

Usage guide for V1.6.8C DA/FA decoder

(Preliminary before complete docs are prepared.)

Important general features:

- * The FA decoder actually DECODES material, and is not an add-on type single ended expander like 3BX, 5BX
- * Better anti-hiss than various 'contraptions', because this decoder mitigates hiss CORRECTLY based on the encoding in the material.
- * Profound reduction in modulation effects from excessive compression on many consumer digital *and analog* recordings
- * Improved stereo image – normalizing to a more realistic ambience.
- * Improved audio processing, updated level detection, better attack/release calculation

New features:

- * Most important: NO features have been removed, but you won't need most of them
- * Semi-automatic decoding, decoding EQ has internal automation to help avoid usage mistakes
- * Semi-automatic inclusion of EQ for multiple sequences of layers
- * Much easier operations for multiple sequences (layers > 4)
- * Numerous frequency additions to post-decoding EQ, for help, type --pi=help
- * Built-in post-decoding EQ modes, including: --pivocal and --pihfcomp
- * Re-emphasizing the stereo image designations: --fw=pop, --fw=wpop --fw=npop and --fw=classical
- * Optimized maximum quality selection. Using --fz=opt or --fx=opt will provide a better quality balance while decode more quickly than --fz=max or --fx=max
- * The pass-through --equalizer (--skip) modes are slightly faster.

Possibly unexpected changes:

- * When using advanced modes, using the '--tone=' switch is now important.

This is a 'make-it-work' document instead of a complete list of commands, even though many ARE listed later in this document.

Decoding operations are now much simpler, there is automatic sequencing, EQ insertion, etc that is based on usage experience. This testing & usage has allowed us to narrow down the decoding operations, making 'decoding' almost trivial in comparison, while post-decoding EQ is still somewhat challenging.

ALL previous commands still function, in all of their 'manual operation' glory. This version is intended to minimize the manual operations, esp for full decoding.

Here are the *primary* variables for 'decoding':

- 1) Calibration level: almost always -44.5, -46, -50, -49 (or 10dB lower, e.g. -54.5, etc)
- 2) Decoder EQ: fcf, fce, (sometimes fcc, fcd)
- 3) Number of 'layers': 1,2,3,4,5,6,8,10
- 4) Stereo image: pop, npop, wpop or classical.

In addition to 'decoding', material often seriously needs post-decoding EQ. Post decoding EQ and 'decoding' can be done in two or more separate passes using the decoder. Some post-decoding EQ itself can be done in multiple steps, for tweaks, etc. Some material (e.g. classical recordings) sometimes don't need post decoding EQ. Very often, pop recordings DO need post decoding EQ. There are some commonly used EQ patterns, plus a few built-in patterns that are complex, but too common to avoid creating the built-ins.

There is 'post processing' available also, including anti-sibilance (very needed for some vocals), and three instances of three kinds of compressors. (One band,

two band and three band compressors). There are multiple compressor instances so that simple compression curves can be crafted.

USAGE

The general command syntax is *exactly* the same as before, but there are new, simplifying features. Most of the older features & FA initiators are not needed, and some features will be deprecated as the documentation is updated. Some of the older features (the --pe switches, --pi switches and the G,g modifiers) still have significant value. A few of the legacy initiators are still potentially helpful.

Most important – it is probably best to ignore every “trick” that you previously learned. The reason is that the ‘tricks’ are now *accurately* built-in. There are ‘permanent’ parts of decoding, where the results cannot be undone. The actual ‘decoding’ operation itself is an irreversable change to the recording. The important permanent decoding steps are choosing ‘calibration level’, ‘decoding EQ’, ‘stereo image’. Decoding quality selection is also permanent, but an important *secondary* decoding step. When learning to use the decoder, the ‘decoding quality’ matters should probably be ignored. Using the advanced decoding quality modes will most likely test your patience while waiting. I usually wait to do a decode with advanced quality AFTER I have tested a decode and gotten good initial results.

Once the decoding parameters are correct, you can use the decoder to produce a ‘rough’ result. The ‘rough’ result will likely need post-decoding EQ which UNDOES the EQ done before the original encoding. Also, some vocals REALLY NEED anti-sibilance control. There are options that enable the anti-sibilance. You’ll know when the anti-sibilance is needed, and there are various parameters that adjust the amount of ‘fixed’ and ‘dynamic’ anti-sibilance.

If you do the ‘decoding’ without the post-decoding EQ, the result will most likely be shrill, or the vocals might be buried in the mud. The post-decoding EQ can fully correct the situation. When doing initial tests, I suggest using a good first guess for the required post-decoding-EQ, then if needed, the post-decoding-EQ can be corrected later.

The general decoding command looks like this:

da-avx -input=<infile.wav> --overwrite -output=<outfile.wav> <rest-of-command>

<rest-of-command> is below:

--tone=<calibration-level> --info=<info-level> --fcs="<nlayers>,auto,decoding-EQ" --fw=<stereo-image> <post-decoding-EQ>

For <post-decoding-EQ>, you’ll most likely want to use one of the equalization macros, like ‘--pi=hf1 or --pi=hf5’, but you can also take ‘full control’ as in the following ‘details’ note:

the **details** of <post-decoding-EQ> usually looks something like this:

--pix6k=n0 -pix9k=n1 -pix12k=n2

or

--pix6k=n0 -pix7p5k=n1 -pix9k=n3 -pix10p5k=n4 -pix12k=n5

(the ‘n’ values are usually negative, and are normally as low as -3, sometimes you might need as high as +1)

