

Sound

In this chapter, you will learn how to:

- Describe the components and measurements of sound
- Use digital audio to record, process, and edit sound
- Use MIDI and understand its attributes, especially relative to digitized audio
- Compare and contrast the use of MIDI and digitized audio in a multimedia production
- List the important steps and considerations in recording and editing digital audio
- Determine which audio file formats are best used in a multimedia project
- Cite the considerations involved in managing audio files and integrating them into multimedia projects

SOUND is perhaps the most sensuous element of multimedia. It is meaningful “speech” in any language, from a whisper to a scream. It can provide the listening pleasure of music, the startling accent of special effects, or the ambience of a mood-setting background. Some feel-good music powerfully fills the heart, generating emotions of love or otherwise elevating listeners closer to heaven. How you use the power of sound can make the difference between an ordinary multimedia presentation and a professionally spectacular one. Misuse of sound, however, can wreck your project. Try testing all 56 of your ringtones on a crowded bus: your fellow passengers will soon wreck your day.

The Power of Sound

When something vibrates in the air by moving back and forth (such as the cone of a loudspeaker), it creates waves of pressure. These waves spread like the ripples from a pebble tossed into a still pool, and when they reach your eardrums, you experience the changes of pressure, or vibrations, as sound. In air, the ripples propagate at about 750 miles per hour, or Mach 1 at sea level. Sound waves vary in sound pressure level (amplitude) and in frequency or pitch. Many sound waves mixed together form an audio sea of symphonic music, speech, or just plain noise.

Acoustics is the branch of physics that studies sound. Sound pressure levels (loudness or volume) are measured in **decibels (dB)**; a decibel measurement is actually the ratio between a chosen reference point on a logarithmic scale and the level that is actually experienced. When you quadruple the sound output power, there is only a 6 dB increase; when you make the sound 100 times more intense, the increase in dB is not hundredfold, but only 20 dB. A logarithmic scale (seen below) makes sense because humans perceive sound pressure levels over an extraordinarily broad dynamic range.



A logarithmic scale is also used for measuring the power of earthquakes (the Richter Scale) and stellar magnitudes (a first magnitude star is

100 times as bright as a typical sixth magnitude star, which is at the limit of human visual perception—magnitude 31.5 is the faintest visible light object detectable by the Hubble Space Telescope). You can recognize logarithmic scales because they use divisions that are multipliers (1, 10, 100, 1000) instead of additions (1, 2, 3, 4).

The decibel scale, with some examples, is shown in Table 4-1; notice the relationship between power (measured in watts) and dB.

dB	Watts	Example
195	25–40 million	Saturn rocket
170	100,000	Jet engine with afterburner
160	10,000	Turbojet engine at 7,000-pounds thrust
150	1,000	ALSETEX splinterless stun grenade
140	100	2 JBL2226 speakers pulling 2,400 watts inside an automobile
130	10	75-piece orchestra, at fortissimo
120	1	Large chipping hammer
110	0.1	Riveting machine
100	0.01	Automobile on highway
90	0.001	Subway train; a shouting voice
80	0.0001	Inside a 1952 Corvette at 60 mph
70	0.00001	Voice conversation; freight train 100 feet away
60	0.000001	Large department store
50	0.0000001	Average residence or small business office
40	0.00000001	Residential areas of Chicago at night
30	0.000000001	Very soft whisper
20	0.0000000001	Sound studio

Table 4-1 Typical Sound Levels in Decibels (dB) and Watts

Sound is energy, just like the waves breaking on a sandy beach, and too much volume can permanently damage the delicate receiving mechanisms behind your eardrums, typically dulling your hearing in the 6 kHz range. In terms of volume, what you hear subjectively is not what you hear objectively. The perception of loudness is dependent upon the frequency or pitch of the sound: at low frequencies, more power is required to deliver the same perceived loudness as for a sound at the middle or higher frequency ranges. You may feel the sound more than hear it. For instance, when the ambient noise level is above 90 dB in the workplace, people are likely to make increased numbers of errors in susceptible tasks—especially when there is a high-frequency component to the noise. When the level is above

