Technical Note TN1081

Understanding the Differences Between Apple and Windows IMA-ADPCM Compressed Sound Files

CONTENTS

If you attempt to play a sound compressed with Apple's IMA compressor on a computer running Windows' IMA decompressor, the sound either does not play or plays incorrectly. The same thing also occurs if you try to play a Windows WAVE file compressed with the IMA-ADPCM algorithm on a Macintosh using the Apple-supplied 'ima4' decompressor. Because the two sound file formats are not interchangeable, many developers working in cross-platform environments want to know what the differences are between Apple IMA and Windows IMA compressed sound files.

This Technote is aimed at developers who want to edit or play back Macintosh IMA-ADPCM files on computers not running the Mac OS or QuickTime for Windows.

Updated: [Dec 06 1996]

A Little Background & Terminology

IMA is an acronym for the Interactive Multimedia Association, which defined and published the **ADPCM** algorithm that Apple's and Microsoft's compressors (and therefore decompressors) are based on.

For purposes of this Note, I refer to Apple's compressor as IMA, because that is generally how Macintosh programmers refer to it, and I refer to Microsoft's compressor as ADPCM because that is how most Microsoft programmers refer to it. I refer to the actual algorithm, regardless of platform, as IMA-ADPCM (because that's its official designation).

The IMA-ADPCM algorithm only defines how to take 16-bit linear samples and compress them down to four bits each. According to page 11 of *Recommended Practices for Enhancing Digital Audio Compatibility in Multimedia Systems revision 3.00* (October, 1992) by IMA, "...tagging the [audio] stream definition and synchronization issues are left to upcoming IMA recommendations," which, to date, have not yet been made. Since it does not define the data stream or file format for the resulting data, Apple and Microsoft have designed file formats that are not exactly the same in the way that data is streamed to the file. The consequence of this is explained in the next section.

The IMA-ADPCM algorithm takes a word (16 bits), and produces a nibble (4 bits), which is then placed sequentially into a buffer for later storage. This has several important advantages. For one thing, it means that while the uncompressed data is endian dependent (being either big-endian or little-endian), the resulting compressed data is endian-independent, which is a big plus when making a cross-platform sound standard.

Back to top

Differences in Data Stream Formats

The reason why the Apple and Microsoft IMA files are not compatible is because of their different data stream formats, not because the actual compression algorithm differs radically. The IMA-ADPCM algorithm does not specifically disallow data other than sound data in the file format, so Apple and Microsoft were able to make design decisions and implement the file stream that worked best for their intended use and to fit within any previous sound file definitions.

Differences in Packet Sizes

The most obvious differences between Apple's and Microsoft's ADPCM compressed files are in the size of the **packet** that

each uses. A packet is the smallest amount of data that can be decompressed by the given the algorithm implementation. All decompression must be done with an integer number of packets.

Apple's IMA compressed files have a fixed packet size of 64 samples per packet, while Microsoft's ADPCM compressed files have a variable number of samples per packet. The packet size is not variable within a WAVE file, but it can, and does, vary between WAVE files.

Since the IMA-ADPCM algorithm is a type of run-length encoding, which doesn't lend itself to easy random access, to allow for random access the encoded data is broken into packets with the predicted sample value and step index put at the front of the packet of samples. This information can be used to restart the decompression from this point in the file.

Back to top

Differences in Encoding/Decoding Stereo Streams and AIFF and WAVE Files

Apple's and Microsoft's ADPCM compressed files also differ in how stereo streams are encoded, and therefore decoded. The IMA standard does not say anything about having predictor bytes, nor does it say how multiple channels of sound should be treated, and that is one place where AIFF and WAVE files differ.

For packet size, Apple chose 64 samples per predictor bytes, and for stereo, Apple defines that a packet of left channel data precedes a packet of right channel data. Apple's IMA compressor puts two bytes at the front of each packet, which are referred to as predictor bytes (not be confused with the predictor value that the IMA-ADPCM algorithm defines). These two bytes contain the step index, a 7-bit value in the low byte, and the most significant 9 bits of the predictor value. Obviously, since the real predictor value is 16 bits, this 9-bit predictor is only an approximation, but it doesn't cause significant loss of quality.

Microsoft, on the other hand, lets the program compressing the sound determine the size of the packet, and the WAVE standard is eight samples of left data followed by eight samples of right data. WAVE files also have a two byte header just like the Mac, but have an 8-bit step index instead of a 7-bit step index and an 8-bit predicted sample approximation instead of a 9-bit approximation.

For more information on the WAVE format, check out http://www.mediatel.lu/workshop/audio/fileformat/files/wave.pdf

The wBlockAlign Field

The WAVE file format has a field called wBlockAlign which tells the program reading the file the minimum number of bytes that must be processed at a time. The only WAVE files compressed with ADPCM that I have studied have had this value set to either 512 or 1024. This field could be set to other values. Setting this value to a number larger than 34 allows the WAVE file to have fewer predictor bytes than the same data as an IMA compressed AIFF file.

Apple's method of small packet sizes gains the ability to start playing sound from more locations in the file, thus giving better random access. Microsoft's method still allows for random access, but with greater jumps between points where sound can begin playing from.

Back to top

Consequences of the Differences Between AIFF and WAVE File Formats

Because of the substantial differences between the AIFF and WAVE formats, the Apple IMA decompression codec will not handle Microsoft's ADPCM compressed WAVE files.

