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. Note that PCM (Pulse Code Modulation) is sometimes called linear encoding and is normal uncompressed audio.

AIFF and AIFF-C: (.aif, .aiff, .aifc) 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. Unlike SoundCap/Edit, 8-bit samples are stored as two's complement values. AIFF-C is basically AIFF with compressed samples.
 pple supports two proprietary 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. 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. Recently, Apple has added support for µ-law and IMA 4:1 sub-formats, which are described below. Files created by Iomega's RecordIt utility are AIFC files using an MPEG layer II codec. SoundApp can play these files without the codec. In addition, the Apple II GS uses ACE 2-to-1 and ACE 8-to-3 compression, but SoundApp does not supports ACE files.

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

Audio CD Tracks: CD audio tracks can be read directly from an audio CD by converting the track files. Since CD audio data is 44.1 kHz, 16-bit, stereo, the resulting files can be quite large. SoundApp can read audio data faster than real-time, depending on the CD-ROM drive being used. Playback of audio CD tracks is also supported as is labeling tracks in a Play List with track names from the CDDB. Note that SoundApp does not yet support playing of audio CD tracks that are not listed in a Finder volume on the Desktop. Enhanced CDs will not have audio CD tracks listed in the Finder using anything before Mac OS 9.0, so these tracks cannot yet be played using SoundApp.

AVR: (.avr) Created by Audio Visual Research, this is a popular sound file format on the 680x0-based Atari ST computers. It can contain data of any sampling rate in mono or stereo at 8- or 16-bits.

DVI ADPCM: (.adpcm) This is the Intel/DVI ADPCM (Adaptive 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 mono at an 8000-Hz sampling rate.

EPOC 32 (Psion Series 5): This format is native to the EPOC 32 operating system which is used on the Psion Series 5 palmtops. It is a more complex format than the Series 3 audio file format and is based on their "Direct File Store" architecture. These files can contain a-law data like the Series 3 files and ADPCM-compressed data. SoundApp does not support the ADPCM variant.

GSM: (.gsm, .au.gsm) This compression algorithm is the European GSM 06.10 standard for full-rate speech transcoding, prI-ETS 300 036, which uses RPE/LTP (residual pulse excitation/long-term prediction) coding at 13 kbit/s. It was developed for the European digital cellular phone system to make the most of tight bandwidth. Basically, what this means is that it analyzes and derives a mathematical formulation of small sections of speech using a model of the human vocal tract. Thus, it is optimized for speech reproduction and is in fact used in many Internet phone applications, although it seems to compress arbitrary sounds relatively well. The ".au.gsm" format consists of a series of 33-byte frames sampled at a mono 8000 Hz. In spite of the fact that the suffix contains ".au" the files are not related to the Sun Audio file format. The WAVE implementation uses a slight variation on the algorithm (they're good with these standards!) and can support mono files at an arbitrary sampling rate.

IMA ADPCM: This is a cross-platform standard from the Interactive Multimedia Association for sound playback. The basic algorithm is the same as in DVI ADPCM. SoundApp currently supports IMA data in WAVE, AIFF-C and 'snd ' resources. Unfortunately, Apple and Microsoft store their data in different ways. (So much for standards!) Both mono and stereo sounds are supported at an arbitrary sampling rate; however, the compression algorithm only accepts 16-bit samples.

IRCAM: (.sf) These files are used by academic music software such as the CSound package and the MixView sound sample editor. These files also specify an arbitrary sampling rate and can contain mono or stereo files. SoundApp only supports 8- and 16-bit linear and 32-bit floating point samples, although the format can contain other encodings.

MIDI: (.mid, .midi, .kar) Musical Instrument Digitial Interface is primarily a standard for communication between musical instruments. General MIDI (GM) is a standard for storing compositions based on what events happened during the performance. It does not contain digitized audio data; instead, it stores only the information about which notes were played in a time-line format. This is similar to the MOD format but without the digitized instrument samples. QuickTime 2.0 and later supports General MIDI data in QuickTime movies. SoundApp can directly play type 0, 1 and 2 MIDI files using QuickTime or OMS 2.1 or later and can also play MIDI data embedded in QuickTime movies. Note that lyric information from karaoke files is not displayed. There are several extensions to the GM standard, including GS (Roland) and XG (Yahama). SoundApp supports both of these extensions. It will also send all System Exclusive data to the selected MIDI output device if you are using the OMS MIDI driver. Using QuickTime 3.0 or later, SoundApp can play files using GS instruments.

MOD: (.mod, .s3m, .mtm, .it) 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 a subset using two different drivers. These include Amiga SoundTracker, NoiseTracker, Protracker, Amiga StarTracker (4- and 8-track), Oktalyzer (4-8 tracks), Amiga MED/OctaMED (4-16 tracks, MMD0/1/2 formats), IBM FastTracker (4-, 6- and 8-track), IBM TakeTracker (1-32 tracks). Using the ZSS driver, SoundApp also supports S3M (ScreamTracker 3), MTM (Multitracker) and IT (Impulse Tracker). Playback of XM or 669 files is not currently supported by either driver.

