

To Everything There Is a Session

A Time to Listen, a Time to Read Multi-session CDs

Dianne Dietrich, Cornell University Library

Alex Nelson, NIST

[Disclaimer](#)

[1. Introduction](#)

[1a. Scope](#)

[2. Background](#)

[3. Generating a multi-session CD on the command line](#)

[4. Behavior patterns in disk imaging outputs of multi-session discs](#)

[5. Accessing content from images of the full disc](#)

[5a. Obtaining the disc offset\(s\) from the physical media](#)

[5b. Obtaining the disc offset\(s\) from imaging metadata](#)

[5c. Obtaining offset values from full User Data disk images](#)

[5d. Conversion to User Data from RAW using bchunk](#)

[5e. Extracting files from images of the full disc](#)

[6. Accessing content from single-session images](#)

[7. Accessing content from single-track images](#)

[7a. Offsets from single-track images](#)

[7b. Conversion to User Data from RAW](#)

[7c. Extracting files from single-track images](#)

[8. Extracting files from physical media](#)

[9. Non ISO 9660 filesystem data](#)

[Conclusion](#)

[Acknowledgements](#)

[Appendix A: Imaging pathways](#)

[Guymager](#)

[dd](#)

[ddrescue](#)

[cdrdao](#)

[Appendix B: Multiple Data Tracks](#)

[References](#)

[Endnotes](#)

Disclaimer

The views and opinions expressed in this article are those of the authors and do not necessarily reflect the official policy or position of any agency of the U.S. Government or Cornell University. Any mention of a vendor or product is not an endorsement or recommendation. Logos and trademarks are copyright their respective owners.

1. Introduction

Many digital preservation organizations have developed workflows to handle optical media that consist of single-session, single-track data CDs, especially at scale (Rothrock et al, 2021). Workflows for single-session CDs consisting of one data track followed by one or more audio tracks also exist, as demonstrated by Geoffrey Brown's work handling mixed-mode [1] CD-ROMs from the Voyager Collection (Brown, 2012).

CDs with multiple sessions can pose significant challenges to workflows and can lead to data errantly not being acquired or reviewed if the user is not aware, especially since guidance written for single-session media may fail in interesting and subtle ways when applied to multi-session discs. The goal of this paper is to provide practitioners a technical framework by which they can develop appropriate analysis and preservation workflows for these types of media. This work improves acquisition workflows, understanding of the varied storage models in optical media schemes at layers before file systems, and, in the case of software installation media, software supply chain review. Reflecting on our own experience with CD-ROM corpora, we saw an unfulfilled need for guidance on preservation techniques for multi-session CDs, since a digital preservation practitioner might encounter this as personal backups on CD-Rs and CD-RWs.

1a. Scope

Within this document, the discussion around optical media is limited to the Compact Disc digital form factor and approximate quantity of storage capacity, that is, under 800 megabytes. While "Optical media" as a general term encompasses other media classes, such as DVDs and the analog LaserDisc format, the technical details of arranging storage reviewed in this article on those media differ from Compact Discs, and are left out of scope.

2. Background

CD standards are detailed in technical standards known as The Rainbow Books, which started in 1980 with the Red Book, the standard for encoding audio on a CD [2]. In 1983, the Yellow Book established the CD-ROM standard [3] for the storage of data. Related, ISO 9660 defines the file system standard for a CD-ROM. The Orange Book (1990) [4] defined the CD-R standard, which included writing multiple sessions of data to a CD; related information can also be found in ECMA-394 (2010) [5]. ISO 13490 (1995) and ECMA-168 (1994) defined how operating systems read CD-Rs with multiple sessions [6]. In 1995, the Blue Book [7] provided the technical specifications of a multi-session disc with both audio and data components, to address the limitations of single-session, mixed-mode CD-ROMs. A Blue Book CD contains exactly two sessions: the first containing audio tracks, and the second containing one data track.

Disc Level									
Session Level	Session 1				Session 2				...
Track Level	Lead in (with TOC)	Track 1	...	Lead out	Lead in (with TOC)	Track 1	...	Lead out	...

Figure 1. Structure of a general multi-session CD. Here TOC refers to Table of Contents. (Adapted from Wikipedia [8]). Individual tracks may be audio or data. (See also ECMA-394, Figures 12–18.)

