

# Apple II File Type Notes



## Developer Technical Support

**File Type:**            **\$D5 (213)**  
**Auxiliary Type:**       **\$0007**

Full Name:    SoundSmith Music Sequence  
Short Name:   SoundSmith document

Written by:    Matt Deatherage

March 1990

Files of this type and auxiliary type contain music sequences used by SoundSmith.

SoundSmith is a music sequencing program that uses the full sound capabilities of the Apple IIGS. SoundSmith uses standard Apple Sampled Instrument format (ASIF) instruments to produce rich stereo sound with a variety of instruments.

For more information on SoundSmith, contact:

Huibert Aalbers  
Travesía Andrés Mellado, 3  
28015 Madrid  
Spain  
Attn: SoundSmith Technical Support  
Phone: (34)-1-5446940

### The File Format

SoundSmith sequences consist of a 600 byte header, followed by three equally-sized blocks containing the notes in the sequence, the effects to be applied to the notes, and parameters for the effects. The file concludes with 30 bytes of stereo information.

### The Header

signature	(+000)	<b>6 Bytes</b>	ASCII bytes "SONGOK". An identifier to SoundSmith that the file is not corrupted.
length	(+006)	<b>Word</b>	The length of each of the three equally-sized blocks that follow the header (Main, Effects1, and Effects2).
tempo	(+008)	<b>Word</b>	The tempo for the song. A note is played each tempo/50th of a second (see the "Playing the Music" section in this Note).
instBlock1	(+020)	<b>InstBlock</b>	The instrument parameters for the first instrument.
instBlock2	(+050)	<b>InstBlock</b>	The instrument parameters for the second instrument.
instBlock3	(+080)	<b>InstBlock</b>	The instrument parameters for the third instrument.

instBlock4	(+110)	<b>InstBlock</b>	The instrument parameters for the fourth instrument.
instBlock5	(+140)	<b>InstBlock</b>	The instrument parameters for the fifth instrument.
instBlock6	(+170)	<b>InstBlock</b>	The instrument parameters for the sixth instrument.
instBlock7	(+200)	<b>InstBlock</b>	The instrument parameters for the seventh instrument.
instBlock8	(+230)	<b>InstBlock</b>	The instrument parameters for the eighth instrument.
instBlock9	(+260)	<b>InstBlock</b>	The instrument parameters for the ninth instrument.
instBlock10	(+290)	<b>InstBlock</b>	The instrument parameters for the tenth instrument.
instBlock11	(+320)	<b>InstBlock</b>	The instrument parameters for the eleventh instrument.
instBlock12	(+350)	<b>InstBlock</b>	The instrument parameters for the twelfth instrument.
instBlock13	(+380)	<b>InstBlock</b>	The instrument parameters for the thirteenth instrument.
instBlock14	(+410)	<b>InstBlock</b>	The instrument parameters for the fourteenth instrument.
instBlock15	(+440)	<b>InstBlock</b>	The instrument parameters for the fifteenth instrument.
musLength	(+470)	<b>Word</b>	Length of the music in SSBlocks.
musList	(+472)	<b>128 Bytes</b>	List of SSBlocks to play. Each block is identified by one byte (i.e., 0 3 5 2 2 n means play block 0, block 3 block 5, block 2, block 2, and block n respectively).

An **SSBlock** is 896 Bytes (64 \* 14 bytes). The Main block is composed of SSBlocks. An **InstBlock** is a 30-byte block of instrument parameters defined as follows:

instName	(+000)	<b>String</b>	ASCII name of the instrument to be used. If this is less than 22 bytes (21 characters plus the length byte), it must be padded to take 22 bytes.
reserved	(+022)	<b>Word</b>	Reserved, set to zero.
volume	(+024)	<b>Word</b>	Volume for this instrument. Although this is a word parameter, legal values range from 0 to 255.
reserved	(+026)	<b>Word</b>	Reserved, set to zero.
reserved	(+028)	<b>Word</b>	Reserved, set to zero.

### The Main block

The main part of the file consists of three equally-sized blocks. The length of each of the three parts is given by the Length field in the header; the entire Main block is 3\*Length bytes long. Bytes in each block are related to each other positionally. For example, the first byte of the Effects1 and Effects2 blocks contain the effects to be applied to the note in the first byte of the Notes block.

The first block is the **Notes** block. Each byte is a MIDI Note number representing the note to play.

The second block is the **Effects1** block. The high nibble of each byte determines which instrument should be used to play the note in the corresponding byte of the Notes block. The low nibble of each byte contains a value to be used by each effect.

The third block is the **Effects2** block, and contains values to be used for the effects listed in the bytes of the Effects1 block.

Table 1 contains currently defined values for the effects and their values. All values not listed are **reserved** and must not be used.

<b>Effects1 byte</b>	<b>Effects2 byte</b>
0 = Arpeggiato	0 = no arpeggiato, \$xy = increment1 of x, increment2 of y
3 = Set Volume	new volume (\$00 - \$FF)
5 = Decrease Volume	volume to subtract from instrument volume
6 = Increase Volume	volume to add to instrument volume
F = Set Tempo	new tempo

---

**Table 1–SoundSmith Effects**

### **Stereo Data**

The file ends with 30 bytes of stereo data. The data is in 15 words, one for each instrument. A value of \$0000 indicates the instrument uses the right channel; a value of \$FFFF indicates the left channel. The first word corresponds to the first instrument, and so on.

### **Playing the Music**

Those wishing to play the music in a SoundSmith file should use an interrupt-driven playback routine. The routine should be called every tempo/50th of a second. When called, the routine should read the next fourteen notes, Effects1 and Effects2 bytes, and play them on voices 1 through 14 using the specified instruments. Since SoundSmith provides 14 voices, you can use the fifteenth DOC oscillator as a timer to generate the required 50 Hz interrupts. When the note value is zero, you should do nothing (do **not** stop the sample). When the note value is 128 (\$80), stop the sample on that voice.

### **Further Reference**

---

- *Apple IIGS Toolbox Reference*, Volume 3