MPEG Audio: (.mp, .mp2, .mp3, .m1a, .m2a, .mpg, .mpeg, .swa) MPEG stands for the “Moving Picture Experts Group,” working under the joint direction of the International Organization for Standardization (ISO) and the International Electro-Technical Commission (IEC). This group works on standards for the coding of moving pictures and associated audio. MPEG audio files can be either layer I, II or III. Increasing layer numbers add complexity to the format and require more effort to encode and decode. However, they also provide higher playback quality for the sample bit rate. SoundApp supports layers I, II and III. To further complicate matters, MPEG files come in two flavors, MPEG-1 and MPEG-2. The encodings for the three layers are mostly the same; however, MPEG-2 streams have lower sampling rates for better fidelity at lower bit rates. Files can have sampling rates of 32000, 44100 and 48000 Hz for MPEG-1 and 16000, 22050 and 24000 Hz for MPEG-2. Actually, there is another flavor that can only be used for layer III data which is frequently called MPEG-2.5. It is not an official standard, but a format designed by Fraunhofer Institut für Integrierte Schaltungen in Germany for low data rate applications at 8000, 11025 and 12000 Hz. Microsoft has adopted the MPEG-2.5 “standard” for use in their NetShow product and also in WAVE files using the NetShow codec. MPEG data can be in stereo or mono and decompresses to 16-bit resolution. MPEG compression is a lossy algorithm based on perceptual encodings, which can achieve high rates of compression without a noticeable decrease in quality. Typical compression rates are around 10-to-1. SoundApp supports MPEG audio playback only on Macs with a PowerPC processor. Finally, Macromedia’s Shockwave streaming audio system uses a layer III encoding with a non-standard header, which SoundApp will ignore. These Shockwave audio files frequently have a “.swa” suffix.

Nomad II: (.nvf) These files are basically 8-kHz, G.721-encoded mono files with a custom header. Support for this format is considered preliminary due to the unknown nature of the header.

PARIS: (.paf) This format the native format for the Ensoniq PARIS digitial audio editing system. PARIS stands for “Professional Audio Recording Integrated System.” It can contain 8-, 16- and 24-bit data in mono or stereo; however, most files are 16- or 24-bit mono.

PSION Series 3 sound: (.wve) This format consists of a short header followed by a-law encoded samples at 8000 Hz. It is used by the PSION Series 3 palmtop personal information manager and uses a “.WVE” suffix.

QuickTime Movies: (.mov) 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 added MIDI tracks via a software synthesizer or external synthesizer in 2.5 and later. QuickTime 2.0 or later and the QuickTime Musical Instruments extension must be installed in order to play QuickTime MIDI files. SoundApp should be able to play and convert from most all audio compression formats, although QuickTime 3.0 doesn’t support conversion from MPEG so SoundApp can’t either.

RecordIt: This format is really a sub-format of AIFF that contains MPEG Layer II encoded data at one of four quality factors.

Sound Blaster VOC: (.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 the 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. Extensions to the format allow support for 16-bit samples and stereo, both of which SoundApp supports.

Sound Designer: Digidesign’s predecessor to the Sound Designer II format. Unlike the second generation format, it does not use resources to store header information. It has a large header, although most of it is used internally by their software. It can only contain mono data. Most files are usually 16-bit, 44.1-kHz.

Sound Designer II: This is a popular 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. Some professional audio applications use a “split stereo” format, which SoundApp supports. This method stores the left and right channel of a sound as separate files. SoundApp requires the left channel to have a “.L” suffix and the right channel to have a “.R” suffix. More information about this format can be obtained from Digidesign.

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 SoundCaps are sometimes referred to as HCOM files because those are 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. MACE-3 and MACE-6 compression is supported for mono 22 kHz files only, which is a limitation of the SoundEdit format. SoundEdit also supports 4:1 and 8:1 compression, but SoundApp does not support these proprietary compression algorithms. 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.

Sun Audio and NeXT: (.au, .snd) 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-, 16-, 24- and 32-bit signed, 32- and 64-bit floating point, 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 linear encoding, whereas µ-law and a-law are logarithmic. This means that the spacing between the different sound levels grows 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.

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.

System and 'snd ': System sound files are simply type 1 'snd ' resources stored with a type of 'sfil' and a creator of 'movr'. The Mac OS provides the familiar icon for them and permits playback in the Finder by double-clicking on them. An '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 or later is required to play 16-bit samples directly. The possible types of compression for 'snd ' resources are the same MACE, IMA and µ-law types used in AIFF-C files.

Windows WAVE: (.wav) This format was created by Microsoft and IBM, and it has unfortunately become a popular standard. Like Sun Audio, 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. SoundApp only supports 8-bit, 16-bit, 32-bit, µ-law, a-law, GSM-, IMA ADPCM-, MS ADPCM- and MPEG-2.5 compressed sounds. IMA and MS ADPCM provide a 4-to-1 compression ratio and GSM provides an approximately 9.7-to-1 compression ratio. 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.

More information about various sound formats can be found in the Audio File Formats FAQ by Guido van Rossum on <ftp://ftp.cwi.nl/pub/audio/>. 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 <mailto:thinman@netcom.com>.

Copyright © 1996-2000 Norman Franke. All rights reserved.