

Usage guide for V1.7.0C,V1.7.0D or V1.7.0E DA/FA decoder

(Preliminary before complete docs are prepared.)

Important general features:

- * WHEN READING THIS, REMEMBER NO MORE USER SPECIFIED POST DECODING EQ!!! This is a great simplification.
- * This is the EASIEST TO USE version ever. A lot of things are now automatically determined.
- * 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
- * **Before you get frustrated, read the 'helpful suggestion' section – it gives good pointers on best decoding choices.**
- * **Good results can be attained by using one of the limited set of 'natural choices'**

New features:

- * Most important: FEW features have been removed, but you won't need most of them
- * **New decoding modes with better MF eq, better LF eq, no need for post decoding EQ.**
- * All of the --pi*, --pix*, --pif* EQ still exists.
- * New totally variable 1st order EQ (BASS: --pvl=<freq>,<dB> TREBLE: --pvh=<freq>,<dB>)
- * 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
- * **New decoding modes – collapsed internal EQ, simplified the modes/initiators (fg, fgG).**
- * **All normal pre-decoding EQ resides in the --fcs switch.**
- * **No user specified post decoding EQ!!!**

Possibly unexpected changes: (carried forward)

- * When using advanced modes, using the '--tone=' switch is still important.
- * The --ingain switch values now add instead of replace each other
- * The --outgain switch values now add instead of replace each other

Explanation for these changes:

- * **Researched useful modes, and re-factored the set of commands to get the best quality with the greatest simplicity.**
- * Removed need for almost any user specified post-decoding EQ.
- * Minimized need for any pre-decoding EQ, and the seldom needed options are built-in to the --fcs command.
- * Improved the lower MF range.

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 even 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: (fg<a,b,c> or fg<^,^^,^^^> or fgG<a,b,c> or fgG<^,^^,^^^>
- 3) Number of 'layers': 1,2,3,4,5,6,8,10 (4, **6**, **8**, 10 most useful.)
- 4) Stereo image: pop, npop, wpop or classical.

Post decoding EQ no longer needed, but slight EQ for 'taste' is certainly valid.

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) still have some value. A few of the legacy initiators are not likely helpful.

Most important – it is probably best to ignore every “trick” that you previously learned. The reason is that MORE ‘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 irreversible 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.

Note: The new EQ sequence using the ‘fg’ initiator is very good at sweetening excessive 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>

No post decoding EQ is needed.

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.

*** Important note: most recordings need -pi1k=1, even classical.**

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,fg^^" --fw=mix --wof=0.8409 (vocal a little crunchy)
--tone=-50 -fcs="8,auto,fg^^" --fw=mix --wof=0.8409 (sounds normal)
--tone=-50 -fcs="4,auto,fg^^" --fw=mix --wof=0.8409 (vocal crunchy, but more of an FA sound.)
(in both of the above, I actually tested with the '*' submode also, which improved the highs.)
* the 'mix' mode is seldom needed!!!

Let It Be (2009 Remasters):

Decoding parameters: --tone=-54.5 -fcs="6,auto,fgG" (still a little compressed)
--tone=-54.5 -fcs="8,auto,fgG" (more emotion)

Nat King Cole Story:

Decoding parameters: --tone=-54,5 -fcs="4,auto,fg^^"

Carpenters 1970 (Close To You) (Carpenters appear to be slightly bass heavy – easy to fix with 'B' submode)

Decoding parameters: --tone=-51 -fcs="6,auto,fgB^^*"

Actual ABBA albums (not Gold):

Decoding parameters: --tone between -54.5 to -50.0 (usually about -50), usually --fcs="6,auto,fg^^"

ABBA Gold (super bright sounding):

Decoding parameters: --tone=-56 --fcs="6,auto,fgG" --pix9k=-2 (*One of the few that needs HF post decoding EQ.*)

London Symphony, 50 Greatest Pieces of Classical Music:

Decoding parameters: --tone=-56 -fcs="6,auto,fgc*" --fw=classical

Some decoding modes (initiators) are ALMOST compatible, but there are subtle optimizations available.

Notes: Getting a proper 'sound' is sometimes not very critical. You can choose several of the FA initiator/modes and get reasonable results. Other times, it can be frustrating because a recording might need a 'pattern' that you haven't used before.

