Brother John

BeSweet-Kommandozeilenreferenz

Version 11.01.2012
http://brother-john.net/besweet-referenz.html

PUBLIC DOMAIN

To the extent possible under law, Brother John has waived all copyright and related or neighboring rights to this work.
This work is published from Germany.

Inhalt

Komplettübersicht der Syntax ..... **III**

1 Grundlagen **1**
1.1 Installation 1
1.2 Die Syntax von BeSweet 3

2 Globale BeSweet-Konfiguration **5**
2.1 Der Abschnitt `-core()` 5
2.1.1 Basisparameter 5
2.1.2 Parameter für VOB-Quellen 7
2.1.3 Zielformat-Parameter 7
2.1.4 Tweaking-Parameter 9
2.2 Der Abschnitt `-split()` 10
2.3 Der Abschnitt `-plugin()` 10

3 Decoder und Soundmanipulation **12**
3.1 Der Abschnitt `-ota()` 12
3.1.1 Normalisierungsparameter 12
3.1.2 Delay-Parameter 14
3.1.3 Samplefrequenz- und Bildrate-Parameter 14
3.2 Der Abschnitt `-soundtouch()` 15
3.3 Die Abschnitte `-shibatch()` und `-ssrc()` 16
3.4 Der Abschnitt `-boost()` 17
3.5 Der Abschnitt `-azid()` 18
3.5.1 Normalisierung und Dynamikkompression 18
3.5.2 Downmix-Parameter 19
3.5.3 Parameter zur Manipulation einzelner Kanäle 21

4 Encoder **23**
4.1 Der Abschnitt `-ogg()` 23
4.2 Der Abschnitt `-lame()` 24
4.2.1 LAME bis Version 3.96 25
4.2.2 LAME ab Version 3.97 25
4.2.3 Spezialparameter 27
4.3 Die Abschnitte `-mp2enc()` und `-toolame()` 28
4.4 Der Abschnitt `-ac3enc()` 29

4.5 Der Abschnitt `-bsn()` für Aften AC3 30
4.6 Der Abschnitt `-bsn()` für Flake FLAC 31
4.7 Der Abschnitt `-bsn()` für Nero AAC 31
4.8 Der Abschnitt `-dimzon()` für Winamp AAC 33

Komplettübersicht der Syntax

Alle Elemente sind anklickbare Links zur jeweiligen ausführlichen Erklärung.

```
BeSweet.exe -core() [-split()] [-plugin()] [-ota()]
[-soundtouch()] [-shibatch()|-ssrc()] [-boost()] [-azid()]
[-ogg()|-bsn()|-lame()|-mp2enc()|-toolame()|-ac3enc()|-dimzon() ]

-core( -input <Quelldatei> -output <Zieldatei>
[-logfile <Logdatei>|-logfilea <Logdatei>]
[-seek|-substream <Stream-Nummer>]
[-2ch|-6ch|-6CH|-6chfloat|-6chwav|-5chaiff|-6chaiff|-6chliff|
-6chogg|-wavmp3|-ddwav|-payload|-6chnothing|-2chnothing]
[-noid3] [-be] )

-split( [-start <sec>] [-end <sec>] )

-plugin( -name <DLL-Name> -func <Name> [-6ch] )

-ota( [-norm <Level>|-g {<dB>|max|peak}|-G <Level>|-hybridgain]
[-d <ms>] [-e <ms>] [-sc <Auflösung>] [-fs <Hz>]
[-r <Quell-fps> <Ziel-fps>] )

-soundtouch( [-r <Quell-fps> <Ziel-fps>|-tempo <Prozent>|
-pitch <Halbtöne>|-rate <Prozent>] [-quick] [-naa] )

-shibatch( [--rate <Hz>] [--normalize] [--att <dB>] )
-ssrc( [--rate <Hz>] [--normalize] [--att <dB>] )

-boost( {/b=<Level>|/b2=<Level>|/b3=<Level>} [/l=<Wert>] )

-azid( [-g {max|<Level>}|--maximize] [-c {none|light|normal|
heavy|inverse}] [-d <vorn>/<hinten>] [-s {stereo|dpl|surround|
dplii|surround2}] [-S {<Pegel>|BSI}] [-C {<Pegel>|BSI}]
[-L <Pegel>] [-l <Pegel>] [-f {0|1}] [-n {0|1}]
[-o <Kanalreihenfolge>] [--ch{<Nummer>|<Kürzel>={light|normal|
heavy|inverse},<dB>] )
```

```
-ogg( [-q <Qualität>|{-b <kbps> [-m <kbps>] [-M <kbps>]}] [-6ch] )

-lame( <Encodingmodus> [<weitere Parameter>] )
-lame( --preset {{[fast] medium|[fast] standard|[fast] extreme|
insane}|cbr <kbps>|<kbps>} [<weitere Parameter>] )
-lame( -v --vbr-{old|new} -V <Qualität> [-b <kbps>] [-B <kbps>]
[<weitere Parameter>] )
-lame( {-b <kbps>|--abr <kbps> [-b <kbps>] [-B <kbps>]}
[<weitere Parameter>] )
-lame( <Encodingmodus> [--resample <Hz>] [--lowpass <kHz>]
[-m {s|j|f|m}] [-p] [-Y] )

-mp2enc( [-b <kbps>|-v <Qualität>] [-m {s|j|d|m}] [-e] )
-toolame( [-b <kbps>|-v <Qualität>] [-m {s|j|d|m}] [-e] )

-ac3enc( [-b <kbps>] [-6ch] )

-bsn( -exe aften.exe {-b <kbps>|-q <Qualität>} [-6chnew] )

-bsn( -exe flake.exe [-level <Level>] [-6chnew] )

-bsn( [-vbr <Qualität>|-abr <kbps>|-cbr <kbps>] [-6chnew]
[-aacprofile_{lc|he|hev2}] [-hintforstreaming] )

-dimzon( -dllname WA_aac.dll -cbr <kbps> [-lc] [-high] [-pns]
[-mpeg2] [-mode {1|2|3|4|5}] [-6chnew] )
```

1 Grundlagen

Dieses Dokument enthält einen Überblick über die Kommandozeilenparameter von BeSweet und deren Zusammenspiel. Die Referenz basiert auf DSPgurus BeSweet Version 1.5 Beta 31, der bsn.dll und der WA_aac.dll von Kurtnoise13. Bei Fragen kann zuerst ein Blick in die Doom9.org-BeSweet-FAQ nicht schaden (Vorsicht, teils veraltet!): http://forum.doom9.org/showthread.php?t=15738

In der Referenz fehlen aktuell die Kommandozeilen für 3GP-AAC, FAAC und Speex.

1.1 Installation

So einfach es anfangs auch erscheint, die BeSweet-Installation ist durchaus mit einigen Tücken behaftet, die wir in diesem Kapitel ansprechen wollen. Um BeSweet vollständig zu installieren, benötigen wir folgende Bestandteile:

► BeSweet
BeSweet v1.5b31 von der BeSweet-Homepage im Abschnitt »beta«.
http://besweet.notrace.dk/

► VOBInput.dll
Falls wir Sound direkt aus den VOB-Dateien transcodieren wollen. Abschnitt »Plugins & Source« der BeSweet-Homepage.
http://besweet.notrace.dk/

► Lame_enc.dll
Falls wir MP3s erzeugen wollen. Beste Anlaufstelle dafür ist RareWares.org. Ich bevorzuge die jeweils aktuelle stabile Version.
http://www.rarewares.org/mp3-lame-bundle.php

► Libvorbis.dll
Falls wir nach Vorbis transcodieren wollen. Auf RareWares.org erhalten wir das aktuelle Paket *libvorbis.dll using aoTuV* in der jeweils aktuellsten Version.
http://www.rarewares.org/ogg-libraries.php

► 2Lame
MP2-Encoding mit 2Lame benötigt die nicht im BeSweet-Paket enthaltene *tooLame.dll.* BeSweet liefert nur den mp2enc MP2-Encoder mit.
http://sourceforge.net/projects/toolame/

► Nero
AAC-Encoding mit Nero bedeutet nicht mehr, das komplette Nero-Paket besitzen zu müssen. BeSweet nutzt den kostenlosen Nero-Kommandozeilenencoder.
http://www.nero.com/deu/technologies-aac-codec.html

► Winamp
AAC-Encoding mit Winamp benötigt zum einen die *WA_aac.dll*, die man im BeLight-Downloadordner in der aktuellen Version findet. Außerdem benötigen wir Winamp.
http://kurtnoise.free.fr/BeLight/
http://www.winamp.com/

► Aften
Falls wir AC-3 encodieren wollen. Das SourceForge-Projekt stellt nur den Quellcode bereit. Ausführbare Pakete erhalten wir auch über SourceForge.
http://win32builds.sourceforge.net/aften/

► Flake
Falls wir FLAC encodieren wollen. Quellcode und ausführbare Dateien stellt das SourceForge-Projekt bereit.
http://sourceforge.net/projects/flake-enc/

► Bsn.dll
Die bsn.dll ist notwendig für Nero, Winamp, Aften und Flake. Die aktuelle Version erhalten wir wieder aus dem BeLight-Downloadordner.
http://kurtnoise.free.fr/BeLight/

Haben wir alle nötige Pakete, entpacken wir zuerst das BeSweet-Archiv in einen Ordner, z. B. *C:\Programme\BeSweet.* Außerdem installieren wir Winamp, wenn wir dessen AAC-Encoder verwenden wollen.

