AUDIO CD-R REFORMATTING AT IUMDS

Patrick Feaster
October 27, 2020

Background[1]

The audio CD is a challenging format from a preservation standpoint. Although the data is encoded digitally, it has none of the practical advantages associated with digital files that can be losslessly migrated, and it must instead be "ripped" as a continuous stream of digital data. Each read of the data is liable to contain different uncorrectable errors in case of damage, resulting in different interpolations and potential glitches.

There's something of a contradiction among recommended practices for reformatting audio CDs.

On one hand, the favored software tools—in particular, Exact Audio Copy (EAC)—rely on having a single drive attempt to read from the disc multiple times, and quite possibly *many* times, stopping and backing up again and again, to move on only after a certain number of identical reads are achieved or after a maximum number of read attempts has been reached. In other words, the disc is put *once* into *one* drive, which then reads each part of it as many times as needed to meet a given benchmark. The logic behind this approach is that a value that's read consistently is more likely to be correct than a value that's read only once or inconsistently. Unfortunately, this process can be very time-consuming and puts significant wear and tear on the drive. Moreover, it's claimed that a given drive's error correction protocol can interpolate identically multiple times, such that consistency isn't necessarily a good measure of accuracy.

On the other hand, it's generally understood that different drives can yield different results, and that even the range of angles at which a disc can be acceptably placed in the *same* drive can have an impact on the optics of data extraction. Thus, it's also recommended that problematic discs be tried in several drives, the idea being to use whichever drive seems to handle it most successfully (perhaps in combination with a tool such as EAC).

However, it's quite possible that two drives could read a given disc *differently* without one or the other being more accurate *as a whole*—or, in other words, both drives might introduce errors *at different points*—especially if problems are due to physical damage or degradation rather than to consistent incompatibilities. The same might also be true of a disc placed into the same drive twice at imperceptibly different angles. The remedial workflow we've developed at IUMDS for handling audio CD-Rs that were failed during an initial reformatting attempt by Memnon takes this observation as its point of departure.

[1] See also the related presentation "Strategies for Reformatting Degraded Audio CDs" at the Association for Recorded Sound Collections virtual conference, May 21, 2020, available online as of this writing (10/27/2020) at https://arsc.aviaryplatform.com/collections/939/collection_resources/25005

Basic Workflow

We begin by checking whether a disc is, in fact, an audio CD-R rather than a data CD-R (which is out of scope for our project). EAC will display differently formatted contents in either case. If the information being reported isn't clear, or a disc isn't recognized by the Windows operating system, we may instead use IsoBuster to view its contents, which is usually sufficient for diagnosis. If a disc turns out to contain data, we record this in the POD as "PROBLEM: data disc," generally with a note as to what file format(s) are present. The disc is then returned with no further action.

If a CD is confirmed as containing audio, we try to obtain an EAC **secure rip** by trying it successively in each of multiple drives. As a rule, we've used the following settings:

- Drive has 'Accurate Stream' feature – yes (true of virtually all "modern" drives)
- Drive caches audio data – yes
- Drive is capable of retrieving C2 error information – no

If a drive caches audio data, EAC will read in overlapping chunks to overcome this; and if it's not capable of retrieving C2 error information, EAC won't rely on this information but will check every sector for consistency. Thus, these settings are chosen to be as conservative as possible and to make no potentially mistaken assumptions about the abilities of a given drive.

Our default drive has been the one mounted in whichever workstation has been used for CD ripping. This has always been a PLDS DVD±RW drive, with DS-8ABSH being a representative model (but with potential for minor variation from workstation to workstation). The first alternate for most of the project has been a Hitachi-LG Data Storage USB drive, model GP65NB60. These two drives are referred to in the metadata associated with specific rips as "PLDS" and "GP65N." Other drives have also seen sporadic use, including an older Compaq USB drive that yielded unusually robust results at first but stopped working a short while into the project.

A first attempt is generally made with the PLDS drive, with each track being ripped to a separate file and those files being saved to a folder named with the barcode number. Any tracks that are reported as successfully ripped is left in the root of that folder. If any tracks contain any reported errors, those are instead moved to a subfolder named "PLDS 1." The disc is then placed in the GP65N drive, and another attempt is made to rip the tracks that contained errors before. Any tracks that still contain errors go into a "GP65N 1" folder. The process then repeats through "PLDS 2" and "GP65N 2." In some cases, one or the other drive may not recognize a given disc. In those cases, even if we're repeatedly using a single drive, we'll remove the disc from it and replace it between rips to accommodate the possibility that slight differences in the physical mounting of the disc may make a difference.

If no rip of a given track is ever reported as error-free, we use an algorithm to compare all the different rip attempts and take the median average of each sample. (Initial experiments took the mode average instead, trying to ascertain a "consensus value," but results appeared to be less successful in practice.) Experience has shown that the different rips line up with each other at the sample level with sufficient reliability to employ averaging in this way, with only rare exceptions. In effect, this approach extends the logic of EAC to the use of multiple placement in multiple drives.