Fixing impairments:

Too much bass: (use B submode)

Distorted highs, but sounds good otherwise: (use '*' submode)

Inadequate highs: add another '^', or start into the letters 'a', 'b' or 'c'.

Too much sibilance: use the 'fg' initiator instead of 'fGg'.
Cannot decode, but sounds FA: try 'H' in desperation.

Understanding the new FA initiator scheme

The new initiator scheme is very simple and logical. The ONLY initiators that are useful: fgG and fg. These initiators are now embedded into the –fcs ‘macro’ command.

The **fgG** (or **fGg**) initiator default high frequency cutoff for the EQ is at 21kHz, while the **fg** initiator cutoff is 24kHz. It wouldn’t seem like these differences are significant, but they are VERY significant.

The actual predominantly useful modes are a limited subset of the below list – look for the ‘Natural modes’ in a later section.

The initiator/mode can be modified by ‘modifiers’. There are 10 modifiers, 2 of the 10 are for input compensation, not directly related to decoding. One modifier, ‘B’ is used seldom, but often enough to be very aware of. The least used modifier is ‘H’, and used on VERY few recordings. I hesitated to add it, but did so for completeness. (The ‘H’ modifier is used for recordings with significant dynamics left over in FA mode, a special FA encoding.)

The possible syntax is:

fg

Base mode. There is 9kHz to 24kHz cut EQ at the beginning of each layer, and an extra 9kHz to 24kHz cut EQ at the beginning of each sequence.

fgG

Base mode. There is 9kHz to 21kHz cut EQ at the beginning of each layer, and an extra 9kHz to 21kHz cut EQ at the beginning of each sequence.

fg^, fg^^, fg^^^

fgG^, fgG^^, fgG^^^

These ‘^’ add EQ ONLY at the **beginning of DECODING** for either the ‘fg’ or ‘fgG’ FA mode initiators. One ^ adds a 9kHz to 18kHz boost at the beginning, before the first layer. Two ^ (^^) adds a 9kHz to 21kHz boost at the beginning, before the first layer. Three ^ (^^^) adds a 9kHz to 24kHz boost at the beginning, before the first layer.

fga, fgb, fgc,

fgGa, fgGb, fgGc

The letters ‘a’, ‘b’, ‘c’ add EQ at the **beginning of each SEQUENCE**. ‘a’ adds a 9kHz to 18kHz boost at the beginning of each sequence, ‘b’ adds a 9kHz to 21kHz boost at the beginning of each sequence, and ‘c’ adds a 9kHz to 24kHz boost at the beginning of each sequence.

‘fgB’,

‘fgGB’

The letter ‘B’ removes the bass EQ needed by most recordings. Some recordings already have the EQ, and if the recording sounds extraordinarily muddy, it probably needs the ‘B’ letter added.

‘fgH’.

‘fgGH

A very special purpose EQ, required for certain esoteric recordings. This adds 6dB of boost at 3kHz. Actual, clean, Supertramp albums are the only recordings that I know that need this. **This just might save you on certain recordings!!!** Never try this until you are desperate.

'fg+'

Adds 18k to 21kHz boost at the beginning. This is mild compensation.

'fg*'

Adds 15k to 24kHz and 18k to 21kHz boost at the beginning. This is relatively strong compensation.

Rules: the 'fg' or 'fGg' modes can have only ONE of '^', '^ ^', '^ ^ ^', 'a', 'b', 'c'. The 'B', '*', '+' modifiers can be added to any other mode.

So, fgaB is valid, fg^^B is valid, fgGB* is valid, fgbB+ is valid, etc. 'B' is optional and probably needed 10% of the time.

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=-50 --fcs="N, auto, fgG"

synonym for 'hfm2' is 'hf2', synonym for 'v3' is 'vocal3'

(Generic command #2)

da-avx --input=infile.wav --overwrite --output=outfile.wav --info=1 --tone=-46.0 --fcs="N, auto, fg^^"

(Generic command #3)

da-avx --input=infile.wav --overwrite --output=outfile.wav --info=1 --tone=-56.0 --fcs="N, auto, fgc*" --fw=classical

(Generic command #4)

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

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

Your best bet at a first guess calibration is -46 on old recordings, on newer recordings: -49 or -50 or -51. Classical recordings can require -tone=-54.5, also -50 or -60. I have seen -56 on normal POP recordings. High quality recordings can have almost any calibration – but are almost always standard values

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. (there is also nclassical, wclassical, wmix, nmix.)