In den BeSweet-Ordner kopieren wir folgende Dateien aus den verschiedenen heruntergeladenen Archiven:

► **VOBInput.dll** und **toolame.dll**,
► **bsn.dll** aus dem Bsn-Archiv,
► **lame_enc.dll** aus dem LAME-Archiv,
► **libvorbis.dll** aus dem Vorbis-Download,
► **aften.exe** aus dem Aften-Archiv, am besten eine mit den passenden Optimierungen für unseren Prozessor,
► **flake.exe** aus dem Flake-Archiv, am besten eine mit den passenden Optimierungen für unseren Prozessor,
► **neroAacEnc.exe** aus dem Nero-Archiv.

Winamp-AAC-Encoding benötigt zwei Dateien im BeSweet-Ordner:

► **WA_aac.dll** aus dem Archiv von der BeLight-Seite,
► **enc_aacplus.dll** aus dem *Plugins*-Unterordner des Winamp-Ordners.

Damit ist BeSweet voll funktionsfähig.

1.2 Die Syntax von BeSweet

Syntax

```
BeSweet.exe -core() [-split()] [-plugin()] [-ota()]
[-soundtouch()] [-shibatch()|-ssrc()] [-boost()] [-azid()]
[-ogg()|-lame()|-mp2enc()|-toolame()|-ac3enc()|-bsn()|-dimzon()]
```

Abschnitte

Nur der `-core()`-Abschnitt muss zwingend immer angegeben werden. Alle anderen sind grundsätzlich optional und beliebig kombinierbar. Lediglich bei den Encoder-Abschnitten darf nur maximal einer definiert werden. Die kürzeste mögliche BeSweet-Befehlszeile lautet:

```
BeSweet.exe -core( -input <Quelle> -output <Ziel> )
```

BeSweet erzeugt dann eine CBR-Stereo-MP3 mit 128 kbit/s.

Leerzeichen

Alle Abschnitte werden jeweils durch ein Leerzeichen getrennt. Außerdem muss zwingend hinter der öffnenden Klammer und vor der schließenden Klammer ein Leerzeichen stehen. Parameter innerhalb eines Abschnitts werden ebenfalls durch

Leerzeichen getrennt. Als Beispiel (die Punkte • stehen für Leerzeichen):

`-core(•-input•quelle.ac3•-output•ziel.wav•-2ch•)•-ota(•-d•80•)`

Dezimaltrenner
BeSweet verwendet unabhängig von den Ländereinstellungen des Systems immer den Punkt als Dezimaltrennzeichen, nie das Komma. D. h. ½ muss immer als `0.5` geschrieben werden, nie als `0,5`.

Standardwerte
Manche Parameter besitzen einen Standardwert (häufig bei `-azid()`). Wird ein solcher Parameter weggelassen, verwendet BeSweet den Standardwert. Soll beispielsweise Azid einen Dolby-Pro-Logic-Downmix durchführen, muss die Option dafür (`-s surround`) nicht unbedingt definiert werden, weil BeSweet bei fehlendem `-s` automatisch `-s surround` setzt. Alle Funktionen ohne Standardwert sind nur dann aktiv, wenn sie explizit angegeben werden. Fehlt z. B. bei OTA der Delay-Parameter (`-d`) , wird auch kein Delay auf die Zieldatei angewendet.

Groß- und Kleinschreibung
Die meisten Abschnitte berücksichtigen bei den Parametern die Groß- und Kleinschreibung. Zum Beispiel sind `-6ch` und `-6CH` im `-core()`-Abschnitt zwei verschiedene Schalter. Deshalb sollte immer genau die in der Referenz angegebene Schreibweise verwendet werden.

2 Globale BeSweet-Konfiguration

2.1 Der Abschnitt `-core()`

Syntax

```
-core( -input -output [-logfile|-logfilea] [-seek|-substream] [-
2ch|-6ch|-6CH|-6chfloat|-6chwav|-5chaiff|-6chaiff|-6chliff|
-6chogg|-wavmp3|-ddwav|-payload|-6chnothing|-2chnothing]
[-noid3] [-be] )
```

Dieser Abschnitt enthält die grundlegende Konfiguration von BeSweet. Das beinhaltet besonders die Definition von Quell-, Ziel- und Logdateien sowie Schalter für Zieldateien, die keinen Encoder-Abschnitt benötigen.

2.1.1 Basisparameter

Die Optionen `-input` und `-output` müssen zwingend angegeben werden. Die restlichen sind optional.

`-input <Quelldatei>`

Pfad und Name der Quelldatei. Natürlich sind auch relative Pfade erlaubt. Pfade, die Leerzeichen enthalten, müssen nach den Windows-Konventionen in Anführungszeichen gesetzt werden. Jokerzeichen (* und ?) im Dateinamen sind nicht erlaubt.

```
-input "D:\Audio\Tonspur 1.ac3"
```

BeSweet akzeptiert folgende Typen von Quelldateien.

► **MPEG-Audio (*.MP3, *.MP2, *.MPA).** MPEG-1 Layer II und Layer III in Mono und Stereo.

► **Dolby Digital (*.AC3).** Mono, Stereo und Multikanal.

► **Ogg Vorbis (*.OGG).** Mono, Stereo und Multikanal.

► **Wave (*.WAV).** Mono-, Stereo- oder Multikanalwaves. 16-Bit-Waves nur bis maximal 2 GByte. 32-Bit-Fließkomma-Waves ohne Größenbeschränkung.

► **PCM-Rohdatenstrom (*.PCM).** »Headerlose Wave«. BeSweet erwartet 16-Bit, Stereo, 48 kHz. Eine andere Samplefrequenz kann über `-ota( -fs <Hz> )` angegeben werden.

► **AVIs (*.AVI).** AVI-Dateien mit AC3-, MPEG- oder PCM-Tonspur. Für PCM sind 8-Bit oder 16-Bit in Mono, Stereo oder Multikanal erlaubt.
► **VOBs (*.VOB).** DVD-Video-Object-Container mit AC3- und/oder PCM-Tonspuren. Für PCM sind 8-Bit oder 16-Bit in Mono, Stereo oder Multikanal erlaubt. 20-Bit und 24-Bit werden nicht unterstützt.
► **BeSweet-Listendatei (*.LST).** Reine Textdatei, in der die Pfade mehrerer Quelldateien enthalten sind (pro Zeile ein Pfad), die zu einer einzigen Zieldatei transcodiert werden sollen. Als Quell-Samplefrequenz wird 48 kHz angenommen. Eine abweichende Frequenz kann über `-ota( -fs <Hz> )` angegeben werden.
► **BeSweet-Muxingdatei (*.MUX).** Reine Textdatei ähnlich der Listendatei. Eine Muxingdatei enthält die Pfade von bis zu sechs Mono-Waves (pro Zeile ein Pfad), die BeSweet wie eine einzelne Multikanal-Wave verarbeiten kann. Die Reihenfolge der Dateinamen muss der korrekten Kanalreihenfolge für die Zieldatei entsprechen.

