Usage guide for V1.6.2D DA/FA decoder

(Preliminary before complete docs are prepared.)

New features:

* Multi layer FA/DA decoding with enhanced speed.
* Much higher quality, even at --normal mode.
* Most of the time, --fx and --fz modes not needed, but still desirable. Diminishing returns beyond '--fx'
* --scrub option along with –fx=plus, --fx=max, --fx=highs, --fz=plus, --fz=max, --fz=highs provide more anti-MD
* Simplified (up-to-9) multi-layer usage.
* Simplified optimal calibration layer setting.
* Only fcf, fcc, fce=G, fcf=G, fcf=g, fce=g, fcc=G (maybe a few others) initiators are needed.
* Shortcut scheme now functional – no need for '--next' sequences.
* No worries about dynamics or layers/clipping between layers.
* Better metadata handling (default no-overwrite ICMT, use FADC if ICMT exists)
* Overwrite ICMT with –comment, --comment="Mesg", --comment="+Addonmessage"
* Simple (weak) anti-sibilance with '--as=xxx' (in linear amount)
* Simple, rough limiter with 'limiter=-xx' (in dB)
* This one is worth learning to use.   IT IS EASY TO USE.

Not fully tested on this release:

Timing line-up of input/output files is only reliable in DA mode.

Most FA material IS encoded multi-layer, and the sound quality of decoding results is substantially improved when doing an accurate multi-layer decoding operation.

Word usage is tricky here: We have decoding LAYERS, but there are calibration (--tone=) levels associated with them.   The layers start with '0', pretty much a very simple FA decoding like we were orginally doing, up to 9 layers (way too large a number to manage.)

We also have decoding LEVELS, those are the calibration (--tone=) levels associated with each layer.   Usually, LAYER 0 (the first one) will start with the lowest calibration LEVEL (-33 or -44.5 but not always – BY FAR, MOST OF THE TIME.   The highest decoding calibration level is usually between -19 and -13, but I have found that higher than-13 is very very seldom used – you are best off just starting with -33 or --34 (secondarily -43 or -44) to begin with.

So, our terminology LAYERS, starts with 0, and always increments up to 9. The layers represent a layering scheme like the 'Russian Doll'.   The LEVELS are associated with each layer, and represent the placement of the 10dB expansion range onto the level of the signal.

Each layer is directly connected to the next (and previous), and there is no clipping possible between the layers. There is no reduction to lower precision, so full numerical precision is maintained between each layer.

USAGE

The general command syntax is *exactly* the same as before, but there are new 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, and the A-K modifiers) still have siginificant value. A few of the legacy initiators are still potentially helpful.

The most important thing to understand is the general command syntax. This 'general' example does not include any extra features and only enables FA decoding. Refer to the *single layer* example below:

> (this command does a very simple single layer decode, simple 'dot' progress)
> **da-avx --input=infile.wav --overwrite --output=outfile.wav --info=1 --fcd --tone=-13**

The above command might not sound optimal for multi-layer material, and might need '--fcd=G' or '--fa' or '--fa=G' instead because of the layering mismatch. However, a single layer decode often can be an improvement. Sometimes also, the tone value might best be --tone=-19+N instead.   When decoding multi-layer material as single layer, sometimes adjusting the '--tone=' value will produce better results.

> (this command does a very OLD FASHIONED simple three layer decode, simple 'dot' progress)
> d**a-avx --input=infile.wav --overwrite --output=outfile.wav --info=1 --fcc --tone=-33 --next --fcd --next --fcd**

> (new-style multi-layer decode)
> d**a-avx --input=infile.wav --overwrite --output=outfile.wav --info=1 –fcs="3, -33, fcc"**

> The above 3 layer command needs some explanation:
> New-style:
>> Input/output files specified as usual. The number of layers is the first arg to '--fcs', the first –tone= value is the second arg. The third arg is the desired initiator.   Only one initiator needs to be specified.   For a single sequence decode, you'll want to start with –4X for 4 layers, or –3X for 3 layers.   Normally, the starting values are: -33, -34.5, -34 or -43, -44.5, -44.

