

# **The SoundFont® 2.0 File Format**

## **A White Paper**

by Dave Rossum

Joint E-mu/Creative Technology Center

Copyright © 1995, 1997, E-mu Systems, Inc. All Rights Reserved.

## **Introduction**

In 1993, E-mu Systems realized the importance of establishing a single universal standard for downloadable sounds for sample based musical instruments. The sudden growth of the multimedia audio market had made such a standard necessary. E-mu's experience as a leader in sample based music synthesis led us to devise the SoundFont® file format standard as a solution.

The SoundFont file format was originally introduced with the Creative Technology Sound Blaster AWE32 product using the EMU8000 synthesizer engine. Since that introduction, E-mu and Creative have made evolutionary improvements in the SoundFont file format standard. Our resulting experience with the issues have given us the confidence to announce public disclosure of the SoundFont file format in its revision 2.0 embodiment.

## **A Brief History of Music Synthesis**

The electronic music synthesizer was invented simultaneously by a number of individuals in the early 1960's, most notably Robert Moog and Donald Buchla. The synthesizers of the 1960's and 1970's were primarily analog, although by the late 70's computer control was becoming popular.

With the advances in consumer electronics made possible by VLSI and digital signal processing (DSP), it became practical in the early 1980's to replace the fixed single cycle waveforms used in the sound producing oscillators of synthesizers with digitized waveforms. This development forked into two paths. The professional music community followed the line of "sample based music synthesizers," notably the Emulator line from E-mu Systems. These instruments contained large memories which reproduced an entire recording of a natural sound, transposed over the keyboard range and appropriately modulated by envelopes, filters and amplifiers. The low cost personal computer community instead followed the "wavetable" approach, using tiny memories and creating timbre changes on synthetic or computed sound by dynamically altering the stored waveform.

During the 1980's, another relatively low cost music synthesis technique using frequency modulation (FM) became popular first with the professional music community, later transferring to the PC. While FM was a low cost and highly versatile synthesis technology, it could not match the realism of sample based synthesis, and ultimately it was displaced by sample based approaches in professional studios.

During the same time frame, the Musical Instrument Digital Interface (MIDI) standard was devised and accepted throughout the professional music community as a standard for the real-time control of musical instrument performances. MIDI has since become a standard in the PC multimedia industry as well.

The professional sample based synthesizers expanded in their capabilities in the early 1990's, to include still more DSP. The declining cost of memory brought to the wavetable approach the ability to use sampled sounds, and soon wavetable technology and sample sound synthesis became synonymous. In the mid '90s wavetable synthesis became inexpensive enough to incorporate in mass market products. These wavetable synthesizer chips allow very good quality music synthesis at popular prices, and are currently available from a variety of vendors. While many of these chips operate from samples or wavetables stored in read only memory (ROM), a few allow the downloading of arbitrary samples into RAM memory.

## **What Is the SoundFont File Format?**

A SoundFont compatible file is, as the name implies, the audio equivalent of a character font. SoundFont compatible files are designed to present the information required to produce wavetable based musical instrument banks in a relatively implementation-independent format. They are also designed to present this information in a manner that is relatively compact and appropriately hierarchical.

The Musical Instrument Digital Interface (MIDI) language has become a standard in the PC industry for the representation of musical scores. MIDI allows for each line of a musical score to control a different instrument, called a preset. The General MIDI extension of the MIDI standard establishes a set of 128 presets corresponding to a number of commonly used musical instruments.

While General MIDI provides composers with a fixed set of instruments, it neither guarantees the nature or quality of the sounds those instruments produce, nor does it provide any method of obtaining any further variety in the basic sounds available. Various musical instrument manufacturers have produced extensions of General MIDI to allow for more variations on the set of presets. It should be clear, however, that the ultimate flexibility can only be obtained by the use of downloadable digital audio files for the basic samples.

The SoundFont file format differs from previous digital audio file formats in that files contain not only the digital audio data representing the musical instrument samples themselves, but also the synthesis information required to articulate this digital audio. A SoundFont compatible file represents a set of musical keyboards, each of which is associated with a MIDI preset. Each MIDI "preset" or keyboard of sound causes the digital audio playback of an appropriate sample contained within the SoundFont file. When this sound is triggered by the MIDI key-on command, it is also appropriately in a manner controlled by the MIDI parameters of note number, velocity, and the applicable continuous controllers. Much of the uniqueness of the SoundFont file format rests in the manner in which this articulation data is handled.