`-output <Zieldatei>`

Pfad und Name der Zieldatei. Natürlich sind auch relative Pfade erlaubt. Pfade, die Leerzeichen enthalten, müssen nach den Windows-Konventionen in Anführungszeichen gesetzt werden.

```
-output "D:\Audio\Tonspur 1 transcodiert.ogg"
```

Die Erweiterung der Zieldatei definiert *nicht* das Zielaudioformat. Das geschieht über die Zielformat-Parameter oder einen Encoder-Abschnitt. BeSweet korrigiert eine falsche Dateierweiterung nicht. Es ist also möglich – wenn auch wenig sinnvoll – z. B. eine Ogg-Vorbis-Datei mit der Endung .ac3 zu erzeugen. Achtung: Existierende Zieldateien werden ohne Nachfrage überschrieben!

`-logfile <Logdatei>`

Pfad und Name der Logdatei, in die BeSweet Informationen über den Transcodingvorgang schreibt. Fehlt diese Option, werden die Infos am Bildschirm ausgegeben.

```
-logfile "D:\Audio\Logfile.log"
```

`-logfilea <Logdatei>`

Ähnlich wie `-logfile`. Überschreibt allerdings existierende Dateien nicht, sondern hängt das neue Log am Ende der Datei an.

```
-logfilea "D:\Audio\Logfile.log"
```

2.1.2 Parameter für VOB-Quellen

-seek

Zeigt die Stream-Nummern (siehe `-substream`) aller in der VOB vorhandenen Tonspuren an. Transcodiert nichts. Dieser Schalter benötigt nur eine über `-input` definierte Quelldatei. Alle anderen Parameter können weggelassen werden.

-substream <Stream-Nummer>

Gibt die Nummer der Tonspur in der VOB an, die transcodiert werden soll. Die Nummern werden als hexadezimale Werte angegeben. Auf der DVD befindet sich AC3 üblicherweise in den Spuren `0x80` bis `0x88`, PCM in `0xA0` bis `0xA7` und MPEG-Audio in `0xC0` bis `0xC7`. Die Spuren `0x89` bis `0x8F` sind für optionale Tonformate wie DTS gedacht, die Standalone-Player ohne passenden Decoder entweder digital ausgeben oder ignorieren. Ist die Quelldatei eine VOB und `-substream` fehlt, decodiert BeSweet automatisch die erste Tonspur in der Datei.
`-substream 0x83`

2.1.3 Zielformat-Parameter

Von Ausnahmen (`-6chogg`, `-wavmp3`, `-ddwav`) abgesehen, werden diese Parameter nur dann verwendet, wenn kein Encoder-Abschnitt definiert ist, wenn also BeSweet nicht zum Transcoding, sondern nur zum Decoding oder zur Soundbearbeitung verwendet wird.

-2ch

Erzeugt eine 16-Bit-Stereo-Wave. Die Kanalreihenfolge dieser Datei ist FL, FR.

-6ch

Erzeugt sechs 16-Bit-Mono-Waves, eine für jeden Quellkanal. Fehlt ein Kanal in der Quelldatei, enthält die entsprechende Wave Stille. Als Kanalreihenfolge der Quelldatei wird angenommen: FL, FR, C, LFE, SL, SR.

-6CH

Erzeugt sechs 16-Bit-Mono-Waves, eine für jeden Quellkanal. Fehlt ein Kanal in der Quelldatei, enthält die entsprechende Wave Stille. Als Kanalreihenfolge der Quelldatei wird angenommen: FL, C, FR, SL, SR, LFE.

-6chfloat

Erzeugt sechs 32-Bit-Fließkomma-Mono-Waves, eine für jeden Quellkanal. Fehlt ein Kanal in der Quelldatei, enthält die entsprechende Wave Stille. Als Kanalreihenfolge der Quelldatei wird angenommen: FL, FR, C, LFE, SL, SR.

-6chwav

Erzeugt eine 16-Bit-Multikanal-Wave. Die Kanalreihenfolge dieser Datei ist FL, FR, C, LFE, SL, SR.

-5chaiff

Erzeugt eine 16-Bit-Multikanal-Big-Endian-AIFF-Datei. Die Kanalreihenfolge dieser Datei ist FL, FR, C, SL, SR.

-6chaiff

Erzeugt eine 16-Bit-Multikanal-Big-Endian-AIFF-Datei. Die Kanalreihenfolge dieser Datei ist FL, FR, C, LFE, SL, SR.

-6chliff

Erzeugt eine 16-Bit-Multikanal-Little-Endian-AIFF-Datei. Die Kanalreihenfolge dieser Datei ist FL, FR, C, LFE, SL, SR.

-6chogg

Erzeugt in Verbindung mit `-ogg()` eine Multikanal-Ogg-Vorbis-Datei. Fehlt der `-ogg()`-Abschnitt, wird mit einer VBR-Qualität von `0.800` transcodiert. Als Kanalreihenfolge der Quelldatei wird angenommen: FL, C, FR, SL, SR, LFE. Der Schalter darf nur gesetzt werden, wenn tatsächlich diese ungewöhnliche Kanalreihenfolge vorliegt.

-wavmp3

Transcodiert in Verbindung mit `-lame()` ins MP3-Format und schreibt zusätzlich einen Wave-Header vor die MP3-Datei. Das ist nützlich, um MP3s mit Programmen wie VirtualDub (aber nicht VirtualDubMod) zu muxen, die nur Waves verstehen. Der Parameter sollte auch nur in solchen Fällen angewendet werden. Fehlt der `-lame()`-Abschnitt, wird eine CBR-Stereo-MP3 mit 128 kbit/s erzeugt.

-ddwav

Erzeugt in Verbindung mit `-ac3enc()` eine Dolby-Digital-Wave. Prinzipiell handelt es sich dabei um eine AC3-Datei mit zusätzlichem Wave-Header, ähnlich wie bei `-wavmp3`. Fehlt der `-ac3enc()`-Abschnitt, wird mit 640 kbit/s encodiert.

-payload

Decodiert nicht, sondern kopiert den unveränderten Datenstrom der Quelldatei. Nützlich, um Spuren aus VOBs zu demuxen, um ohne Decoding Dateien zu splitten, Delays einzufügen usw.

2.1.4 Tweaking-Parameter

-6chnothing

Decodiert eine 6-Kanal-Tonspur, ohne die Daten auf die Festplatte zu schreiben. Wird von AacMachine benutzt, wenn Spuren aus einer VOB normalisiert werden.

-2chnothing

Decodiert eine Stereo-Tonspur, ohne die Daten auf die Festplatte zu schreiben. Wird von AacMachine benutzt, wenn Spuren aus einer VOB normalisiert werden.

-noid3

Verhindert das Schreiben des ID3V1-Tags, den BeSweet normalerweise der Zieldatei hinzufügt.

-be

Zwingt BeSweet, die Quelldatei als 16-Bit-Big-Endian-Wave anzusehen.

2.2 Der Abschnitt -split()

Syntax
```
-split( [-start] [-end] )
```

Diese Funktion erlaubt es, nur einen Teil der Quelldatei zu bearbeiten. Dabei können -start und -end auch einzeln angegeben werden. Fehlt -start, wird als Startpunkt der Dateianfang gesetzt, fehlt -end, ist der Endpunkt das Dateiende.

Beide Werte sind absolut, d. h. ein Startpunkt bei 6 Sekunden und ein Endpunkt bei 10 Sekunden ergibt eine 4 Sekunden lange Zieldatei.

-start <sec>

Setzt den Anfangspunkt für die Bearbeitung der Quelldatei auf den angegebenen Sekundenwert. Sekundenbruchteile sind erlaubt. Das Beispiel unten setzt den Startpunkt auf 10 Sekunden, 80 Millisekunden.

```
-start 10.080
```

-end <sec>

Setzt den Endpunkt für die Bearbeitung der Quelldatei auf den angegebenen Sekundenwert. Sekundenbruchteile sind erlaubt. Das Beispiel unten setzt den Endpunkt auf 90 Sekunden, 525 Millisekunden.

```
-end 90.525
```

2.3 Der Abschnitt -plugin()

Syntax
```
-plugin( -name -func [-6ch] )
```

BeSweets Funktionalität kann mit externen Plugins erweitert werden. Deren Funktionen werden im `-plugin()`-Abschnitt aufgerufen.

-name <DLL-Name>

Gibt den Dateinamen des Plugins an.
`-name MyPlugin.dll`

-func <Name>

Ruft die Funktion `<Name>` des Plugins auf.