Stereo ADPCM WAVE files have the channels interleaved differently than AIFF files and have fewer predictor bytes, which is a significant enough difference that the Apple IMA decompressor will not decompress them and produce the same sound as the Microsoft ADPCM decompressor would. The Microsoft ADPCM decompressor will not decompress a stereo IMA AIFF file as the Apple IMA decompressor would unless it is modified to take into account the different channel interleaving and predictor bytes.

Because the Apple IMA compressor makes use of the high bit of the low byte in the predictor word for the predicted sample (to double its precision) and the Microsoft IMA compressor treats this bit as part of the value of the step index, it is likely that the step index will be invalid most of the time if the Microsoft ADPCM decompressor is dealing with an Apple IMA compressed sound.

The step index is likely to be invalid because, depending on what the value of the predicted sample is, the high bit of the low byte of the predictor word (the least significant bit of the predicted sample) may or may not be set. Because the Microsoft ADPCM decompressor takes all 8 bits of the low byte of the predictor word, instead of only the low 7 bits (which are all that are needed; the step index's possible range is from 0 to 88, inclusive), having the high bit set will automatically make the index out of range, which may cause the decompressor to stop decompressing data.

Differences in Resulting Data at the Nibble Level

Inspection of the same data compressed using Apple's IMA compressor and Microsoft's ADPCM compressor shows differences in the resulting data at the nibble level. The difference in the resulting data is often not major: if the data is different, it is usually only off by one value plus or minus. Examining the public Microsoft IMAADPCM.C file shows that the imaadpcmFastEncode and imaadpcmSampleEncode algorithms are not exactly the same as that published by IMA, but I have not attempted to trace execution of sample data to determine where the algorithm is producing different data. When played back through the correct decompressor, the two sounds sound identical.

The Bottom Line

What this means is that if you were to go to all the work of converting from one file format to the other -- for example, converting a WAVE file to AIFF file -- you would find that when you decompressed the sound, the resulting sound would be slightly different, though the change might not be audible.

Back to top

Options for Cross-Platform Compressed Sound

If you are a developer and wish to have only one sound file format for your cross- platform product, what are your options when it comes to cross-platform compressed sound?

At this point in time, you have three options:

- go with QuickTime sound-only movies,
- use both AIFF and WAVE IMA-ADPCM compressed sounds,
- or don't use IMA-ADPCM compressed sounds.

Each solution has its own advantages and drawbacks. Only you can determine the right combination for your intended market.

Back to top

Summary

As you can see, using cross-platform sounds doesn't have to be difficult. It all revolves around the proper selection of a file format. IMA may not the panacea that it was intended to be.

Back to top

Playing a Compressed WAVE File on the Mac: A Sidebar

You can play a compressed WAVE file on the Mac, but probably not with the default functionality of the Sound Manager. You have to do all of the sound header parsing yourself, just as you do for an uncompressed sound, and then you have data which you may or may not be able to directly pass to the Sound Manager.

If the WAVE is formatted using ulaw, then your program doesn't have to do anything special. Since the ulaw file is processed on a byte by byte basis, and there is no endian difference between the same data as an AIFF or WAVE file, the standard Mac ulaw decompressor can deal with this data without a problem.

On the other hand, you cannot play IMA-ADPCM compressed WAVE files as simply as you could play a ulaw WAVE files because of the difference in the actual data stream of a sound compressed with the Mac's IMA compressor versus the same sound compressed with the Windows' IMA-ADPCM compressor.

You have to deal with Windows' IMA-ADPCM compressed WAVE sounds just as you would any sound which required a custom decompressor. Your program does all the decompression. This can be done either by writing a decompression component for the Mac (in which case any program can use it), or by having a decompression function in your program.

If you write your own sound decompressor ('sdec'), you can use any Sound Manager routine that will play an arbitrarily compressed sound. You must make sure, however, to specify that the sound header explicitly states that the sound is compressed with your compressor, so that the Sound Manager will call your 'sdec'.

If you choose not to write a decompression component and you can decompress the sound completely, you can use any Sound Manager call that takes a buffer of uncompressed sound. If you can't decompress the sound completely, you will have to decompress it in chunks and use `SndPlayDoubleBuffer` or `bufferCmd's` to play each chunk.

Playing an Uncompressed WAVE File Via the Sound Manager

To play an uncompressed WAVE file via the Sound Manager is relatively easy. You only need to parse the sound's header and then give the Sound Manager buffers of properly formatted data to play. Because the headers of a AIFF and WAVE are very similar and the data is stored in very much the same way in both files, parsing a WAVE header is no more difficult than parsing an AIFF header. Microsoft documents the format in *Multimedia Programming Interface and Data Specification v1.0*. In the `SndPlayDoubleBuffer` sample code (on Apple's Developer ToolChest CDs), there is also quick-and-dirty code that shows how to parse a WAVE header.

Once you have parsed the WAVE header, you can play the data in much the same way you would play AIFF data. Remember that the WAVE file's data is stored little-endian, but if you have 8-bit (mono or stereo) sounds, that means nothing: a byte in little-endian is the same as a byte in big-endian. If you use 16-bit (mono or stereo) sound, however, an endian conversion will have to take place before you can play the sound.

Back to top

References

Multimedia Programming Interface and Data Specification v1.0

For more information on the WAVE format, check out http://www.mediatel.lu/workshop/audio/fileformat/files/wave.pdf

Back to top

Downloadables

 Acrobat version of this Note (180K). Download

Back to top

Technical Notes by API | Date | Number | Technology | Title
Developer Documentation | Technical Q&As | Development Kits | Sample Code