SoundFont compatible files are formatted using the "chunk" concepts of the standard Resource Interchange File Format (RIFF) used in the PC industry. Use of this standard format shell provides an easily understood hierarchical level to the SoundFont file format.

## **Issues in a Universal Synthesizer Data Format**

The General MIDI standard was an attempt to define the available instruments in a MIDI composition in such a way that composers could produce songs and have a reasonable expectation that the music would be acceptably reproduced on a variety of synthesis platforms. Clearly this was an ambitious goal; from the two

operator FM synthesis chips of the early PC synthesizers, through sampled sound and “wavetable” synthesizers and even “physical modeling” synthesis, a tremendous variety of technology and capability is spanned. The fact that many composers are disappointed in the results of the General MIDI standard is not surprising.

The task attempted by the SoundFont format is relatively simpler, but still by no means trivial. A SoundFont compatible file represents information to be loaded by a specific type of synthesizer technology - the sampled sound or modern “wavetable” synthesizer. Like General MIDI, the SoundFont format assumes only minimal basic capabilities of the synthesizer, but supports enhancements in an upwardly compatible manner. Most of the issues in the design of SoundFont compatible files are based around determining a format which can appropriately encapsulate minimal capabilities in a machine independent format, and yet allow for greater complexity as it becomes available.

Even something as seemingly straightforward as presenting the sample data itself is not a trivial issue. What resolution or word size(s) should be supported? Should data compression be employed, and if so, what method should be used? Are there any standards that must be followed by the samples themselves such that they can be reproduced with optimal fidelity on a variety of synthesis hardware platforms? How should the looping of samples be handled? Is there information unnecessary to the reproduction of the sound yet useful for future editing which should be carried? All of these questions must be considered in the determination of the digital audio format itself, which is the simplest portion of the SoundFont file format standard.

At the heart of the SoundFont file format is the hierarchical structure of the preset articulation data. When a musician presses a key on a MIDI musical instrument keyboard, a complex process is initiated. The key depression is simply encoded as a key number and “velocity” occurring at a particular instant in time. But there are a variety of other parameters which determine the nature of the sound produced. Each MIDI “channel” or keyboard of sound is associated at any instant to a particular bank and preset, which determines the nature of the note to be played. Furthermore, each MIDI channel also has a variety of parameters in the form of MIDI “continuous controllers” that may alter the sound in some manner. The sound designer who authored the particular preset determined how all of these factors should influence the sound to be made.

Sound designers use a variety of techniques to produce interesting timbres for their presets. Different keys may trigger entirely different sequences of events, both in terms of the synthesis parameters and the samples which are played. Two particularly notable techniques are called layering and multisampling. Multisampling provides for the assignment of a variety of digital samples to different keys within the same preset. Using layering, a single key depression can cause multiple samples to be played.

## **The Philosophy Behind the SoundFont Format**

The SoundFont format is designed to specifically address the concerns of wavetable (sampling) synthesis. The goals of the format are to be a general, extensible, and portable data interchange standard for reproduction on a variety of differing wavetable synthesis engines.

The SoundFont format is a file interchange format. While it is practical in many cases to navigate the data structures in real time, runtime considerations have been subsidiary to the other beneficial properties of the format.

Portability considerations have precluded any attempt to compress the data. The vast majority of data volume in a SoundFont compatible file is the digital audio data itself. This data does not easily lend itself to conventional lossless data compression schemes. Use of a lossy compression scheme, such as that used in MPEG and other “perceptually based” encoders, opens up difficult questions with respect to the fidelity of the data when reproduced by synthesis engines based on a variety of differing technologies. SoundFont files thus use conventional 16 bit linear coding for all sample data, which provides adequate fidelity for all users.

This philosophy of sacrificing data compactness to ensure portability and fidelity of the medium has been extended to the articulation data as well. The SoundFont format provides adequate resolution in all parameters for the most exacting use.

