DYSON-HESS
Noise Reduction Decoder System

John S. Dyson and Richard L. Hess

COPYRIGHT

dhnrds.com
contact@dhnrds.com

Special thanks to:
George Massenburg, Bob Katz, Bob Ludwig, Tom Fine,
Don Ososke, Phyllis G Diangelo, and Mary Beth Hess

WARRANTY

This software is sold on an as-is basis with no guarantees that it will perform to your expectations. We have thoroughly tested the software and believe that it is a quality product, but there may be incompatibilities with your system.

We will attempt to solve any problems, but our liability is limited to the purchase price of your license. We cannot and will not be held liable for any consequential damages.

WARNING

We specifically disallow the use of this software on original master files for which you do not have a backup (or two) on separate drives. It is not that we do not trust the software, but in the rush to get things done, it is too easy to possibly delete an original file, and we cannot be responsible for that, and don't want that to happen to you.

That is why we do not mention the piping and redirect commands for the decoders, as that approach can overwrite files without warning.

Table of Contents

Blank

INTRODUCTION

The Dyson-Hess Noise Reduction Decoder System is a stand-alone application suite designed to provide for high-quality reproduction of analog audio recordings encoded with several compressor-expander (compander)-type noise reduction systems. These systems, first introduced in 1966, improve the perceived signal-to-noise ratio by changing the signal dynamics in a controlled manner, reducing (compressing) them during the recording process and increasing (expanding) them in a complementary way during the playback process. These are generically referred to as “compander” (COMPressor expANDER) systems.

It appears that 17 different systems were sold for both professional and consumer use, with widely varying market acceptance. Two of them appear to be compatible with a third widely sold system, so that leaves what we believe are 15 systems, each requiring a different decoder in order to properly reproduce the original recordings.

We have great respect for the original engineers and the companies who made these systems and appreciate their use of the best technology available at the time to improve upon a medium in a way that was deemed necessary by a large percentage of the users of analog magnetic recording. None of the companies involved with the production of this equipment has offered—even with continued prodding—a software decoder for their systems, and all of the hardware decoders have been out of production for about a decade at least and the latest models were introduced in 1990 or before. We acknowledge that these companies are active and provide valuable products today and we do not call our modules by the specific brand name. We have not received any blessing or permission from the companies involved with the original hardware units and these software decoders are the result of measurement of hardware decoders and analysis of the hardware circuitry as well as extensive listening tests.

Our software is meant to be accurate decoders for the tapes that were encoded with various noise reduction systems. We are neither claiming nor implying that the original encoder/decoder manufacturers are endorsing these software decoders. We avoid mentioning the trademarked names to honour that respect.

As stated above, we look at this as a system or a suite of software applications. There will be a Windows graphical user interface (GUI) that will control multiple decoder modules. The decoder modules also are able to run in command-line interface (CLI) mode. We currently have available a decoder for the pioneering and highly successful “DA” system. We have begun work on the GUI. There is a vast number of calculations involved in the lower-distortion modes, modes that were never feasible in the hardware domain.

The GUI will allow the operator to set up how the decoder(s) will operate. Once set, it will run in batch mode, overnight, if necessary to process the desired files.

PROJECT ROAD MAP

This project is intended to provide professional-level decoding of recordings made with several noise reduction compander systems. We see three steps in the road map:

1. The DA project and the GUI should be ready mid 2019, the stand-alone DA decoder with CLI is available for public beta testing now.
2. Throughout 2019, we hope to finish decoders for DB, DC, X-I, X-II, and C4.
3. In 2020, we will be evaluating the feasibility of undertaking DSpR and DS decoders.

The delay in undertaking the DSpR and DS decoders is due to their huge complexity over the other decoders. We estimate that the DSpR will be an order of magnitude more complex than the DA decoder (which has been in the works for a year and a half), and the DS is based on the DSpR technology. We wish to complete the other decoders first.

As you can see, we are addressing both professional and consumer decoders. The reality of archive work is that the consumer compander systems make their way into archives as well as the professional systems. While it is practically impossible to know with 100% certainty the distribution of recordings made with all 15 different systems, in Mr. Hess's experience of digitizing tapes for over two decades (and making them for three decades prior), these eight systems are all that he has ever used, and he has a number of hardware channels of each.