80 dB, it is quite impossible to use a telephone. Experiments by researchers in residential areas have shown that a sound generator at 45 dB produces no reaction from neighbors; at 45 to 55 dB, sporadic complaints; at 50 to 60 dB, widespread complaints; at 55 to 65 dB, threats of community action; and at more than 65 dB, vigorous community action, possibly more aggressive than when you tested your ringtones on the bus. This neighborhood research from the 1950s continues to provide helpful guidelines for practicing rock musicians and multimedia developers today.

Human hearing is less able to identify the location from which lower frequencies are generated. In surround sound systems, subwoofers can be placed wherever their energy is most efficiently radiated (often in a corner), but midrange speakers should be carefully placed.

There is a great deal more to acoustics than just volume and pitch. If you are interested, many texts will explain why middle C on a cello does not sound like middle C on a bassoon; or why a five-year-old can hear a 1,000 Hz tone played at 20 dB, while an older adult with presbycusis (loss of hearing sensitivity due to age) cannot. Your use of sound in multimedia projects will not likely require highly specialized knowledge of harmonics, intervals, sine waves, notation, octaves, or the physics of acoustics and vibration, but you do need to know how to record and edit sounds on your computer and incorporate them into your multimedia work.

Digital Audio

Digital audio is created when you **represent the characteristics of a sound wave using numbers**—a process referred to as **digitizing**. You can digitize sound from a microphone, a synthesizer, existing recordings, live radio and television broadcasts, and popular CD and DVDs. In fact, you can digitize sounds from any **natural or prerecorded source**.

Digitized sound is **sampled sound**. Every **n th fraction of a second**, a **sample** of sound is taken and stored as **digital information in bits and bytes**. The **quality** of this digital recording depends upon how often the samples are taken (**sampling rate** or frequency, measured in **kilohertz**, or thousands of samples per second) and how many numbers are used to represent the value of each sample (**bit depth, sample size, resolution, or dynamic range**). The more often you take a sample and the more data you store about that sample, the finer the resolution and quality of the captured sound when it is played back. Since the quality of your audio is based on the **quality of your recording** and not the device on which your end user will play the audio, digital audio is said to be **device independent**.

The three sampling rates most often used in multimedia are **44.1 kHz (CD-quality)**, **22.05 kHz**, and **11.025 kHz**. Sample sizes are either **8 bits or 16 bits**. The larger the sample size, the more accurately the data will

describe the recorded sound. An 8-bit sample size provides 256 equal measurement units to describe the level and frequency of the sound in that slice of time. A 16-bit sample size, on the other hand, provides a staggering 65,536 equal units to describe the sound in that same slice of time. As you can see in Figure 4-1, slices of analog waveforms are sampled at various frequencies, and each discrete sample is then stored either as 8 bits or 16 bits (or more) of data.

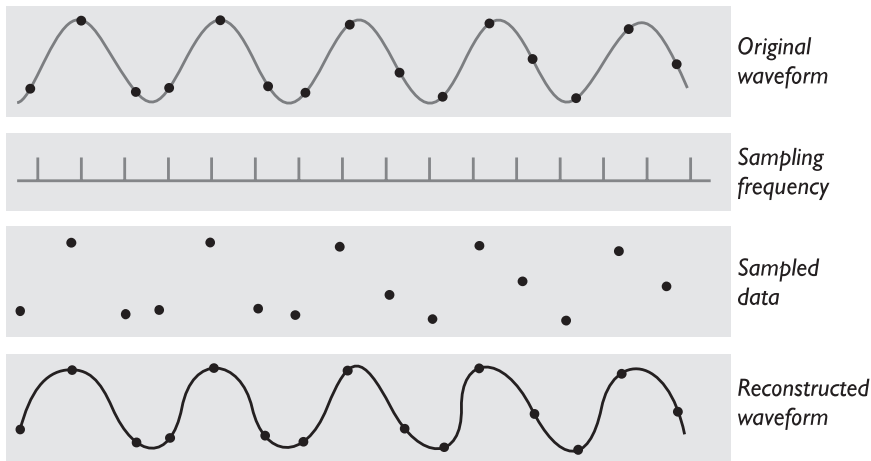


Figure 4-1 It is impossible to reconstruct the original waveform if the sampling frequency is too low.

