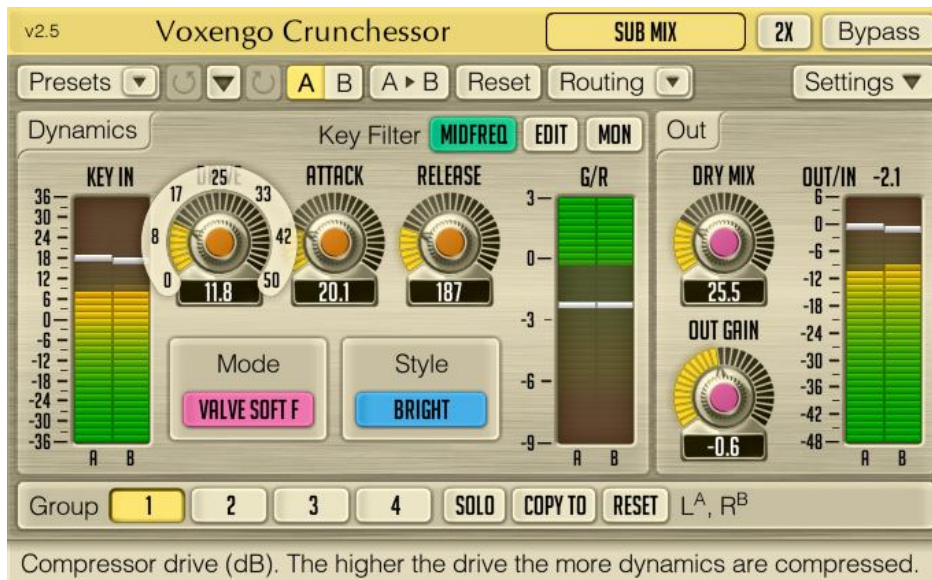




Voxengo Crunchessor User Guide



Software version 2.5

<http://www.voxengo.com/>

Contents

Introduction 3

 Features 3

 Compatibility 3

User Interface Elements 4

 Dynamics 4

 Mode 4

 Style 4

 Key Signal Filter 5

 Output 5

 Level Meters 5

Credits 6

 Beta-Testers 6

Questions and Answers 7

Introduction

Crunchessor is a general-purpose track compressor for professional audio production applications. One of its main advantages is the ease of tuning, which at the same time instantly delivers an excellent sonic performance. Another remarkable feature of Crunchessor is its valve-type processing, which is applied in parallel to compression. This makes Crunchessor an ideal choice for musicians and producers who are fond of analog compression sound and its warmth.

Beside this, Crunchessor offers you a selection of compression coloration styles--by itself a pretty unique feature for a compressor. By choosing an appropriate coloration style, you can add the character most suitable for the sound material you are working with. As an option, you can disable both coloration and valve warming processing to get a neutral and clean compression sound.

Crunchessor also accepts external side-chain source signals and allows you to filter key signal sources before they are used for compression.

Being a general-purpose compressor, Crunchessor can be used for any kind of sound material: drums, vocals, bass, guitars, synth sounds, etc. Crunchessor can be suitable both for single track and whole mix processing. Full multi-channel operation is supported.

Features

- Easy compression tuning
- Valve warmth
- Coloration styles
- External side-chaining
- Key signal filtering
- Multi-channel processing
- Internal channel routing
- Channel grouping
- Mid/side processing
- Up to 8x oversampling
- 64-bit floating point processing
- Preset manager
- Undo/redo history
- A/B comparisons
- Contextual hint messages

Compatibility

This plug-in can be loaded into any audio host application that conforms to the VST or AudioUnit plug-in specification.

This plug-in is compatible with Windows XP, Windows Vista (32- and 64-bit) and Mac OS X 10.4.11 and above (Intel and PowerPC) computers (2 GHz dual-core or faster processor with at least 1 GB of system RAM recommended). A separate binary distribution file is available for each target computer platform for each plug-in format.

User Interface Elements

Note: Most interface elements (buttons, labels) located on the top of the user interface and on the bottom are standard among all Voxengo plug-ins and do not require much learning effort. For an in-depth description of these and other standard user interface elements and features please refer to the “Voxengo Primary User Guide”. Learned once it will allow you to feel comfortable with all pro audio plug-ins from Voxengo.

Dynamics

The “Drive” parameter controls the amount of compression being applied to the incoming signal.

The “Attack” and “Release” parameters control the time transfer function of the compressor. Higher attack times allow transients to pass through while lower attack times compress transients better.

Mode

The “Mode” switch allows you to choose between several valve-type and clean processing modes. The “Valve” processing modes apply a great deal of warmth during compression, and doing so they can adjust the original tone of the sound.

The “Clean” mode gives a more neutral sound, which can be particularly useful for mastering applications.

The “Crisp” modes offer a brighter sound in comparison to the “Valve” modes. “Crisp” modes also feature a distinctive attack and release dynamic response.

Both “Valve” and “Crisp” modes come in varying “strength”. The “F” and “S” postfixes denote “fast” and “slow” attack modes, respectively. While both “Crisp” and “Valve” modes produce a transient saturation, the “Crisp” modes are much lighter sounding than the “Valve” modes are.

All modes except “Crisp S” modes additionally feature a slight program-dependent attack behavior.