Any EQ done against the recording must be undone by the FA decoder, or the results will be less optimal. ‘Decoding’ has been made much easier than before,

but the difficult part is '<post-decoding-EQ>'

The long command lines in the meta-examples might seem to be overwhelming, but I will give some real-world examples (actually tested):

For the Best of Carly Simon:

```
Decoding parameters:  --tone=-50 --fcs="6,auto,fce" --fw=mix
MF Post Decoding EQ:  --pi1k=1
HF Post Decoding EQ:  --pix6k=-2 --pix9k=-1 --pix12k=-2 (--pi=hfm2)
Post Decoding EQ modifier:  --pivocal (--pi=vocal or --pi=vocal1)
```

Let It Be (2009 Remasters):

```
Decoding parameters:  --tone=-49 --fcs="8,auto,fce"
MF Post Decoding EQ:  --pi1k=1
HF Post Decoding EQ:  --pix6k=-1 --pix9k=-1 --pi12k=-1 (--pi=hfm1)
```

Nat King Cole Story:

```
Decoding parameters:  --tone=-50 --fcs="6,auto,fce"
MF Post Decoding EQ:  --pi1k=1
HF Post Decoding EQ:  --pix6k=-1 --pix7p5k=-1 --pix9k=-1 --pix10p5k=-1 --pix12k=-1 (--pi=hfm3)
Post Decoding EQ modifier:  --pivocal (--pi=vocal or --pi=vocal1)
```

The above are examples only, but show the diversity of post-decoding EQ. The above are three very common kinds of decodes. By the time that this version of the decoder is distributed, each of the above 'HF Post Decoding EQ' will have the abbreviations below. The EQs shown below are by far the most common (if needed.) (If you use the --info=3 mode or higher, the decoder will give a complete dump of the current equivalent EQ parameters.)

(Basic HF post-decoding EQ, in order of decreasing HF)

```
--pi=hfm0:  --pix6k=-1 --pix12k=-1
--pi=hfm1:  --pix6k=-1 --pix9k=-1 --pix12k=-1
--pi=hfm2:  --pix6k=-2 --pix12k=-2
--pi=hfm3:  --pix6k=-2 --pix9k=-1 --pix12k=-2
--pi=hfm4:  --pix6k=-1 --pix7p5k=-1 --pix9k=-1 --pix10p5k=-1 --pix12k=-1
--pi=hfm5:  --pix6k=-2 --pix9k=-2 --pix12k=-2
--pi=hfm6:  --pix6k=-1 --pix7p5k=-1 --pix9k=-1 --pix10p5k=-1 --pix12k=-1 --pix13p5k=-1
```

Available modifiers: if you use '-' after the '=', for example --pi=hfm0, then a --pix4p5k=-1 filter is added. For example, for --pi=hfm1 is used, then the set of EQ will be: '--pix4p5k=-1 --pix6k=-1 --pix9k=-1 --pix12k=-1'.

(HF compensation for losses, in increasing boost)

```
--pi=hfc0:  --pix18k=3 --pif18k=-1
--pi=hfc1:  --pix15k=3 --pif15k=-2 --pix18k=3 --pif18k=-1
--pi=hfc2:  --pix15k=3 --pif15k=-1 --pix18k=3 --pif18k=-1
```

(Vocal EQ, in order of increasing HF):

```
--pi=vocal0  --pix3k=1 --pix6k=-1
--pi=vocal1  --pix3k=1 --pix9k=-1
--pi=vocal2  --pix3k=1 --pix4p5k=1 --pix6k=-1 --pix7p5k=-1
--pi=vocal3  --pix3k=1 --pix4p5k=1 --pix7p5k=-1 --pix9k=-1
```

--pi=vocal4 --pix3k=1 --pix4p5k=1 --pix9k=-1 --pix10p5k=-1

--pi=vocal is the same as --pi=vocal3

Available modifiers: if you use '+' after the '=' (e.g. --pi+=vocal0), then the first frequency is changed to --pix2p75k=1 instead of --pix3k=1. If the modifier is '**' then the first frequency is changed to --pix2p5k=1 instead of --pix2p75k=1 for '+', or --pix3k=1 for no modifier.

So, the command lines for the above examples (excluding the filenames and program names):

For the Best of Carly Simon:

da-avx --input=<in.wav> --output=<out.wav> --tone=-50 --fcs="6,auto,fce" --fw=mix --pi1k=1 --pi=hfm2 --pi=vocal

Let It Be (2009 Remasters):

da-avx --input=<in.wav> --output=<out.wav> --tone=-49 --fcs="8,auto,fcf" --pi1k=1 --pi=hfm2

Nat King Cole Story:

da-avx --input=<in.wav> --output=<out.wav> --tone=-50 --fcs="6,auto,fcf" --pi1k=1 --pi=hfm3 --pi=vocal

Lonfon Symphony, 50 Greatest Pieces of Classical Music:

da-avx --input=<in.wav> --output=<out.wav> --tone=-54.5 --fcs="8,auto,fce" --fw=classical

(In the case of 'classical' mode only, you can use --fcs="8,auto,fce=classical" instead of --fw=classical. The other image modes like 'wpop' must be specified in the '--fw' switch.)

Notes:

Getting a proper 'sound' doesn't appear to be very critical. You can choose several of the --pi options and get good results. I have found that '--pi=vocal' is very important when needed.

You CAN use the longer forms of the commands, but I suggest a small guide of the '--pi' options sitting next to you when listening. Even if you don't use the built-in options, they can be a good guideline.

