

# SimSynth

Copyright© 1994 by David Billen

## Introduction and overview

This is a general introduction to SimSynth. For technical reference, see the help file which is accessible from within the program. If you do not know how to program an analog synthesizer, and would like to learn, see the HOWTO.WRI file...

### What is SimSynth?

**SimSynth is a sound synthesizer. It allows you to create new sounds. SimSynth does for sound what a paint program does for graphics. SimSynth generates RIFF wave format files, (.WAV files), which can be played from a Windows™ compatible wave device, or converted for use in DOS applications.**

### System Requirements

#### Minimum requirements:

- Microsoft Windows version 3.1 or above
- 80386 (or better) processor
- 4+ megabytes of RAM
- Math co-processor
- Sound card

#### Recommended:

- 80486 50mhz (or better) processor
- 8 megabytes of RAM
- Stereo sound card

### A note to beginners

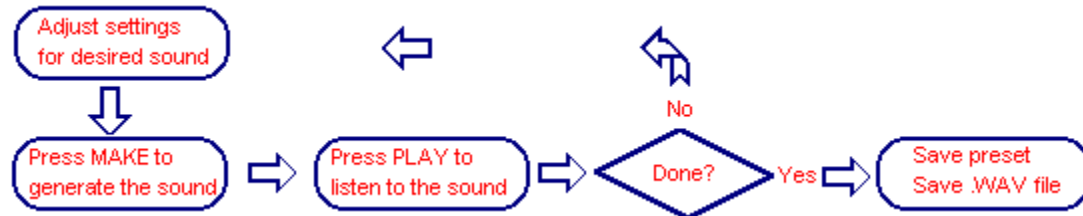
If you are not familiar with the basics of sound synthesis, and are interested in learning, see the "HOWTO.WRI" file. That document presents a general introduction to synthesizer programming.

You should know that SimSynth isn't nearly as difficult to use as a first glance at the SimSynth screen might imply. There are quite a few parameters to adjust, but once you know what each one does, you've pretty well learned it all!

## How to use SimSynth

First, you need to know what all of the parameters do. See the notes for beginner and experienced synth programmers in this document. Then see the parameter reference for details.

Using SimSynth is a matter of adjusting the parameters to configure the sound you wish to create. Then pressing the MAKE button to generate the sound, and the PLAY button to test it.



SimSynth is divided into six sections:

**OSC:** This section is used to configure up to three oscillators and the LFO, (which is mostly programmed the same as a *high* frequency oscillator). These can be programmed like traditional analog oscillators, or, you can configure them to play a .WAV file.

**SVF:** This section is used to configure the filter.

**ENV:** This section is used to program up to three envelopes. It is also the means of controlling the amplitude and duration of the sound.

**EFX:** This section allows the optional selection of a stereo effect.

**GEN:** This section has a few options to specify how the sound should be generated. Most importantly, it contains a large button labeled "MAKE". The MAKE button generates the sound according to the parameter settings in all of the sections. In fact, *no changes to the sound are heard unless you regenerate the sound by pressing the MAKE button!* This section also contains load and save buttons for loading and saving a preset.

**The output section:** This section configures the output. It's parameters are regenerated when the MAKE button is pressed, but are not saved with a preset. It contains options such as output sample rate, stereo/mono, etc. Most importantly, it contains a large button labeled "PLAY". The PLAY button plays the last sound generated with the MAKE button. Note: *if you change parameters and then press PLAY, you do not hear the changes. MAKE must be pressed to first regenerate the sound!* This section also contains a save button to save the currently generated sound as a .WAV file.

If you are using the DEMO version of SimSynth, the save button is not enabled. You can experiment with SimSynth freely, and even save presets, but when you are ready to create .WAV files, you need to order the "real" version.

## **Note for experienced synthesizer programmers**

See the PARAM.HLP file, (accessible from within the program), for a parameter by parameter reference.

The analog synthesizer metaphor was held to pretty firmly, but there are a couple of digital features thrown in. In particular, see the parameter reference for a discussion of pulse width, frequency control over LFO and NOISE, and .WAV oscillators.

SimSynth uses a State Variable Filter (SVF). This type of filter was popular on older Oberheim™ synthesizers. A SVF produces low pass, band pass, high pass, and notch output, all at the same time. A single cutoff frequency, and emphasis setting controls them all. As is traditional, the output mix is controlled with two sliders. One is labeled "L" on one end, and "H" on the other. It fades between low pass output, ("L"), and high pass output, ("H"). Setting this slider to center selects notch output. The other slider is labelled "L/H" on one end, and "Band" on the other. This fades between the output of the other slider, and band pass output. Emphasis tightens the width of the band pass output.

If you're not familiar with a SVF filter, you might get started by setting both sliders to the left, giving only low pass filter. This is what most sounds require anyway. You can experiment later.

