while writing to the recorder, whereas the `cdrcp.exe` process requires reading one continuous image from the source device.

Fine Tuning

The user is encouraged to experiment with both parameters, Buffer Number and Buffer Size. These parameters are located at field number 12 and 13 in `frmtopt.cdr` file for the Image on-the-fly process or field number 6 and 7 in `cpopt.cdr` file for the `cdrcp.exe` process.

The following is a table describing the Buffer Numbers and the Buffer Sizes HyCD Engine currently uses:

	Windows		SUNOS		Solaris	
	Buffer Number	Buffer Size	Buffer Number	Buffer Size	Buffer Number	Buffer Size
Sony	200	16	200	16	200	16
Philips 2X, Kodak 2X	200	16	50	128	100	32
Yamaha 4X	200	16	50	128	100	32
Kodak 6X	100	32	50	128	100	32

Currently, we use the same numbers for `cdrcp2.exe` and `cdrcp.exe`. The user is encouraged to experiment with different numbers.

Bar Code Reading Support

Bar Code Reading Support

The Kodak PCD Writers support bar code reading capability. With the HyCD Engine, you can obtain bar code information from each CD-ROM before recording. The bar code information is appended to the file `barcode.cdr` under current working directory. To activate this feature, set the 15th field of `cpopt.cdr` to 1.

The format of `barcode.cdr` is as follows:

Bar_Code Result_of_write Result_of_validate

where

Bar_Code:

contains bar code if it reads successfully;

???-???-??? if it fails to read the bar code

Result_of_write:

1 = Success;

0 = Failure;

? = Process cancelled or operation not performed

Result_of_validate:

1 = Success; 0 = Failure;

? = Process cancelled or operation not performed

Example 1:

This example has 4 discs been written and validated successfully with the bar code read correctly and recorded in `barcode.cdr`.

```
5341-1642-2095 1 1
5341-1642-2096 1 1
5341-1642-2097 1 1
5341-1642-2098 1 1
```

Example 2:

This example fails to read bar code, but CD was written and validated successfully.

```
???-???-??? 1 1
```

Example 3:

This example fails to read bar code, CD was written successfully but validation fails.

```
???-???-??? 1 0
```

Example 4:

This example fails to read bar code, and CD writing failed and validation aborted.

```
????-????-???? 0 ?
```

Example 5:

This example shows CD with bar code 5341-1642-2092 writing succeeded, but validating failed.

```
5341-1642-2092 1 0
```

Example 6:

This example shows CD with bar code 5341-1642-2092 writing succeeded, but did not perform validation or validation process was cancelled in the middle.

```
5341-1642-2093 1 ?
```

Example 7:

This example shows CD with bar code 5341-1642-2092 writing failed, but did not perform validation or validation process was cancelled in the middle.

```
5341-1642-2094 0 ?
```


Reference

frmtopt.cdr

Line	Default	Comments	Options
1	5	CD-ROM Format Default to 1 on PC, 5 on UNIX systems.	1 = ISO 9660 2 = High Sierra 4 = RRIP 8 = HFS 32 = Joliet File System Possible Combinations: 5 = ISO 9660 + RRIP 9 = ISO + HFS 13 = ISO + HFS + RRIP 37 = ISO + RRIP + Joliet 41 = ISO + Joliet + HFS 45 = ISO + Joliet + RRIP + HFS
2	1	Symbolic Links	For UNIX: If source file is a symbolic link with absolute path, 0 = Keep It as Absolute 1 = Change It to Relative Path For PC: Default to 1
3	1	Directory Depth Control	0 = Limit Depth 8 1 = No limit
4	0	Customize If yes, cdrp1 will stop after composing the file list. If not, cdrp1 will invoke cdrp2 right after composing the file list.	0 = No 1 = Yes