> Old-style: (Use only for complex decodes)
>> The multi layer control groupings are separated by the '--next' switch. The '--next' switch starts a new layer/grouping. The first grouping starts by default. The second grouping starts after the first --next switch and the third group starts after the 2nd --next switch.
>> Each grouping uses the same defaults from the previous grouping. Since the first grouping starts with an '--fcc' initiator, then unless another initiator is used in the second grouping, then the '--fcc' will be the default.   Notice that the 2nd and third '--fcc' aren't actually needed, because the defaults are inherited from the previous grouping.   The 2nd grouping inherits the '--fcc' initiator from the first grouping, and the 3rd grouping inherits the '--fcc' from the 2nd grouping.
>>
>> Also, the --tone values are inherited, but instead of a direct inheritance, the --tone value is incremented by 10dB for every step.
>>
>> (Exact equivalent command for the three old style layer decode above, but taking more advantage of inheritance)
>> **da-avx --input=infile.wav --overwrite --output=outfile.wav --info=1 --fcd --tone=-39+N --next --next**

Understanding the new FA initiator scheme

The new initiator scheme is very simple and logical. The –fcc, --fcf, --fcf=G, --fcf=g 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 semi-common initiators are in 'italics'. The others are best ignored unless desperate.

Most to least highs:

--fa	FAMF	*(not on --fcs)*
--fa=G	*FAMF + 9kHz*	*(not 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)

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 –fcs="3, -33, fcf"

(Generic command #2)

da-avx --input=infile.wav --overwrite --output=outfile.wav --info=1 –fcs="3, -34.5, fcc"

(Generic command #3)

da-avx --input=infile.wav --overwrite --output=outfile.wav --info=1 –fcs="3, -34, fce=G"

(Generic command #4)

da-avx --input=infile.wav --overwrite --output=outfile.wav --info=1 –fcs="3, -34.5, fcf=classical"

Before becoming expert, it is best to keep one sequence of layers. That is, start with a low value like -33 or -43, then only go up to -13dB (20 or 30dB higher than the start.) It is best to try 3 layers, and if the sound is a bit 'bunched up', then try 4 layers.

For 4 layers, start with -43, -44.5 or -44.
For 3 layers, start with -33, -34.5 or -34.

Sometimes, the starting point can be –36 or -46 or even -39 or -49, but not often.

Experimenting with the 'initiator' like '**fcf'** or '**fcc**'. I suggest staritng with '**fcf**'. If the result is too 'sharp' or 'intense', then try **fcc**. If the result is too dull, then try **fce**. Sometimes, the material is not tamed even with **fcc**. Sometimes **fce=G** or**fcf=g** might be much better. This is a matter of experience and understanding.

Choosing the correct 'calibration' or 'tone' number like -43 or -34.5, that is a matter of stability of the sibilance and lack of gating.
Choosing the correct initiator is more for the general harshness, HF balance.

I STRONGLY suggest starting with fewer layers at first. At first, don't try more than 3 or 4 layers until you have a feel for the decoder.

Helpful Suggestion

The decoder actually has a slight tilt downwards vs frequency, but still sometimes sounds 'hot' on the high end.   I suggest that sometimes the fadditional EQ switches at the end of all over switches might be helpful: **--pe6k=K,-0.375 –pe7p5k=K,-0.375 –pe9k=K,-0.375 –pe10p5k=K,-0.375 –pe12k=K,-0.375**. You can pick and choose which switches and you can use other gain values.