The lower MF --pi1k=1 is almost always needed, and sometimes --pi750=1 is also needed (both.) If neither is needed, then I suggest reviewing your parameters very carefully. Since the FA encoding creates a nasty lower midrange boost, the bass cut that compensates on the encoding side needs an equivalent slight bass boost.

Fixing impairments:

The most irritating impairment for me is what I call 'slimey sound'. This happens when the upper midrange is too strong. 'Slimey sound' can happen because the calibration level is too low, and also can happen because of the decoding EQ allowing too many highs through. Paradoxically, 'Slimey sound' can also happen if the lows are attenuated too much on decoding. For example, using 'fcc' instead of 'fcf' or needing 'fce', but still using 'fcf' will sometimes also cause 'Slimey sound'.

A fixable audible problem for classical (and some pop music) is that the strings just do not sound correct. This can happen because fcf is used instead of fce, when fce is needed. Also, the strings can sound wrong if the calibration is too high or too low.

Comparing with the old decoding EQ:

--fce is similar to $-fcd=G$
--fcf is similar to $-fcd=Gg$
--fcc is similar to $-fcd=g$

--fcx appears never to be useful for pure decoding, but might help to compensate in certain situations.

Understanding the new FA initiator scheme

The new initiator scheme is very simple and logical. The --fcf, --fce initiators are the most often used. These initiators are now embedded into the --fcs 'macro' command, but **without the '--' switch indicator**.

There is a pattern for the sound of each initiator. Most of the time, you'll have to choose the initiator by listening for the correct sound. Here are the hints – each progressive initiator allows LESS highs through. So, the first one will allow the most highs through, that is, do less HF shelving cut. **The most common initiators are in bold.** The other initiators are best ignored unless desperate.

Most to least highs:

--fa	FAMF	(fa on --fcs)
--fa=G	FAMF + 9kHz	(fa=G on --fcs)
--fcd	FAMF + 3kHz/9kHz	(fcd on --fcs)
--fce	FAMF + 3kHz/18kHz	(fce on --fcs)
--fce=G	FAMF + 3kHz/18kHz + 9kHz/18kHz	(fce=G on --fcs)
--fce=g	FAMF + 3kHz/18kHz + 9kHz/inf	(fce=g on --fcs)
--fcf	FAMF + 3kHz/21kHz	(fcf on --fcs)
--fcd=G	FAMF + 3kHz/9kHz + 9kHz/inf	(fcd=G on --fcs)
--fcf=G	FAMF + 3kHz/21kHz + 9kHz/21kHz	(fcf=G on --fcs)
--fcf=g	FAMF + 3kHz/21kHz + 9kHz/inf	(fcf=g on --fcs)
--fcc	FAMF + 3kHz/inf	(fcc on --fcs)
--fcc=G	FAMF + 3kHz/inf + 9kHz/inf	(fcc=G on --fcs)
--fcc=g	FAMF + 3kHz/inf + 9kHz/inf	(fcc=g on --fcs)
--fcx	FAMF + 3.18kHz/10.5kHz	(fcx on --fcs)
--fcx=G	FAMF + 3.18kHz/18kHz	(fcx=G on --fcs)
--fcx=G+	FAMF + 3.18kHz/18kHz & 12kHz+3dB	(fcx=G on --fcs)
--fcx=G*	FAMF + 3.18kHz/18kHz & 15kHz+3dB	(fcx=G on --fcs)
--fcx=gG	FAMF + 3.18kHz/21.5kHz	(fcx=Gg on --fcs)
--fcx=g	FAMF + 3.18kHz/inf	(fcx=g on --fcs)
--fcx=g+	FAMF + 3.18kHz/inf & 12kHz+3dB	(fcx=g on --fcs)
--fcx=g*	FAMF + 3.18kHz/inf & 15kHz+3dB	(fcx=g on --fcs)
--cdd	additional 3.18kHz/10.5kHz	(external to --fcs, used with --fcx)

These should be the only FA initiators inside of --fcs command:

fcd	FAMF + 3kHz/9kHz	(fcd on --fcs)
fce	FAMF + 3kHz/18kHz	(fce on --fcs)
fcf	FAMF + 3kHz/21kHz	(fcf on --fcs)
fcc	FAMF + 3kHz/inf	(fcc on --fcs)

I have only tested 20-30 CDs, but my results show that fcf and fce and once in a while, fcc, are the only needed initiators.

Notes about the 'G/Gg/g' modifiers

(Auxiliary info — info not needed for initial learning, this is background info)

The 'G' style modifiers basically set the high frequency limit for the 1st order LPF when used with the various 'fcX' switches. There are exceptions to the rules below, but generally the rules are:

G	18kHz rolloff limit
gG (or Gg)	21.5kHz rolloff limit
g	no limit (keeps rolling off)
+	Add 12kHz/+3dB
*	Add 15kHz/+3dB

The '+' modifier is needed very often after 'G' or 'g'.

The '*' modifier is available in case it is needed.

Near equivalents:

3.18kHz (avoid)	3kHz in single filter	3kHz in mixed filters (avoid)
fcx	fcd (3kHz to 9kHz)	fcd
fcx=G	fce (3kHz to 18kHz)	fed=G
fcx=Gg	fcf (3kHz to 21.5kHz)	fed=Gg
fcx=g	fcc (3kHz to 60kHz)	fed=g

The single filter and mixed filter versions are slightly different, best to use single filter version of the 3kHz base EQ.

