

# SONICCHARGE BitSpeak

version 1.6.1

---

## Table of Contents

---

Introduction.....	3
User Interface .....	4
Bitspeek Versus Vocoders.....	6
Using Bitspeek in your Host .....	6
Requirements .....	9
Change History.....	9
Credits and Contacts .....	10
Copyrights And Trademarks .....	11

# Introduction

## INTRODUCTION

Bitspeek is a real-time pitch-excited linear prediction codec effect. Right now, you are probably thinking, “oh, another one of those”? Or perhaps not. Chances are that you have never heard about “linear prediction”, although most of us use it daily when we talk on our cell phones. Linear prediction coding is a voice compression technology that appeared in commercial products in the seventies and was implemented in some well-known speaking toys of the early eighties.

We have applied this technology to create a VST/AU effect plug-in that analyzes audio, extracts parameters, like pitch, volume, and formant data, and then resynthesizes the audio using a simple oscillator, noise, and filter architecture.

Ever heard the robotic voice in Microtonic that reminds you to purchase? That is an example of what Bitspeek can sound like. But there is more to it. We have added playback parameters that adjust the pitch and tonal quality of the sound as well as support for MIDI and a beat-synchronized “formant freezing effect”. Despite having only a few simple controls, this box can produce a broad range of sounds from cheap speaking toys to high-end vocoder and talkbox effects.

/ Magnus Lidström

## User Interface

---



### Rate (kHz)

**Bitspeak** performs its calculations at a designated fixed sample-rate, regardless of the sample-rate your project is running at. The possible settings are 8 kHz, 11 kHz, 22 kHz, and 44 -> 48 kHz (*the last mode will adjust to the project sample-rate and select a rate between 44 and 48 kHz*). Notice that **Bitspeak** is still compatible with any host sample-rate by performing an automatic sample-rate conversion. The Rate choice affects several other internal parameters in the DSP algorithms, and changing Rate will change the sound dramatically. (*Notice that the 44 kHz mode may require a lot of CPU. In many cases, the 22 kHz mode works just as well.*)

### Frame Rate

The audio signal is analyzed and processed in blocks called “frames”. For each frame, **Bitspeak** estimates the pitch, volume, and formants of the incoming audio, as well as the balance between “voiced” audio (*e.g., vowels*) and “voiceless” (*e.g., the noise in a consonant*). By lowering the frame rate, the analysis will be performed more rarely, and you will achieve a cheaper toy-like sound. You can also “freeze” the audio by dragging Frame Rate down to 0. The parameter range is 0 to 80 Hz (*if Sync is off*), and higher rates require more CPU than lower ones.

### Sync

Enable Sync to make **Bitspeak** “freeze” frames in sync with the tempo of your music. When Sync is enabled, you may select various time-synchronized rates (*1/8, 1/16, etc.*) with the Frame Rate slider.

## MIDI

Enable MIDI to control the pitch and envelope of the effect via MIDI. You will need a host that is capable of transmitting MIDI to effect plug-ins to use this feature. Please read the documentation for your host for information on how to set it up. When enabled, **Bitspeak** will only play when it receives MIDI, and it will transpose the pitch according to the MIDI notes it receives. Turn down the Tracking parameter to zero to achieve a vocoder / auto-tune like sound. If you set the Frame Rate to zero (*with Sync disabled*), **Bitspeak** will “freeze” the formants on MIDI note on, allowing you to create interesting “stroboscopic” audio effects. Finally, **Bitspeak** supports Pitch Wheel messages (*one octave up and down*), and the MIDI Sustain Pedal can also be used to “freeze” frames while playing.

## Pitch

You can transpose the outgoing audio by -36 to +36 semitones (*-3 to +3 octaves*). Hold down the shift-key while turning the knob to make finer adjustments down to a single cent in precision.

## Tracking

Determines how the source signal pitch affects the synthesized audio, from 0% to 200%. At 100%, the processed audio will follow the original audio pitch intonation as precisely as possible. (*Sometimes though, the tracking detects the wrong octave, especially on source material with an extremely low pitch.*) At 0%, the pitch will stay fixed and produce a robotic vocoder-like quality.

## Detune

There is a second oscillator which can be used to achieve a fat detuned sound or for chord-like effects. The second oscillator is transposed from the first by +0 to +1200 cents, representing a range of up to one octave.

## Noise

This parameter adjusts the balance of “voiced” vs. “voiceless” sound. At the default setting +/- 0%, **Bitspeak** attempts to follow the balance of the source signal, so that “voiced” sounds (*like vowels*) produce distinct tones while “voiceless” sounds (*such as consonants*) produce noise. By turning Noise all the way down to -100%, all noise will be removed from the output audio. By turning Noise up to +100%, the output audio will consist only of filtered noise (*sounding like a loud whisper*).