Since High-Com, High-Com II, and CBS-CX are related, just tweaked in different ways, and CBS-CX was widely used on LaserDiscs, we are in discussions with an expert with these systems and may consider software decoders for the High-Com family of formats. Mr. Hess has a Telefunken High-Com and a Nakamichi High-Com II decoder. The Hungarian Ex-Ko and JVC ANRS systems should decode reasonably well with the DB decoder. Mr. Hess has a hardware decoder for Sanyo Super D, for which we are **NOT** planning to make a software decoder. Additionally, we are **NOT** planning on offering software decoders for the Burwen Noise Eliminator (which was never commercially released), the JVC Super ANRS, nor the Toshiba ADRES systems. All of these formats, except CBS-CX are mentioned in Mr. Hess's 2018 slide deck from his presentation at the October AES Conference in New York City. This is available at www.dhnrds.com.

COMPATIBLE FILE TYPES

All decoders in this system will use the following file types. Individual decoder instructions will note variations.

1. The decoder requires WAV files. Mono or stereo files are acceptable. Multi-channel files (beyond stereo) are not currently supported.
2. It will operate at and has been tested at 44.1 kHz through 192 kHz. For a variety of reasons, the conventional sampling frequencies in order of performance from best to worst are: 96 kHz, 192 kHz, 48 kHz, and 44.1 kHz.
3. Input files can have data types of: 16 bit signed integer, 24 bit signed integer, and 32 bit floating. The output file is 24 bit signed integer or 32 bit floating, depending on input file bit depth, but can be forced to 32 bit floating, using `--floatout`.
4. WAV file options are: Broadcast Wave (BEXT), rf64, and bw64.
5. See the section, **Command Line Switches—Advanced** for additional decoder-specific options.
6. If 16 bit output files are needed, they should be carefully normalized and then redithered in a DAW.
7. While we do provide some means of level adjustment into and out of the decoder, if overs are noticed, the best choice is to use the `--floatout` option and deal with any issues in the DAW.

Audio File Bit Depths		
Input File	Output File	
	Normal	with `--floatout`
16 bits	24 bits	32 bits float
24 bits	24 bits	32 bits float
32 bits float	32 bits float	32 bits float

DYSON-HESS NOISE REDUCTION DECODER TYPE DA (DHDA)

QUICK TESTING SETUP

This section is meant for using the demonstration version for a quick test.

1. Copy the appropriate exe file plus the three dll files from the distribution into a folder.
2. Place copies of the original material to be decoded in the same folder.
3. Open a command prompt and navigate to that folder. Enter:
4. `da-avx --tonel=-13 --toner=-13 --info=1 --finalplus --floatout --input="infile.wav"`
5. followed by `Enter`
6. Use `da-win` instead of `da-avx` for computers without the AVX2 instruction set.
7. The calibration tone parameters (`--tonel=x --toner=x`) should match the calibration tone on the recording to be decoded. If the tone is the same for both channels, `--tone=x` can be used, but we suggest the discipline of assigning the individual channel tone values.

> Use your DAW for measuring the tone levels and make certain to check the slow meter calibration with the included –20 dBFS test tone.

> Provide a `tone=` parameter for both channels for mono files.

- Info provides evidence that the program is running.
- Finalplus is the best quality option. Basic and Finalize are alternates.
- Floatout provides the output in a 32-bit floating point format.
- Infile is the name of your source file, with extension (must be wav).
- The output file name is automatically generated by appending `_DHNRDS_DA_<program-version>_DECODED`.

FORMAL WINDOWS INSTALLATION

In order to properly use the CLI of the DHNRDS DA module, some steps need to be taken:

1. Select a fast 64 bit Windows computer with adequate available storage to store a number of audio files.
    - If experimenting with different decoder modes each mode will produce a file as large or larger than the original file, so have enough disk space available.
    - The software should work on all versions of Windows from Windows 7 Service Pack 1. It is regularly being run under Windows 10.
    - 8 GB of RAM is a recommended minimum.
    - To use `da-win`, a processor that supports at least the SSE2/3 instruction set is required.
    - To use `da-avx`, a processor that supports the AVX2 instruction set is required.
    - Use the CPU-Z utility from www.cpuid.com to identify the available instruction sets.
    - The da-win version has been tested on a Silvermont ATOM and the da-avx should work well on Haswell or better (that is a 4000 series CPU and will probably work on a 3000 series). It is used regularly on a Skylake i7-6700K processor. We expect it to work on recent, high-performance AMD processors, but that has not been tested.
2. Create a folder on the system drive, typically C:
3. C:\Program Files\DHNRDS\
4. Make certain this folder is included in your PATH statement or specifically referenced in the command line.
5. Un-Zip the distribution file to that folder. There are two executables and three DLLs plus some license information (repeated below) and a DAW calibration tone.

RUNNING THE CLI VERSION

Here is an example of a working command line for preparing a file for further mastering use.