Style

The “Style” selector gives you an opportunity to select an appropriate sonic character (coloration) of the compressor. The selected “character” is most evident on the transient moments of the sound, i.e. when the compressor actually starts to work (reduces gain). Note that generally this is a very subtle control, so if in doubt you may simply leave it at “Neutral”. Here is a list of all “Style” options and what kind of filtering they apply:

- Neutral: no additional filtering.
- Clear: up to -3dB of gain at 1kHz, up to 3dB of gain at 64Hz.
- Bright: up to 3dB of gain at 10kHz.
- Dark: up to -3dB of gain at 8kHz, up to -3dB of gain at 9kHz.

- Lo-Pump: up to 6dB of gain at 64Hz.
- Punch: up to 3dB of gain at 5kHz, up to 3dB of gain at 64Hz.
- Smooth: up to 3dB of gain at 1kHz, up to -6dB of gain at 10kHz.
- Snappy: up to 6dB of gain at 2.2kHz, up to 6dB of gain at 6kHz.
- Warm: up to 4dB of gain at 1kHz.

Key Signal Filter

The “Key Filter” switch enables the key signal filtering. When this switch is enabled, the key signal (be it external side-chain or not) used for compression characteristic calculation will be internally processed by a filter. Filter options are:

- Off: no filtering.
- BDrum: low-pass filter which filters everything out above 128 Hz.
- HiFreq: filter which increases power of higher frequencies a lot.
- HiHat: high-pass filter which filters everything out below 6 kHz.
- LoFreq: filter which increases power of lower frequencies a lot.
- MidFreq: a wide parametric filter centered around 1 kHz, having 6 dB of gain. Such emphasis applied on the center of the audible frequency spectrum can be beneficial when one wants to reduce various warbling effects caused by the extremely powerful transients in the high frequency end. During such moments the higher frequencies start to “duck” the lower frequencies causing an unstable “warbling” sound in the lower end. Enabling the Filter is most useful for wide-band signals like complete mixes.
- SDrum: band-pass filter with center around 1.5 kHz.

By using the “Mon” switch you may enable filtered key signal monitoring. The “Edit” button opens the “Key Filter Editor” window where you can precisely adjust the filtering parameters, and store key filter presets if necessary.

Output

The “Dry Mix” parameter specifies amount of original unprocessed signal being routed to the output before the final output gain is applied. By increasing this parameter you may reduce the overall distortion: hence one of the tactics when using this plug-in can be getting a high distortion amount at first and then adjusting the “Dry Mix” parameter to achieve a required proportion between the clean and processed signals. In other terms, this parameter engages “parallel compression”.

The “Out Gain” parameter changes overall output signal level of the plug-in. This parameter also affects the level of the key signal when it is being monitored.

Level Meters

Crunchessor features three RMS level meters, with scales shown in deciBel. Peak level indication is present on all meters. The “Key In” meter estimates the level of the key signal used for compression characteristic calculation. The “G/R” is a gain reduction meter showing gain reduction changes relative to the 2-second average gain reduction. The “Out” meter shows plug-in’s master output level.

Credits

This plug-in was produced by Aleksey Vaneev in Syktyvkar, Komi Republic, Russia.

DSP algorithms and internal signal routing code were created by Aleksey Vaneev.

Graphics user interface code and the “standard” graphics design were created by Vladimir Stolypko.

Plug-in is implemented in multi-platform C++ code form and uses “zlib” compression library (written by Jean-loup Gailly and Mark Adler), filter design equations by Magnus Jonsson and Robert Bristow-Johnson, VST plug-in technology by Steinberg, AudioUnit plug-in SDK by Apple, Inc. (used under the corresponding licenses granted by these parties).

Voxengo Crunchessor Copyright © 2004, 2009 Aleksey Vaneev.

Beta-Testers

Crunchessor was beta-tested by these persons:

Ben Williams

Jay Key

Laurent Bergman

Mads Ljungdahl

Matthew Fagg

Michael Anthony

Murray McDowall

Niklas Silen

Steffen Bluemm

Questions and Answers

Q. If I wanted to do some low level compression with Crunchessor, for example, I want a 1.5:1 down to -30 dB. It seems impossible with this plug as the more drive I add the more ratio I get. I was wondering if it would be possible to add threshold and compression ratio controls in one of the future versions?

A. Crunchessor uses a kind of logarithmic scale for compression – it is hardly possible to define the 1.5:1 ratio precisely. Crunchessor was designed to be as simple at operation as possible, so any special threshold and ratio controls are not planned for implementation.

Q. How much version 2 of Crunchessor is different from version 1, sound-wise?

A. Version 2 of Crunchessor fully implements compression algorithm of version 1. However, version 2 features a lot of “fine-tuning” applied to this algorithm, so in some cases version 2 algorithm may perform differently and usually better than version 1. It should be also noted that preset format of version 2 is incompatible with version 1 presets, but both version 1 and version 2 plug-ins can be used in the same project without conflicting each other, allowing for a smooth transition from version 1 to version 2.

Q. Unless I missed it somewhere, I don't see any documentation with Crunchessor 2.0 explaining how to setup the side-chaining routing.

A. Please read the “Voxengo Primary User Guide” for more information on how to setup side-chaining in Voxengo plug-ins.

Q. What would you suggest for a setting on orchestration mixes (for the final post-processing – can I use Crunchessor effectively?) Because it is orchestral it needs to be transparent.

A. Yes, of course, in many cases you may use Crunchessor for final post-processing. You may use Crunchessor's “Clean” mode coupled with the “Flat” coloration style as a starting point. You may get a pretty transparent sound if you use a long attack (above 25 milliseconds) and a long release (above 250 milliseconds) times.

However, compression may not be the best process to apply on orchestrations as it shrinks music material's dynamic range which is important for orchestral music.

Happy Mixing!