Generality of the synthesis engine capabilities is also inherent in the SoundFont format structure. The data hierarchy allows a single MIDI key depression to trigger an arbitrary number of sonic events. The basic SoundFont format structure is capable of expressing arbitrary networks within the modulation capabilities, and even within the signal processing capabilities themselves.

While the SoundFont format enumerates its parameters, these enumerations are extensible to provide even more extensive modulation capabilities as wavetable synthesis engines improve. As such, the SoundFont format structure will not become obsolete with future generations of wavetable synthesis hardware or even with software based synthesizers.

## **The SoundFont 2.0 Preset Hierarchy**

A SoundFont compatible file contains a single SoundFont compatible Bank. A SoundFont compatible Bank comprises a collection of one or more MIDI presets, each with unique MIDI preset and bank numbers. SoundFont compatible Banks from two separate files can only be combined by appropriate software which must resolve preset identity conflicts. Because the MIDI bank number is included, a SoundFont compatible Bank can contain presets from many MIDI banks. This is useful if the MIDI bank numbers are used as “variations”, but if the feature is misused, confusion over between MIDI banks and SoundFont compatible Banks can result.

A SoundFont compatible Bank contains a number of information strings, including the SoundFont Format Revision Level to which the Bank complies, the sound ROM, if any, to which the Bank refers, the Creation Date, the Author, any Copyright Assertion, and a User Comment string.

Each MIDI Preset within the SoundFont compatible Bank is assigned a name, a MIDI Preset # and a MIDI Bank #. A MIDI Preset represents an assignment of sounds to keyboard keys; a MIDI Key-On event on any given MIDI Channel refers to one and only one MIDI Preset, depending on the most recent MIDI Preset Change and MIDI Bank Change occurring in the MIDI Channel in question.

Each MIDI Preset in a SoundFont compatible Bank comprises an optional Global Preset Parameter List and one or more Preset Layers. The Global Preset Parameter List contains any default values for the Preset Layer Parameters.

A Preset Layer contains the applicable Key and Velocity Range for the Preset Layer, a list of Preset Layer Parameters, and a reference to an Instrument. The Preset Layer Parameters, whether defined in the Preset Layer or as defaults, additively modify the Instrument Parameters, allowing a single Instrument to be used to give a variety of sounds.

Each Instrument contains the applicable Key and Velocity Range for the Instrument, an optional Global Instrument Parameter List and a reference to one or more Instrument Splits. The Global Instrument Parameter List contains any default values for the Instrument Split Parameters.

Each Instrument Split contains the applicable Key and Velocity Range for the Instrument Split, an Instrument Split Parameter List and a reference to a Sample. The Instrument Split Parameter List, plus any default values, contains the absolute values of the parameters describing the articulation of the notes.

Each Sample contains Sample Parameters relevant to the playback of the Sample Data and a pointer to the Sample Data itself.

## **The SoundFont 2.0 Parameters**

The SoundFont 2.0 format provides an extensible list of Parameters, comprised of two types, Generators and Modulators. These names do not refer to the audio function of the parameters, but instead to their relationship in the data structure. A Generator is a direct input function to the synthesis model; a Modulator is a connection from a dynamic data source such as a MIDI Continuous Controller to a Generator. One additional parameter type is the Sample Parameters, which describe the nature of the sample data.

Typical SoundFont 2.0 format Generators are LFO Delays and Frequencies, Envelope Time parameters, Pitch Tuning, Filter Cutoff Frequency and Resonance, Attenuation, and the Amount that Envelopes and LFOs are applied to Pitch, Filter Cutoff Frequency, and Amplitude.

Typical SoundFont 2.0 format Modulators are the application of Pitch Wheel to Pitch, Modulation Wheel to Vibrato Depth, etc.

Typical SoundFont 2.0 format Sample Parameters include the Original Sample Rate of the sample, the Original Sampled Key Number of a pitched sample, any Pitch Correction required to bring the sample into tune, and the Sample Start, End, and Loop points.

## **Parameter Units**

Great care has been taken in the design of the SoundFont 2.0 format to ensure that the parameter units are precisely and correctly specified.