The value of each sample is rounded off to the nearest integer (**quantization**), and if the amplitude is greater than the intervals available, clipping of the top and bottom of the wave occurs (see Figure 4-2). Quantization can produce an unwanted background hissing noise, and **clipping** may severely **distort** the sound.

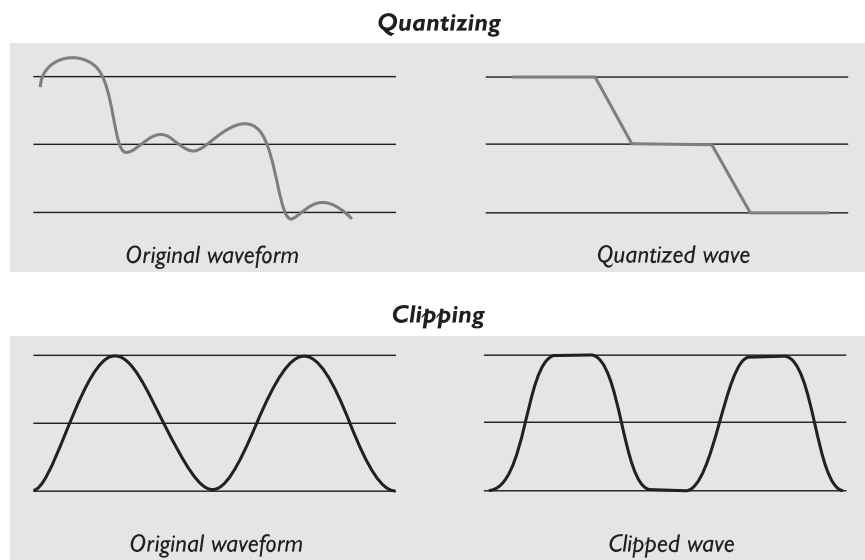
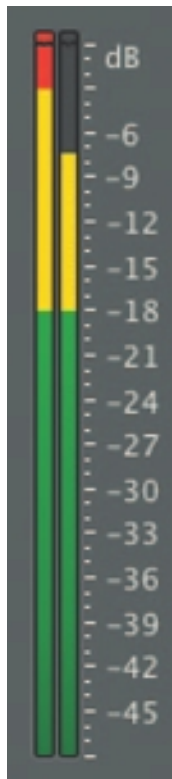


Figure 4-2 Examples of quantizing and clipping

I have a 20-second sample of a song which I play to my class at 8K, 22K, 44K, and 48K, and I have the students listen and compare quality. They comment that 8K does not sound all that bad *until* they hear the 44K and 48K.

They also see (hear) very little difference between 44K and 48K.

Dennis Woytek, Assistant
Professor of Multimedia
Technology,
Duquesne University



Making Digital Audio Files

Making digital audio files is fairly straightforward on most computers. Plug a **microphone** into the microphone jack of your computer. If you want to digitize archived analog source materials—music or sound effects that you have saved on videotape, for example—simply plug the “Line-Out” or “Headphone” jack of the device into the “Line-In” jack on your computer. Then use audio digitizing software such as **Audacity** (see Figure 4-3), to do the work.

You should focus on **two crucial aspects** of preparing digital audio files:

- Balancing the need for **sound quality against file size**. Higher quality usually means larger files, requiring longer download times on the Internet and more storage space on a CD or DVD.
- Setting proper **recording levels** to get a good, clean recording.