In 2017, Johan van der Knijff of the Open Preservation Foundation shared findings from the Dutch National Library showing that disk images created from Blue Book CDs failed their validation step, even though nothing in the imaging process indicated that the capture had failed in any way (Van der Knijff, 2017). Van der Knijff explained that this was happening because the second data track required a physical offset value in order to be correctly parsed. With a disc in hand, it is possible to determine the location of the data track and the appropriate offset value. Van de Knijff described how to extract content from a disk image by generating a derivative image file left-padded with zeros to represent the physical offset. In our 2018 talk (Dietrich and Nelson, 2019), we built on the foundation of van de Knijff's work, providing examples of the user's perspective in imaging a Blue Book CD and attempts to read the resulting disk images, showing how each tool we tried (and, at times, different options within the same tool) yielded different output formats. We also showed that each kind of output required different post-processing actions in order to extract content. We concluded by offering one suggestion for determining the correct offset value of a Blue Book CD-ROM if the original disc was no longer available that appeared to successfully identify the correct offset value for the set of testing discs we sourced from our collections.

In reflecting on that work, we noted the similarities between Blue Book CDs and multi-session CD-R/Ws: Understanding that the offset value was necessary for a data track located in any session subsequent to the first was undoubtedly crucial. One important question remained for us, however. What findings from working with Blue Book CDs did not hold when applied to multi-session CDs? The goal of this paper is to provide additional detail on multi-session CDs to aid digital preservation practitioners in accurate and complete captures of the data they contain that can be applied in optical media workflows.

3. Generating a multi-session CD on the command line

This section is intended to illuminate the inner workings of a multi-session CD by guiding the reader through the creation of one using Ubuntu 20.04 LTS.

The steps illustrated below will generate a “data disc” as defined by ECMA-394: “a disc on which every Session contains one or more Data Tracks.” The test CD that we use throughout this paper will be a data disc constrained to one track per session. In the test CD, each data track will contain only ISO 9660 filesystem data.

The standards define abilities to include multiple data tracks per session, but our experience with disc authoring software frequently showed incomplete or inconsistent support for the capabilities provisioned by the standards. See Appendix B for a discussion on creating discs with multiple data tracks in a single session.

The purpose of this section is not to attempt to re-create the functionality of CD authoring tools, nor replicate all possibilities for multi-session CDs users may have in their collections. These steps make use of command line tools installed on an Ubuntu 20.04 LTS workstation. There are many options within the tools used here – **xorrisofs** [9] and **mkisofs** [10] – that can be used to generate a testing CD. The parameters specified in the following example detail the steps necessary to generate a basic disc that can be used for testing tools and workflows. The following examples specify **xorrisofs**, but **mkisofs** has equivalent parameters and can be used to generate a test CD using the same technique. (The documentation for **xorrisofs** also includes documentation for generating multiple-session CDs.)

To understand the sample commands below, the directory `first_session` contains files selected for the first session of the disc. In this sample multi-session CD, each session contains a single text file named after the session it is in: `SESSION1.TXT` is in the directory `first_session`. The contents of `SESSION1.TXT` are “This is track 1 in session 1.” We will use this convention for subsequent sessions (i.e., files for the second session will be indicated by a directory named `second_session`, and it will contain one file called `SESSION2.txt` containing text “This is track 2 in session 2.” and so on).

The first step to making a multi-session CD is to generate an ISO file with the data for the first session by running the command indicated below:

```
xorrisofs -o first_session01.iso first_session/
```

The next step is to burn `first_session01.iso` to blank CD-R or CD-RW media using the following command [11]:

```
cdrecord -multi -v -eject -speed=4 first_session01.iso
```

Next, determine what the offset for the second session must be and use that information to generate an ISO file with the data for the second session. This information is provided by the **cdrecord** command, when the freshly-burned disc is inserted back into the closed tray. This command records the necessary data:

```
cdrecord -msinfo > msinfo_after_session_1.txt
```

This command will not produce the correct result if the CD (that was just burned to) is not currently loaded into the drive. In the event of error, `msinfo_after_session_1.txt` will be empty.

The following command generates the ISO file with the data for the second session:

```
xorrisofs -o second_session02.iso -C $(head -n1  
msinfo_after_session_1.txt) -M /dev/cdrw second_session/
```

Next, burn `second_session02.iso` to the media with the data from the first session:

```
cdrecord -multi -v -eject -speed=4 second_session02.iso
```

Keep repeating the ISO generating step that calculates the next offset from the physical media and burning steps until the last session, and then run **cdrecord** without the **-multi** flag. Forgetting to calculate the session offsets with each ISO generating step will result in an unusable disc.

