Voxengo CRTIV Shumovick User Guide

Version 1.2

https://www.voxengo.com/product/crtivshumovick/

Contents

Introduction

CRTIV Shumovick plug-in for professional music production applications produces a creative dynamic noise-floor padding effect. This effect is most effective on beats and synth sounds used in electronic music production – EDM, hip-hop, and many others. The noise-floor effect created by this plug-in is correlated to the spectral content of the sound being processed.

As a result, on beats, this creates mellow noise bursts that make bassdrum sounds a lot softer, with a bit of stereo-field enhancement effect. On synths, on the other hand, such noise-floor effect may boost the power and density of the sound. Shumovick can be also effective when used on sterile-sounding basslines where it can add a bit of "pillowy" noise background. In practice, Shumovick can be also used on acoustic drum buses adding "vintage" feel and aggression to them. Shumovick is, however, not as effective when used on full mixes, but can be also used on full mixes in some music genres.

CRTIV Shumovick is a relatively CPU-demanding effect plug-in, so it requires a higher-end computer processor for a comfortable use. However, the CPU requirements can be lowered by adjusting plug-in's processing band count.

Features

- Noise-floor signal envelope controls
- Noise-floor signal equalization
- Selectable processing band count
- 64-bit floating point processing
- Preset manager
- Undo/redo history
- A/B comparisons
- Contextual hint messages
- All sample rates support
- Zero processing latency

Compatibility

This audio plug-in can be loaded into any audio host application that conforms to the AAX, AudioUnit, VST or VST3 plug-in specification.

This plug-in is compatible with Windows (32- and 64-bit Windows XP, Vista, 7, 8, 10 and later versions, if not announced otherwise) and macOS (10.11 and later versions, if not announced otherwise, 64-bit Intel processor-based) computers (2.5 GHz dual-core or faster processor with at least 4 GB of system RAM required). A separate binary distribution file is available for each target computer platform and audio plug-in specification.

User Interface Elements

Note: All Voxengo plug-ins feature a highly consistent user interface. Most interface elements (buttons, labels) located at the top of the user interface are the same in all Voxengo plug-ins. For an in-depth description of these and other standard features, and user interface elements, please refer to the "Voxengo Primary User Guide".

Envelope

This group of knobs affects envelope characteristics used to shape the internal noise signal sent to output. The algorithm derives envelope from the input spectral signal.

The "Attack" parameter specifies the attack time of the envelope signal, in milliseconds. Lower values produce an envelope with sharp transients – useful to react on high-hats, for example. Higher values reduce transient response, this can be useful when smooth noise "padding" is suitable for the source signal.

The "Release" parameter specifies the release time of the envelope signal, in milliseconds. Lower values produce an envelope with fast fall time, with a bit of "stuttering" flavor. Higher values produce a longer fall time, but this may have an effect of "noise overcrowding".

The "Dynamics" parameter adjusts the dynamic response of the envelope signal. This parameter works as either a compressor of dynamics (lower values) or an expander of dynamics (higher values).

Noise

This group of knobs affects noise signal's spectral shape. These controls work like a Baxandall equalizer.

The "Lo Gain" parameter controls the gain of lower frequencies of the noise signal.

The "Hi Gain" parameter controls the gain of higher frequencies of the noise signal. Lowering this value usually allows you to get a smoother-sounding result. The "+3/0/-3" selector chooses between different spectral slopes (in decibel per octave), affecting the "brightness" of the noise signal.

The "Bands" parameter controls the number of bands the noise signal is split into. This parameter affects both the CPU resources required for effect processing and the overall sonic quality of the produced noise signal. Note that this parameter also slightly affects loudness of the noise in a non-linear way.

Levels

The "Wet Gain" parameter controls the loudness of the noise signal.

The "Dry Gain" parameter controls the loudness of the dry input signal. The "mute" switch can be used to disable the dry signal in the output mix. Muting the dry signal permits use of the plug-in as a "send" effect.

Meter

This block displays the output signal level in decibel. “OL” indicator reacts on signal levels above 0 dBFS.

Credits

DSP algorithms, internal signal routing code, user interface layout by Aleksey Vaneev.

Graphics user interface code by Vladimir Stolypko. Graphics elements by Vladimir Stolypko and Scott Kane.

This plug-in is implemented in multi-platform C++ code form and uses “zlib” compression library (written by Jean-loup Gailly and Mark Adler), “LZ4” compression library by Yann Collet, filter design equations by Magnus Jonsson and Robert Bristow-Johnson, VST plug-in technology by Steinberg, AudioUnit plug-in SDK by Apple, Inc., AAX plug-in SDK by Avid Technology, Inc., Intel IPP and run-time library by Intel Corporation (used under the corresponding licenses granted by these parties).