Helpful Suggestion

Assume from the start that

-tone=-46.0 -fcs="4,auto,fgG^^" as a first guess.

Remember that some recordings need the bass EQ removed. That is done by using the 'B' submode.

Here are two ways to start choosing the correct FA initiator mode: One by 'natural' modes (mathematically clean) and the other by statistics.

Natural modes: (modes on the same line are equally natural):

fGg, fg	basic decoding mode
fGg^^, fg^^^	Normal cancellation of EQ at start of decoding.
fg^^	Represents a 'natural' EQ for FA at start of decoding.
fgc, fgGb	Normal cancellation of EQ for every sequence (group of layers)
fgb	Represents a 'natural' EQ for FA for each sequence

Modes above are most likely the correct choice and encountered.

Modes above in normal type haven't been encountered.

This list of steps below is just a suggestion. You will figure out your own way of choosing the EQ. I can usually select close to the correct EQ in about 2-3 minutes at most. Sometimes, there are difficult cases with very low calibration or other odd needs.

I often try fgG or fgG^^ first – it gives me an idea of the next required step.

- 1) First, preliminary 90% tries: '**fgG^^**' or '**fg^^^**' or '**fgc**', or *possibly* '**fgb**', '**fg**'.
- 2) Always remember the 'B' submode when the bass is too strong.

- 3) Except for the specific examples above, use the 'letter' modes like 'a' 'b' 'c' only if you have exhausted the '^' modes.
- 4) Remember '*' if the sound is a little fuzzy on the high end. Calibration can also cause a fuzzy high end.

The rest of these are secondary & optimizations. I can usually get good results with the above, but sometimes there are minor defects. Below is a somewhat disorganized set of ideas for better results.

- 5) On High-res material, I have sometimes been successful with 'fg' alone instead of 'fg^^^'. (The difference between three '^' (^^^) and two '^' (^^) are fairly small.)
- 6) Try 'fgG' alone.
- 7) If the sound is 'dull', then try '^' first -- this is because '^' is a natural add-on to 'fgG'
- 8) If the sound becomes too bright, then drop to a single '^', or still dull, then '^'
- 9) If the sound is good, but missing an 'edge' (mildly dull), then try '+', then '*'.
- 10) Try 'fg^^^' next, if none of the fgG work well.
- 11) If the sound is 'edgy', then try '^' next instead of '^^^' -- this is because '^' is a natural EQ to FA, while '^^^' is a natural EQ to 'fg'.
- 12) If the sound is still too bright, then try 'fg'. If you have too much brightness at 'fg', the calibration is probably too low, or the recording has too much HF.
- 13) There is also the 'a','b','c' suffix, and you can try those like 'fgGb' or 'fgc', with similar sequences etc.

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

New mode: --fx=opt, --fz=opt... These are an ideal mix of 'max' and the default, depending on which layer, automatically.

Absolutely the best quality is obtained by the --fz=max mode, runs full anti-MD on all layers.

There are even higher modes, DO NOT TRY TO USE THEM (--xpp=max or somesuch), unless you have more than 20 cores. I suggest using "--fx", "--fz" or "--fx=opt" or "--fz=opt", 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.

I have a computer with 10 cores, and --fz=max is just barely tolerable, but I avoid it. If you need to clean the highs, then try --fz=plus.

Other 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. --outgain values now add together.
- 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. --ingain values now add together.
- 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)

These switches are mostly useful when NOT in auto mode. **You probably don’t normally need to use these.** Perhaps the –b18 or –b15 series of switches might be helpful on recordings that have an HF rolloff, but unlikely. **Most useful pre-decoding EQ selections are built-in to the –fcs command complex now.**

Using these switches are normally an integral part of FA decoding when NOT in auto mode, 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.

If you want 'max' quality, it might be best to use the 'opt' version of the switches.

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 (deprecated)

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=<Iratio>

--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

These should be used mostly for 'tweaking' the sound, not for decoding per-se.

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 used to be necessary for accurate decoding. Now, these might be useful for tweaking the sound. 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)

New 1st order **variable** EQ commands

These provide both 1st order both bass and treble:

HF shelf: --pvh=freq,#dB

LF shelf: --pvl=freq,#dB

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 5 to 15. 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>