1. Open a command prompt (easiest way is to search for "`cmd`" or right click from the Windows icon on the toolbar or press the `Windows key + X`. For the last two to work, Command Prompt must be enabled (switch off) rather than PowerShell in `Settings>Personalization>Taskbar`.)
2. Change to your data drive (let's assume D:) by pressing
3. `D:<Enter>`
4. Change to the working directory for the current project:
5. `cd \Projects <Enter>`
6. `cd NR_Encoded_Project <Enter>`
7. Assemble the command line(s) in Notepad (one line for each). Here is an example:
8.
9. `"C:\Program Files\DHNRDS\da-avx" --tonel=-13.5 --toner=-13.7 --info=1 --finalize`
10. `--floatout --input="infile.wav" --output="outfile.wav"`
11. Copy the text from notepad and paste into the Command Prompt window.
12. NOTE: You may stack multiple commands this way. Edit and save in Notepad then copy and paste into the CMD window. It's like a batch file that you activate by pasting. This is much faster for Mr. Hess than editing and saving a batch file. Environment variables can also be used.
13. Note, "outfile.wav" MUST be a different name from "infile.wav."
14. If no "outfile.wav" is specified, a default suffix is added to the "infile.wav" file name.

CALIBRATING PLAYBACK TONE

This process will not change with the GUI. We will let you use your DAW of choice to measure the calibration tone.

There is often a signature "warble tone" on the tape. It is this tone that should be measured.

It is important to measure using an RMS meter calibrated as per AES 17. We are incorporating a –20 dBFS file in the decoder ZIP file. Once your meter is reading –20 dBFS on our supplied tone, then read the calibration tone on your tape.

If you measured the tones on the tape as –14.5 dBFS on the left channel and –14.3 dBFS on the right channel, then the tone portion of the command line would look like:

`--tonel=-14.5  --toner=-14.3`

FINE-TUNING THE CALIBRATIONS AND MODES

On some projects, it may make sense to run the decoder in the planned mode with the `--tone` offset by a small amount (~0.5 or 1.0 dB) above and below the noted setting to confirm the optimum `--tone` values.

Once the calibration tone is confirmed, run some samples in the three operating modes. We believe that the `--finalize` and

`--finalplus` modes offer substantial benefits in sound quality as compared to `--basic` by reducing intermodulation distortion. None of these controls are duplicates of what is available in typical DAW or audio edit/repair applications, but

rather they optimize distortion-reducing algorithms which minimize the distortions added by the fast gain control of the noise-reduction system.

Please review the other switches under Command Line Switches—Fine Adjustments, below, to see if any of these would be appropriate for the current project.

CALIBRATING PLAYBACK WITHOUT A TONE

Some variation of this procedure will be required for the tapes encoded with the earliest (Model 301) encoders, as the signature reference tone was not used on these earliest hardware encoders. The system relied on accurate alignment to the Ampex standard reference level of 185 nWb/m at 700 Hz (NAB mark on some hardware encoders' meters). In other cases, a cross-reference was made to the BASF/AGFA 320 nWb/m calibration level (the DIN marking on some hardware encoders' meters). In all cases, the decoder needs to be aligned to the 185 nWb/m calibration tone. If only the BASF/AGFA 320 nWb/m tone is available, we suggest based on work by Jay McKnight to start by setting the DHDA tone reference value to 3.8 dB below the measured tone.

Starting points for tapes without calibration tones is a complex subject. The best way of doing this is by playing a calibration tape on the playback machine to generate a starting point.

For reel-to-reel machines, there were two calibration standards. The first was that the noise reduction system was calibrated to 185 nWb/m and the second was that it was calibrated to the operating level of the tape machine. So, if you have tones, but no noise reduction calibration tone, start with the 1 kHz level lineup tone level (in dBFS, using the proper meter, confirmed with the –20 dBFS test tone provided with the decoder). If the tape was recorded at 185 nWb/m then there is no further guesswork involved. If the tape was recorded at 250 nWb/m, then that could be the NR tone level as well, or the NR tone could be 2.6 dB BELOW the tape lineup tone. 20*log(185/250)

If the tape is a German tape, recorded with the G320 reference fluxivity, then the noise reduction calibration tone will theoretically be 4.76 dB below the level calibration tone (20*log(185/320)), although this is approximate. Jay McKnight (www.mrltapes.com) has written extensively on the calibration of tape systems. As mentioned above, perhaps calibrating the noise reduction to 3.8 dB below the G320 calibration tone would be more appropriate.

If there are no tones on the current tape and no tones on related tapes in a set, then starting with an old Ampex alignment tape at reference fluxivity (185 nWb/m) would be a good start. If you suspect an elevated operating level (reference fluxivity), applying a correction factor for that becomes a decision. More than one attempt might be necessary.