The precise definition of parameters is important so as to provide for reproducibility by a variety of platforms. Varying hardware platforms may have differing capabilities, but if the intended parameter definition is known, appropriate translation of parameters to allow the best possible rendition of SoundFont compatible files on each platform is possible.

For example, consider the definition of Volume Envelope Attack Time. This is defined in the SoundFont 2.0 format as the time from when the Volume Envelope Delay Time expires until the Volume Envelope has reached its peak amplitude. The attack shape is defined as a linear increase in amplitude throughout the attack phase. Thus the behavior of the audio within the attack phase is completely defined.

A particular synthesis engine might be designed without a linear amplitude increase as a physical capability. In particular, some synthesis engines create their envelopes as sequences of constant dB/sec ramps to fixed dB endpoints. Such a synthesis engine would have to simulate a linear attack as a sequence of several of its native ramps. The total elapsed time of these ramps would be set to the attack time, and the relative heights of the ramp endpoints would be set to approximate points on the linear amplitude attack trajectory. Similar techniques can be used to simulate other SoundFont 2.0 format parameter definitions when so required.

SoundFont 2.0 format parameter units have been designed to allow specification equal or beyond the Minimum Perceptible Difference for the parameter. For example, all units of frequency are in “Absolute Cents.” The unit of a “cent” is well known by musicians as 1/100 of a semitone, which is below the Minimum Perceptible Difference of frequency. Absolute Cents are defined by the MIDI key number scale, with 0 being the absolute frequency of MIDI key number 0, or 8.1758 Hz.

Absolute Cents are used not only for pitch, but also for less perceptible frequencies such as Filter Cutoff Frequency. While few synthesis engines would support filters with this accuracy of cutoff, the simplicity of having a single perceptual unit of frequency was chosen as consistent with the SoundFont 2.0 format philosophy. Synthesis engines with lower resolutions simply round the specified Filter Cutoff Frequency to their nearest equivalent.

A particularly important feature of the SoundFont 2.0 format parameter units is their correspondence with perception. For example, Envelope Decay Time is measured not in seconds or milliseconds, but in a logarithmic unit which we call “TimeCents.” An absolute timecent is defined as 1200 times the base two logarithm of the time in seconds. A relative timecent is 1200 times the ratio of the times.

Specification of Envelope Decay Time in timecents allows additive modification of the decay time. For example, if a particular Instrument contained a set of Instrument Splits which spanned Envelope Decay Times of 200 msec at the low end of the keyboard and 20 msec at the high end, a Preset could add a relative timecent representing a ratio of 1.5, and produce a Preset which gave a decay time of 300 msec at the low end of the keyboard and 30 msec at the high end. Furthermore, when MIDI Key Number is applied to modulate Envelope Decay Time, it is appropriate to scale by an equal ratio per octave, rather than a fixed number of msec per octave. This means that a fixed number of timecents per MIDI Key Number deviation are added to the default decay time in timecents.

## **Modulation in the SoundFont Format**

An important aspect of realistic music synthesis is the ability to modulate instrument characteristics in real time. This can be done in two fundamentally different ways. First, signal sources within the synthesis engine itself, such as low frequency oscillators (LFOs) and envelope generators can modulate the synthesis parameters such

as pitch, timbre, and loudness. But also, the performer can explicitly modulate these sources, usually by means of MIDI Continuous Controllers (CCs).

The SoundFont 2.0 format provides tremendous flexibility in the selection and routing of modulation by the use of the Modulation parameters. Each Modulation parameter specifies a modulation signal Source, for example a particular MIDI Continuous Controller, and a modulation Destination, for example a particular SoundFont format generator such as filter cutoff frequency. The specified Modulation Amount determines to what degree (and with what polarity) the source modulates the destination. An optional Modulation Transform can non-linearly alter the curve or taper of the Source, providing additional flexibility. Finally, a second Source can be optionally specified to be multiplied by the Amount.

By using the modulator scheme extremely complex modulation engines can be specified, such as those used in the most advanced sampled sound synthesizers. In the initial implementation of the SoundFont 2.0 format, several default modulators are defined. These modulators can be turned off or modified by specifying the same Source, Destination and Transform with zero or non-default Modulation Amount parameters.