5	1	Disc Optimization	1 = No Change 2 = Disc Size 3 = Dir Size
6	0	Reserved	
7	0	Reserved	
8	0	Reserved	
9	1	Reserved	
10	0	Reserved	
11	0	Reserved	
12	1	Buffer Number	Depends on the recorder and the OS. Please use default values generated from GUI as a reference.
13	8	Buffer Size	Depends on the recorder and the OS. Please use default values generated from GUI as a reference.
14	0	Enforce directory Name	0 = No 1 = Yes
15	0	Auto Convert Directory Name	0 = No 1 = Yes
16	0	Enforce File Name	0 = No 1 = Yes
17	0	Auto Convert File Name	0 = No 1 = Yes
18	1	Keep trailing semi-colon and version	0 = Exclude 1 = Keep

19	X:Y:Z// e.g., 0:2:3//	CD-ROM Recorder Device Name	X, Y, Z are numbers where X = Disc Size; 0 = 74 min. 1 = 63 min. Y = CD-ROM Recorder Type 0 = Sony 900E 1 = Philips, Kodak, Plasmon 2 = Kodak 600 3 = Yamaha 100/102 4 = Sony 920S/924S 5 = Ricoh/Plextor 6 = Teac 50s/55s 7 = Dynatek Crossfire/ Replicator 8 = Philips 2600 9 = Yamaha 400c 10 = Mitsumi CR-2600 11 = Ricoh 6200/Philips OmniWriter 12 = Yamaha 400t 13 = Philips 361 14 = Sony 926 Z = Record Speed 0 = 1X 1 = 2X 2 = 4X 3 = 6X Destination Is Not A CD-ROM Recorder: Unix = // PC = \\ Mac = ::
----	-----------------------------	--------------------------------	---

20	//	Boot File Name	No Boot File: Unix = // PC = \\ Mac = ::
21	/tmp/wk/ test.vol//	Volume File Name	
22	//	Conversion File Name	No Conversion File: Unix = // PC = \\ Mac = ::
23	/tmp/wk//	Working Directory	
24	/tmp/wk/ test.cfl//	File List File Name	
25	/tmp/wk/ test.log//	Log File Name	
26	/tmp/wk/ test.sfl//	Source List File Name	
27	/dev/rsd2c// or X:Y//	Destination File/Device Name. X = Controller ID. Y = SCSI ID.	For SUN Systems: Virtual Disk = Device Name Tape = X:Y CD-R = X:Y Image File = File Name For PC Systems: Virtual Disk = X:Y Tape = X:Y CD-R = X:Y Image File = File Name For SGI systems: All devices = X:Y Image File = File Name

28	R//	Device Type	R = Other Devices/File A = ANSI Tape Drive D = Dynatek Crossfire When destination is image file or virtual disk, this field should appear like this : R:X:Y:Z// X = Controller ID. Y = SCSI ID. Z = CDROM Recorder Type (Reference line 19)
29	0	Validation Mode	0 = Do Not Validate. 1 = Validate Against Destination. Do Not Perform Write.
30	0	Virtual Disk Label	0 = Overwrite 1 = Preserve
31	0	Disc Image Block 0-15	0 = Clear 1 = Preserve
32	0	Virtual Write	0 = No 1 = Yes
33	0	Session Control	0 = Single Session 1 = Interim Session 2 = Final Session
34	0	Include . Files	For UNIX Only 0 = No 1 = Yes
35	0	Include Hidden Files	For PC Only 0 = No 1 = Yes
36	0	Use AUFS Filter	For PC, set to 0 = No For Unix, set to 1 = Yes

