About Sound Formats

SoundApp supports many different file formats, but what exactly are they? The following is a brief description of the various sound formats that SoundApp supports:

SoundCap: This is a Macintosh sound format created for use with an early audio digitizer. Version 4.3 of the application circa 1986 is the latest I've seen. It was written by Mark Zimmer and Tom Hedges from Fractal Software. It supported two basic flavors of sounds, compressed and uncompressed. Both types had 'FSSD' as the file type and 'FSSC' as the creator. Uncompressed files are just a series of 8-bit unsigned bytes in the data fork. Compressed files store information pertaining to sampling rate and a checksum. Sampling rates are limited to 5.6, 7.4, 11.1 and 22.2 kHz, and compression is done with a Huffman algorithm. Compressed files are sometimes referred to as HCOM files because that is the first four characters of the file.

SoundEdit: This is the same file type as uncompressed SoundCap for mono sounds. In addition, it adds to the resource fork some information about colors, labels, looping segments and the format. The most useful for playback is the 'INFO' resource, which stores the sampling rate, limited to the same four as SoundCap. Stereo files consist of the left and right channels stored back-to-back in the data fork. The 'INFO' resource specifies the lengths of each channel, which can be different. SoundEdit came with the MacRecorder sound digitizer from Farallon and later by MacroMedia. SoundEdit Pro and SoundEdit 16 are more recent incarnations, and they support a much larger format suite, including up to 48-kHz samples and 16-bit resolution. They shed the limitations inherent in the original format. SoundApp does not currently support SoundEdit Pro or SoundEdit 16 files.

Studio Session Instrument: This format is primarily used with Super Studio Session and stores digitally sampled instruments. There are two types: compressed and uncompressed. Compressed instruments have the same format as compressed SoundCap files, and uncompressed instruments are likewise similar to uncompressed SoundCap files, with the addition of an eight-byte header.

AIFF and AIFF-C: AIFF stands for Audio Interchange File Format and was developed by Apple for storage of sounds in the data fork. It has been adopted by SGI and some other specialized applications. The Macintosh OS includes support for playing and creating AIFF files. More information about the format can be found in Inside Macintosh VI or Inside Macintosh: Sound. In addition, the format specification can be found at various places on the Internet. AIFF is a very flexible file format, allowing the specification of arbitrary sampling rates, sample size, number of channels, and application-specific format chunks which can be ignored by other applications. AIFF-C is basically AIFF with compressed samples. Apple supports two types of compression on the Macintosh, MACE 3-to-1 and MACE 6-to-1. Both are lossy compression algorithms, but provide reasonable quality with a great space savings. In addition, the Apple II GS uses ACE 2-to-1 and ACE 8-to-3 compression, but SoundApp only supports MACE-compressed files. Unlike SoundCap/Edit, samples are stored as two's complement values. Lossy compression means the compressed sound will not sound exactly like the original sound, much like JPEGs do not look exactly like the original picture.

System 7 and 'snd ': System 7 sound files are simply type 1 'snd ' resources stored with a type of 'sfil' and a creator of 'movr'. System 7 provides the familiar icon for them and permits playback in the Finder by double-clicking on them. A 'snd ' is a type of resource which consists of a series of commands for use by the Sound Manager. In addition to digitized sound samples, 'snd ' resources can contain direct frequency-modulated and wave table-based sounds. Any number of the three types can be combined with various effects to produce complex sound files. Simple Beep is an example of a non-digitized 'snd '. There are two types of 'snd ' resources, amazingly called type 1 and type 2. Type 1 is the format described above and is referred to as the System sound format. Type 2 is for use with HyperCard and can contain only a sampled (digitized) sound. SoundApp can play both types but will only convert sampled sounds. For more information on 'snd ' files consult Inside Macintosh VI or Inside Macintosh: Sound. A familiarity with the Resource Manager would also be helpful. 8-bit samples are stored as unsigned bytes, like SoundCap/Edit, but 16-bit samples are signed, like AIFF. Stereo 'snd ' resources are also possible, but Sound Manager 3.0 is required to play 16-bit samples directly. The two types of compression for 'snd '

resources and the same MACE types used in AIFF files.