## Mix

A simple dry/wet mix control. The dry signal is “latency compensated” to be in phase with the wet signal.

## Legacy Mode

Version 1.5 of **Bitspeak** features a vastly improved volume tracking algorithm and introduces support for stereophonic processing. For compatibility reasons, you can turn these new algorithms on and off by clicking the LEGACY MODE text in the top right corner of **Bitspeak**’s display. *The improvements in version 1.5 are most apparent on noisy material with sharp transients, like drums.*

## Context Menu

You can right-click (or control-click on Mac) anywhere in the user interface to bring up a menu with some standard functions like undo, redo, copy, paste, zoom the interface, etc.

## Bitspeek Versus Vocoders

---

Vocoder is short for “voice encoder”, and technically that means any device that is capable of breaking down a voice signal into a set of parameters from which you can reconstruct the voice intelligibly. With this terminology, **Bitspeek** may be called an “LPC vocoder”. (*LPC is short for “linear prediction coding”.*)

However, in musical contexts, the vocoder we all know is a device consisting of a filter bank controlled from a modulation source (*e.g., speech*). This filter bank is fed with an arbitrary carrier signal (*typically from a synthesizer*) that is played independently of the modulation source.

This method is very different from how **Bitspeek** works. In **Bitspeek**, a built-in synthesizer tracks and follows the fundamental frequency, volume and noise level of its input. The synthesized tone is passed into a formant filter that is not implemented as a bank of bandpass filters like in conventional vocoders. Instead, the analysis is divided into time frames (*typically around 10ms each*). For each frame, the algorithm quickly constructs a resonant filter that closely represents the formants of the input.

In other words, whereas conventional vocoders work with several bandpass filters at stationary frequencies, **Bitspeek** creates animated filters whose peaks are precisely positioned in the spectrum. On the other hand, the signal is chopped up in discrete time frames, whereas filter bank vocoders vary formants smoothly over time.

## Using Bitspeek in your Host

---

When not running in “Legacy Mode”, **Bitspeek** works with stereophonic sound. It analyzes the stereo image of the source signal and extracts two parameters: panning and stereo width (*corresponding to the amplitude balance and the correlation of left and right signals*). **Bitspeek** will then attempt to mimic the stereo image with the built-in synthesizer.

The oscillator is only monophonic (*but panned*), *whereas* the noise is stereophonic and copies the stereo width from the source. This solution opens up for some interesting pseudo-reverb effects.

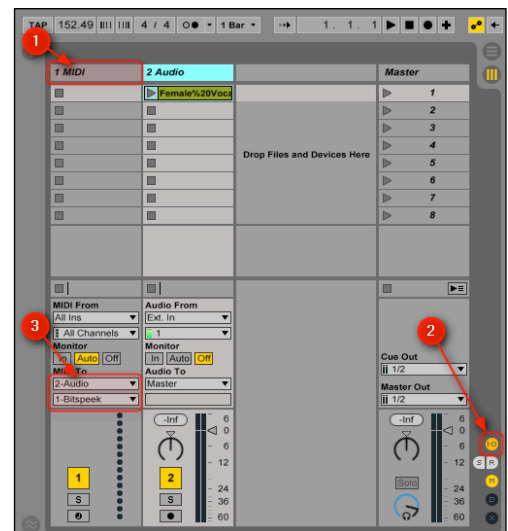
**Bitspeek** requires “lookahead” and adds a latency of around 13 ms. Most modern hosts compensate for this latency during playback.

Playing the pitch of **Bitspeek** with MIDI notes is great fun, and most hosts support routing of MIDI effects. Here are a few quick instructions on how to set things up in some popular hosts. In all examples, MIDI is set to “On” and Tracking to 0% in the **Bitspeek** interface.

## Ableton Live

(Add Bitspeek to the effect chain on an audio or instrument track. Then turn the MIDI switch to “On” and Tracking down to 0% in the Bitspeek interface.)

1. Create a MIDI track.
2. Bring up the I-O parameters if they are hidden.
3. Assign “MIDI To” to the audio track that contains **Bitspeek** and make sure “MIDI To” is assigned to the **Bitspeek** effect and nothing else.