37	0	Use PCFS Filter	For PC, set to 1 = Yes For Unix, set to 0 = No
38	1	ISO Level	Level 1 = 1 Level 2 = 2
39	0	Virtual Disk Type	For PC = 0 For Mac = 1
40	2	Macintosh Allocation Block Size	Valid Sizes are: 2K, 4K, ..., 20K
41	0	Create PC boot disc	1 = True 0 = False
42	/home/ cd_i_app//	CD-I application path	For Video CD on SGI platform only
43	my volume//	Macintosh Volume Name	
44	0	Reserved	
45	0	Place conversion file (.cnv) on CD-ROM or not	1 = place on CD-ROM; 0 = do not place on CD-ROM
46	0	First Writable Address for Multi-session feature	can be obtained from cdrscan2.exe program
47	0	Reserved	
48	0	Reserved	
49	0	Writing Method	0 = Track At Once 1 = Disc At Once 2 = Fixed Packet Writing 3 = Variable Packet Writing

Note: For each field that is not a pure numeric field, the field name must be ended with two separators; i.e., 2 slashes (//) for Unix platform, 2 back slashes (\\) for PC platform or 2 colons (::) for Macintosh platform.

project.vol

Line	Field Name	Default
1	System:	HyCD Publisher
2	Volume:	Volume_1
3	VolumeSet:	1_of_1
4	Publisher:	Creative_Digital_Research
5	Copyright:	copywrit.doc
6	Abstract:	abstract.doc
7	Bibliograph:	bib.doc
8	Creation:	i.e. 7/1/1995 11:34:54.99 -32
9	Expiration:	i.e. 7/1/1995 11:34:54.99 -32
10	Modification:	i.e. 7/1/1995 11:34:54.99 -32
11	Effective:	i.e. 7/1/1995 11:34:54.99 -32
12	VolumeSeq:	1
13	VolumeSetSize:	1

cancel.cdr

Line	Default	Comments	Options
1	0	Current State	0 = Initial State 1 = Cancel Operation by user.

stop.cdr

Without Kodak Disc Transporter:

Line	Default	Comments	Options
1	0	Current State	0 = Initial State 1 = cdrp1.exe is done. 2 = cdrp2.exe or cdrpc.exe or cdrprf.exe or cdrscan.exe or cdrscan2.exe is done. -1 = Error in cdrp1.exe or cdrp2.exe or cdrpc.exe or cdrprf.exe or cdrscan.exe or cdrscan2.exe. -2 = Confirm Cancellation.

With Kodak Disc Transporter

Line	Default	Comments	Options
1	0	Current State	0 = Initial State For CDRP1: 0 = continue; 1 = done; -1 = error; -2 cancelled. For CDRP2: 1 = continue; 3 = done; 4 = transporter_done; -3 = error; -4 = cancelled. For CDRCP: 1 = continue; 3 = done; 4 = transporter_done; -3 = error; -4 = cancelled. For CDRPRF: 0= continue; 2= done; -1= error; -2 = cancelled.

progress.cdr

`progress.cdr` is used for showing the progress of different programs. The following describes its interface with each components of the CDR Engine:

During `cdrp1.exe`:

Line	Default	Comments
1	100	Mode: Scanning Source
2		Number of Files Scanned
3		File Name Currently Scanned
4	1	End of Progress File Mark

During `cdrp2.exe`: Composing Image

Line	Default	Comments
1	200	Mode: Composing Image
2		Percent of Completion
3	1	End of Progress File Mark

During `cdrp2.exe`: Writing Image

Line	Default	Comments
1	1	Mode: Writing Image
2		Percent of Completion
3	00:00:00	Starting Time
4	00:00:00	Elapsed Time
5	00:00:00	Estimated Completion Time
6	1	End of Progress File Mark

During `cdrcp.exe`: If Image is a CD-ROM Image Recognized by the Engine

Line	Default	Comments
1	1	Interpretable Mode
2		Percent of Completion
3	00:00:00	Starting Time
4	00:00:00	Elapsed Time
5	00:00:00	Estimated Completion Time
6	1	End of Progress File Mark

During `cdrcp.exe`: If Image is a Not CD-ROM Image Recognized by the Engine

Line	Default	Comments
1	-1	Non-interpretable Mode
2		Number of KByte Read So Far
3	00:00:00	Starting Time
4	00:00:00	Elapsed Time
5		Average Transfer Rate (KByte/Sec.)
6	1	End of Progress File Mark