Setting Proper Recording Levels

A distorted recording sounds terrible. If the signal you feed into your computer is too “hot” to handle, the result will be an unpleasant crackling or background ripping noise. Conversely, recordings that are made at too low a level are often unusable because the amount of sound recorded does not sufficiently exceed the residual noise levels of the recording process itself. The trick is to set the right levels when you record.

Any good piece of digital audio recording and editing software will display digital meters to let you know how loud your sound is. Watch the meters closely during recording, and you’ll never have a problem. Unlike analog meters that usually have a 0 setting somewhere in the middle and extend up into ranges like +5, +8, or even higher, digital meters peak out. To avoid distortion, do not cross over this limit. If this happens, lower your volume (either by lowering the input level of the recording device or the output level of your source) and try again. Try to keep **peak levels** between **-3 and -10**. Any time you go over the peak, whether you can hear it or not, you introduce distortion into the recording. In digital meter displays, if you see red, you are over the peak.

Editing Digital Recordings

Once a recording has been made, it will almost certainly need to be edited. Shown in Figure 4-3 with its special effects menu, Audacity is a **free open-source sound editing application for Windows, Macintosh, and Linux** (<http://audacity.sourceforge.net>). With such a tool you can create sound tracks and digital mixes. The basic sound editing operations that most multimedia producers need are described in the paragraphs that follow.

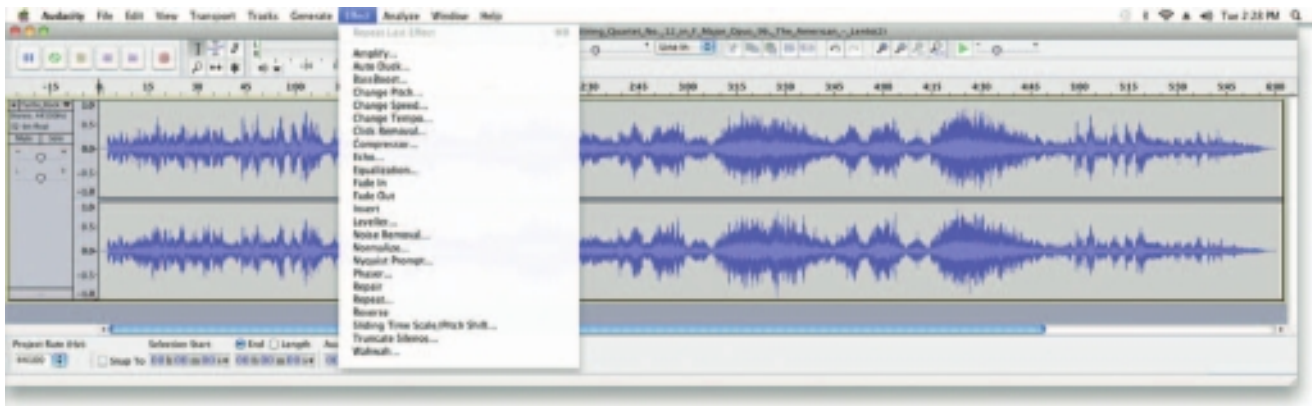


Figure 4-3 Audacity is an open-source, cross-platform editing tool for digitizing sound (<http://audacity.sourceforge.net/>).

Trimming Removing “dead air” or blank space from the front of a recording and any unnecessary extra time off the end is your first sound editing task. Trimming even a few seconds here and there might make a big difference in your file size. Trimming is typically accomplished by dragging the mouse cursor over a graphic representation of your recording and choosing a menu command such as **Cut, Clear, Erase, or Silence**.

Splicing and Assembly Using the same tools mentioned for trimming, you will probably want to remove the **extraneous noises** that inevitably creep into a recording. Even the most controlled studio voice-overs require touch-up. Also, you may need to assemble **longer recordings** by cutting and pasting together many shorter ones. In the old days, this was done by splicing and assembling actual pieces of magnetic tape.