## **Quality issues**

One of the advantages in using SimSynth is the high quality waves it produces. Bypassing the sampling process eliminates a great deal of noise and distortions, as well as optimizing the amplitude for the best possible signal to noise ratio.

All sounds are generated internally as 16 bit, 44.1 kHz two channel sounds. If the settings in the output section specify a lower sample rate, the wave is reduced in size by averaging. The result is a wave which is "oversampled". This goes a long way to reducing the type of aliasing distortion which can result from violating Nyquist's theorem, (which states that the sample rate should be double the highest frequency sampled). But it can't work miracles. If you create a sound which contains frequencies above 5.5125 kHz, and set the output rate to 11.025 kHz, the result will at best have the excessive frequencies removed by the oversampling process. At worst there will be distortion. The only way to produce high quality sounds that are compatible with inexpensive sound cards, is to explicitly use low frequencies. (For game type effects, you can actually take advantage of the aliasing distortion! Who's to really say that a plasma resonance cannon doesn't make that zipper noise?)

8 bit waves are derived from the 16 bit generated waves by dividing each sample by 256. Many presets sound better if you enable the 16 bit setting, but mostly, 16 bit output is highly useful as an editing format.

Stereo comes only from the effects. If mono output is selected, and effects are used, the two channels are mixed into one. This can cause phase cancellation. If you are producing mono sound effects consider using a wave editor to achieve echo effects, etc.

## Questions and answers

**Q** I'm working on a preset, and I'm not getting any sound output. What's wrong?

**A** Check the frequency for the oscillator. A sine or triangle wave at too low a frequency is inaudible. Check that the filter cutoff is not out of audible range. Insure that you do not have all three oscillators muted. Also, pulse width set to an extreme, (i.e. 1% or 99%) for a square wave is sometimes inaudible.

**Q** I am using the echo effect, and would like to hear my echo. But the wave only lasts until the envelope expires. What can I do?

**A** The amp envelope determines the length of the sound. Set the level for the last amp envelope stage you are using to 1, instead of zero. Set the stage after that to the amount of extra time required, and set its level to zero. The last stage will be spent decaying from 1 to zero, which is silent, allowing time for the echo to fade off.

**Q** Is there anything I can do to speed up the sound generation?

**A** Any of the "modulation" parameters slow the generation, (these are the envelope or LFO track for oscillator or filter). Be careful that these sliders are set perfectly to zero when you are not using them, (rather than just knocking them down to some insignificant amount). Each oscillator slows the generation, (if you are using an additional oscillator to get an octave, use the "plus 1" switch instead.) Hard filter slows the generation considerably. Preamp to the filters and the Amp boost switch in the GEN section both using floating point math to correct "over drive". Using either of these slows the generation.

Use the "x2" (times 2) switch in the GEN setting when appropriate. This causes the sound to generate just as quickly, but play twice as long, (and deep). For low frequency sounds, there's little reason not to use it. For high frequency sounds, there is a loss of quality, so, there is a trade-off to consider.

**Q** I figured out that if I set the L/H (lowpass/highpass output mix) slider to zero, I get a notch filter. What use is a notch filter?

**A** One use is that it seems to simulate a great doppler effect when you sweep it with envelope or LFO.

**Q** I E-mailed a preset. When the recipient attempted to generate it, SimSynth was unable to access the .WAV file that I had used for an oscillator. Why?

**A** The data from a .WAV file which has been selected as an oscillator is not saved along with the preset. Only the file name is. If the .WAV file does not exist when the preset is loaded, SimSynth can't make the sound.

When SimSynth loads a preset which uses a .WAV oscillator, it first tries to find the file in the very same directory where it was selected. If it can't find it there, it tries to find the .WAV file in the same directory from which the preset was loaded.

The solution is to copy any .WAV files which are used as oscillators to the same directory in which you are keeping the preset that uses them. If you mail out the preset, be sure to send the needed .WAV files along with it.

Ditto for .OSC files.

**Q** I'm getting distortion. Any suggestions?

**A** Are you using lots of emphasis? Especially on square waves? If so, try using a hard filter. The hard filter causes the sound to be filtered a second time. This can clean up distortion caused by the emphasis.

Hard filters slow down the sound generation. It is sometimes desirable to use normal filter while testing a sound, then switch to hard filter only to generate the finished wave.

**Q** I'm getting "drop outs" in the sound. Any suggestions?

**A** Are you using detune with square waves? Detune adds a second, slightly sharp frequency to the oscillator. As the two resulting, out of phase wave forms occasionally meet at just the right spot, you get a brief moment of near perfect phase cancellation. The only solution is to adjust the pulse width a little. This slightly alters the tone, but it is the only answer.