-6ch

Liefert 6-Kanal-Input an das Plugin. Wird dann benötigt, wenn das Plugin den Downmix (z. B. auf Pro Logic) durchführt.

3 Decoder und Soundmanipulation

3.1 Der Abschnitt `-ota()`

Syntax
`-ota( [-norm|-g|-G|-hybridgain] [-d] [-e] [-sc] [-fs] [-r] )`

OTA steht für »overall track adjustment«. Hauptsächliches Einsatzgebiet ist die Normalisierung von Audioströmen und das Einfügen von Delays.

3.1.1 Normalisierungsparameter

`-norm <Level>`

Normalisiert die Datei vor dem Encoding (Pre-Gain-Normalisierung) auf einen bestimmen Prozentwert, wobei ein 2-Pass-Verfahren zum Einsatz kommt. Im ersten Durchgang wird die Datei nach dem höchsten Pegel durchsucht. Aus diesem und dem angegebenen `<Level>` errechnet sich der passende Gain-Wert, der im zweiten Durchgang während des Encodings angewendet wird.

`<Level>` ist eine Fließkommazahl zwischen `0` und `1`, wobei `1` für 100 % steht, also die maximal mögliche Normalisierung.

`-norm 0.97`

`-g {<dB>|max|peak}`

Erhöht oder senkt die Lautstärke der Datei vor dem Encoding (Pre-Gain) um einen festen Dezibel-Wert. `5db` z. B. hebt alle Samples um 5 dB an, `-5db` senkt um 5 dB ab.

`-g -5db`

`max` arbeitet mit dem höchsten Wert, der den Ton noch nicht verzerrt, und ist identisch zu `-norm 1`.

`-g max`

`peak` nimmt den Wert, der im Peak-Chunk der Quellwave angegeben ist. Existiert dieser Wert nicht oder ist die Quelldatei keine Wave, verhält sich `peak` genauso wie `max`.

`-g peak`

Im Vergleich zu Azids `-g`-Funktion arbeitet OTAs `-g` etwas langsamer, dafür aber genauer, und normalisiert beim Einsatz einer Listendatei alle Quellen insgesamt, so dass in der Zieldatei keine Sprünge in der Lautstärke auftreten können.

`-G <Level>`

Normalisiert die Datei nach dem Encoding (Post-Gain-Normalisierung) auf einen bestimmten Prozentwert. Der Wert wird als Fließkommazahl zwischen `0` und `1` angegeben, wobei `1` für 100 %, also die maximal mögliche Normalisierung, steht.

Sollte die Zieldatei nicht MP2, MP3 oder Vorbis sein, verhält sich `-G` genauso wie `-norm`. Das funktioniert jedoch nicht bei AAC. Für dieses Format muss immer ausdrücklich Pre-Gain (`-g` oder `-norm`) definiert werden.

```
-G 0.97
```

`-hybridgain`

Kombiniert einen festen Pre-Gain-Wert und Post-Gain-Normalisierung, abhängig vom Format der Quell- und Zieldatei. Die folgende Tabelle gilt nur für Stereo-Zieldateien. Handelt es sich beim Ziel um eine Multikanal-Datei, beträgt der Pre-Gain-Wert in jedem Fall 0 dB.

Quelldatei	DNR [1]	Pre-Gain bei Dynamikkompression [2]				
		keine	leicht	normal	schwer	invers
6-Kanal-AC3	ein	7 dB	8 dB	10 dB	14 dB	5 dB
Stereo-AC3	aus	0 dB	0 dB	0 dB	0 dB	0 dB

1) Dialog Normalization Reduction, Azids `-n` 2) Azids `-c`

Nach dem Encoding wird die Datei per Post-Gain auf 97 % normalisiert. Für MP3 wird außerdem LAMEs `scale`-Option auf 1 gesetzt.

Ist die Quelldatei keine AC3 oder das Zielformat nicht MP2, MP3 oder Vorbis, verhält sich `-hybridgain` genauso wie `-g max`. Das funktioniert jedoch nicht bei AAC. Für dieses Format muss immer ausdrücklich Pre-Gain (`-g` oder `-norm`) definiert werden.

3.1.2 Delay-Parameter

-d <ms>

Fügt am Anfang der Zieldatei die angegebene Anzahl Millisekunden an Stille ein oder schneidet (bei negativen Werten) soviel am Anfang der Datei weg.

```
-d 112
-d -80
```

-e <ms>

Fügt am Ende der Zieldatei die angegebene Anzahl Millisekunden an Stille ein.

```
-e 1600
```

-sc <Auflösung>

Setzt die minimale Sampleauflösung für Delay-Operationen. Wird von AacMachine benutzt, um vollständige AAC-Frames als Delay einzufügen.

3.1.3 Samplefrequenz- und Bildrate-Parameter

-fs <Hz>

Übergeht BeSweets automatische Samplefrequenz-Erkennung und erzwingt den angegebenen Hertz-Wert.

```
-fs 44100
```

Diese Funktion wandelt keine Samplefrequenzen um! BeSweet geht dann nur ohne weitere Prüfung davon aus, die Quelldatei hätte die angegebene Frequenz. Nützlich ist das für Quellmaterial mit unbekannter Samplefrequenz, z. B. PCM.

-r <Quell-fps> <Ziel-fps>

Konvertiert den Ton in eine andere Bildrate. Dabei ändern sich Tonhöhe und Länge der Audiospur. Die Funktion wird beispielsweise für NTSC-nach-PAL-Konvertierungen benötigt, also dann, wenn man die Bildrate des Videos ändert.

Die Werte sind Integer-Zahlen, wobei BeSweet die letzten drei Ziffern als Nachkommastellen behandelt. Das folgende Beispiel würde demnach eine Konvertierung von 23,976 fps auf 25,000 fps bewirken.

```
-r 23976 25000
```

Die Konvertierung funktioniert bei 6-Kanal-Zieldateien nur eingeschränkt. Möglich ist lediglich die Erzeugung von 5.1-AC3s und separaten Mono-Waves.

3.2 Der Abschnitt `-soundtouch()`

Syntax

```
-soundtouch( [-r|-tempo|-pitch|-rate] [-quick] [-naa] )
```

Mithilfe dieses Abschnitts kann die Geschwindigkeit und Tonhöhe des Audiostroms angepasst werden, leider nur für Mono- und Stereo-Zieldateien. Als Ausweichmöglichkeit bietet sich für 5.1-Transcodings unter Umständen OTAs `-r` an. Generell dürfte `-soundtouch()` eine höhere Qualität liefern als `-ota( -r )`.

`-r <Quell-fps> <Ziel-fps>`

Konvertiert den Ton in eine andere Bildrate, ohne dabei die Tonhöhe zu verändern. Die Funktion wird beispielsweise für NTSC-nach-PAL-Konvertierungen benötigt, also dann, wenn man die Bildrate des Videos verändert.

Die Werte sind Integer-Zahlen, wobei BeSweet die letzten drei Ziffern als Nachkommastellen behandelt. Das folgende Beispiel würde demnach eine Konvertierung von 23,976 fps auf 25,000 fps bewirken.

```
-r 23976 25000
```

`-tempo <Prozent>`

Ändert die Geschwindigkeit der Audiospur um den angegebenen Prozentwert, ohne die Tonhöhe zu verändern. Werte können zwischen `-95` und `+5000` liegen. Negative Werte verlangsamen den Ton, positive beschleunigen ihn.

```
-tempo -20
```

`-pitch <Halbtöne>`

Ändert die Tonhöhe der Audiospur um die angegebene Anzahl Halbtöne, ohne die Geschwindigkeit zu verändern. Werte können zwischen -60 und +60 liegen. Negative Werte senken die Tonhöhe, positive erhöhen sie.

```
-pitch +15
```

-rate <Prozent>

Ändert die Geschwindigkeit der Audiospur um den angegebenen Prozentwert, wobei sich auch die Tonhöhe entsprechend ändert. Die Funktion arbeitet analog zu `-r` im `-ota()`-Abschnitt. Werte können zwischen `-95` und `+5000` liegen. Negative Werte verlangsamen den Ton, positive beschleunigen ihn.

```
-rate -20
```

-quick

Verwendet einen schnelleren Algorithmus zur Berechnung. Darunter leidet jedoch die Qualität.