Volume Adjustments If you are trying to assemble ten different recordings into a single sound track, there is little chance that all the segments will have the same volume. To provide a consistent volume level, select all the data in the file, and raise or lower the overall volume by a certain amount. Don’t increase the volume too much, or you may distort the file. It is best to use a sound editor to **normalize** the assembled audio file to a **particular level**, say 80 percent to 90 percent of maximum (without clipping), or about **-16 dB**. Without normalizing to this rule-of-thumb level, your final sound track might play too softly or too loudly. Even pros can leave out this important step. Sometimes an audio CD just doesn’t seem to have the same loudness as the last one you played, or it is too loud and you can hear clipping. Figure 4-4 shows the normalizing process at work in **Adobe’s Soundbooth**.

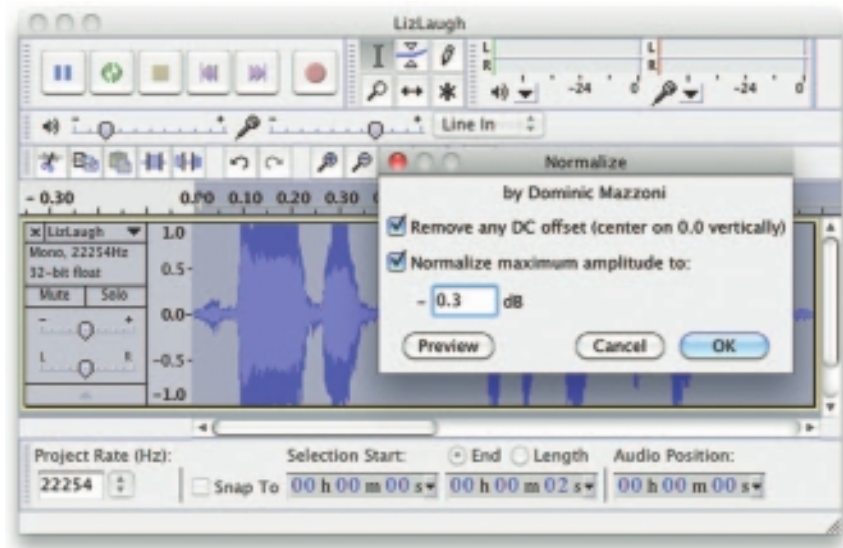


Figure 4-4 Normalizing evens out the sound level in an audio file.

Format Conversion In some cases, your digital audio editing software might read a format different from that read by your presentation or authoring program. Most sound editing software will save files in your choice of many formats, most of which can be read and imported by multimedia authoring systems. **Data may be lost when converting formats.** If, for example, you have a Digital Rights Management (DRM)-protected M4P file downloaded from the iTunes store and burn that file to an Audio CD track, the DRM data will be lost because the Audio CD format does not provide for DRM data. The now-unprotected tune on the CD can then be ripped into a playable **MP3** format.

Resampling or Downsampling If you have recorded and edited your sounds at 16-bit sampling rates but are using **lower rates and resolutions** in your project, you must **resample** or **downsample** the file. Your software will examine the existing digital recording and work through it to **reduce the number of samples.** This process may save considerable disk space.

Fade-ins and Fade-outs Most programs offer **enveloping capability**, useful for long sections that you wish to fade in or fade out gradually. This enveloping helps to **smooth** out the very beginning and the very end of a sound file.

Equalization Some programs offer **digital equalization (EQ)** capabilities that allow you to modify a **recording's frequency content** so that it sounds **brighter** (more high frequencies) or **darker** (low, ominous rumbles).

Time Stretching Advanced programs let you alter the length (in time) of a sound file without changing its pitch. This feature can be very useful, but watch out: most **time-stretching** algorithms will severely degrade the audio quality of the file if the length is altered more than a few percent in either direction.