Utilities that list out the contents of each generated ISO file show how each session's file listing display includes the files from the prior session. For example, a view of the first session of the test CD described above would include `SESSION1.txt`, a view of the second session of the test CD would include `SESSION1.txt` and `SESSION2.txt`, a view of the third session of the test CD would include `SESSION1.txt`, `SESSION2.txt`, and `SESSION3.txt`, and a view of the fourth session would include `SESSION1.txt`, `SESSION2.txt`, `SESSION3.txt`, and `SESSION4.txt`.

4. Behavior patterns in disk imaging outputs of multi-session discs

It can be difficult to know when a multi-session CD is present in a workflow. This section elaborates on tool behavior of multi-session CDs created by disk imaging processes.

If the provenance of a disk image is unknown, there are a few possibilities for how it might be structured. The following list details categories of outputs that are possible given ECMA-394.

1. RAW disk image of an entire disc, including data tracks from all sessions
2. Disk image of the User Data portion of an entire disc, including data tracks from all sessions (referred to after this as a User Data disk image)
3. RAW session image of an individual session within a disc
4. User Data session image of an individual session within a disc
5. RAW track image of an individual data track within a session
6. User Data track image of an individual data track within a session

Using the imaging tools listed in Appendix A, we were able to generate disk images in all of the above categories except for category 4.

We repeat, as we believe it deserves clarification: If we consider the general, abstract class of "Disk images," we are considering two *disjoint* subclasses of disk images, the RAW disk image and the User Data disk image. We also define three disjoint subclasses of a more-general "Image", the disk image, session image, and track image. A RAW image will include error correction and detection data from the physical disc and its size will be a multiple of 2,352 bytes. A User Data disk image does not include this information, and only includes the "User Data" [12] portion of each sector. Its size will be a multiple of 2,048 bytes.

Detecting that one has a disk image of a multi-session CD using various parsing tools can be subtle. The lack of an explicit error message does not necessarily mean the tool successfully displayed the contents of that disk image file. The table below summarizes behaviors of tools when presented with a disk image file that matches the description in the middle column. The table specifically does not indicate how the disk images were created. It only provides detail on the behavior of the tools we tested to parse our test CD generated in Section 3 (i.e., multiple sessions, each session containing only one data track with ISO 9660 filesystem data).

Tool	Input Image Format	Tool Behavior
------	--------------------	---------------

<pre>mount -t iso9660 -o loop CD.iso mount_point</pre>	Image of any level (full disc, session, or track), RAW	Non-applicable. The mechanism to create a loopback device (/sbin/losetup) does not support conversion from RAW sectors to UserData sectors (i.e. flag-value --sector-size 2352 is rejected).
<pre>mount -t iso9660 -o loop CD.iso mount_point [13]</pre>	Image of the full disc, User Data	File listing at the mount point only includes the first track of the first session; listed file contains expected data.
<pre>mount -t iso9660 -o loop,sbsector=[offset to last session on disc] CD.iso mount_point [14] (See also Section 5 for description of the sbsector option.)</pre>	Image of the full disc, User Data	File listing at the mount point includes all files from all tracks; all files contain expected data.
<pre>mount -t iso9660 -o loop CD.iso mount_point</pre>	Track image from any of the sessions but the first, User Data	Error message: can't read superblock on /dev/loopX
<pre>mount -t iso9660 -o loop,sbsector=[offset to session of track on disc] CD.iso mount_point</pre>	Track image from any of the sessions but the first, User Data	Error message: wrong fs type, bad option, bad superblock on /dev/loopX, missing codepage or helper program, or other error.
<pre>mount -t iso9660 -o loop,sbsector=[offset] CD.iso mount_point</pre>	Track image from any of the sessions but the first, User Data, prepended with zeros corresponding to the session offset	File listing at the mount point includes files from all previous tracks and sessions back to the first; however, all files not originally included on the track/session specified by the offset contain no data. See Section 7 for further detail.
<pre>isoinfo [15]</pre>	Image of the full disc, User Data	With no offset provided, display metadata for first session data only with -I;