Cassette machines were, in theory, better standardized with a 200 nWb/m reference fluxivity, and using a calibration tape in lieu of a tone on the current tape is often a good starting point.

Most incoming files of digitized tapes seem to have a noise reduction calibration tone in the neighbourhood of

–13 dBFS to –15 dBFS. We are encouraging transfers of noise reduction encoded tapes, especially X-I and X-II, with a noise reduction calibration tone of –20 dBFS.

Blank

DHDA REFERENCE

COMMAND LINE SWITCHES—FILES

While input and output redirection works well, using the command line switches for input and output files seems to be slightly less prone to errors that might overwrite an original file. We strongly recommend putting the filenames in double quotes as some filename-legal characters may cause confusion to the system, although this is not technically required.

`--input="infile.wav"` Specify input file name–redirection/pipe superseded

`--output="outfile.wav"` Specify output file name–redirection/pipe superseded [If not provided, the system will use the input filename and appends the following to the file name: `_DHNRDS_DA_<program-version>_DECODED`].
Decoder will not overwrite existing files, see `--overwrite`, below.

`--overwrite` The default is to NOT overwrite existing files. This switch allows creating a new output file with the same name as a file that already exists, overwriting the original. Please note that stdout always allows overwriting with no program control over that which is why we do not specifically detail the redirect and piping syntax, although it exists.

`--floatout` Force 32 bit floating point output, even if integer input. Many mastering engineers prefer to work from 32 bit files. This may solve an issue should the decoded output rise above 0 dBFS in integer file mode.

`--info=xxx` Display real-time running input/output levels and per channel gain.
`--info=1`, small amount of real-time info.
`--info=10`, also provides per channel info.
`--info=110` provides internal info at startup.

> It is always useful to have one of these options selected to monitor the processing.

COMMAND LINE SWITCHES—MODES

These three switches set the overall decoding mode. The quality increases as one moves down the list. The trade-off is processing time.

`--basic` Fastest decoding mode and the default. Generally matches the sound of hardware decoding.

`--finalize` Contains distortion reducing code to remove artifacts of the decoding process.

`--finalplus` Provides an additional level of distortion reducing code to further remove artifacts of the decoding process.

COMMAND LINE SWITCHES—CALIBRATION AND LEVELS

Use these command line switches to set the processor calibration tone level. Note, that the command line is parsed left to right and a later command may override an earlier one.
A calibration level must be set for both channels.
Mono files should use `--tone=xxx` or both `--tonel=xxx` and `--toner=xxx`.

`--tone=xxx` Set tone level in dB, both Left & Right.

`--tonel=xxx` Set tone level for Left channel. (alias `--tl`)

`--toner=xxx` Set tone level for Right channel. (alias `--tr`)

The following commands are used to trim the gains. The input gain controls will interact with the `--tone` settings, and that must be taken into account. The purpose of these are to adjust levels if there is the possibility of peak clipping. The preferred approach, since the decoder works in 32 bit floating point, is to use the `--floatout` switch and adjust the level of the output file. **These all default to 0 dB and are offsets from the normal levels.**

`--ingain=xxx` The input gain in dB, both Left & Right.

`--ingainl=xxx` The input gain in dB, Left only. (alias `--ingl`)

`--ingainr=xxx` The input gain in dB, Right only. (alias `--ingr`)

`--outgain=xxx` The output gain in dB, both Left & Right.

`--outgainl=xxx` The output gain in dB, Left only, (alias `--outl`)

`--outgainr=xxx` The output gain in dB, Right only, (alias `--outr`)

COMMAND LINE SWITCHES—FINE ADJUSTMENTS

`--filtercomp=off` Filter Compensation is normally on by default in finalize and finalplus modes. The hardware units have some subtle timing differences between frequency bands which partly accounts for their distinct sound. Setting `--filtercomp=off` provides a closer emulation of the hardware decoder's sound, while we believe the default mode provides something closer to the original signal.

`--hpf` This option does a pretty quick roll-off below 20 Hz. At 20 Hz the response is down about 1 dB, but below that it becomes steeper. The Q is pretty well controlled to avoid making the rumble worse. It is NOT the most effective rumble remover, and more meant to be a DC remover, but is there should it be needed. Wsing this switch may actually better match the rolloff of the hardware decoders, but the designers felt that leaving this switch off provides a better representation of the original recording.

COMMAND LINE SWITCHES—BROADCAST WAVE (BEXT) SUPPORT

`--bext="string"` Supports appending to added bext information.

`--bextorig="string"` If doesn’t exist, create new bext block. Use originfo.

`--bextorigref="string"` Use for origref information.