During `cdrprf.exe`:

Line	Default	Comments
1	-1	Reserved
2		Number of KByte Read So Far
3	00:00:00	Starting Time
4	00:00:00	Elapsed Time
5		Average Data Transfer Rate (KByte/Sec.)
6		Minimum Data Transfer Rate (KByte/Sec.)
7	1	End of Progress File Mark

During `cdrcp.exe` and `cdrp2.exe`: When recorder is closing session in a real write

Line	Default	Comments
1	2	Close session
2	0	Percentage of Completion
3	0	First Session: 0 = No, this is not the first 1 = Yes, this is the first
4	1	End of Progress File Mark

cpopt.cdr

Line	Default	Comments	Options
1	/tmp//	Working Directory Path	
2	/dev/rsd2c// or X:Y// or X:Y:1//	Source File/Device Name. X = Controller ID. Y = SCSI ID.	For SUN Systems: Virtual Disk = Device Name Tape = X:Y CD-ROM = X:Y:1 Image File = File Name For PC Systems: CD-ROM = X:Y:1 All other devices = X:Y Image File = File Name For SGI systems: CD-ROM = X:Y:1 All other devices = X:Y Image File = File Name
3	/dev/rsd2c// or X:Y//	Destination File/Device Name. X = Controller ID. Y = SCSI ID.	For SUN Systems: Virtual Disk = Device Name Tape = X:Y CD-R = X:Y Image File = File Name For PC System: All devices = X:Y Image File = File Name For SGI systems: All devices = X:Y Image File = File Name
4	R:R//	Device Type	R = Other Devices/File A = ANSI Tape Drive

5	0	Validate	0 = run CDRCP 1 = validate destination with source if destination is not CD Recorder; automatic validation after create if destination is CD Recorder
6	1	Buffer Number	
7	8	Buffer Size	

8	<p>X:Y:Z// e.g. 0:2:3//</p>	<p>CD-ROM Recorder Device Name</p>	<p>X, Y, Z are numbers where X = Disc Size; 0 = 74 min. 1 = 63 min. Y = CD-ROM Recorder Type 0 = Sony 900E 1 = Philips, Kodak, Plasmon 2 = Kodak 600 3 = Yamaha 100/102 4 = Sony 920S/924S 5 = Ricoh/Plextor 6 = Teac 50s/55s 7 = Dynatek, Crossfire/ Replicator 8 = Philips 2600 9 = Yamaha 400c 10 = Mitsumi CR-2600 11 = Ricoh 6200/Philips OmniWriter 12 = Yamaha 400t 13 = Philips 3610 14 = Sony 926 Z = Record Speed 0 = 1X 1 = 2X 2 = 4X 3 = 6X Destination Is Not A CD-ROM Recorder: Unix = // PC = \\ Mac = ::</p>
---	-------------------------------------	--	--

9	/tmp/t1.log//	log file name	
10	0	Preserve Virtual Disk Label Information	0 = Overwrite 1 = Preserve
11	0	Disc Image Block 0-15	0 = Clear 1 = Preserve
12	0	Virtual Write when Destination is CD-R device	0 = No 1 = Yes
13	0	Session Control	0 = Single Session 1 = Interim Session 2 = Final Session
14	0	Virtual Disk Type	0 = for PC 1 = For Mac
15	0	Bar Code Reading Support	0 = No Bar Code Reading Support 1 = Has Bar Code Reading Support
16	/cdr/mickey.tcd//	Track Selection and Attribute File	*

Note: for each field that is not a pure numeric field, the field name must be ended with 2 separators; i.e., 2 slashes (//) for Unix platform, 2 back slashes (\\) for PC platform or 2 colons (::) for Macintosh platform.