		<p>-p only reports one Path Table.</p> <p>When using -T with offset to session: display metadata for that session with -I; display start of Path Table for that session with -p.</p>
isoinfo	Image of the full disc, RAW	Non-applicable - isoinfo does not support RAW images.
isoinfo	Session image of any of the sessions but the first, RAW	Non-applicable - isoinfo does not support RAW images.
isoinfo	Track image from any of the sessions but the first, User Data	<p>With no offset provided, error message: "Short read on old image."</p> <p>When using -N with offset to specific session: display metadata for that session with -I; display start of Path Table for that session with -p</p>
fiwalk [16]	Image of the full disc, RAW or User Data	Reports out file listing for first session only.
fiwalk	Session image of any of the sessions but the first, RAW	TSK_Error 'Cannot determine file system type' using a variety of sector offset values and sector sizes.
fiwalk	Track image from any of the sessions but the first, RAW or User Data	TSK_Error 'Cannot determine file system type' ... using a variety of sector offset values and the automatically-used sector size [17].
fls [18]	Image of the full disc, RAW or User Data	Reports out file listing for first session only. Options for specifying an image offset (-o) yield the following error: "Cannot determine file system type."
fls	Session image of any of the sessions but the first, RAW	"Cannot determine file system type."

<code>fls</code>	Track image from any of the sessions but the first, Raw or User Data	With no extra parameters, error message: "Cannot determine file system type." With -b 2048 (defining sector size) and -o (with offset value), "Sector offset supplied is larger than disk image."
<code>tsk_recover -a [19]</code>	Image of the full disc, RAW or User Data	Files Recovered: 1 File corresponding to contents in the first track of the first session only.
<code>tsk_recover -a -o [sector offset corresponding to session]</code>	Image of the full disc, RAW or User Data	Cannot determine file system type (Sector offset: 11702)Files Recovered: 0
<code>tsk_recover -a</code>	Session image of any of the sessions but the first, RAW	Cannot determine file system type (Sector offset: 0)Files Recovered: 0
<code>tsk_recover -a -o [sector offset corresponding to session]</code>	Session image of any of the sessions but the first, RAW	Cannot determine file system type (Sector offset: 11702)Files Recovered: 0
<code>tsk_recover -a</code>	Track image from any of the sessions but the first, RAW or User Data	Cannot determine file system type (Sector offset: 0)Files Recovered: 0
<code>tsk_recover -a -o [sector offset corresponding to session]</code>	Track image from any of the sessions but the first, RAW or User Data	Cannot determine file system type (Sector offset: 11702)Files Recovered: 0

5. Accessing content from images of the full disc

This section will illustrate how to use **isoinfo** to extract content from an image of a full disc, whether the image is RAW or User Data. The technique for extracting content in this and the next sections 5 through 7 focuses only on images that contain an ISO 9660 file system.

First, obtain the appropriate offsets; either with the physical media in hand, described in 5a; or from imaging metadata, described in 5b; or from the image itself, described in 5c.

If you have a RAW disk image, follow the directions for converting to a User Data disk image as described in 5d. The process of using **bchunk** [20] will generate track-level images; follow the directions for extracting content in 7. If you have a User Data disk image, use the technique described in 5e.

It is also possible to extract content from an image of the full disc by using **mount** and the value of the session offset (in 2KiB sectors) for the last session on the disc. The following command worked using a full disk image generated from our test CD:

```
mount -t iso9660 -o loop,sbsector=[offset to last session] \
CD.iso mount_point
```

5a. Obtaining the disc offset(s) from the physical media

If the original physical disc is still accessible, the command line utility **cd-info** [21] will display the offset(s) present on that media. Sample output from the multi-session CD created in Section 3 is below:

```
Disc mode is listed as: CD DATA (Mode 2)
CD-ROM Track List (1 - 4)
  #: MSF      LSN   Type   Green? Copy?
  1: 00:02:00 000000 data   false no
  2: 02:38:02 011702 data   false no
  3: 04:14:04 018904 data   false no
  4: 05:50:06 026106 data   false no
170: 05:54:08 026408 leadout (59 MB raw, 58 MB formatted)
Media Catalog Number (MCN): 00000000000000
Last CD Session LSN: 26106

-----
CD Analysis Report
CD-ROM with ISO 9660 filesystem
ISO 9660: 184 blocks, label `ISOIMAGE'
Application:
Preparer   : XORRISO-1.5.2 2019.10.26.180001, LIBISOBURN-1.5.2,
LIBISOFS-1.5.2, LIBBURN-1.5.2
Publisher  :
System     :
Volume     : ISOIMAGE
Volume Set :
session #2 starts at track 2, LSN: 11702, ISO 9660 blocks: 174
ISO 9660: 174 blocks, label `ISOIMAGE'
session #3 starts at track 3, LSN: 18904, ISO 9660 blocks: 174
```

```
ISO 9660: 174 blocks, label `ISOIMAGE`  
session #4 starts at track 4, LSN: 26106, ISO 9660 blocks: 174  
ISO 9660: 174 blocks, label `ISOIMAGE`
```

From this output, we see the Logical Sector Numbers (LSNs) corresponding to our sessions are 0, 11702, 18904, and 26106. These are the numbers we would use, e.g., in the `sbsector` option to **mount**.