The 3.18kHz is a single filter implementation.

The only EQ that appears to be correct is: fcf, fce, fcc, fcd. Fcf and fce are most correct.

Simple, Generic Decoding Commands

These examples are reasonable 'starter' multi-layer decoding commands. These generally gives improved results on perhaps 2/3s (or even more) of FA material.

There are two general command-types that could be deemed 'generic' decode commands. The major variant beyond the two commands is the choice of -33+N or -39+N for the tone value:

(Generic command #1)

da-avx --input=infile.wav --overwrite --output=outfile.wav --info=1 --tone=-44.5 --fcs="N, auto, fce" --pi=hfm2

(Generic command #2)

da-avx --input=infile.wav --overwrite --output=outfile.wav --info=1 --tone=-46.0 --fcs="N, auto, fcf" --pi=hfm3 --pi=vocal

(Generic command #3)

da-avx --input=infile.wav --overwrite --output=outfile.wav --info=1 --tone=-49.0 --fcs="N, auto, fcf" --pi=hfm4

(Generic command #4)

da-avx --input=infile.wav --overwrite --output=outfile.wav --info=1 --tone=-54.5 --fcs="N, auto, fce" --fw=classical

(In the above commands, N – number of layers, can be 1,2,3,4,5,6,8,10... 4,5,6,8 are most common. If just learning, try N='4' first.)

Your best bet at a first guess calibration is -44.5 or -46, on newer recordings: -49 or -50. Classical recordings can require --tone=-54.5.

Use the --fw switch now for specifying the 'classical' stereo image. 'Pop' is the default, but there is also 'wpop' (wide pop), 'npop' (narrow pop) and 'mix' for something in between classical and pop for the stereo image.

Helpful Suggestion

Assume from the start that

`-tone=-46.0 -fcs="4,auto,fcf" --pix6k=-1 -pix9k=-1 -pix12k=-1` as a first guess.

Or when using the `-pi` macro

`-tone=-46.0 -fcs="4,auto,fcf" --pi=hfm1`

If the correct decoding parameters have been used, more often than not, post decoding EQ switch of '`--pi1k=1`' (first order 3dB bass boost at 1kHz) creates a near-perfect bass balance. Some recordings are also helped by an additional `-pi750=1`. Do NOT overuse the 1st order EQ. It is best to avoid doing too much first order EQ!!!

When trying to find the correct sound, I have found that changing the decoding EQ should be done with care (only choose fcf or fce, then fcc if you really need.) *Using 'auto' mode can save lots of frustration*, as it does special EQ at the beginning of each sequence of layers. There is a way of attaining more control by using the `-bxxk` switches directly. In 'auto' mode, it might be helpful to add the `-b15k/-b18k` switch in certain limited cases, and might be a lifesaver.

My own steps for finding the correct settings are generally:

- 1) The super-highs are mostly controlled by the decoding <FA-initiator> (or decoding EQ, same thing). So, if the super-highs don't sound correct, then try fce or fcc. (fce will make the sound more bright, fcc will dull the sound.)
- 2) The midrange will sound wrong if the calibration (the `-tone=`) is wrong. If the upper midrange is a bit too strident, try increasing the calibration. If there is 'gating' or the midrange is too dull, then try decreasing the calibration.
- 3) Once the sound is **generally** correct, then you can tweak the 'hfm' argument of the `-pi` switch. If the sound is too bright, then increase the `-pi=hfm1`, one step at a time... If the sound is too dull, then try `-pi=hfm0`. If that still doesn't sound right, then you might have to use a lower calibration or use fce instead of fcf.

The above steps are iterative. It can be very helpful to run the decoding tests in real-time, perhaps using the `-splay` or `-play` switches as appropriate for your system. (`--splay` uses SoX, `--play` uses an ad-hoc player program 'aoplay' as distributed with the decoder.)

If you are in doubt about the post-decoding EQ, then just do the EQ, produce a .wav file. Next, do multiple post-decoding EQ passes with the `-` equalizer switch, which disables decoding but still supports EQ, anti-sibilance and compression as desired.

About 'Higher Quality' modes

About higher quality modes. More than likely, they aren't needed. Most recordings have already been damaged by NR encoding/decoding, and the normal higher quality modes only help a little. On pristine material, the improvement is tremendous. Simply add the '--fx', or '--fz' switches for slightly improved quality at the cost of a LOT of CPU. The amount of CPU usage is dependent on the number of layers, and can be very very slow when using especially '--fz'. Frankly, I am not patient enough to use '--fz' unless for testing purposes.

Here are the 'higher quality' modes in order of increasing quality:

--fx, --fx=plus, --fx=highs, --fx=max

--fz, --fz=plus, --fz=highs, --fz=max

There is a --scrub switch that enables stronger scrubbing of MD.

There is also a --dh=off switch, which turns on the Hilbert modulation removal, and can help produce amazing results, but is VERY VERY SLOW.

Absolutely the best quality is obtained by the --fz=max mode, which internally enables both --scrub and --dh=off. By the time you get this usage guide, --scrub will automatically force --dh=off. I believe that separating --scrub and --dh=off was a mistake.

There are even higher modes, DO NOT TRY TO USE THEM, unless you have more than 20 cores. I suggest using "--fx", "--fz" or "--fx --scrub" or "--fz --scrub", in this order of decreasing speed. If you are intrepid, only running 6 layers or less, and have a very fast computer, then you might try '--xpp=max' or just '--xpp' alone.