`--bextdesc="string"` Use for bext desc information.

COMMAND LINE SWITCHES—SPECIAL CONTROLS

In the case of the DA system there are at least two different conditions where special controls are needed. Some users of the early DA system pulled the LF and MF band cards when encoding tapes. That was not carried through to the later single-card systems (other than removing components on the boards). These switches can emulate that mode.

Additionally, we have seen some single card versions which have had a failure in one or more bands. These controls can adjust both the on/off status of the band and its relative thresholds. There is also a single command that can permit listening to one band at a time.

The command, without a parameter, turns the band processing off, e.g. `--lfoff` would stop the low frequency noise reduction processing. With a parameter, it adjusts the reference level for the band, e.g. with a setting of `--lfoff=-10` the `--tone` calibration for the low frequency band would be reduced by 10 dB relative to the global `--tone` setting.

The complete set of commands (usable with or without a parameter as described above. In the longer form of the commands, the initial `l` or `r` refers to left and right, respectively.

`--lfoff=`, `--llfoff=`, `--rlfoff=`	Adjusts the Low Frequency band, below 80 Hz.
`--lfoff`, `--llfoff`, `--rlfoff`	Disable LF NR (below 80 Hz).
`--mfoff=`, `--lmfoff=`, `--rmfoff=`	Set tone level offset for 80 Hz to 3 kHz.
`--mfoff`, `--lmfoff`, `--rmfoff`	Disable MF NR (80 Hz to 3 kHz).
`--hf0off=`, `--lhf0off=`, `--rhf0off=`	Set tone level offset for 3 kHz to 20+ kHz.
`--hf0off`, `--lhf0off`, `--rhf0off`	Disable HF0 NR (3 kHz to 9 kHz).
`--hf1off=`, `--lhf1off=`, `--rhf1off=`	Set tone level offset for 9 kHz to 20+ kHz.
`--hf1off, --lhf1off, --rhf1off`	Disable HF1 NR (9 kHz to 20+ kHz).
`--one=x`	Pass only one band, where x is the band number

LF=`1`, MF=`2`, HF0=`3`, HF1=`4`

COMMAND LINE SWITCHES—ADVANCED

Sox is a very useful tool that makes using the decoder much easier. It provides input/output sampling frequency flexibility (decoupling it from the processing sampling frequency used by the program.) Also, it gives excellent file type flexibility.

Using the decoder with flac for example:

Almost any flac file can be used for input, and almost any flac type can be specified for output, so here is how you do it:

```
sox infile.flac --type=wav --encoding=floating-point --bits=32 - samplingfrequency -v 96k | da-avx --tone=-13.00 --info | sox - outfile.flac
```

Notice that the sampling frequency is specified on the input sox command, and the data type is also specified. This is done to allow any acceptable flac file for input, while providing the program with the highest quality input/output & sampling frequency for its processing. (So, 96k is your best bet for quality. 192K is a close second, 48K is faster and quite acceptable, but 44.1k will sometimes sound ‘crunchy’.)

```
sox infile.flac --type=wav --encoding=floating-point --bits=32 - samplingfrequency -v 96k | da-avx --tone=-13.00 --mfoff --info | sox - outfile.flac
```

Normal ‘flac’ to ‘flac’ use, but the MF channel will have no gain control.

DHDA GENERAL INFORMATION

TIMING

All testing on the current versions of the decoder show the output files to match the timing of the input files within +/- 188 µs,. The timing should be very close to sample accurate using the same mode and the same sampling frequency on the same computer. To be safe, it is preferred to decode stereo pairs as a stereo file. We do not believe that this is an issue for multi-tracks that are encoded on all the tracks. We suggest that if some tracks are encoded and some not, that you confirm the timing checks between the files that have been decoded and the files that did not need decoding. Remember, the differences we're discussing are on the same order as moving a microphone 0.063 m (2.6 in).

HIGHER-THAN-EXPECTED NOISE LEVELS

Some older encoded recordings may have a higher than expected background noise level. One example of this might be an encoded safety master of an original unencoded master tape. In this case, there may be a perception of a slight increase in the noise level with the DHDA decoder as opposed to the hardware decoder. The hardware decoder's perceived improvement in noise performance actually comes with a reduction in bandwidth for low level signals. We suggest avoiding an attempt to further match the DHDA decoder to the HW decoder. We suggest that running a pass of gentle noise reduction on the decoded recording in a product such as those made by iZotope and Cedar is a better and more controlled solution. The DHDA is not optimized for single-ended noise reduction, but only to decode the original encoded signal, and thereby reducing the noise that would have been added by the tape which was encoded. The other products specialize in single-ended noise reduction.