Our basic "concatenator," **concat.exe**,[2] reads the table of contents from the disc (which still needs to be mounted in an accessible drive) and checks it against the file durations of all ripped tracks, reporting any discrepancies. It generates two pairs of WAV and CUE files: a Preservation (Pres) version containing all the separate rips in their original form and a Preservation Intermediate (PresInt) version which contains only a single copy of each track, averaged from multiple rips where appropriate (i.e., in cases where the script doesn't find a file in the root folder but finds multiple files in subfolders). The averaging simply arranges the samples of all rips of the same track into an array and takes the median value for each sample. Samples missing from any rip are disregarded. A problematic rip will sometimes conclude with spurious values of zero, so strings of concluding zeroes are also disregarded if any of the available rips contains non-zero values for those samples. The Pres file contains one copy of each track in order (the longest available file; or, if there are multiple files of that length, the first one it finds while going through the subfolders), and then the remaining contents of each subfolder in turn. The CUE sheet associated with the Pres file includes the subfolder name in the name of each track sourced from it, e.g., "Track11(GP65N 1)." The CUE sheet associated with the PresInt file marks any track pulled from a subfolder or averaged from multiple rips, e.g., "Track11(error)." An accompanying TXT file begins with a statement as to whether the Table of Contents matched the available rips—almost always "No discrepancy found between ripped audio and TOC," to the point that the check may not in fact be particularly useful. The TXT file then presents the EAC reports for each rip appended in chronological order. In rare instances where no EAC report was generated (due to manual interruption or to a computer crash or reboot) there may also be a note inserted to this effect.

For each audio CD-R reformatted by IUMDS, we submit five files to the Packager: a Preservation WAV and CUE, a Preservation Intermediate WAV and CUE, and a TXT report.

Modified Workflows for Special Situations

Sometimes a secure rip will end up taking too long for comfort; we've run EAC on some damaged tracks for a week straight without having it complete the rip. For these cases, we tried some different approaches, at first centered on other EAC modes ("fast" and "burst").

But in the meantime, we also encountered a second situation EAC couldn't handle at all: namely, audio CD-Rs that hadn't been closed after being burned. These won't show up in a Windows operating system, and EAC is unable to recognize or access them. However, IsoBuster can detect and read such discs, so that's what we use instead (via "Extract Audio to Wave file"). Because IsoBuster doesn't report errors as EAC does, we always rip every track multiple times and average the results. Subfolder names contain the letters "iso," e.g., "PLDS iso 1." Because the Windows operating system doesn't recognize a disc of this kind, we can't compare its Table of Contents with the duration of rips as we usually do. So for these situations we use an alternate version of the concatenator (**concat2.exe**) which skips that step (which, as mentioned above, might not be all that useful in practice anyway). The number of tracks on a CD needs to be entered manually into this version of the concatenator so that it will know how many tracks to look for, and the first line of reports made using this script reads: "Opted not to check ripped audio against TOC." This other concatenator also exports a reference image for each track in which each pixel represents a sample and brightness represents discrepancies among rips, helping us make a quick assessment as to whether a CD ripped with IsoBuster had many errors or not. A relatively black image

[2] Written in MATLAB and run from an executable compiled for Windows 10.

should be accurate, while a relatively white image is likely to contain glitches. These image files are stored only temporarily and not included in submission packages.

Once this alternative **IsoBuster / concat2.exe** workflow was in place, we also began using it by default for other cases in which an EAC secure rip failed to yield results in a timely fashion. As the project progressed, it emerged as our standard "Plan B."

Another special situation involves "hybrid" discs containing both audio and data tracks. Four of these turned up among discs received by IUMDS. It appears that Memnon routinely reformatted only the audio portion of such hybrid discs, so we have no means of gauging how many others passed through MDPI with their data tracks ignored. As of this writing (10/27/2020), we've ripped the audio and copied the data from each of the four hybrid discs at IUMDS, but we haven't yet resolved what (if anything) to do with the data portions, e.g., whether to try to submit them in separate ZIP files.

Higher-Level Strategy

Broadly speaking, our approach to audio CD-Rs has involved cycling through discs in multiple passes, with each pass involving more aggressive methods (as described above), such that we picked off the "easy" discs at the beginning, addressed "harder" discs later on, and ended up with an accumulation of "difficult" or "impossible" discs at the end. Partway through the project there was a new infusion of "easy" discs which came to IUMDS because they were identified only after Memnon had ceased work on this format (allowing the NOA license which they had been using to expire). These were the discs we were working on at the time of a temporary shutdown due to the COVID-19 pandemic, which also coincided with a failure of the CD ripping workstation. A replacement workstation finally entered operation in August 2020, and the remaining "easy" discs were completed over the next month, leaving us to get the best results we could from the remaining "difficult" discs by the end of the project in December 2020.