Switches that you'll likely need

For this document, other than the simple examples above, we need to depend on the older, broken documentation. That IS bad, and I am planning an update soon. However, in the interim, I am listing in a simple form, the switches that you'll need. This is admittedly VERY primitive – I haven't even updated the man page yet, because I haven't chosen the commands to add and remove.

These are the switches that I normally use, and I have been working on the most difficult material for testing reasons!!!

Input/Output switches:

- input=<infile.wav>**
Specifies audio file to read/decode.
- overwrite --output=<outfile.wav>**
Switch combination that writes the output file.
- info=1 or --info=2 (--info=3, info=11, etc supplies some more useful details.)**
Gives real-time display of program activity. I STRONGLY suggest using one of these switches.
At --info=3, the EQ used at each step, and the calibrations for each step are displayed.
- outgain=<xxdB>**
This is a general purpose facility to support additional/less output gain. Since the output level can be higher than the input level, it might be necessary to use less than 0dB gain on output. This switch gives that capability.
- ingain=<xxdB>**
This is a general purpose facility to support additional/less input gain. Be very careful with this switch – it is important to maintain 0dB gain from digital source material, or the calibration numbers will be wrong.
- fgain=<xxdB>**
Final gain after all compressors/limiters/etc. Adds to --outgain when a compressor/limiter is not used.
- floatout**
For greater dynamic range, the decoder can produce floating point .wav files. Some consumer programs choke on FP files though.
- intout**
Force creating 24bit integer output files
- chsw**
Switch output left and right channels.

Mode/Tuning switches:

- fcf, --fcc, --fce, --fcf=G/g/gG, --fce=G/g/gG, --fce=G (deprecated)**
(Not needed anymore.) These are the EQ modes, that describe the kind of FA EQ curve being used on the current decoding layer. The modifier 'classical' is also needed for many classical recordings. To specify 'classical' and 'G', then use something like '**--fcf=classical,G**'.
- fcs="<nlevels>,<toneval>,<FA init>"**
- fcs="<nlevels>,auto,<FA init>"**
This is now the preferred initiator method for simple FA decoding operations.
There are several built-in automatic capabilities/helpers which are enabled when replacing 'tone' with the literal word 'auto'. Since there is no longer a place in the --fcs command to specify the tone level, then make sure that you add --tone=<toneval> before the --fcs command. I moved the calibration level back into the '--tone=' switch because it makes more sense for the 'auto' mode. Setting the calibration (toneval) inside of the

–fcs command will forever be valid and still useful. Sometimes the ‘auto’ mode is not best.

A good first choice for ‘nlevels’ is 4, perhaps 6. The typically used <FA init> are fcf, fce, fcc. I haven’t run into fcd yet, but probably is needed somewhere. “auto” mode automatically invokes certain usage of the –b switches, which would otherwise using the ‘–b9k’ series can be tedious and error prone.

--tone=<calibration level>

This specifies the calibration level used, and is almost standardized. Typical ‘tone’ or ‘calibration’ levels are (in order of most to least likely):

–X4.5, –X6.0, –X0, –X9, –X4*, –X5*, where X is 4, 5, 6, or 3. So, the first calibration to try would be **–44.5**, then **–46.0**, then **–50**, then **–49**. --tone is most often used for DA decoding or for manual FA decoding. **The calibration used to be sometimes set in the –fcs command, and can still be done that way. However, the place for the calibration number in the –fcs command is now the designator for the ‘auto’ mode, and the using the –tone= switch is back in vogue.**

--equalizer (--skip)

This supports running the post-decoding EQ (e.g. --pe, --pi), compression (--c1, --c2, --c3) and anti-sibilance (--as), and/or stereo image width manipulation (--wof) without doing a decoding operation. This is very useful so that the time consuming decoding might be done once, since it is easier now, then do the more tricky post decoding EQ in quicker iterations.

Pre-decoding EQ switches: (pEQ)

Using these switches are normally an integral part of FA decoding, and the –b9k version is usually the minimum necessary. The –b18k version is especially useful for compensating the loss of HF from legacy HW. Normally, using the auto mode will mitigate most of the usage of these switches, but –b15k and –b18k are still useful for bandwidth compensation. This form of EQ is NOT operational in ‘--equalizer’ mode.

--b3k6k, --b3k9k, --b3k18k, --b3k22k, --b3kxxk

--b9k18k, --b9k22k, --b9kxxk

--b12k18k, --b12k22k, --b12kxxk

--b15k18k, --b15k22k, --b15kxxk

--b18k22k, --b18kxxk

These 1st order shelving boost EQ are placed immediately before each layer where you specify them. If in auto mode, then will automatically be placed at the beginning of each sequence of layers. The normal usage is for beginning of each set of layers like this for 6 layers:

<pEQ> <calibration -44.5> <calibration -34.5> <calibration -24.5> <calibration -14.5>

<pEQ> <calibration -54.5> <calibration -44.5>

(Note: when you see something like –b9kxxk, that means that the 1st order shelving boost goes from 9k to near-inf, which happens to be 60k)

--b3k, --b9k, --b12k, --b15k, --b18k

These are the same as the fully specified <pEQ> as above, but the decoder automatically selects the best version of each <pEQ>. The choice is based upon the type of <FA initiator>. For fce, the correct version is <--bYYk18k>, for fcf, the correct version is <--bYYk22k>, then for fcc, the correct version is <--bYYkxxk>.

--cde

Does a CD de-emphasis before the decoder.

