

Determining Features of Synthesizers

Getting synthesizer data

You can determine certain information about the capabilities of a synthesizer by using the **SndControl** function. For example, you can determine whether a particular synthesizer supports a particular initialization option (some synthesizers do not support all initialization options). This can be most useful if you want your application to run under the enhanced **Sound Manager** as well as under earlier versions where the playback synthesizers do not have the same output characteristics. By first determining whether the intended synthesizer supports the desired output characteristics, you can avoid requesting characteristics that are not available. Because you generally need to know about the capabilities of a synthesizer before you actually create a sound channel, you can call **SndControl** even if no channel has been created for the synthesizer.

To determine whether an initialization option is supported by a particular synthesizer, call the **SndControl** function and pass it the `availableCmd` command. The following code example illustrates how to determine if the sampled sound synthesizer supports stereo sound output.

Using the availableCmd command to check for stereo sound

```
// Assuming inclusion of MacHeaders
#include <Sound.h>

// Prototype your routine like this prior to calling it
Boolean StereoAvailable (void);

Boolean StereoAvailable()
{
    SndCommand mySndCmd;
    OSErr myErr;

    // Prototype for DoError function
    void DoError(OSErr);

    mySndCmd.cmd = availableCmd;
    mySndCmd.param1 = 0; // unused on input
    mySndCmd.param2 = initStereo; // test for stereo

    myErr = SndControl(sampledSynth, &mySndCmd);
    if ( myErr != noErr )
        DoError(myErr);

    return (mySndCmd.param1 != 0);
}
```

The **SndControl** function requires two parameters. The first parameter indicates the resource ID of the synthesizer whose characteristics are to be determined. The second parameter is a sound command. In the case illustrated, the `cmd` field of that sound command is set to availableCmd. The `param2` field of the sound command contains the initialization parameter in question. (The initialization parameters are discussed in the separate section entitled

Initializing Sound Channels) The *param1* field of the sound command is unused on input. If **SndControl** returns successfully, then *param1* contains 1 if the synthesizer has the requested characteristics and 0 otherwise.

To determine which version of a synthesizer is available, call the **SndControl** function with the versionCmd command. Neither *param1* nor *param2* of the sound command passed to **SndControl** is used on input. If the function returns successfully, the version is returned in *param2* of the sound command. For example, version 2.0 of a synthesizer would be returned as 0x00020000. The following code example illustrates how to use the versionCmd command.

Using the versionCmd command

```
// Assuming inclusion of MacHeaders
#include <Sound.h>

// Prototype routine to get version like this prior to calling it
void GetVersion(long *);

void GetVersion (long *version)
{
    SndCommand mySndCmd;           // the sound command record
    OSErr myErr; // Error checking variable

    // Prototype for error handling procedure
    void DoError(OSErr);

    mySndCmd.cmd = versionCmd;
    mySndCmd.param1 = 0;           // unused on input
    mySndCmd.param2 = 0;           // unused on input

    // determine version of sampled sound synthesizer
    myErr = SndControl(sampledSynth, &mySndCmd);
    if ( myErr )
        DoError(myErr);
    else
        *version = mySndCmd.param2;
}
```