## **The SoundFont Format Generators**

While the list of SoundFont format Generators is arbitrarily expandable, the SoundFont 2.0 format standard provides a basic list which are implemented in the AWE32 product line. The basic pitch, filter cutoff and resonance, and attenuation of the sound can be controlled. Two envelopes, one dedicated to control of volume and one for control of pitch and/or filter cutoff are provided. These envelopes have the traditional attack, decay, sustain, and release phases, plus a delay phase prior to attack and a hold phase between attack and decay. Two LFOs, one dedicated to vibrato and one for additional vibrato, filter modulation, or tremolo are provided. The LFOs can be programmed for depth of modulation, frequency, and delay from key depression to start. Finally, the left/right pan of the signal, plus the degree to which it is sent to the chorus and reverb processors is defined.

## **The SoundFont Format Modulators**

The Modulator construct is new to the SoundFont 2.0 format Standard, and only a few defaults are currently supported. These include the standard MIDI controllers such as Pitch Wheel, Vibrato Depth, and Volume, as well as MIDI Velocity control of loudness and Filter Cutoff.

## **The SoundFont Format Sample Parameters**

The Sample Parameters represented in SoundFont 2.0 format carry additional information which is not expressly required to reproduce the sound, but is useful in further editing the SoundFont compatible bank. The original Sample Rate of the sample and pointers to the Sample Start, Sustain Loop Start, Sustain Loop End, and Sample End data points are contained in the Sample Parameters. Additionally, the Original Key of the sample is specified in the Sample Parameters. This indicates the MIDI key number to which this sample naturally corresponds. A null value is allowed for sounds which do not meaningfully correspond to a MIDI key

number. Finally, a Pitch Correction is included in the Sample Parameters to allow for any mistuning that might be inherent in the sample itself.

## **The SoundFont 2.0 Format Specification**

The SoundFont 2.0 File Format Specification is publicly available. The specification may be obtained electronically from the SoundFont web page at [www.soundfont.com](http://www.soundfont.com).

The specification may also be obtained by contacting E-mu technical support at (408) 438-1921 or by contacting Creative Technologies technical support at (408) 428-6600.

## **Future Enhancements**

The SoundFont 2.0 format represents a first level of capability for the SoundFont compatible file standard. The SoundFont 2.0 format is fully upward compatible with many enhancements, providing more generators and modulators within the SoundFont format structure.

The Joint E-mu/Creative Technology Center is assuming responsibility for managing the SoundFont format. We anticipate both internal and external requests for enhancements to the SoundFont format standard, in fact there are many pending internal enhancement requests at present. These will be evaluated, and as resources allow, will be incorporated into the standard. In particular, we realize that there will be requests for enhancements beyond the capabilities of the E-mu/Creative product line, and we explicitly intend to support incorporation of these within the standard.

## **Summary**

Introduced in 1993, the SoundFont wavetable synthesis bank format has become a standard with the proliferation of the Sound Blaster AWE32 which uses the EMU8000 wavetable synthesis chip. The SoundFont format standard is now being publicly disclosed in its revision 2.0 embodiment.

SoundFont compatible files, in a manner analogous to character fonts, enable the portable rendering of a musical composition with the actual timbres intended by the performer or composer. The SoundFont format is a portable, extensible, general interchange standard for wavetable synthesizer sounds and their associated articulation data.

A SoundFont compatible bank is a RIFF file containing header information, 16 bit linear sample data, and hierarchically organized articulation information about the MIDI presets contained within the bank. Parameters are specified on a precisely defined, perceptual relevant basis with adequate resolution to meet the best rendering engines. The structure of the SoundFont format has been carefully designed to allow extension to arbitrarily complex modulation and synthesis networks.

The SoundFont format will be supported by a variety of tools and example code produced by Creative Technology and the Joint E-mu/Creative Technology Center.



The SoundFont 2.0 format will be the industry standard for wavetable synthesis banks well into the next millennium.

E-mu®, E-mu Systems®, and SoundFont® are registered trademarks of E-mu Systems, Inc.

Sound Blaster™ and AWE32™ are trademarks of Creative Technologies, Ltd.

All other brand and product names listed are trademarks or registered trademarks of their respective owners.