TECHNOLOGY

The decoder is "feed-forward," but implemented so that the effects of unfolding the feedback compressor are accurate. This decoder is actually implemented as two separate devices running in parallel – one for the left channel and one for the right.

The general set of time constants is quite dynamic because of the unfolding from feedback to feed-forward, so the extra time domain 'boost' caused by the feedback scheme doesn't normally happen in a feed-forward scheme. This is all compensated for quite accurately, so the dynamics in the decoding should be nearly perfect. Also, the time constants for the two HF channels are different than the LF and MF channels. BUT, very importantly, the attack/decay times are NOT really time constants, but are dynamically calculated. Likewise certain characteristics of the input band filters are also dynamically calculated. (For example, the high pass filter for the 9 kHz to 20+ kHz band isn't a simple `HPF=s^2/(s^2+s*wc/q+wc^2)` type equation, nor are the other bands.) The equation shape is actually a little different, and some of the variables actually change from sample to sample. The decoder is a VERY dynamic piece of DSP software, currently running around 10,000 lines of code.

OPERATIONAL BENCHMARKS

Running times for version 6.5 of the DHDA decoder will vary between about one quarter real time real time for "basic," about three quarters real time for "finalize," and just a bit faster than real time for "finalplus" running on a 4 GHz Intel Core i7-6700K with 24 GB of RAM under Windows 10 (1809).

BATCH COMMAND REFERENCE

One can write a batch file, but it is far easier to simply keep the same commands in a text file mybatch.txt, edit it, copy all of it and paste it in on the command line.

On later versions of Windows (e.g. Windows 10), PowerShell is the default option (rather than command prompt) in the right-click WindowsLogo and also for the WindowsLogo+X key combination. This can be changed in Settings>Personalization>Taskbar. Using Search to find Cmd or entering it in the run box always works. Please note that "rem" starts a comment line. You can paste the entire blocks into the command prompt. The rem lines are in normal text instead of `code text` to make them more readable.

Here are a few command stack examples:

RUNNING THE SAME FILE IN THREE DIFFERENT MODES:

```
rem you can put a change directory command in here
rem cd "\MyRoot\MyName\MyDirectory\"
Set FILEBASE="MyFilenameWithoutTheExtension"
rem change the following for your file
Set DHDATONEL=-13.2
Set DHDATONER=-13.6
Set DHDARUNFILE="C:\Program Files\dhnrds\da-win.exe"
rem if file is in the same directory as the data remove rem to use:
rem Set DHDARUNFILE="da-win.exe"
rem or use
rem Set DHDARUNFILE="C:\Program Files\dhnrds\da-avx.exe"
rem or this
rem Set DHDARUNFILE="da-avx.exe"
rem select which of the two is applicable to your machine, or just put the filename without the path if it's in
rem the same folder/directory. This is defaulting to da-win which works in all cases.
Set DHDAMODE="--basic"
%DHDARUNFILE%  --tonel=%DHDATONEL% --toner=%DHDATONER% --info=110 %DHDAMODE% --input=%FILEBASE
%.wav --output=%FILEBASE%_DHDA_DECODED_%DHDAMODE%_L%DHDATONEL%_R%DHDATONER%.wav
Set DHDAMODE="--finalize"
%DHDARUNFILE%  --tonel=%DHDATONEL% --toner=%DHDATONER% --info=110 %DHDAMODE% --input=%FILEBASE
%.wav --output=%FILEBASE%_DHDA_DECODED_%DHDAMODE%_L%DHDATONEL%_R%DHDATONER%.wav
Set DHDAMODE="--finalplus"
%DHDARUNFILE%  --tonel=%DHDATONEL% --toner=%DHDATONER% --info=110 %DHDAMODE% --input=%FILEBASE
%.wav --output=%FILEBASE%_DHDA_DECODED_%DHDAMODE%_L%DHDATONEL%_R%DHDATONER%.wav
rem at this point you can repeat all the commands with another file.
*** End of Command File***
```

MODIFICATIONS

Consider separating the commands into a "run" group at the top, then a "file" group for each file, and then an "execute" group for each version of that file. In this way, typing is minimized. It works out easier in the end to include environment variables in the execution string so you can use them both in the command and the output filename.

> If you have an unused variable, set it to a single space after the equals sign.