-naa

Schaltet Anti-Aliasing aus, um Geschwindigkeit auf Kosten der Qualität zu gewinnen.

3.3 Die Abschnitte `-shibatch()` und `-ssrc()`

Syntax

```
-shibatch( [--rate] [--normalize] [--att] )
-ssrc( [--rate] [--normalize] [--att] )
```

Diese Abschnitte dienen zur Umwandlung der Samplefrequenz. Allerdings sind einige Bugs bekannt. So funktioniert die Umwandlung von 44,1 kHz auf 48 kHz nicht korrekt. Für solche Fälle sollte ein externes Programm wie SSRC.exe verwendet werden. Beide Schreibweisen, `-shibatch()` und `-ssrc()`, sind identisch.

--rate <Hz>

Gibt die Zielsamplefrequenz in Hertz an.

```
--rate 24000
```

--normalize

Normalisiert die Lautstärke auf 100 %. Anstatt dieses Schalters sollten die Normalisierungsoptionen von `-ota()` oder `-azid()` verwendet werden.

--att <dB>

Dämpft die Zieldatei um die angegebene Anzahl Dezibel.
`--att 3`

3.4 Der Abschnitt -boost()

Syntax
`-boost( {/b|/b2|/b3} [/l] )`

Dieser Abschnitt dient genauso wie Azids `-c` zur Dynamikkompression. Damit sich die beiden Funktionen nicht gegenseitig stören und Artefakte im Sound erzeugen, sollte immer nur eine allein verwendet werden.

/b=<Level>

Definiert die Stärke der Dynamikkompression unter Verwendung eines Algorithmus von LigH. Werte liegen zwischen 1 und 10. Diese Funktion eignet sich mit einem Wert um 3 besonders für Stereo-Quellen.
`/b=3`

/b2=<Level>

Definiert die Stärke der Dynamikkompression unter Verwendung eines Algorithmus von DSPguru. Werte liegen zwischen 1 und 10. Diese Funktion eignet sich mit einem Wert um 4 besonders für 6-Kanal-Quellen.
`/b2=4`

/b3=<Level>

Definiert die Stärke der Dynamikkompression unter Verwendung eines Algorithmus von Tera. Werte liegen zwischen 1 und 10.
`/b3=5`

/l=<Wert> Standardwert: `0.99`

Setzt den maximal erlaubten Pegel der Zieldatei als Prozentwert zwischen `0` und `1`. Also müssten 95 % als `0.95` angegeben werden. Der Audiostrom wird nicht normalisiert. Der Parameter gibt nur an, welchen Wert die Lautstärke maximal erreichen darf.

```
/l=0.95
```

3.5 Der Abschnitt -azid()

Syntax

```
-azid( [-g|--maximize] [-c] [-d] [-s] [-S] [-C] [-L] [-l] [-f]
[-n] [-o] [--ch] )
```

Azid decodiert AC3-Dateien und führt falls gewünscht einen Downmix von 6-Kanal-AC3s durch.

Liegt eine AC3 als Quelle vor und der Azid-Abschnitt wird weggelassen, verwendet BeSweet die unten angegebenen Standardeinstellungen. Ein Spezialfall ist dabei der Parameter `-d`, den BeSweet je nachdem setzt, wie viele Kanäle das Zielformat erwartet. Für MP3 würde demnach immer ein Stereo-Downmix durchgeführt, für Vorbis oder AAC hängt das von den Einstellungen des Encoder-Abschnitts ab.

3.5.1 Normalisierung und Dynamikkompression

-g {max|<Level>}

Führt eine Pre-Gain-Normalisierung durch, ähnlich wie OTAs `-g` und `-norm`. `max` ist identisch zu `--maximize`. Die Funktion durchsucht die Datei erst nach dem höchstmöglichen Gain-Wert und wendet diesen dann an (2-Pass-Verfahren wie bei OTAs `-norm`).

```
-g max
```

`<Level>` kann auf verschiedene Arten verwendet werden. Ein Wert zwischen `0%` und `99%` arbeitet ähnlich wie `-g max`, nur wird nicht der höchstmögliche Gain-Wert auf die Datei angewendet, sondern der angegebene Prozentsatz des Maximums.

```
-g 97%
```

Eine negative Zahl mit dem Zusatz db arbeitet ähnlich wie die Prozentangabe. Hier wird allerdings der höchstmögliche Gain-Wert abzüglich der definierten Anzahl Dezibel angewendet.

`-g -3.2db`

Eine positive Zahl mit dem Zusatz db hebt die Lautstärke um einen festen Dezibel-Wert an.

`-g 4db`

Im Vergleich zu OTAs `-g` arbeitet Azids `-g` schneller, dafür aber etwas ungenauer. Außerdem normalisiert Azid beim Einsatz von Listendateien die Quellen einzeln, so dass in der Zieldatei Sprünge in der Lautstärke auftreten können. Um das zu vermeiden, sollte in diesem Fall immer über `-ota()` normalisiert werden.

`--maximize`

Führt eine Pre-Gain-Normalisierung durch, die identisch zu `-gmax` arbeitet. Erst wird die Datei nach dem höchstmöglichen Gain-Wert durchsucht und die Lautstärke dann so weit angehoben (2-Pass-Verfahren wie bei OTAs `-norm`).

`-c {none|light|normal|heavy|inverse}` Standardwert: none

Komprimiert die Dynamik der Quelldatei. D. h. die Spanne zwischen lautestem und leisestem Ton wird reduziert. Die Funktion besitzt fünf Modi.

`none`	Keine Dynamikkompression.
`light`	Leichte Kompression. Dieser Modus komprimiert 50 % (-6 dB) schwächer als `normal`.
`normal`	Normale Kompression. Die Decoder in Standalone-Playern verwenden in der Regel diesen Modus.
`heavy`	Starke Kompression. Eignet sich für eine laute Umgebung oder weniger hochwertige Hifi-Anlagen/Boxensysteme.
`inverse`	Umkehrung des `light`-Modus, d. h. laute Samples werden nochmals verstärkt, leise weiter gedämpft.

3.5.2 Downmix-Parameter

`-d <vorn>/<hinten>` Standardwert: siehe Kapitelanfang

Gibt an, für wie viele Zielkanäle Azid downmixen soll. Wichtig: Ein evtl. vorhandener LFE-Kanal wird hier nicht berücksichtigt. Für Stereo-Downmixing ist zusätzlich

der `-s`-Parameter interessant. Werte werden in der Form `Kanäle vorn/Kanäle hinten` angegeben. Erlaubt sind: `1/0`, `2/0`, `3/0`, `1/1`, `2/1`, `3/1`, `1/2`, `2/2` und `3/2`.

```
-d 2/0
```

`-s {stereo|dpl|surround|dplii|surround2}` Standardwert: `dpl`

Legt fest, in welchem Modus ein Stereo-Downmix durchgeführt wird. Dieser Parameter wirkt sich nur dann aus, wenn `-d 2/0` angegeben ist.

`stereo`	Normaler Stereo-Downmix.
`dpl`	Downmix kompatibel zu Dolby Pro Logic.
`surround`	wie `dpl`
`dplii`	Downmix kompatibel zu Dolby Pro Logic II.
`surround2`	wie `dplii`

`-S {<Pegel>|BSI}` Standardwert: `BSI`

Gibt an, mit welcher Pegelveränderung die hinteren Surround-Kanäle in die vorderen Kanäle links und rechts gemixt wird. Der Pegel wird in Dezibel angegeben. Der Wert BSI veranlasst BeSweet, Die Pegelveränderung entsprechend dem im BSI-Abschnitt der AC3-Datei gespeicherten Wert zu setzen.

```
-S BSI
-S -1.5db
```

`-C {<Pegel>|BSI}` Standardwert: `BSI`

Gibt an, mit welcher Pegelveränderung der Center-Kanal in die vorderen Kanäle links und rechts gemixt wird. Der Pegel wird in Dezibel angegeben. BSI setzt den Pegel entsprechend den in der AC3 gespeicherten BSI-Infos.

```
-C -2db
```

`-L <Pegel>` Standardwert: `0db`

Gibt an, mit welcher Pegelveränderung der LFE-Kanal (Basskanal) in die vorderen Kanäle links und rechts gemixt wird. Der Pegel wird in Dezibel angegeben.

```
-L -3db
```

-l <Pegel> Standardwert: `0db`

Gibt an, mit welcher Pegelveränderung der Quell-LFE-Kanal in den Ziel-LFE-Kanal gemixt wird. Der Pegel wird in Dezibel angegeben.
`-l -2.5db`

-f {0|1} Standardwert: `0`

Filtert beim Downmix auf Stereo die hinteren Surroundkanäle. Der Filter ist ein »2nd order butterworth filter«, der abhängig von der Frequenz eine ansteigende Phasenverschiebung durchführt. Bei 7 kHz erreicht diese 90 Grad und –3 dB.