~~IMPORTANT HINT: (allowing more highs with –fcd at the first, lower initiators).~~

~~NO LONGER VALID – left here to show the change. All of the differences in initiators VS layers is now compensated.~~

~~Very often, the proper initiator sequence doesn't use the same initiator for every layer.   The usual sequence is either the same initiator (e.g. always using –fcd or –fcd=G), or starting with –fcd at the lower (first) levels of decoding, and then –fcd=G for the higher levels.   When/if you allow a wrap-around, like from 19+N back to 49+N, and if the first (lowest) initiator was something like –fcd, then the first wrap-around initiator would also be –fcd…   Here is an example of this:~~

~~da-avx –input=infile.wav –output=outfile.wav –fcd –tone=-39+N –next –fcd=G –tone=-29+N –next –fcd=G –tone=-19+N –next –fcd –tone=-49+N~~

~~Note the above, where the first, -39+N layer is –fcd, the next two –29+N, 19+N layers are –fcd=G…   The last, wraparound layer at -49+N is –fcd again. So, the sequence is: --fcd (-39), --fcd=G(-29), --fcd=G(-19) and then at wrap around –fcd(-49)…   That is, wrap around is defined as the calibration (tone=) level going from a higher (-19dB) level back to a lower (-49dB) level.~~

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 are even higher modes, DO NOT TRY TO USE THEM.   I suggest using "–fx", "--fz" or "–fx –scrub" or "–fz –scrub", in this order of decreasing speed.

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 listening 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**
  Gives real-time display of program activity. I STRONGLY suggest using one of these switches.

--outgain=<xxdB>
  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.

--**floatout**
  For greater dynamic range, the decoder can produce floating point .wav files. Some consumer programs choke on FP files though.

Mode/Tuning switches:

--fcf, --fcc, --fcf=G, --fce=G, --fcf=g, --fce=G, --fcc=G
  (Never used with the –fcs switch.) 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,tone,FA init"
  This is now the preferred initiator for simple FA decoding operations. Example –fcs switch would be: --fcs="3,-33,fce" or –fcs="4,-44.5,fcc". One uses either a single **'--fcs'** switch (**preferred**) or a group of –fcf FA intiators along with –next switches (not preferred for simple decodes.)

--tone=<calibration level>
  This specifies the calibration level used, and is almost standardized. The calibration is almost always -13.55365 or a multiple of 10dB lower. The calibration could be –19.55365 or multiple of 10dB lower instead. To make typing easier, instead of -13.55365, one can type –13+N, or -19+N, where 'N' fills in the .55365. This '+N" form also applies to the 10dB offsets also, so you can specify –33+N just as easily.

--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.

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.

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:

--wia=<Iratio>
--woa=<Oratio>
--wof=<Eratio>

These switches override the default 'Iratio=2.0', 'Oratio=0.50' and 'Eratio=1.0'. Normally, Oratio is 1/Iratio. Usually Iratio is either 2.0 or 1.414. If the stereo image needs to be modified after the decoding, then Eratio can modify the stereo width. Numbers like 1.414, 1.19, 0.8409, 0.707 might be needed. Note that relations to sqrt 2 are most commonly used.
When 'classical' mode is operative, then Iratio=1.0 and Oratio=1.0.
Normally, for both classical and normal modes, --wof=1.414 by default, which you can change as needed.

Output equalization switches

This list does not contain full descriptions yet.

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 operative
- mode 3 – mixed Q=0.5/0.707/0.8509, <optQ> is not operative
- mode 4 – mixed Q=0.5/0.8409 spread over +-250Hz, <optQ> is not operative
- mode K (5) – mixed Q=0.5/0.707/0.8409 spread over +-250Hz, <optQ> is not operative

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

For the simple 1st order pi1k (bass), pi3k (treble), pi6k (treble), pi9k (treble) switches:

--pi1k=<0 or 1>	1kHz 3dB boost
--pi3k=< -1 or 1>	3kHz to 9kHz cut or boost
--pi9k=< -1 or 1>	9kHz to 18kHz cut or boost

New auxiliary switches

--limiter=-NN.NN (dB)

This limiter switch attempts to limit the output to the specified level. The behavior is not quite accurate 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 is definitely a tweakable parameter until the code is made more solid. It is definitely true that this mode DOES help.

--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. You might have to adjust the setting, and is mostly useful for very loud sibilance, and is not very effective against ‘fake sibilance’ where they used an approx 6kHz peaking filter to brighten the vocals. It DOES work against the peaking filter sibilance (like often used on Karen Carpenters vocals), but does help. 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. This is an EXPERIMENTAL tool, but just might help.