--comment or **--comment="msg"**

Normally the decoder will not overwrite the 'ICMT' list item if it already exists. Adding the '--comment' switch with no arguments will cause the decoder to overwrite the command line information into the 'ICMT' list item, even if the item already exists in the source .wav file. Using an argument supports a manually specified comment into the 'ICMT' list item. If the 'ICMT' already exists, and the '--comment' switch was not used, then the decoder will create a nonstandard 'FADC' list item.

--next

~~(Never used with the --fcs switch.)~~ This opens up a new decoding layer, just like doing another decode in sequence. The decoding modes default from the previous decoding modes, except the new '~~--tone=~~' value is incremented by 10dB. Normally, you should start with the lowest decoding calibration '~~--tone=~~' value first, and then let the program auto-increment the values to either 14.5 or 13. After that, then the program automatically resets the increments back to the previous starting value for '~~--tone=~~' and cycles through again. It is probably best to specify all of the parameters unless you understand the defaulting and auto-increment mechanism.

--nextda

~~This opens up a true DA decoder instead of FA. No details yet, but follows the same rules, except the default calibration is always '~~--tone=~~13.55365'. This can be overridden.~~

--wide=<bw-mult>

Specifies the bandwidth of the processing. This bandwidth is the 'bw-mult' times the internal sample rate. Normally, for 44.1kHz input, the internal sample rate is 66.15kHz. So, if the bw-mult is specified to be 0.30 on a normal CD file input, then the bandwidth will be 66.15kHz * 0.30 or 19.845kHz.

--maxfreq=<bandwidth>

Specifies the bandwidth of processing in 'Hz.' There freq response can never be higher than about 42kHz. The actual maximum frequency response can never be greater than 'internal clock rate' * 0.45. This would be about 41-42kHz for 96k based frequency sources, and 38kHz for 88.2kHz based frequency sources. This is useful for dealing with noisy SACDs where there is a lot of hash above 24kHz.

Mode/Tuning switches (cont'd):

--fx, --fz

These are the anti-MD mode specifiers. The old specifiers are still available and theoretically provide higher quality at GREAT cost of CPU. I suggest that if you want to use an anti-MD mode, then '--final' gives good first order improvements, and '--fx' is probably all that you really need. '--fz' is really good, and really does ferret out distortions, but is VERY VERY CPU expensive, esp when running multiple layers.

There are options that support incremental improvement over the gross-level anti-MD. The --fx and --fz switches support these options:

--fx=plus	(splits the two HF bands, mitigates some intra-band MD creation)
--fx=highs	(additional splitting of the HF bands, further mitigating the intra-band MD creation)
--fx=max	(splits the MF band also, with further MD mitigation.)
--fz=plus	(splits the two HF bands, mitigates some intra-band MD creation)
--fz=highs	(additional splitting of the HF bands, further mitigating the intra-band MD creation)
--fz=max	(splits the MF band also, with further MD mitigation.)
--fz=opt	(Better optimization of CPU time and quality than --fz=max)
--fx=opt	(Better optimization of CPU time and quality than --fx=max)

--scrub

This switch is used in conjunction with the --fx and --fz switches. This allows those switches to enable anti-MD mode. I

Very useful information switch:

--df

This switch means 'dump filters'. This is very important because it also shows the calibration levels (--tone= values) for each layer. It can help if loosing track of what the tone= values are...

Stereo image switches:

--fw=<stereo-width-name>

This switch is now used in-lieu of being contained in the FA initiator set of strings. The useful options are:

classical	the stereo width is not modified when decoding.
pop (default)	the stereo width is modified during decoding
wpop	the stereo width is modified during decoding, and the resulting width is made slightly more wide.
npop	the stereo width is modified during decoding, and the resulting width is made slightly more narrow
mix	the stereo image width is a hybrid between classical & pop

--wia=<lratio>

--woa=<Oratio>

--wof=<Eratio>

the --fw switch sets these settings based upon the type of stereo image. Sometimes the values must be modified manually. The only one of these that would commonly be used: --wof. I use --wof very regularly, but will probably more often use --fw=wpop and --wf=npop in the future. --wof is operative during --equalizer mode.

Output/post-decoding equalization switches

For the 2nd order 'peXX' switches:

mode = 1,2,3,4,K

mode 1 – simple filter, <optQ> IS operative

mode 2 – mixed Q=0.5/0.8409, <optQ> is not needed

mode 3 – mixed Q=0.5/0.707/0.8509, <optQ> is not needed

mode 4 – mixed Q=0.5/0.8409 spread over +-250Hz, <optQ> is not needed

mode K (5) – mixed Q=0.5/0.707/0.8409 spread over +-250Hz, <optQ> is not needed

Three (3) instances of each of the –pe equalizers can now be used. Also, these are operational in –equalizer mode.

Each of the mixed modes do sound different. Sometimes mode 2 sounds better than mode 3 and vice versa. Same as mode 4 vs. mode 5 (K). It is tricky to tell beforehand which of the 2,3,4,K filters will sound best. Usually, it is best to mix the filter types if you are going to use EQ at nearby frequencies. For example, if you need an EQ at 6kHz, 9kHz and 12kHz, it might be best to use type '4' for 6kHz, type 'K' for 9kHz and type '2' for 12kHz.

The type 4 & 5 EQ aren't available at frequencies 120Hz and below. It makes sense, because one cannot create a negative Hz filter, and calculating 120Hz – 250Hz does create a negative frequency.

