

Playing Frequencies Issuing commands

You can play frequencies one at a time by using the **SndDoCommand** or **SndDoImmediate** function to issue freqDurationCmd sound commands. A sound plays for a specified duration at a specified frequency. You can use any of the three available playback synthesizers to play the sound. If you use the wave-table synthesizer or sampled sound synthesizer, then a voice must previously have been installed in the channel. (See the section called **Installing Voices Into Channels** for instructions on installing wave tables and sampled sounds as voices.)

You can also play frequencies by issuing the freqCmd command, which is identical to the freqDurationCmd command, except that no duration is specified when you issue freqCmd.

Note: A freqDurationCmd command continues playing until another command is available in the sound channel. Therefore, to play a single frequency for a specified duration, you should issue freqDurationCmd followed immediately by quietCmd.

See **Manipulating a Sound That Is Playing** for further details on quietCmd.

When you use the freqDurationCmd command and are using a sampled sound as the voice, freqDurationCmd starts at the beginning of a sampled sound. The freqDurationCmd command plays the sound between the loop points specified in the sampled sound header to extend the sound to the specified duration. There must be an ending-point for the loop specified in the header in order for freqDurationCmd to work properly.

The structure of a freqDurationCmd command is slightly different from that of most other sound commands. The *param1* field contains the duration of the sound, specified in half-milliseconds. (A duration of 2000 represents a duration of 1 second. The maximum duration is a duration of 32,767, or about 16 seconds.) The *param2* field specifies the frequency of the sound. The frequency is specified as a MIDI value. The following Listing demonstrates the use of the freqDurationCmd command.

//Listing: Using the freqDurationCmd command

// Assuming inclusion of MacHeaders

#include <Sound.h>

// Prototype your routine like this prior to calling it

void MyPlayFreqDuration (void);

void MyPlayFreqDuration()

{

SndChannelPtr mySndChan; // pointer to a sound channel

SndCommand mySndCmd; // a sound command

OSErr myErr;

 // Prototype for error handling routine

 void DoError(OSErr);

```

mySndCmd.cmd = freqDurationCmd;
mySndCmd.param1 = 2000;           // duration in half-milliseconds
mySndCmd.param2 = 60;            // frequency

// play the sound
// assume that mySndChan points to a valid sound channel
myErr = SndDoCommand(mySndChan, &mySndCmd, false);
if ( myErr )
    DoError(myErr);
}

```

The Table below shows the decimal values that can be sent with a freqDurationCmd or freqCmd command. Middle C is represented by a value of 60. These values correspond to MIDI values.

To calculate a duration, use the following formula:

$\text{duration} = (2000 / (\text{repetitions per minute} / 60)) * \text{repetitions per sound}$

You can rest a channel for a specified duration by issuing a restCmd command. The duration, specified in half-milliseconds, is passed in the *param1* field of the sound command.

Table MIDI values

	A	A#	B	C	C#	D	D#	E	F	F#	G	G#
Interval 1					1	2	3	4	5	6	7	8
Interval 2	9	10	11	12	13	14	15	16	17	18	19	20
Interval 3	21	22	23	24	25	26	27	28	29	30	31	32
Interval 4	33	34	35	36	37	38	39	40	41	42	43	44
Interval 5	45	46	47	48	49	50	51	52	53	54	55	56
Interval 6	57	58	59	60	61	62	63	64	65	66	67	68
Interval 7	69	70	71	72	73	74	75	76	77	78	79	80
Interval 8	81	82	83	84	85	86	87	88	89	90	91	92
Interval 9	93	94	95	96	97	98	99	100	101	102	103	104
Interval 10	105	106	107	108	109	110	111	112	113	114	115	116
Interval 11	117	118	119	120	121	122	123	124	125	126	127	