5b. Obtaining the disc offset(s) from imaging metadata

Some imaging utilities automatically generate metadata that includes offset information for sessions present on a CD. While technically, the cue sheet syntax [22] does not include fields that would supply that value, some programs supply this information in the unstructured-text comments of their respective cue sheets. Lines starting with REM are comments that some utilities use to store offset values.

5c. Obtaining offset values from full User Data disk images

For User Data images of full multi-session discs, obtaining the offset of any session after the first may be possible using **isoinfo** through testing every possible option. To find all of the possibilities, divide the disk image file size by the sector size (2,048). For each possible offset, run

```
isoinfo -i CD.iso -T $sector_num
```

Any sector value that does not yield the error message, “CD-ROM is NOT in ISO 9660 format” or “Short read on old image” is a usable offset value. Beware that using **-N** with an incorrect offset value may not result in an error message, but instead yield an incomplete listing of files.

5d. Conversion to User Data from RAW using bchunk

One method of converting a RAW image to a User Data image is by post-processing it using the command **bchunk**. This utility accepts a RAW image and a cue sheet and uses these to generate User Data images for data tracks and “native CD audio” or WAV files for audio tracks (quoted directly from **bchunk** man page).

A sample **bchunk** command follows:

```
bchunk raw_disc_image.bin cd.cue derivative_iso_basename
```

Not all disk imaging utilities supply a cue sheet as output by default, and in some cases, creation of a cue sheet must be handled separately from the image creation process.

5e. Extracting files from images of the full disc

In our testing, we found the command line tool **isoinfo** is able to extract files from an image of the full disc using the corresponding session offset(s).

Consider the test multi-session CD created in Section 3. If the file `SESSION1.txt` was added in the first session as the first track, `SESSION2.txt` was included in the second session as the second track of the disc, and `SESSION3.txt` was included in the third session as the third track, (and so on), in order to extract `SESSION2.txt` from an image of the full disc, one would need the offset for the second session. The offset for the third or fourth session would not work, even though when listing files from a specific track using a session offset, that file would appear to be listed (because all of the files from each of the previous sessions' tracks are listed).

We used the following command to get a complete listing of all of the files in each track from all of the sessions of our test CD.

```
isoinfo -i CD.iso -T 26106 -l
```

When working with an image of the full disc, **isoinfo** requires the **-T** flag (instead of the **-N** flag for the single-track image).

We used the following command to extract an individual file from the image and redirect it to a file named `session4.txt`:

```
isoinfo -i CD.iso -T 26106 -x '/SESSION4.TXT;1' > session4.txt
```

The `;1` in the filename display is for the ISO file version and may not be present on all multi-session CDs. If it is present in the **isoinfo** listing, it should be used when invoking the specific commands to extract files, as shown in our example below.

Ensure that the full absolute pathname within the disc's last session's file system is accurate (the **-f** flag will provide the full path to be used when invoking the **-x** flag).

With the commands exercised so far, it is possible to see how in some CD authoring programs, it was possible to “delete” a file in a subsequent session. When working with a multi-session disc where the creator did this, the file listing of the track in the last session will not include those so-called deleted files. They can still be located in an image of the full disc, and within the metadata, by using the offset to the session where that track was included. Do carefully consider how to handle this “deleted, but recoverable” (Casey et al., 2019) data when working with real-world collection material (Lassere and Whyte, 2021).

6. Accessing content from single-session images

In our testing, we were only able to generate single-session RAW images. Since **isoinfo** only handles User Data images, it cannot be used directly with a RAW image. Use the technique for converting the single-session RAW image into single-track User Data images described in 7b first, obtain session offsets (either by following 5a, 5b, 5c, or 7a), and then proceed to the techniques outlined in 7c.

7. Accessing content from single-track images

In order to extract content from single-track images, you must have the image of the track where the file was originally included on the disc in order to extract content, as well as the session offset for that particular track.

Depending on your starting point, it may be possible to obtain session offsets using the techniques available with the disc in hand (5a), or using imaging metadata (5b), or from an image of the full disc (5c). If these techniques aren't possible, try the technique in 7a.