## Apple Logic Pro X

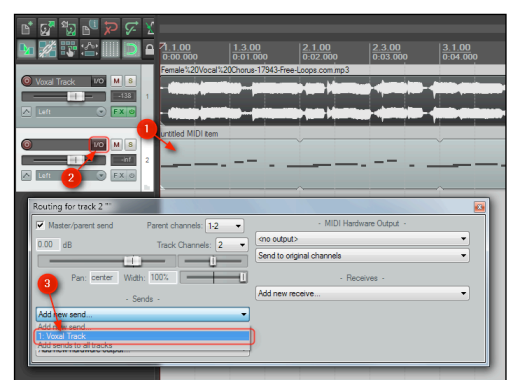
1. Create a new instrument track.
2. Click the “Plug-In” button and select **Bitspeek** under MIDI-controlled Effects. (Turn the MIDI switch “On” in Bitspeek and turn down Tracking to 0%.)
3. Select your audio track from the Side Chain menu in the top right corner of the plug-in window.
4. You can mute the audio track output since it is now already passing through the instrument track.



## Cockos Reaper

(Add Bitspeek to the effect chain on an audio or instrument track. Then turn the MIDI switch to “On” and Tracking down to 0% in the Bitspeek interface.)

1. Insert a new track and then add a “New MIDI item”.
2. Click the “I/O” button for the MIDI track.
3. In the Routing window, choose to “Add new send...” and select the track with the **Bitspeek** effect you wish to control.



## Steinberg Cubase

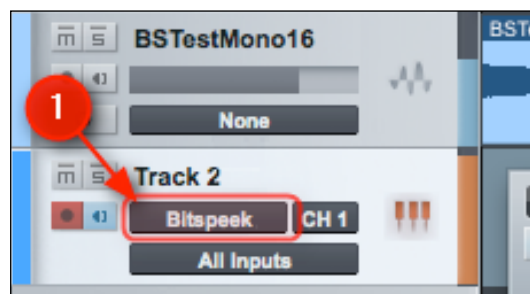
(Add Bitspeek to the effect chain on an audio or instrument track. Then turn the MIDI switch to “On” and Tracking down to 0% in the Bitspeek interface.)

1. Create a new MIDI track.
2. Select **Bitspeek** as MIDI destination for the new track.

## PreSonus Studio One

(Add Bitspeek to the effect chain on an audio or instrument track. Then turn the MIDI switch to “On” and Tracking down to 0% in the Bitspeek interface.)

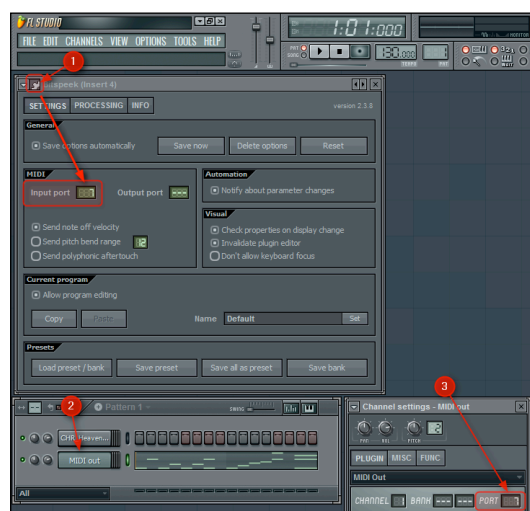
1. Add an Instrument track and select **Bitspeek** as the destination for the new track.



## Image-Line FL Studio

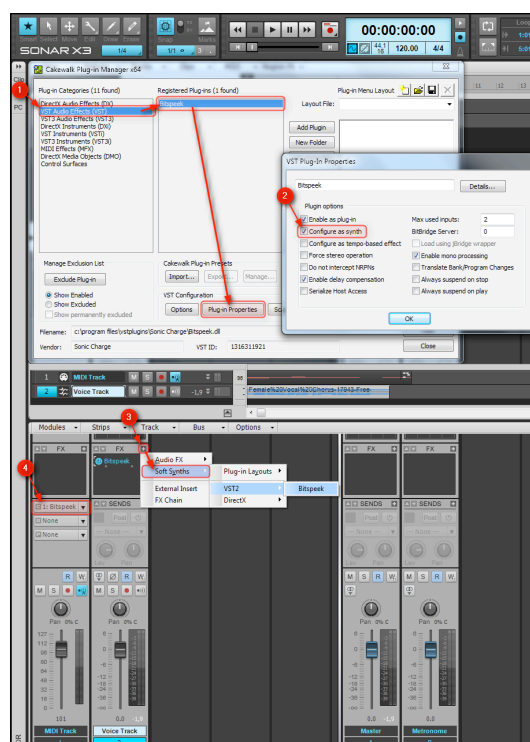
(Add Bitspeek to the effect chain on an audio or instrument track. Then turn the MIDI switch to “On” and Tracking down to 0% in the Bitspeek interface.)