Hauptsächlich findet dieser Schalter Anwendung, um einen korrekten Pro-Logic-Downmix sicherzustellen und/oder um Phasing-Probleme (Ton klingt ausgewaschen) bei schlecht gemasterten Tonspuren zu vermeiden.

`-f 1` schaltet die Funktion ein, `-f 0` oder keine Angabe schaltet sie aus.

-n {0|1} Standardwert: `0`

Aktiviert »Dialog Normalization Reduction« (DNR). 5.1-AC3-Dateien enthalten in den BSI-Infos eine Angabe, wie weit die subjektiv empfundene Lautstärke der Dialogspur (Center-Kanal) unter dem maximalen Pegel liegt. Der Schalter `-n` ändert den Dialogpegel unter Berücksichtigung der BSI-Info auf –31 dB. Eine Normalisierung wird dadurch nicht beeinträchtigt, da DNR vor allen Normalisierungsfunktionen angewendet wird.

Empfehlenswert ist DNR nur für zwei Szenarios. Erstens: Um mehrere 5.1-AC3s mit unterschiedlichen Dialogleveln in eine einzelne Zieldatei zu transcodieren. Zweitens: Um unterschiedliche Dialoglevel innerhalb einer einzelnen Quell-AC3 anzugleichen (möglicherweise eine TV-Aufnahme mit lauten Werbeblöcken). DNR wirkt sich nicht aus, wenn AC3s ohne unterschiedliche Dialoglevel transcodiert werden (sollte für die meisten DVD-Audiospuren der Fall sein). Außerdem sind AC3s ohne Center-Kanal (z. B. 2/0) nicht betroffen.

`-n 1` schaltet die Funktion ein, `-n 0` oder keine Angabe schaltet sie aus.

3.5.3 Parameter zur Manipulation einzelner Kanäle

Für ein normales AC3-Transcoding sind diese Parameter meistens unnötig. Wichtig wird `-o`, wenn sich die Kanalreihenfolgen von Quell- und Zieldatei unterscheiden.

-o <Kanalreihenfolge>

Definiert, in welcher Reihenfolge die Kanäle in die Zieldatei geschrieben werden. Diese Funktion wird nur benötigt, wenn sich die Kanalreihenfolgen von Quell- und Zieldatei unterscheiden. Die Kanäle werden mit Komma getrennt und mit Kürzeln bezeichnet, die ihre Position in der *Quelldatei* angeben.

`l` FL-Kanal der Quelldatei (vorne links).
`r` FR-Kanal der Quelldatei (vorne rechts).
`c` C-Kanal der Quelldatei (vorne Mitte).
`lfe` LFE-Kanal der Quelldatei.
`sl` SL-Kanal der Quelldatei (hinten links)
`sr` SR-Kanal der Quelldatei (hinten rechts)

Eine Anwendung dieses Parameters ist das Transcoding von 5.1-AC3 nach 5.1-Vorbis. AC3 hat eine Kanalreihenfolge von FL, FR, C, LFE, SL, SR. Vorbis dagegen erwartet FL, C, FR, SL, SR, LFE. In diesem Fall müsste der `-o`-Parameter folgendermaßen verwendet werden:

```
-o l,c,r,sl,sr,lfe
```

--ch<Nummer>={light|normal|heavy|inverse},<dB>

Legt für einzelne Kanäle individuelle Einstellungen zu Dynamikkompression und Gain fest. `<Nummer>` bezeichnet die Kanalposition von `0` bis `5` in der Zieldatei. Die Nummer richtet sich also danach, was eventuell mit `-o` definiert wurde. Der Wert vor dem Komma gibt die Stärke der Dynamikkompression an. Es werden die gleichen Werte wie bei `-c` verwendet. Der Wert nach dem Komma gibt einen festen Gain-Wert in Dezibel an, der für diesen Kanal gilt. Er wird mit dem Zusatz db versehen. Um beispielsweise für Kanal Nummer 4 (in AC3-Standardreihenfolge ist das SL) eine leichte Kompression zu verwenden und die Lautstärke um 2,5 dB abzusenken, müsste der Parameter so aussehen:

```
--ch4=light,-2.5db
```

--ch<Kürzel>={light|normal|heavy|inverse},<dB>

Dieser Parameter arbeitet analog zu `--ch<Nummer>`, bezieht sich aber auf die Quellkanäle. Die `<Kürzel>` sind die gleichen wie bei `-o`.

Das Beispiel von oben (SL-Kanal mit leichter Kompression und 2,5 dB Dämpfung) könnte identisch mit diesem Parameter konfiguriert werden:

```
--chsl=light,-2.5db
```

4 Encoder

Diese Abschnitte sind nicht kombinierbar, d. h. es darf nur maximal einer verwendet werden. Wird kein Encoder-Abschnitt definiert, sollte ein Zielformat-Parameter im `-core()`-Abschnitt gesetzt werden.

4.1 Der Abschnitt `-ogg()`

Syntax
`-ogg( [-q|{-b [-m] [-M]}] [-6ch] )`

Encodiert ins Ogg-Vorbis-Format. Vorbis ist stark auf echtes VBR im Constant-Quality-Modus ausgerichtet, deshalb sollte möglichst immer der `-q`-Parameter verwendet werden, um Qualität und Bitrate der Zieldatei zu definieren.

Der Abschnitt kann leer als `-ogg()` angegeben werden. Dann wird mit einer Durchschnittsbitrate von 160 kbit/s encodiert (enspricht `-ogg( -b 160 )`).

-q <Qualität>

Setzt das Qualitätslevel, mit dem die Zieldatei encodiert wird. Die Obergrenze liegt bei `1.000`. Als Untergrenze gilt `-0.200`.

Die übliche Vorbisnotation benutzt eine Qualitätsskala von –2 bis 10, BeSweet dagegen von –0,2 bis 1. Umrechnungen sind einfach möglich, indem man den BeSweet-Wert mit 10 multipliziert oder den Vorbis-Wert durch 10 dividiert. Also entspricht BeSweets `0.250` in Standardnotation `2.50` und umgekehrt.

`-q 0.200`

Für Qualitäten kleiner als `0` zeigt BeSweet an `Avarege Bitrate: 160` (ja, einschließlich des Tippfehlers). Transcodiert wird aber mit der korrekten Qualität.

-b <kbps>

Setzt die nominale Bitrate der Vorbis-Datei in kbit/s (ABR-Modus).

`-b 128`

`-m <kbps>`

Setzt die minimal erlaubte Bitrate in kbit/s (in Verbindung mit `-b`).
`-m 64`

`-M <kbps>`

Setzt die maximal erlaubte Bitrate in kbit/s (in Verbindung mit `-b`).
`-M 160`

`-6ch`

Erzeugt eine 6-Kanal-Ogg-Vorbis mit der Kanalreihenfolge FL, C, FR, SL, SR, LFE. Hat die Quelldatei eine andere Reihenfolge, müssen die Kanäle entsprechend umgeleitet werden.

Funktioniert nicht in Verbindung mit `-ssrc()`, `-boost()` und `-ota( -g )`.

4.2 Der Abschnitt `-lame()`

LAME erzeugt MP3-Dateien. Je nach Version der verwendeten *lame_enc.dll* sollten andere Parameter fürs Encoding verwendet werden. Entsprechend beschäftigt sich Kapitel 4.2.1 mit LAME bis Version 3.96 (i. d. R. nicht mehr zu empfehlen), Kapitel 4.2.2 mit LAME ab 3.97. Anschließend werden unter 4.2.3 Zusatzoptionen besprochen, die nur unter speziellen Umständen interessant sind.

Der `-lame()`-Abschnitt kann leer als `-lame()` angegeben werden. Dann wird eine CBR-Stereo-MP3 mit 128 kbit/s erzeugt (identisch mit `-lame( --preset cbr 128 )`).