Very often, during decoding, a *first order* boost at 1kHz (--pi1k) is needed, and sometimes even another boost at 750Hz (--pi750), however I found that given the EQ on most recordings, *first order* filters aren't helpful at 500Hz or below. The pe375Hz filter is very very helpful to create a good bass balance, especially the –pe375=K,0.75 type filter – can be *wonderful*. For the deepest bass, a small change at –pe45=2,0.75 (or -0.75) can be helpful.

I used to use –pe1k for the needed lower MF boost on most decodes, but I found that the 1st order –pi EQ works better there. However, it is perfectly valid to mix both 1st order and 2nd order EQ at the same frequency.

Bass EQ:

--pe225=<mode>,<dBchange>,<optQ>	22.5Hz
--pe45=<mode>,<dBchange>,<optQ>	45Hz
--pe90=<mode>,<dBchange>,<optQ>	90Hz
--pe120=<mode>,<dBchange>,<optQ>	120Hz
--pe375=<mode>,<dBchange>,<optQ>	375Hz
--pe500=<mode>,<dBchange>,<optQ>	500Hz
--pe750=<mode>,<dBchange>,<optQ>	750Hz
--pe1k=<mode>,<dBchange>,<optQ>	1kHz

Treble EQ:

--pe3k=<mode>,<dBchange>,<optQ>	3kHz
--pe4p5k=<mode>,<dBchange>,<optQ>	4.5kHz
--pe6k=<mode>,<dBchange>,<optQ>	6kHz

--pe7p5k =<mode>,<dBchange>,<optQ>	7.5kHz
--pe9k =<mode>,<dBchange>,<optQ>	9kHz
--pe10p5k =<mode>,<dBchange>,<optQ>	10.5kHz
--pe12k =<mode>,<dBchange>,<optQ>	12kHz
--pe15k =<mode>,<dBchange>,<optQ>	15kHz
--pe18k =<mode>,<dBchange>,<optQ>	18kHz

Special MF EQ: (+ or – about 1.5dB between 1k & 3k), default is +1
--pemf=1 (default), --pemf=0 (old style), --pemf=-1 (for symmetry only)

Simple 1st order pi1k (bass), pi3k (treble), pi6k (treble), pi9k (treble), pi12k switches

These filters below can be used multiple times on a command line, and will sum up to the total number (- and + cancel, just like any math operations.)
These are also all operational in –equalizer mode.

--pi500=<0 to 3>	500Hz 3dB BASS boost	(For recordings that appear 'thin' sounding)
--pi750=<0 to 3>	750Hz 3dB BASS boost	(For recordings that appear 'thin' sounding)
--pi1k=<0 to 3>	1kHz 3dB BASS boost	(Needed on most decoding operations)
--pi2k=<-6 to 6>	2kHz to 9kHz TREBLE cut or boost	
--pix2k=<-6 to 6>	2kHz to 21.5kHz TREBLE cut or boost	
--pif2k=<-6 to 6>	2kHz to 100kHz TREBLE cut or boost	
--pi2p25k=<-6 to 6>	2.25kHz to 9kHz TREBLE cut or boost	
--pix2p25k=<-6 to 6>	2.25kHz to 21.5kHz TREBLE cut or boost	
--pif2p25k=<-6 to 6>	2.25kHz to 100kHz TREBLE cut or boost	
--pi2p5k=<-6 to 6>	2.5kHz to 9kHz TREBLE cut or boost	
--pix2p5k=<-6 to 6>	2.5kHz to 21.5kHz TREBLE cut or boost	
--pif2p5k=<-6 to 3>	2.5kHz to 100kHz TREBLE cut or boost	
--pi2p75k=<-6 to 6>	2.75kHz to 9kHz TREBLE cut or boost	
--pix2p75k=<-6 to 6>	2.75kHz to 21.5kHz TREBLE cut or boost	
--pif2p75k=<-6 to 6>	2.75kHz to 100kHz TREBLE cut or boost	
--pi3k=<-6 to 6>	3kHz to 9kHz TREBLE cut or boost	
--pix3k=<-6 to 6>	3kHz to 21.5kHz TREBLE cut or boost	
--pif3k=<-6 to 6>	3kHz to 100kHz TREBLE cut or boost	
--pi3p25k=<-6 to 6>	3.25kHz to 9kHz TREBLE cut or boost	
--pix3p25k=<-6 to 6>	3.25kHz to 21.5kHz TREBLE cut or boost	
--pif3p25k=<-6 to 6>	3.25kHz to 100kHz TREBLE cut or boost	
--pi4p5k=<-6 to 6>	4.5kHz to 9kHz TREBLE cut or boost	
--pix4p5k=<-6 to 6>	4.5kHz to 21.5kHz TREBLE cut or boost	
--pif4p5k=<-6 to 6>	4.5kHz to 100kHz TREBLE cut or boost	
--pi6k=<-6 to 6>	6kHz to 9kHz TREBLE cut or boost	
--pix6k=<-6 to 6>	6kHz to 21.5kHz TREBLE cut or boost	
--pif6k=<-6 to 6>	6kHz to 100kHz TREBLE cut or boost	
--pi7p5k=<-6 to 6>	7.5kHz to 9kHz TREBLE cut or boost	
--pix7p5k=<-6 to 6>	7.5kHz to 21.5kHz TREBLE cut or boost	
--pif7p5k=<-6 to 6>	7.5kHz to 100kHz TREBLE cut or boost	
--pi9k=<-6 to 6>	9kHz to 18kHz TREBLE cut or boost	