If the imaging tool that you used generated track-level User Data images, or you generated such images using **bchunk** (described in 5d) you can use the technique described in 7c to extract content from those images. If you have single-track RAW disk images, convert them to User Data using the technique in 7b.

It is also possible to extract files from single track-images using the technique referenced in (Van der Kniff, 2017 and [23]) of prepending an image with sectors of zero-bytes corresponding to the session offset and mounting the resulting image. We confirmed this in our review of `mount` in the table in Section 4. The same caveat applies here too: not every file that will be listed under the resulting mount point will contain data captured from the physical disc. In our testing, only the files originally added in that track will have the expected data.

7a. Offsets from single-track images

For single-track images, **isoinfo** combined with the **-p** flag will report the start block of the Type-L path table [24]. Noting the ISO 9660 spec [25], the path table immediately follows the volume descriptor set, which consists of at least two volume descriptors – a primary volume descriptor and a volume descriptor set terminator, which is itself preceded by 16 sectors designated as a system area. From this information, it should be possible to backtrack and determine the value of the original offset for that session.

In testing the single-track disk images created from our sample multi-session disc, subtracting the path table location from the number of volume descriptors and the 16 sectors of the system area consistently yielded an offset value that was one sector greater than the value reported by

utilities that read the physical disc. Thus, our formula for determining the offset of the session was calculated by the following formula:

Type-L path table location - number of volume descriptors - 17

7b. Conversion to User Data from RAW

One method of creating a User Data disk image from a RAW image is described in 5d. It assumes an image of the full disc and the existence of a cue sheet generated from the original physical media. If all you currently have is a track image and no cue sheet information, it might be possible to generate a cue sheet that represents an abstract single data track in order to use as input to **bchunk**. Save the following text as `file.cue` (with each indentation level being two single-whitespace characters):

```
FILE "data_track.bin" BINARY
  TRACK 01 MODE1/2352
    INDEX 01 00:00:00
```

Once this is generated, it is possible to use the following **bchunk** command to convert a RAW image to User Data:

```
bchunk data_track.bin file.cue derivative_iso_basename
```

Under the scope of images we inspect in this paper—one data track per session—this "default" supplement should enable conversion to User Data. The "MODE1" string may need to be changed to "MODE2", because User Data within a mode-1 sector appears at a different location than a mode-2 sector. If you use the wrong mode, the resulting image will not appear to have an ISO9660 file system, so there is effectively no chance of getting incorrect file extractions with the incorrect mode - the file system will not function.

7c. Extracting files from single-track images

The process of extracting files from single-track images is similar to what is described in 5e. Use **isoinfo** with the offset to the session that has the track where the file is originally added to the disc and the pathname. Unlike in 5e, use the **-N** flag (instead of the **-T** flag) with the session offset.

Working with an image of a track from the last session of our test multi-session CD, we used the following command to get a listing of all of the files listed in the fourth file system. (This will not necessarily be the listing of every file in every track, but for our test CD which did not exercise "deletion," this will be a complete listing.)

```
isoinfo -i Session4.iso -N 26106 -l
```

We have provided sample output of this command below:

```
[      24 00] SESSION1.TXT;1  
[  11726 00] SESSION2.TXT;1  
[  18928 00] SESSION3.TXT;1  
[   26130 00] SESSION4.TXT;1
```

We used the following command to extract a specific file:

```
isoinfo -i Session4.iso -N 26106 -x '/SESSION4.TXT;1' >  
session4.txt
```

This example works because the file `SESSION4.txt` was added in the fourth track (within the fourth session), but generally having only an image of the last track does not mean you will be able to extract every file referenced in the file listing from that image.

One way to check whether a file is extractable from an image is to observe the corresponding session offset value and compare it against the values in the column to the left of the file name in the **isoinfo** file listing output. If the leftmost number in the bracketed area is less than the session offset that will be supplied to **isoinfo**, then the extraction will fail. In the above example, 26106 is the offset value for the fourth session, and the number next to the file name listing for `SESSION2.TXT` is 11726. That means that any **isoinfo** command that supplies an offset value for the fourth session (26106) will not be able to successfully extract `SESSION2.TXT`.

8. Extracting files from physical media

With the media in hand, it is possible to directly access files from a multi-session CD. On the graphical interfaces for Windows 10, Ubuntu 20.04 LTS, and macOS Big Sur, when presented with our test data-only multi-session CD, all of the files from every track in each session of the disc were displayed and accessible.