4.2.1 LAME bis Version 3.96

Syntax
```
-lame( [--preset] [<weitere Parameter>] )
```

--preset {{[fast] medium|[fast] standard|[fast] extreme|insane}|cbr <kbps>|<kbps>}

Definiert Modus und Qualität an, mit der LAME die MP3-Datei encodiert.

Konstante Bitrate wird mit `cbr` und einem Wert in kbit/s angegeben. Zulässig sind `8`, `16`, `32`, `40`, `48`, `56`, `64`, `80`, `96`, `112`, `128`, `160`, `192`, `224`, `256` und `320`. Encoding mit einer konstanten Bitrate von 192 kbit/s sieht so aus:

```
--preset cbr 192
```

Außerdem benutzt das benannte Preset `insane` eine konstante Bitrate von 320 kbit/s. Die Angabe `--preset insane` ist identisch zu `--preset cbr 320`.

Eine Durchschnittsbitrate (ABR) wird ohne Zusatz durch einen beliebigen Wert in kbit/s zwischen `77` (für LAME-Version 3.90) bzw. `72` (für Version 3.96) und `320` angegeben. Encoding mit einer durchschnittlichen Bitrate von 135 kbit/s sieht so aus:

```
--preset 135
```

Variable Bitrate wird durch ein benanntes Preset angegeben. Von der niedrigsten zur höchsten Bitrate (= Qualität) heißen diese `medium`, `standard`, `extreme`. Jedem der Presets kann ein `fast` vorangestellt werden, um den leicht ungenaueren, dafür aber deutlich schnelleren neuen VBR-Modus zu benutzen.

```
--preset standard
--preset fast medium
```

Anstatt `--preset` kann auch `--alt-preset` verwendet werden.

```
--alt-preset standard
--alt-preset 150
```

4.2.2 LAME ab Version 3.97

Syntax
```
-lame( -v --vbr-{old|new} -V [-b] [-B] [<weitere Parameter>] )
-lame( {-b|--abr} [-b] [-B] [<weitere Parameter>] )
```

BeSweet definiert standardmäßig 128 kbit/s als Minimalbitrate. Bei niedrigen VBR-Einstellungen (etwa ab `-V 5`) oder kleinen ABR-Bitraten (etwa ab 140 kbit/s)

ist es deshalb sinnvoll, den Wert mit dem Paramter `-b` zu senken. Ansonsten ist eine endgültige Bitrate von unter 128 kbit/s nicht erreichbar.

`-v`

Aktiviert den VBR-Modus. Muss zwingend in Verbindung mit `--vbr-old` bzw. `--vbr-new` und `-V` verwendet werden, und zwar in der oben in der Syntax angegebenen Reihenfolge.

`--vbr-{old|new}`

Wählt den VBR-Modus, wobei `--vbr-old` für den älteren und langsameren Standard-VBR-Modus steht, `--vbr-new` für den schnelleren neuen Modus. Muss zwingend in Verbindung mit `-v` und `-V` verwendet werden, und zwar in der oben in der Syntax angegebenen Reihenfolge.

`-V <Qualität>`

Definiert die verwendete VBR-Qualitätsstufe. Gültige Werte sind ganze Zahlen zwischen `0` und `8`. Dabei steht `0` für die höchste, `8` für die niedrigeste Qualität/Bitrate.

Muss zwingend in Verbindung mit `-v` und `--vbr-old` bzw. `--vbr-new` verwendet werden, und zwar in der oben in der Syntax angegebenen Reihenfolge.

```
-v --vbr-new -V 2
```

`-b <kbps>`

Encodiert mit der angegebenen Bitrate im CBR-Modus. Zulässig sind `8`, `16`, `32`, `40`, `48`, `56`, `64`, `80`, `96`, `112`, `128`, `160`, `192`, `224`, `256` und `320`.

```
-b 128
```

Im VBR- und ABR-Modus definiert der Parameter die minimal mögliche Bitrate für ein Audioframe. Gültige Werte sind `32`, `40`, `48`, `56`, `64`, `80`, `96`, `112`, `128`, `160`, `192`, `224`, `256` und `320`. Bei Verwendung von sowohl `-b` als auch `-B`, muss die Minimalbitrate immer kleiner sein als die Maximalbitrate.

```
-v --vbr-new -V 7 -b 32
--abr 128 -b 64
```

-B <kbps>

Setzt im VBR- und ABR-Modus die maximal mögliche Bitrate für ein Audioframe. Gültige Werte sind 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256 und 320. Bei Verwendung von sowohl -b als auch -B, muss die Minimalbitrate immer kleiner sein als die Maximalbitrate.

```
-v --vbr-old -V 3 -B 224
--abr 96 -b 32 -B 256
```

--abr <kbps>

Encodiert mit der angegebenen durchschnittlichen Bitrate (ABR-Modus). Zulässige Werte für <kbps> liegen zwischen 72 und 320.

4.2.3 Spezialparameter

Syntax

```
-lame( <Encodingmodus> [--resample] [--lowpass] [-m] [-p] [-Y] )
```

--resample <Hz>

Ändert die Samplefrequenz der Zieldatei auf den angegebenen Wert in Hertz. Für echtes Resampling sollte -shibatch() verwendet werden. Dieser Parameter dient nur dazu, das automatische Downsampling von Lame bei niedrigen Bitraten (unter 88 kbit/s) zu verhindern. Erlaubte Werte sind 8000, 11025, 12000, 16000, 22050, 24000, 32000, 44100, 48000. Um für ein Encoding mit weniger als 88 kbit/s eine Quell-Samplefrequenz von z. B. 44100 Hz beizubehalten, müsste geschrieben werden:

```
--resample 441000
```

--lowpass <kHz>

Definiert die Frequenz für die Anwendung des Lowpass-Filters in Kilohertz. Alle Frequenzen über dem angegebenen Wert werden verworfen und nicht encodiert. Maximal möglicher Wert ist die Hälfte der Samplefrequenz. Bei einer 48-kHz-Quelle darf der Wert also maximal 24 betragen. Sinnvoll sind nur kleinere Werte, da eine 48-kHz-Datei keine höheren Frequenzen als 24 kHz enthalten kann.

`--lowpass 18.5`
Lowpass-Filter werden von Lame automatisch passend zur Qualitätseinstellung verwendet. Deshalb sollte `--lowpass` außer in besonderen Ausnahmefällen nicht verwendet werden.

`-m {s|j|f|m}`

Standardwert: `j`

Der Parameter wird nur für einen Downmix auf Mono benötigt. Dann muss angegeben werden:
`-m m`
Die restlichen drei Möglichkeiten gelten für Stereo-Zieldateien und legen den Modus fest, in dem die Kanäle encodiert werden. Der Standard `j` sollte bei Stereo-Encodings immer beibehalten werden.

`s`	normales Stereo.
`j`	Joint-Stereo.
`f`	Erzwungenes (forced) Joint-Stereo.

`-p`

Berechnet und speichert eine 16-Bit-CRC-Prüfsumme für jedes Audioframe (Fehlerschutz). Die Prüfsumme ist Teil der angegebenen Bitrate, d. h. für die Audiodaten bleiben entsprechend weniger Bits übrig.

`-Y`

Ändert die ATH-Anpassungsmethode, so dass Rauschen in sfb21 ignoriert wird.

4.3 Die Abschnitte `-mp2enc()` und `-toolame()`

Syntax

```
-mp2enc( [-b|-v] [-m] [-e] )
-toolame( [-b|-v] [-m] [-e] )
```

Der angegebene Name dieses Abschnitts legt fest, ob MP2enc oder 2Lame als MP2-Encoder verwendet wird. Es stehen bei beiden die gleichen Optionen zur Verfügung. Dieser Abschnitt kann leer als `-mp2enc()` bzw. `-toolame()` angegeben werden. Dann wird im Dual-Channel-Modus mit 192 kbit/s CBR encodiert.

-b <kbps> Standardwert: 192

Encodiert im CBR-Modus mit der angegebenen Bitrate in kbit/s.
`-b 192`

-v <Qualität>

Encodiert im VBR-Modus mit dem angegebenen Qualitätslevel. Werte können zwischen `0` und `9` liegen. Vorsicht! Der VBR-Modus ist experimentell und ohne Funktionsgarantie.
`-v 5`

-m {s|j|d|m} Standardwert: d

Definiert den verwendeten Stereo-Modus. `s` für normales Stereo, `j` für Joint-Stereo, `d` für Dual-Channel und `m` für Mono.
`-m s`

-e

Fügt dem encodierten Stream Prüfsummen als Fehlerschutz hinzu.