Sun Audio (AU) and NeXT: Internally, these are the same formats. SoundApp differentiates between them by file type or suffix merely for the user's benefit. The format specifies arbitrary sampling rates and multi-channel sounds. It supports a number of sound encodings, including μ-law, a-law, various linear formats of varying sample sizes, floating point samples, native DSP samples and G.72x ADPCM compression. SoundApp supports μ-law, a-law, 8-bit signed, 16-bit signed, G.721 ADPCM and both versions of G.723 ADPCM. Each μ-law sample is stored in 8 bits, but the meaning of the sample is different. Normal sound formats use a linear encoding, whereas μ-law and a-law are logarithmic. This means that the spacing between the different sound levels grow progressively larger as the values increase. This format provides a larger dynamic range than normal 8-bit samples, approximately equivalent to 12-bit samples. However, it suffers from more noise than linear encodings. The G.721, G.723-24 and G.723-40 ADPCM formats are CCITT standards for compression of 8000-Hz 14-bit samples into a 32-, 24- or 40-kbps data stream. These compressed formats are not very popular due to the extremely slow decompression rates. Most files start with the four-character signature, '.snd', but there are some older headerless AU files. These are assumed to be μ-law encoded, mono at 8000 Hz. A ".al" suffix will force the sound to be a-law, if it does not have a header. The U.S. telephone system uses μ-law encoding for digitization, whereas the European telephone systems use a-law encoding.

Windows WAVE: This format was created by Microsoft and IBM, and it has unfortunately become a popular standard. Like AU, it specifies an arbitrary sampling rate, number of channels and sample size. It also specifies a number of application-specific blocks within the file. It has a plethora of different compression formats, although the Microsoft ADPCM is the most popular. SoundApp only supports 8-, 16-bit and MS ADPCM-compressed sounds. MS ADPCM provides 4-to-1 compression. All data fields and 16-bit samples are stored in little-endian notation, as Intel processors require. All other formats supported by SoundApp use big-endian notation which means the high-bytes come first in the data stream.

Sound Blaster VOC: This is the format used by the Creative Voice SoundBlaster hardware used in IBM-compatible computers and is optimized for that hardware. It specifies sampling rate as a multiple of an internal clock and is not as flexible as the other general formats. Data can be segmented and portions of silence can be added. SoundApp supports both of these features, but not the looping feature.

Amiga IFF (8SVX): This is the dominant format on the Commodore Amiga platform. It can specify an arbitray sampling rate but ony supports mono 8-bit sounds. It also supports a 2-to-1 lossy compression format which uses a unique Fibonacci delta compression.

Sound Designer II: This is a format for professional sound editing on the Macintosh. It can specify arbitrary sampling rates and supports multiple channels and data sizes. Information regarding the specifics of the sound are stored in three 'STR ' resources. Like VOC, 8SVX and WAVE, samples are encoded as signed values. More information about this format can be obtained from Digidesign. SoundApp only supports version II files.

DVI ADPCM: This is the Intel/DVI ADPCM (Adaptave Differential Pulse Code Modulation) format. It is a 4-to-1 compressed 16-bit file format. It is unique among the various ADPCM formats in that it's very fast, and like all ADPCM formats it is lossy. The version of the format that SoundApp supports plays at a 8000-Hz sampling rate.

Amiga MOD: This is not really a sound format but a music format. It stores digitized instruments and contains a musical score which produces a lengthy composition with a very small amount of data. There have been various extensions to this format, but SoundApp only supports those which Sound Trecker 2.2 supports. These include Amiga SoundTracker, StarTracker, NoiseTracker, ProTracker (4-track), Amiga StarTracker (4- and 8-track), Oktalyzer (4-8 track), Amiga MED/OctaMED (4-16 track MMD0/1/2 formats), IBM FastTracker (4-, 6- and 8-track), IBM TakeTracker (1-32 track). SoundApp does not support MTM or S3M formats. Native code will be used for MOD playback on a Power Macintosh.

QuickTime Movies: This is the Apple standard for time-based multimedia files. Versions 1.x support

moving pictures, sound and later versions support text. QuickTime 2.0 supports MIDI data, but cannot convert it to a sound file. The QuickTime extension is required in order for SoundApp to convert movies to sound files. SoundApp does not support the playing of QuickTime files at this time.

Audio CD Tracks: Using QuickTime 1.6 or later, SoundApp can extract the digitial data from audio CDs. It will bring up a dialog box which allows various parameters to be set for the conversion. Tracks are first converted to audio-only QuickTime movies and then to the desired format. Audio tracks take up a huge amount of space.

More information about various sound formats can be found in the Audio File Formats FAQ by Guido van Rossum at guido@cwi.nl. A UNIX program called SOX can convert various formats between each other and provides source code, and it has been ported to a variety of other computers. More information about it can be obtained from its author, Lance Norskog, at thinman@netcom.com.