* For field # 16, there are 3 cases:

1. Source = CD, Destination = CD or Track File; then this field contains the name of the file which contains the track information of the source CD that is to be copied directly to CD or to track files residing on the hard disk drive. For example,

16	/cdr/mickey.tcd//
----	-------------------

and the content of “/cdr/mickey.tcd” is as follows:

Field #	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Track 5	//	1	5	1	0	0	2	0	66705	3848	2352	0	0	0
Track 9	//	1	9	1	0	0	2	0	109205	8423	2352	0	0	0

where fields information is as follows:

Field #	Meaning
Field # 1	Reserved (always 2 separators; that is, 2 slashes (//) for Unix platform, 2 back slashes (\\) for PC platform or 2 colons (::) for Macintosh platform.)
Field # 2	Session number
Field # 3	Track number
Field # 4	Audio track or data track? 1=audio track; 0=data track
Field # 5	Mode: mode 1 or mode 2 of data track
Field # 6	Form: form 1 or form 2 of mode 2 of data track
Field # 7	Channel: 2 channels or 4 channels of audio track
Field # 8	Pre-emphasis: 1=pre-emphasis; 0=no pre-emphasis
Field # 9	Starting block number of this track
Field # 10	Total number of blocks of this track
Field # 11	Bytes per block
Field # 12	Session TOC type: 0=audio; 1=others
Field # 13	Session closed? 1=closed; 0=not closed
Field # 14	Disc finalized? 1=finalized; 0=not finalized

- Source = Track File, Destination = CD; then this field contains the name of the file which contains the track information that is to be copied to CD. For example,

16	/my_cd.tfl//
----	--------------

and the content of “/my_cd.tfl” is as follows:

Field #	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	/t5.cdr//	0	0	1	0	0	2	0	0	3848	2352	0	0	0
	/t9.cdr//	0	0	1	0	0	2	0	0	8423	2352	0	1	0

where fields information is as follows:

Field #	Meaning
Field # 1	Data/Audio Track File Name (full path ending with 2 separators; that is, 2 slashes (//) for Unix platform, 2 back slashes (\\) for PC platform or 2 colons (::) for Macintosh platform.)
Field # 2	Reserved (always 0)
Field # 3	Reserved (always 0)
Field # 4	Audio track or data track? 1=audio track; 0=data track
Field # 5	Mode: mode 1 or mode 2 of data track
Field # 6	Form: form 1 or form 2 of mode 2 of data track
Field # 7	Channel: 2 channels or 4 channels of audio track
Field # 8	Pre-emphasis: 1=pre-emphasis; 0=no pre-emphasis
Field # 9	Reserved (always 0)
Field # 10	Total number of blocks of this track
Field # 11	Bytes per block
Field # 12	Session TOC type: 0=audio; 1=others
Field # 13	Session closed? 1=closed; 0=not closed
Field # 14	Disc finalized? 1=finalized; 0=not finalized

- For combinations of other Sources and Destinations, this field is Reserved: always 2 separators; that is, 2 slashes (//) for Unix platform, 2 back slashes (\\) for PC platform or 2 colons (::) for Macintosh platform.

prfopt.cdr

Line	Default	Comments	Options
1	0	Source	0 = File List 1 = Virtual Disk 3 = CD-ROM Reader 4 = Image File
2	/dev/rsd2c or X:Y	Destination File/Device Name. X = Controller ID. Y = SCSI ID.	For SUN Systems: Virtual Disk = Device Name Tape = X:Y CD-R = X:Y Image File = File Name For PC System: Virtual Disk = X:Y Tape = X:Y CD-R = X:Y Image File = File Name For SGI systems: All devices = X:Y Image File = File Name
3		Working Directory Name: Includes Full Path Name	
4		Log File Name: Includes Full Path Name	

