Tech Note 0132

.WAV FILE FORMAT

The .WAV file format is commonly used in pcs that use the Microsoft Windows operating system. These files are sometimes also used in acoustic experiments. This tech note describes how to convert a 16-bit integer sound file to a .WAV file, and explains some of the technical details of the .WAV file construction.

CONVERTING TO A .WAV FILE

To convert a 16-bit integer file to .WAV format, add a 44-byte header to the binary file as shown below. Note that four-byte numbers (unsigned longints) are stored on disk as 11 22 33 44 where 44 is msb and two-byte numbers (unsigned integers) are stored as 11 22 where 22 is the msb. These are Intel conventions and are taken care of automatically by compiled programs.

4 bytes	"RIFF"
4 bytes	number of bytes following this field (size of sound file + 36)
4 bytes	"WAVE"
4 bytes	"fmt " (last character is a space)
4 bytes	16 (size of the format chunk)
2 bytes	1 (data format code, 1 = PCM)
2 bytes	1 (number of channels)
4 bytes	samples per second
4 bytes	number of bytes sent per second (same as previous, since it's played as mono)
2 bytes	2 (number of bytes in a sample)
2 bytes	16 ("alignment": number of bits per sample)
4 bytes	"data"
4 bytes	number of bytes of wave data (should be even for 16-bit samples)

CONVERTING A .WAV FILE TO 16-BIT

Converting a .WAV file to 16-bit integer format is a little bit trickier for any of the following reasons:

- additional fields can be included in the file besides the sound data (see RIFF format details, below)
- data can be in stereo and must be split to separate files
- data may only be 8-bit, requiring a conversion to 16-bit in your program

Fortunately, all this information is encoded in the file and can be dealt with as needed.

TECHNICAL DETAILS OF .WAV FILES

The .WAV waveform file is in standard Microsoft RIFF format (Resource Interchange File Format). The RIFF format is a tagged-file structure.

The building block of a RIFF file is called a chunk, defined using C syntax, looks like the following:

```
typedef unsigned long DWORD;        // 4 bytes
typedef unsigned char BYTE;         // 1 byte
typedef DWORD FOURCC;               // four-character code
typedef struct {
   FOURCC ckID;         // four character chunk ID
   DWORD cksize;        // the size of field <ckdata>
   BYTE ckData[ckSize]; // the actual data of the chunk.  If the size of the
                        chunk is an odd number of bytes, a pad byte with value
                        zero is written after ckdata, which is not counted in
                        cksize
   }
```

Two types of chunks, the "LIST" and "RIFF" chunks, may contain nested chunks, or subchunks. The LIST and RIFF chunks have an additional four-byte field, the form type, which identify the format of the data stored in the file.

Typically:

RIFF (size) (form-type) <chunk-data>
LIST (size) (list-type) <chunk-data>

The <chunk-data> could be a WAVE form data chunk. If the form-type is WAVE:

<chunk-data> = <fmt-chunk><data-chunk>

<fmt-chunk> = <wave-format><format-specific>

<wave-format>

```
struct {
   WORD wFormatTag;        // format category
   WORD nchannels;         // number of channels
   DWORD nSamplesPerSec;   // sampling Rate
   DWORD nAvgBytesPerSec;  // bytes per second
   WORD nBlockAlign;       // block alignments
}
```

<format-specific>

```
struct {
UNIT nBitsPerSample;    // sample size
}
```

<data-chunk> = data(<wave-data>), currently the data are in PCM (Pulse Code Modulation) format. For 8-bit WAVE files, mono data are saved byte-by-byte, and stereo data are saved as channel-1-byte, channel-2-byte. Each data byte is saved as unsigned char. For 16-bit (or more than 8 bits) WAVE files, the channel orders are the same, but since every datum has two bytes, the lower-order byte is saved first. Data more than 8-bit are saved as int.

Every chunk has a standard tag (8 bytes, 4 bytes for chunk ID, four bytes for chunk data size in bytes).

Below is a typical examples showing RIFF chunk and a subchunk:

RIFF {size1}WAVE

fmt {size2}{wave-format-chunk-body}

data {size3}{data-chunk-body}

where

size1: total size in this RIFF chunk, right after this size DWORD;
size2: size of the wave format chunk after size 2, 16-byte, as mentioned above;
size3: total size of the wave data after size3.

The Microsoft Multimedia Development Kit (MDK) includes a RIFF file-viewing utility called *FileWalker*.