4.4 Der Abschnitt -ac3enc()

Syntax
`-ac3enc( [-b] [-6ch] )`

Encodiert ins AC3-Format. Aufgrund schwerer Qualitätsprobleme ist `-ac3enc()` nicht sinnvoll. Als Alternative steht Aften bereit.

Bei der Verwendung von `-ac3enc()` muss immer über Azid normalisiert werden. Eine Normalisierung über OTA wird ignoriert. Dieser Abschnitt kann leer als `-ac3enc()` angegeben werden. Dann wird eine 6-Kanal-AC3 mit 384 kbit/s erstellt.

-b <kbps> Standardwert: 384

Setzt die Bitrate in kbit/s, mit der die AC3-Datei encodiert wird. Standard auf der DVD sind hierbei 448 und 384 kbit/s für 6-Kanal-AC3s und 192 kbit/s für Stereo-AC3s.

```
-b 448
```

-6ch

Veranlasst ac3enc, eine 6-Kanal-AC3 zu erzeugen. Ohne diesen Parameter richtet sich ac3enc nach der Kanalanzahl der Quelldatei.

4.5 Der Abschnitt -bsn() für Aften AC3

Syntax

```
-bsn( -exe aften.exe {-b|-q} [-6chnew] )
```

Encodiert ins mit Aften ins AC3-Format. Die Option `-exe aften.exe` ist zwingend erforderlich, um den Aften-Encoder zu aktivieren.

-b <kbps>

Encodiert mit der angegebenen konstanten Bitrate im CBR-Modus. Maximal sind 640 kbit/s möglich.

```
-b 384
```

-q <Qualität>

Encodiert mit der angegebenen Qualität im VBR-Modus. Gültige Werte sind ganze Zahlen zwischen `0` und `1023`.

```
-q 500
```

-6chnew

Veranlasst Aften, eine 6-Kanal-AC3 zu erzeugen. Ohne diesen Parameter wird Stereo angenommen. Aften richtet sich nicht nach der Kanalanzahl der Quelle, sondern ausschließlich nach diesem Schalter.

4.6 Der Abschnitt -bsn() für Flake FLAC

Syntax
```
-bsn( -exe flake.exe [-level] [-6chnew] )
```

Verwendet den Flake-Encoder, um ins verlustlose FLAC-Format zu encodieren. Die Option `-exe flake.exe` ist zwingend erforderlich, um den Flake-Encoder zu aktivieren.

-level <Level>

Legt fest, wie intensiv die Kompression arbeiten soll. Je höher das Level, desto kleiner die Zieldatei und desto länger die Encodingzeit. Gültige Werte sind ganze Zahlen zwischen `0` und `12`.
`-level 8`

-6chnew

Veranlasst Flake, eine 6-Kanal-FLAC zu erzeugen. Ohne diesen Parameter wird Stereo angenommen. Flake richtet sich nicht nach der Kanalanzahl der Quelle, sondern ausschließlich nach diesem Schalter.

4.7 Der Abschnitt -bsn() für Nero AAC

Syntax
```
-bsn( [-vbr|-abr|-cbr] [-aacprofile_{lc|he|hev2}] [-6chnew]
[-hintforstreaming] )
```

Verwendet den kostenlosen Nero-CLI-Encoder, um AAC-Audio zu encodieren. Die Zieldatei wird immer in den MP4-Container verpackt.

AAC ist stark auf echtes VBR im Constant-Quality-Modus ausgerichtet, deshalb sollte möglichst immer der Parameter `-vbr` verwendet werden, um Qualität und Bitrate der Zieldatei zu definieren.

-vbr <Qualität>

Setzt das Qualitätslevel der Zieldatei im VBR-Modus. Gültige Werte liegen zwischen `0.00` (niedrige Qualität) und `1.00` (hohe Qualität). Eine Ausnahme ist 6-Kanal-HE-AAC, wofür ein Maximum von `0.45` gilt.

```
-vbr 0.25
```

-abr <kbps>

Encodiert mit der angegebenen durchschnittlichen Bitrate im ABR-Modus. Zulässig sind positive ganze Zahlen.

```
-abr 80
```

-cbr <kbps>

Encodiert mit der angegebenen konstanten Bitrate im CBR-Modus. Zulässig sind positive ganze Zahlen.

```
-cbr 100
```

-aacprofile_{lc|he|hev2}

Standardwert: Automatik

Erzwingt ein Encoding mit dem angegebenen AAC-Profil. `-aacprofile_hev2` ist nur für Stereo-Zieldateien möglich und für extrem niedrige Bitraten gedacht. LC und HE können dagegen sowohl mit Stereo als auch mit 5.1 arbeiten.

`lc`	LC (Low Complexity).
he	HE (High Efficiency) = LC + Spectral Band Replication (SBR)
hev2	HE v2 = LC + SBR + Parametric Stereo (PS)

Wird die Option weggelassen, verwendet Nero eine von der Qualität bzw. Bitrate und Kanalanzahl abhängige Automatik, um das AAC-Profil einzustellen. Die folgende Tabelle zeigt die Grenzwerte, wie sie in Nero 1.1.34.2 gelten.

	`-vbr`	`-abr` und `-cbr`
Stereo-Zieldatei	Bis 0,15 HE v2, bis 0,30 HE, darüber LC.	Bis 39 kbit/s HE v2, bis 84 kbit/s HE, darüber LC.
5.1-Zieldatei	Bis 0,30 HE, darüber LC.	Bis 254 kbit/s HE, darüber LC.

-6chnew

Veranlasst Nero, eine 6-Kanal-AAC zu erzeugen. Ohne diesen Parameter wird Stereo angenommen. Nero richtet sich nicht nach der Kanalanzahl der Quelle, sondern ausschließlich nach diesem Schalter.

-hintforstreaming

Fügt der Zieldatei für Streaming benötigtes Hinting hinzu.

4.8 Der Abschnitt -dimzon() für Winamp AAC

Syntax

```
-dimzon( -dllname WA_aac.dll -cbr [-lc] [-high] [-mode]
[-mpeg2] [-pns] [-6chnew] )
```

Verwendet den Winamp-Encoder, um AAC-Audio zu erstellen. Die Zieldatei ist ein roher AAC-Datenstrom, d. h. nicht in einen MP4-Container verpackt. Winamp unterstützt ausschließlich CBR-Encoding. Die Option `-dllname WA_aac.dll` ist zwingend notwendig, um den Winamp-Encoder zu aktivieren.

-cbr <kbps>

Definiert die Bitrate der Zieldatei. Gültige Werte sind ganze Zahlen. Minimum und Maximum hängen entsprechend der folgenden Tabelle von Encodingmodus und Kanalanzahl ab.

Modus	Stereo	6-Kanal
LC	96–256 kbit/s	160–320 kbit/s
HE	16–256 kbit/s	96–213 kbit/s
HE High	96–256 kbit/s	nur Stereo
HE + PS	16–48 kbit/s	nur Stereo

-lc

Aktiviert das AAC-Profil *Low Complexity* (LC). Ohne diesen Schalter encodiert Winamp im HE-Modus (High Efficiency).

-high

Verwendet eine für hohe Bitraten optimierte Variante des HE-AAC-Modus. Arbeitet nur mit Stereo.

-mode {1|2|3|4|5}

Definiert den Encoding-Modus für Mono- und Stereo-Zieldateien.

1 Mono.
2 Joint Stereo.
3 Normales Stereo.
4 Parametric Stereo (PS). Nur in Verbindung mit HE-AAC.
5 Dual Channel.

-mpeg2

Erzeugt einen MPEG-2-AAC-Bitstream. Ohne diesen Schalter wird MPEG-4-AAC verwendet.

-pns

Aktiviert Perceptual Noise Substitution (PNS).

-6chnew

Veranlasst Winamp, eine 6-Kanal-AAC zu erzeugen. Ohne diesen Parameter wird Stereo angenommen. Winamp richtet sich nicht nach der Kanalanzahl der Quelle, sondern ausschließlich nach diesem Schalter.