--pix9k=< -6 to 6>	9kHz to 21.5kHz TREBLE cut or boost	
--pif9k=< -6 to 6>	9kHz to 100kHz TREBLE cut or boost	
--pi10p5k=<-6 to 6>	10.5kHz to 9kHz TREBLE cut or boost	
--pix10p5k=< -6 to 6>	10.5kHz to 21.5kHz TREBLE cut or boost	
--pif10p5k=< -6 to 6>	10.5kHz to 100kHz TREBLE cut or boost	
--pi12k=< -6 to 6>	12kHz to 18kHz TREBLE cut or boost	
--pix12k=< -6 to 6>	12kHz to 21.5kHz TREBLE cut or boost	
--pif12k=< -6 to 6>	12kHz to 100kHz TREBLE cut or boost	
--pi13p5k=<-6 to 6>	13.5kHz to 9kHz TREBLE cut or boost	
--pix13p5k=< -6 to 6>	13.5kHz to 21.5kHz TREBLE cut or boost	
--pif13p5k=< -6 to 6>	13.5kHz to 100kHz TREBLE cut or boost	
--pi15k=< -6 to 6>	15kHz to 18kHz TREBLE cut or boost	
--pix15k=< -6 to 6>	15kHz to 21.5kHz TREBLE cut or boost	
--pif15k=< -6 to 6>	15kHz to 100kHz TREBLE cut or boost	
--pi18k=< -6 to 6>	18kHz to 18kHz TREBLE cut or boost (noop)	
--pix18k=< -6 to 6>	18kHz to 21.5kHz TREBLE cut or boost	
--pif18k=< -6 to 6>	18kHz to 100kHz TREBLE cut or boost	
--pi1k=<0 to 3>	1kHz 3dB BASS boost	(Needed on most decoding operations)

VERY USEFUL POST DECODING EQ MACROS (All operational in -equalizer mode)

(Basic HF post-decoding EQ, in order of decreasing HF)

```
--pi=hfm0      --pix6k=-1 --pix12k=-1
--pi=hfm1      --pix6k=-1 --pix9k=-1 --pi12k=-1
--pi=hfm2      --pix6k=-2 --pix12k=-2
--pi=hfm3      --pix6k=-2 --pix9k=-1 --pix12k=-2
--pi=hfm4      --pix6k=-1 --pix7p5k=-1 --pix9k=-1 --pix10p5k=-1 --pix12k=-1
--pi=hfm5      --pix6k=-2 --pix9k=-2 --pix12k=-2
--pi=hfm6      --pix6k=-1 --pix7p5k=-1 --pix9k=-1 --pix10p5k=-1 --pix12k=-1 --pix13p5k=-1
```

(HF compensation for losses, in increasing boost)

```
--pi=hfc0      --pix18k=3 --pif18k=-1
--pi=hfc1      --pix15k=3 --pif15k=-2 --pix18k=3 --pif18k=-1
--pi=hfc2      --pix15k=3 --pif15k=-1 --pix18k=3 --pif18k=-1
```

(Vocal EQ, in order of increasing HF):

```
--pi=vocal0    --pix3k=1 --pix4p5k=1 --pix6k=-1 --pix7p5k=-1
--pi=vocal1    --pix3k=1 --pix4p5k=1 --pix7p5k=-1 --pix9k=-1
--pi=vocal2    --pix3k=1 --pix4p5k=1 --pix9k=-1 --pix10p5k=-1
--pi=vocal is the same as --pi=vocal1
```

New auxiliary switches (operational in –equalizer mode)

--limiter=-NN.NN (dB)

This limiter switch attempts to limit the output to the specified level. The behavior is not quite accurate (perhaps +-1.0dB) because of the simplified anti-distortion DA style detector. The default value is equivalent to using -3dB. The actual limiting will be at a dB or so higher than what you specify. The default -9dB should be **relatively** safe for 0dB clipping, but no guarantee. This function now works very well. .

--as=<level> (anti-sibilance)

This anti-sibilance mode is primitive but is insidiously subtle when used correctly. It is implemented both by parallel and cascade notch filters at different frequencies. This is a 'soft' control, and a good default value to use might be 0.50 to 10. You might have to adjust the setting, and is mostly useful for very loud sibilance, and is also very effective against 'fake sibilance' where they used an approx 6kHz peaking filter to brighten the vocals. Be careful, because the setting is not effective over a wide dB range, and also can create a weird kind of dynamic compression at high frequencies. REALLY WORKS WELL!!!!

Compressors (prelim) (operational in –equalizer mode)

--c1="<cratio>,<threshold>,<mingain>,<maxgain>,<release-time>"

One, single band

--c2="<cratio>,<threshold>,<mingain>,<maxgain>,<release-time>"

Two bands, split frequency is 275Hz

--c3="<cratio>,<threshold>,<mingain>,<maxgain>,<release-time>"

Three bands, split frequencies are 275Hz, 2.75kHz

Up to three compressors of each type can be utilized. Levels are in dB.

These are compressors based on a set of detectors, with a soft dynamic attack and a dynamic release. The '<release-time>' is a guideline, where the slowest release speed can be between 0.040 to 10 seconds.

The <cratio> is best between 1.19 to 3 for the compression ratio. These compressors are not well debugged for sound quality YET, so higher compression ratios can produce 'grain' in the sound.

The <threshold> is the level where the gain increases below the level, and the gain decreases above that level.

The <mingain> is the highest gain allowed when the level drops below the <threshold>

The <maxgain> is the lowest gain allowed when the level goes above the <threshold>