RUNNING A GROUP OF MULTITRACK FILES

rem This conveniently places all the settings at the top of the file
rem Filename structure is **Fileprefix##Filesuffix**.wav where ## is a two-digit track number.
rem Do not include the extension. It defaults to .wav.

```
Set DHDAFileprefix=”MyfilePrefix”
Set DHDAFilesuffix=”MyfileSuffix”
Set DHDADecodedSuffix=”DHDA_071A_Decoded”
Set DHDARUNFILE=”C:\Program Files\dhnrds\da-avx.exe”
```

rem ***CHANGE THE ABOVE AS PER THE PREVIOUS EXAMPLE***

```
Set DHDAMode=”--finalize”
Set DHDATone01=-13
Set DHDATone02=-13
Set DHDATone03=-13
Set DHDATone04=-13
Set DHDATone05=-13
Set DHDATone06=-13
Set DHDATone07=-13
Set DHDATone08=-13
Set DHDATone09=-13
Set DHDATone10=-13
Set DHDATone11=-13
Set DHDATone12=-13
Set DHDATone13=-13
Set DHDATone14=-13
Set DHDATone15=-13
Set DHDATone16=-13
Set DHDATone17=-13
Set DHDATone18=-13
Set DHDATone19=-13
Set DHDATone20=-13
Set DHDATone21=-13
Set DHDATone22=-13
Set DHDATone23=-13
Set DHDATone24=-13
```

rem END OF INPUT SECTION

```
Set DHDATrack=01
Set DHDATone=%DHDATone01
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
Set DHDATrack=02
Set DHDATone=%DHDATone02
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
```

```
Set DHDATrack=03
Set DHDATone=%DHDATone03
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
Set DHDATrack=04
Set DHDATone=%DHDATone04
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
Set DHDATrack=05
Set DHDATone=%DHDATone05
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
Set DHDATrack=06
Set DHDATone=%DHDATone06
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
et DHDATrack=07
Set DHDATone=%DHDATone07
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
Set DHDATrack=08
Set DHDATone=%DHDATone08
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
Set DHDATrack=09
Set DHDATone=%DHDATone09
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
Set DHDATrack=10
Set DHDATone=%DHDATone10
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
Set DHDATrack=11
Set DHDATone=%DHDATone11
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
Set DHDATrack=12
Set DHDATone=%DHDATone12
```

```
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
Set DHDATrack=13
Set DHDATone=%DHDATone13
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
Set DHDATrack=14
Set DHDATone=%DHDATone14
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
Set DHDATrack=15
Set DHDATone=%DHDATone15
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
Set DHDATrack=16
Set DHDATone=%DHDATone16
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
Set DHDATrack=17
Set DHDATone=%DHDATone17
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
Set DHDATrack=18
Set DHDATone=%DHDATone18
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
Set DHDATrack=19
Set DHDATone=%DHDATone19
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
Set DHDATrack=20
Set DHDATone=%DHDATone20
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
Set DHDATrack=21
Set DHDATone=%DHDATone21
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
```

```
Set DHDATrack=22
Set DHDATone=%DHDATone22
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
Set DHDATrack=23
Set DHDATone=%DHDATone23
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
Set DHDATrack=24
Set DHDATone=%DHDATone24
%DHDARUNFILE% --tone=%DHDATone% --info=110 %DHDAMODE% --input=%DHDAFilePrefix%%DHDATrack%
%DHDAFileSuffix%.wav --output=%DHDAFilePrefix%%DHDATrack%%DHDAFileSuffix%_DHDADecodedSuffix_
%DHDAMode%_%DHDATone%.wav
```

LOGGING A RUN

If you wish to log this run, place some variation of this at the top of the command file:

```
echo: >> u:\decoderlog.txt
echo: >> u:\decoderlog.txt
echo _____________ %date% _____________ >> u:\decoderlog.txt
%DHDAFilePrefix%XX%DHDAFileSuffix%_DHDADecodedSuffix_%DHDAMode% >> u:\decoderlog.txt
echo: >> u:\decoderlog.txt
```

Then, place some variation of this line before each run line, but after the Set commands for each run:

```
echo Track %DHDATrack%    %time% >> u:\decoderlog.txt
```

Then at the end of the file after the last command, place this:

```
echo END  %date%   %time% >> u:\decoderlog.txt
```

ABOUT DYSON-HESS

CONTACT INFORMATION

email: contact@dhnrds.com

web: dhnrds.com

BIOGRAPHY—JOHN S. DYSON

John Dyson has a BSEE degree from Purdue University, utilizing it to its fullest, and beyond. He is an operating systems & real-time software developer, also with expertise in analog electronic circuit design and a background in digital signal processing.

His engineering background goes back to the 1970's time frame when he was a coop student at Naval Avionics Facility Indianapolis, the location where the WWII era Norden bombsight was developed and worked in the group who created/patented the scene matching (SMAC) guidance of the Tomahawk cruise missile.

After that, he developed hardware & software for a building automation company whose product was used to remotely and automatically manage the HVAC in buildings including at University of Illinois at Champaign/Urbana and a major supermarket chain in Chicago.