When copying files directly from the media, if a multi-session CD includes file systems other than ISO 9660, copying directly from the media may not include the originating file system features a current workstation's file system does not support, (e.g., resource forks from the source file system not being supported in a destination file system in a Windows environment).

9. Non ISO 9660 filesystem data

We attempted to generate a test multi-session CD that included HFS file systems using **mkisofs** and the technique outlined in Section 3, but were unsuccessful. (**xorrisofs** only supports embedding HFS+ partitions, and could not be used for this test.) According to the

documentation for **mkisofs**, “only files from the last session will be in the HFS volume...**mkisofs** can not add existing files from previous sessions. However, if each session is created with the **-part** option, then each session will appear as separate volumes when mounted on a Mac.”. We did find one resource that suggested HFS discs cannot be “multi-session”, but instead must be “multivolume.” [25] Without confirmation through tests, this information suggests that images of individual sessions with HFS data can be accessed through specific file system tools such as **HFSExplorer** [26] or the **hfsutils hmount** [27] command; however it is unclear how those same tools might work on an image of the full disc.

Conclusion

When working with optical media, workflows that are unaware of, or do not account for, formats such as multi-session discs may risk losing data. Multi-session CD-R/Ws may pose significant challenges if treated in the same manner as single-session, single-track media.

Acknowledgements

The authors gratefully acknowledge Bez Thomas for supplying the inspiration for the title of this article.

Appendix A: Imaging pathways

This section describes some of the pathways by which one can create an image of a multi-session CD. This might be helpful in understanding the provenance of images if the physical media is no longer available.

Guymager

When imaging our sample multi-session CD, **Guymager** [28] created a User Data disk image of the entire disc. While conducting testing, Guymager logged a significant number of bad sectors, even though the resulting image was understandable (using the techniques described in this paper) with **isoinfo**.

dd

When imaging our sample multi-session CD, **dd** [29] gave the following error:

```
dd: error reading `dev/cdrom': Input/output error
```


The resulting file was understandable (through **file** [30] and **disktype** [31]) as a disk image containing ISO 9660 filesystem data, but it contained only the data from the track of the first session; thus, it appeared to generate a single-track User Data disk image. This contrasts with the behavior exhibited by the GUI interface for accessing the mounted disk.

ddrescue

We issued the following command to test **ddrescue** [32] (installed from the `gddrescue` package) on our sample CD:

```
ddrescue -n -b 2048 /dev/cdrom <ISO file> <log file>
```

The utility reported 7 bad areas and 864 read errors for one of the authors, and 6 bad areas and 861 errors when tried by the other; but in both cases produced a disk image that contains ISO 9660 filesystem data, as indicated by running both **disktype** and **file**. This image contained data from all tracks from all sessions' data; thus, it appeared to generate a full disc User Data image. To confirm, when we supplied the appropriate offsets we could obtain file listings for all four sessions (using **isoinfo**), as well as read the contents of all of the files.

cdrdao

This utility will create a RAW image and toc-file for a single session only (see: **--session** flag); thus, it is generating a single-session RAW disk image. The default filename if none is given for the output is `data.bin`. A sample command is given below:

```
cdrdao read-cd --read-raw --session 2 --datafile cd.bin cd.toc
```

The toc-file output, which is specific to the **cdrdao** [33] utility, includes the same technical data as a cue sheet (tracks, pre-gaps, etc.) in a different structure and layout. The command line utility called **toc2cue** [34] can convert this toc-file into a cue sheet.

Appendix B: Multiple Data Tracks

ECMA-394 [5] defines a “data disc” as “a disc on which every Session contains one or more Data Tracks” suggesting that an individual CD may contain multiple data tracks. However, it was difficult to find examples to support that these kinds of disc were common, or that their creation was supported by CD burning software. To give an example, in the dialogue box for CD burning software (PCMag, n.d.), the option to leave a session open to add new tracks is not available if the user had previously selected to generate a data CD. While there were examples for generating mixed-mode CDs in the **man** pages for **cdrecord** [11] and **cdrdao** [32], there were

no examples given in either **man** page to generate a disc with multiple data tracks within the same session. Further, the **man** page for **cdrecord** notes that, “Many operating systems are not able to read more than a single data track, or need special software to do so.”

We attempted to generate two testing discs: one CD that consisted of a single session and nine data tracks and another CD that consisted of three sessions, each with three data tracks per session.