5	1000	Maximum Profiling Limit (in MB)	Profile will display result once the limit is reached, or when it reaches end of disc or when user stops it.
6	5	Minimum File Size (in KB)	<p>A file must be larger than the minimum File Size in order to be calculated into the average access speed.</p> <p>This will eliminate the inaccurate access speed calculation where the user may have many small files with an average speed of 1 sec. to read each file.</p>

scanopt.cdr

Line	Default	Comments	Options
1	/tmp/wk//	Working Directory	
2	// or X:Y:Z//	To scan SCSI devices or to scan tracks of CD. X = Controller ID. Y = SCSI ID. Z = CD reader type.	If //, then scan SCSI devices; if X:Y:Z//, then scan tracks of CD. For CD reader type: 1 = Regular reader 2 = Philips 2000 and below & Kodak 225 recorder 3 = Kodak 600 recorder 4 = Yamaha 100/102 5 = Sony 920S/924S 6 = Ricoh/Plextor 7 = Teac 50s/55s 8 = Dynatek, Crossfire/ Replicator 9 = Philips 2600 10 = Yamaha 400c 11 = Mitsumi CR-2600 12 = Ricoh 6200/Philips OmniWriter 13 = Yamaha 400t 14 = Philips 3610 15 = Sony 926 500 = Philips PCA80SC 501 = Plextor PX-4XCE/PX- 8XCS/PX-12XSx/PX-20TSx

3	/tmp/test.log//	Log File Name: Includes Full Path Name	
4	0	Scan CD or Eject tray/ caddy	Scan CD if it's 0; Eject tray/caddy if it's 1

Example of scanopt.cdr for “scanning SCSI devices”:

Line	content
1	/shome/user/testing//
2	//
3	/shome/user/testing/test.log//
4	0

Result of “scanning SCSI devices”, output to a file scsidev.cdr:, for example:

Card id	SCSI id	Device Type	Vendir id	Product id	Firmware
0	0	0x00	FUJITSU	M2694ES-512	811F
0	5	0x04	YAMAHA	CDR102	0.90
0	6	0x05	TOSHIBA	XM-4101TASUNSLCD	3424

- OR -

Card id	SCSI id	Device Type	Vendir id	Product id	Firmware
0	1	0x00	SEAGATE	ST51080N	0943
0	2	0x00	MICROP	21112-15MQ1001901	HQ30
0	3	0x00	QUANTUM	MAVERICK 540S	0905
0	4	0x04	KODAK	KHSW/OB-01	0342
0	5	0x08	KODAK	DISC TRANSPORTER	1.17

where “Device Type” is as follows:

Device Type	Device
0x00	Disk Device
0x01	Tape Device
0x02	Printer
0x03	Processor
0x04	WORM
0x05	CD-ROM Device
0x06	Scanner Device
0x07	Optical Memory Device
0x08	Medium Changer Device
0x09	Communications Device
others	Unknown Device

Example of scanopt.cdr for “scanning tracks from CD”:

Line	content
1	/shome/user/testing//
2	0:5:4// *
3	/shome/user/testing/test.log//
4	0

* Use Yamaha recorder as reader to scan tracks of CD where Yamaha recorder is at SCSI card 0, SCSI id 5.

Result of “scanning tracks from CD”, output to a file toc.cdr, for example:

Field #	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Track 1	//	1	1	1	0	0	2	0	0	23092	2352	0	0	0
Track 2	//	1	2	1	0	0	2	0	23092	14678	2352	0	0	0
Track 3	//	1	3	1	0	0	2	0	37770	20973	2352	0	0	0

Track 4	//	1	4	1	0	0	2	0	58743	19941	2352	0	0	0
Track 5	//	1	5	1	0	0	2	0	78684	18378	2352	0	0	0
Track 6	//	1	6	1	0	0	2	0	97062	12493	2352	0	0	0
Track 7	//	1	7	1	0	0	2	0	109555	16693	2352	0	0	0
Track 8	//	1	8	1	0	0	2	0	126248	12558	2352	0	0	0
Track 9	//	1	9	1	0	0	2	0	138806	16978	2352	0	1	1

where fields information is as follows:

Field #	Meaning
Field # 1	Reserved (always //) *
Field # 2	Session number
Field # 3	Track number
Field # 4	Audio track or data track? 1=audio track; 0=data track
Field # 5	Mode: mode 1 or mode 2 of data track
Field # 6	Form: form 1 or form 2 of mode 2 of data track
Field # 7	Channel: 2 channels or 4 channels of audio track
Field # 8	Pre-emphasis: 1=pre-emphasis; 0=no pre-emphasis
Field # 9	Starting block number of this track
Field # 10	Total number of blocks of this track
Field # 11	Bytes per block
Field # 12	Session TOC type: 0=audio; 1=others
Field # 13	Session closed? 1=closed; 0=not closed
Field # 14	Disc finalized? 1=finalized; 0=not finalized

- * For each field that is not a pure numeric field, the field name must be ended with two separators; i.e., 2 slashes (//) for Unix platform, 2 back slashes (\\) for PC platform or 2 colons (::) for Macintosh platform.

Example of scanopt.cdr for “ejecting tray/caddy”:

Line	content
1	/shome/user/testing//
2	0:5:4//
3	/shome/user/testing/test.log//
4	1

scan2opt.cdr

Line	Default	Comments	Options
1	/tmp/wk//	Working Directory	
2	X:Y:Z//	X = Controller ID. Y = SCSI ID. Z = CD recorder type.	For CD recorder type: 1 = Philips 2000 and below & Kodak 225 2 = Kodak 600 3 = Yamaha 100/102 4 = Sony 920S/924S 5 = Ricoh 1420C 6 = Teac 50s/55s 7 = Dynaktek, Crossfire/ Replicator 8 = Philips 2600 9 = Yamaha 400c 10 = Mitsumi CR-2600 11 = Ricoh 6200/Philips OmniWriter 12 = Yamaha 400t 13 = Philips 3610 14 = Sony 926
3	/tmp/test.log//	Log File Name: Includes Full Path Name	

Example of scan2opt.cdr for “obtaining First Writable Address”:

Line	content
1	/shome/user/testing//
2	0:5:3//
3	/shome/user/testing/test.log//

Result of “obtaining First Writable Address”, output to a file fstwradd.cdr, for example:

Line	content (First Writable Address)
1	32256

fmtree.cdr

Line	Default	Comments	Options
1	/tmp/wk//	Working Directory	
2	X:Y:Z//	To scan or erase CD X = Controller ID Y = SCSI ID Z = Recorder Type	For CD recorder Type: 11 = Ricoh 6200/Philips OmniWriter 13 = Philips 3610
3	/tmp/t.log	Log File Name: Include Full Path Name	
4	1	Scan disc only or scan and erase	0 = scan disc only 1 = scan disc and erase
5	0	Format Method	0 = Complete Format 1 = Quick Format

Example of fmtree.cdr for “Scan and Erase CD completely”:

Line	Content
1	/shome/user/testing//
2	0:5:11//
3	/shome/user/testing/t.log
4	1
5	0

Result of “Scan disc only”, output to a file rewrite.cdr, for example:

Line	Content
1	1

“1” means disc is Rewriteable, “0” means disc is not Rewriteable

Index

B

Buffer Number 37
Buffer Size 37

C

CDR Publisher 3
CDRENGHOME 11
communication files 14
Conventions 8

D

Device Driver 6

E

Engine Commands 12
environment variable 11

H

High Sierra 6
HyCD Engine 3

I

Image File 6
Image on-the-Fly 6
ISO 9660 6

L

License 6

M

Mastering 6
Multi-Buffer I/O 37, 43

P

parameter files 14
POSIX 6
Premastering 6

R

Rock Ridge Extension 6
RRIP 6

S

SCSI 6
Swap Space 38
System Speed 21

T

Terminology 6

V

Validate 20
Virtual Disk 6
Virtual Memory 38