1. Select a free input port under the MIDI section in the plug-in settings.
2. Add a “MIDI Out” channel.
3. In the channel setting, select the same port number as you did for **Bitspeek**.



## Cakewalk

1. Enter the Cakewalk Plug-in Manager, select **Bitspeek**, and click “Plug-in Properties”.
2. Turn on “Configure as synth” and click OK. **Bitspeek** should now show up under VST Instruments (VSTi).
3. Insert **Bitspeek** in the FX chain as a “Soft Synth” instead of an “Audio FX”. (Turn the MIDI switch “On” in Bitspeek and turn down Tracking to 0%.)
4. Insert a MIDI track and select **Bitspeek** as output for the new track.



# Requirements

---

The minimum requirements for installing and running **Bitspeek** are:

- Microsoft Windows XP or OS X 10.9 — Mavericks
- A host that supports 32 or 64-bit VST 2.4 or AudioUnit 2 plug-ins
- 2GHz Pentium IV or equivalent
- 20MB of free disk space

# Change History

---

## Version 1.6.1 (2020-08-24)

- Added support for time-limited licensing.
- Made a workaround to handle a rare Windows problem with generating a unique machine-id.

## Version 1.6 (2020-03-04)

- Scalable GUI and retina support.
- New algorithm for the “system unique identifier” used for authorization. Hopefully fixing the problem where the plug-in became unregistered spontaneously.
- (Mac) Solved a compatibility problem with DAWs that are built with recent Apple SDK's, e.g. Cubase 10.5.
- (Mac) Notarized installer for Catalina.
- (Mac) New 64-bit compatible uninstaller.
- (Mac) 64-bit Audio Unit no longer depends on the “Component Manager”. This means you should not need to restart after installation.
- (Mac) Preferences and registration data is now shared with “sandboxed” DAWs like GarageBand (*meaning Authenticator works with these DAWs too*).
- (Mac) Fixed a problem where under certain conditions the preferences data could stay locked if the DAW crashed, requiring a full system restart.
- Lots of other minor bug and compatibility fixes.

## Version 1.5 (2014-12-16)

- Vastly improved volume tracking algorithm (*old patches open with legacy mode enabled for backwards compatibility*).
- Stereophonic processing.
- Dry / wet mix knob.
- New skin.
- Rebuilt GUI from scratch using our latest in-house framework (*giving you features like undo/redo, copy/paste, and more*).

- Right-click context-menu on knobs and sliders to set exact values with text.
- Recalls last used program(s) when creating a new instance.
- Changed trial to be fully functional for 3 weeks instead of having regular audio dropouts.
- Supports **Sonic Charge Authenticator** for easier registration.
- Many minor bug-fixes.

## **Version 1.0.2 (2011-10-07)**

- 64-bit support.
- Changed to stereo I/O configuration (although the effect is monophonic)
- Improved compatibility with older VST 2.3 hosts and wrappers.
- Many other minor compatibility improvements.

## **Version 1.0.1 (2011-01-01)**

- Solved a problem that prevented registration from working if you had not installed **MicroTonic** or **Synplant** before **Bitspeak**.

## **Credits and Contacts**

---

**Sonic Charge Bitspeak** v1.0 - v1.6.1 (2011 - 2020)

Created by:

Magnus Lidström

Graphical design and additional development:

Fredrik Lidström

**Sonic Charge** website:

<https://soniccharge.com>

---

## Copyrights And Trademarks

---

The **Sonic Charge Bitspeek** software and documentation are owned and copyright by **NuEdge Development** 2011-2020, all rights reserved.

Symbiosis version 1.2 - 1.4, Copyright © 2010-2019, NuEdge Development / Magnus Lidström. All rights reserved. Released under the “New Simplified BSD License”.

libpng versions 1.2.6 - 1.6.35, Copyright © 2004-2018 Glenn Randers-Pehrson.

zlib version 1.2.3 - 1.2.11, Copyright © 1995-2017 Jean-loup Gailly and Mark Adler.

The Steinberg VST PlugIn SDK are copyright Steinberg Soft- und Hardware GmbH. VST is a trademark of Steinberg Soft- und Hardware GmbH.

Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

Mac, Mac OS X, OS X, and macOS are trademarks of Apple Computer, Inc.

**Bitspeek** software and documentation are protected by Swedish copyright laws and international treaty provisions. You may not remove the copyright notice from any copy of **Bitspeek**.

Please, read the end-user license agreement enclosed in the package for a lot more legal mumbo-jumbo.

The contractor/manufacture for **Sonic Charge Bitspeek** is:

NuEdge Development / Magnus Lidström  
Högbergsgatan 16  
S-116 20 Stockholm  
Sweden