In the 1980's through 1990's he worked at AT&T/Bell Labs, was a significant contributing team member on satellite TV, HDTV and pre-inexpensive-GPS vehicle location projects. During the AT&T work, he was given permission to participate in his hobby FreeBSD operating systems project -- the predominant follow-on to Berkeley BSD "unix". On the FreeBSD project, he wrote, rewrote, and improved significant parts of the OS kernel, including the virtual memory subsystem, also meticulously optimizing the kernel for the Intel X86 related platforms. As a result of the successful FreeBSD project, he was offered and accepted participation in the late 1990's tech boom, working on a network computing project for a Larry Ellison owned company, NCI/Liberate Technologies.

Through the 2000s, he had participated in the design/development of the DIRECTV set top boxes at Thomson/Technicolor Corp.

His most recent personally motivated projects, audio noise reduction and processing, resulted from his long term interest in high quality audio recordings being available to the masses. John's success is enabled by learning and adapting to each new situation, accumulating new EE, SW & DSP engineering skills as needed.

John is a cooperative development team member, and embraces working with other creative, positive, and well intentioned individuals.

BIOGRAPHY—RICHARD L. HESS

Richard Hess bought his first tape recorder in 1963. He received his BS in Communications from St. John's University in 1973. During that time, Richard was involved with quadraphonic sound and was awarded US Patent 3784746.

In 1974, Richard joined ABC Television in New York City as a systems design engineer. He implemented ABC's first audio sweetening operation (for *Wide World of Sports*). He was involved in creating a new master control for all the audio feeds entering and exiting the plant. His final major project was designing the production audio and intercom facilities for the new home of WABC-TV at 7 Lincoln Square.

From 1973-1982 he also made on-location recordings at many locations, most notably St. Thomas Church Fifth Avenue including six LP albums (2.5 were re-released on Priory CDs).

In 1981 he left New York for Toronto to join McCurdy Radio Industries, and became Director of Engineering. He worked to improve the noise performance of McCurdy's audio consoles.

Upon the sale of McCurdy in 1983, Richard joined National TeleConsultants in Glendale, California, where he worked until 2004. He held many titles including Vice President and Principal Consultant. Major responsibilities included: Redesign of 23 radio studios for Voice of America in Washington, DC; new station facilities for KPBS radio and TV in San Diego; Audio and Control systems design for Voice of America Shortwave Broadcast Stations in Morocco and Thailand; the first HDTV release control room for DIRECTV; new station facilities for KTTV-TV in Los Angeles; and work on portions of the ESPN Digital Center in Bristol, Connecticut.

In 2004, he moved back to the Toronto area and began working full-time at his own audio tape restoration business, which continues today. Projects have included Andrew Morton's recordings of Princess Diana for NBC; the audio recording archives of Marie-Lynn Hammond (four CDs), Stingband (2 CDs), and Nancy White (1 CD); the album masters and other tapes for Manowar; Stan Rogers's album masters (all re-released on CD); the entire Huggett Family archive (including two albums produced by George Martin); and seismic instrumentation recordings of the 1980 Mt. St. Helens eruption for the U.S. Geological Survey.

In 2008, he published "Tape Degradation Factors and Challenges in Predicting Tape Life" in the ARSC *Journal*.
http://www.richardhess.com/tape/history/HESS_Tape_Degradation_ARSC_Journal_39-2.pdf

He presented on tape recorder play head azimuth at the ARSC 50th Conference in Bloomington, Indiana, in 2016 and at the AES Archiving Conference at the Library of Congress in Culpeper, Virginia, in 2018. He is presenting on Noise Reduction Systems at the AES Convention in New York City in October, 2018.

Richard is a life member of the AES, past vice-president of the AES LA Section, and a member of ARSC.

GENERAL INFORMATION

SUPPORT FOR OTHER COMPUTER PLATFORMS (UPDATED)

There are command-line versions of the software available upon request for Linux. Mr. Dyson is developing in a Linux environment. We are investigating how to leverage that into an Apple Macintosh environment.

More news later on this front. We have heard your requests for the Macintosh option.

COPYRIGHT AND LEGAL

APPLICATION SOFTWARE COPYRIGHT

LGPL LICENSE (FOR PORTIONS OF THE CYGWIN DISTRIBUTION)

GNU LESSER GENERAL PUBLIC LICENSE

incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the \ Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but maydiffer in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

<END OF LGPL>

VECMATHLIB-LICENSE

Copyright (c) 2012, 2013 Erik Schnetter <eschnetter@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

<END OF VECMATHLIB-LICENSE>

End of Document