Initializing Sound Channels Setting Channel Charateristics

When you first create a sound channel with **SndNewChannel**, you can request that the channel have certain characteristics as specified by a **sound-channel initialization parameter.** For example, to indicate that you want to allocate a channel capable of producing stereo sound, you might use the following code:

```
myErr=SndNewChannel(&mySndChan, sampledSynth, initStereo, NULL);
```

These are the currently recognized constants for the sound-channel initialization parameter.

```
initChanLeft        //left channel--sampledSynth only
initChanRight       //right channel--sampledSynth only
initChan0           //channel 1--wave table only
initChan1           //channel 2--wave table only
initChan2           //channel 3--wave table only
initChan3           //channel 4--wave table only
initMono            //mono channel--sampledSynth only
initStereo          //stereo channel--sampledSynth only
initMACE3           //3:1 compression--sampledSynth only
initMACE6           //6:1 compression--sampledSynth only
initNoInterp        //no linear interpolation
initNoDrop          //no drop-sample conversion}
```

Constant	Description
initChanLeft	Play sounds through the left channel of the Macintosh audio jack.
initChanRight	Play sounds through the right channel of the Macintosh audio jack.
initChan0	Play sounds through the first channel of the wave-table synthesizer.
initChan1	Play sounds through the second channel of the wave-table synthesizer.
initChan2	Play sounds through the third channel of the wave-table synthesizer.
initChan3	Play sounds through the fourth channel of the wave-table synthesizer.
initMono	Play sounds through both channels of the Macintosh audio jack and the internal speaker. This is the default channel mode.
initStereo	Play sounds through both channels of the Macintosh audio jack and the internal speaker. A stereo sound contains left and right samples that are interleaved (that is, left, right, left, right, and so forth). Note that some machines cannot play stereo sounds.

initMACE3	Assume that the sounds to be played through the channel are MACE 3:1 compressed. The loadCmd command and the **SndNewChannel** function calculate CPU loading based on MACE 3:1 overhead. A noncompressed sound plays normally, even through a channel that has been initialized for MACE.
initMACE6	Assume that the sounds to be played through the channel are MACE 6:1 compressed. The loadCmd command and the **SndNewChannel** function calculate CPU loading based on MACE 6:1 overhead. A noncompressed sound plays normally, even through a channel that has been initialized for MACE.
initNoInterp	Do not use linear interpolation when playing a sound back at a different frequency from the sound's recorded frequency. Using the *initNoInterp* initialization parameter decreases the CPU load for this channel. Sounds most affected by the absence of linear interpolation are sinusoidal sounds. Sounds least affected are noisy sound effects like explosions and screams.
initNoDrop	Do not use drop-sample conversion when playing a sound back.

Note: Most Macintosh computers play *only* the left channel of stereo sounds out the internal speaker. Some machines (for example, the Macintosh SE/30 and Macintosh IIsi) mix both channels together before sending a signal to the internal speaker. You can use the **Gestalt** function to determine if a particular machine mixes both left and right channels to the internal speaker. All models of the Macintosh, however, play stereo signals out the headphone jack.

Because MACE is extremely CPU-intensive, using the initMACE3 and initMACE6 options reserves considerably more time for a channel than does using the other options. If you can determine whether MACE sounds will be used for a given channel, then the CPU loading values will be much more accurate.

The initialization parameters are additive. To initialize a channel for stereo sound with no linear interpolation, simply pass an initialization parameter that is the sum of the desired characteristics, as follows:

```
myErr=SndNewChannel(&mySndChan, sampledSynth,
                    initStereo+initNoInterp, NULL);
```

Note that the call to **SndNewChannel** is really only a request that the **Sound Manager** open a channel having the desired characteristics. It is possible that the parameters requested cannot be provided without consuming too much CPU time. See **Managing the CPU Load** for a method of determining when a call to **SndNewChannel** succeeds. In general, you should initialize a sound channel for the most processor-intensive case (that is, monophonic sound with linear interpolation and MACE 3:1 compression) unless you know exactly what kind of sound is to be played.

When the **Sound Manager** does succeed in opening a new sound channel with the requested characteristics, it links that channel to the desired playback synthesizer. The synthesizer reacts to that command by allocating any private memory it needs and performing other necessary initialization procedures.

You can alter certain initialization parameters, even while a channel is actively playing a sound, by issuing the reInitCmd sound command. For example, you can change the output channel from left to right, as follows:

```
mySndCmd.cmd = reInitCmd;
mySndCmd.param1 = 0;                    //unused
mySndCmd.param2 = initChanRight;        //new init parameter

myErr= SndDoImmediate(mySndChan, &mySndCmd);
if (myErr != noErr) DoError(myErr);
```

The reInitCmd command accepts the initNoInterp constant to toggle linear interpolation on and off; it should be used with uncompressed sounds only. If an uncompressed sound is playing when you send a reInitCmd command with this constant, linear interpolation begins immediately. You can also pass initMono, initChanLeft, or initChanRight to pan to both channels, to the left channel, or to the right channel. This affects only monophonic sounds.

Note that the **Sound Manager** remembers the settings you pass and applies them to all further sounds played on that channel.