In order to generate the first test CD, we used the following command:

```
cdrecord -v -eject -speed=4 track01.iso track02.iso track03.iso \
track04.iso track05.iso track06.iso track07.iso track08.iso \
track09.iso
```

In order to generate a testing disc with three sessions and three data tracks per session we used the methodology outlined in Section 3, with each **cdrecord** step including three data tracks that were each generated (when applicable, using the offset of the previous session) with **xorrisofs**.

We noticed the output of **cd-info** with both testing discs in the drive reported nine “sessions”, with one data track per “session”. This discrepancy between the layout we intended and the tool output confounded our ability to establish ground truth, and without understanding our initial inputs into the workflow, we chose to not pursue this line of inquiry further.

References

Brown, Geoffrey. [Developing Virtual CD-ROM Collections: The Voyager Company Publications](#). International Journal of Digital Curation (7): 2 (2012). <https://doi.org/10.2218/ijdc.v7i2.226>

Casey, Eoghan, Nelson, Alex, and Hyde, Jessica. [Standardization of file recovery classification and authentication](#), Digital Investigation (31) (2019). <https://doi.org/10.1016/j.diin.2019.06.004>

Dietrich, Dianne and Alex Nelson. [The Emperor's New Grooves: Recognizing Multisession CD-ROM tracks not captured in disk images](#). Talk at BitCurator User Forum 2019.

Lassere, Monique and Jess M Whyte. [Balancing Care and Authenticity in Digital Collections](#). Journal of Critical Library and Information Studies (3): 2 (2021). <https://doi.org/10.24242/jclis.v3i2.125>

PCMag. [Definition of track-at-once](#). PCMag Encyclopedia. Accessed 2022-12-02. <https://web.archive.org/web/20220127030639/https://www.pcmag.com/encyclopedia/term/track-at-once>

Rothrock, Michelle, Alison Rhonemus, Nick Krabbenhoeft. [Assessing High-volume Transfers from Optical Media at NYPL](#). *code4lib Journal* (51): 2021-06-14.

Van der Knijff, Johan. [Imaging CD-Extra / Blue Book discs](#). Open Preservation Foundation. 25 April 2017.

Endnotes

[1] [Mixed Mode CD - Wikipedia](#)

[2] [Compact Disc Digital Audio - Wikipedia](#)

[3] [Yellow Book \(CD-ROM standards\) - Wikipedia](#)

[4] [Orange Book \(CD standard\) - Wikipedia](#)

[5] [ECMA-394](#)

[6] [ISO 13490 - Wikipedia](#), [ECMA-168](#)

[7] [Blue Book \(CD standard\) - Wikipedia](#)

[8] [Track \(optical disc\) - Wikipedia](#)

[9] [xorrisofs\(1\) - Linux man page](#)

[10] [mkisofs\(8\) - Linux man page](#)

[11] [cdrecord\(1\) - Linux man page](#)

[12] [ECMA-130](#) (see Figure 11)

[13] [mount\(8\): mount filesystem - Linux man page](#)

[14] Attempts to use the `session=x` option in `mount` with an image of the full disc were unsuccessful; only the file from the first track of the first session was listed at the mount point.

[15] [isoinfo\(1\) - Linux man page](#)

[16] [Fiwalk - Forensics Wiki](#)

[17] Our attempts to force TSK tools to use a sector size, any power of 2, were disregarded by the TSK library on encountering the ISO9660 file image. RAW data was also translated into

User Data for analysis. Hence, we did not experiment further with attempts to force sector sizes in TSK.

- [18] [FLS\(1\) manual page](#)
- [19] [TSK RECOVER\(1\) manual page](#)
- [20] [bchunk\(1\) - Linux man page](#)
- [21] [cd-info\(1\) - Linux man page](#)
- [22] [Cue sheet \(computing\) - Wikipedia](#)
- [23] [\[Libcdio-devel\] Re: Retrieving DATA session from multisession audio disc](#)
- [24] https://en.wikipedia.org/wiki/ISO_9660#Path_tables
- [25] [ISO 9660 - OSDev Wiki](#)
- [25] [Information on CD Formats](#)
- [26] [Catacombae - HFSExplorer](#)
- [27] [HFS Utilities Home Page](#)
- [28] [Guymager](#)
- [29] [dd\(1\): convert/copy file - Linux man page](#)
- [30] [file\(1\): determine file type - Linux man page](#)
- [31] [disktype Documentation](#)
- [32] [GNU ddrescue Manual](#)
- [33] [cdrdao\(1\) - Linux man page](#)
- [34] [toc2cue\(1\) - Linux man page](#)