Digital Signal Processing (DSP) Some programs allow you to process the signal with reverberation, multitap delay, chorus, flange, and other special effects using **digital signal processing (DSP)** routines.

Being able to process a sound source with effects can greatly add to a project. To create an environment by placing the sound inside a room, a hall, or even a cathedral can bring depth and dimension to a project. But a little can go a long way—do not overdo the sound effects!

TIP Once a sound effect is processed and mixed onto a track, it cannot be further edited, so always save the original so that you can tweak it again if you are not happy.

Reversing Sounds Another simple manipulation is to reverse all or a portion of a digital audio recording. Sounds, particularly spoken dialog, can produce a surreal, otherworldly effect when played backward.

Multiple Tracks Being able to edit and combine multiple tracks (for sound effects, voice-overs, music, etc.) and then merge the tracks and export them in a “final mix” to a single audio file is important.

File Size vs. Quality

Remember that the sampling rate determines the frequency at which samples will be taken for the recording. Sampling at higher rates (such as 44.1 kHz or 22.05 kHz) more accurately captures the high-frequency content of your sound. **Audio resolution** (such as 8- or 16-bit) determines the accuracy with which a sound can be digitized. Using more bits for the sample size yields a recording that sounds more like its original.

WARNING The higher the sound quality, the larger your file will be.

Stereo recordings are more lifelike and realistic because human beings have two ears. Mono recordings are fine but tend to sound a bit “flat” and uninteresting when compared with stereo recordings. Logically, to record stereo you need two microphones (left and right), and the sound file generated will require twice as much storage space as the mono file for the same length of play time.

Table 4-2 provides some commonly used sampling rates and resolutions, with resulting file sizes.

Sampling Rate	Resolution	Stereo or Mono	Bytes Needed for 1 Minute	Comments
44.1 kHz	16-bit	Stereo	10.5MB	CD-quality recording; the recognized standard of audio quality.
44.1 kHz	16-bit	Mono	5.25MB	A good trade-off for high-quality recordings of mono sources such as voice-overs.
44.1 kHz	8-bit	Stereo	5.25MB	Achieves highest playback quality on low-end devices such as most of the sound cards in Windows PCs.
44.1 kHz	8-bit	Mono	2.6MB	An appropriate trade-off for recording a mono source.
22.05 kHz	16-bit	Stereo	5.25MB	Darker sounding than CD-quality recording because of the lower sampling rate, but still full and “present” because of high bit resolution and stereo. Preferred for CD-ROM projects.
22.05 kHz	16-bit	Mono	2.5MB	Not a bad choice for speech, but better to trade some fidelity for a lot of disk space by dropping down to 8-bit.
22.05 kHz	8-bit	Stereo	2.6MB	A popular choice for reasonable stereo recording where full bandwidth playback is not possible.
22.05 kHz	8-bit	Mono	1.3MB	A thinner sound than the previous choice, but very usable. Any Macintosh or any MPC can play back this type of file. About as good as listening to your TV set.
11 kHz	8-bit	Stereo	1.3MB	At this low a sampling rate, there are few advantages to using stereo.
11 kHz	8-bit	Mono	650KB	In practice, probably as low as you can go and still get usable results; very dark and muffled.
5.5 kHz	8-bit	Stereo	650KB	Stereo not effective.
5.5 kHz	8-bit	Mono	325KB	About as good as a bad telephone connection.

Table 4-2 One-Minute Digital Audio Recordings at Common Sampling Rates and Resolutions

TIP The only reason to digitize audio at a higher specification than can be used by the target playback device is for archiving it. As playback technologies and bandwidth improve over time, you may wish (someday) for higher-quality original files when you upgrade a product. Save the originals!

Consumer-grade audio compact discs provide stereo at a sampling rate of 44.1 kHz and 16-bit resolution. Sound studios using high-end equipment digitally record and edit performances at much higher sampling rates and depths than this target distribution platform, and the final mix is downsampled before mass replication.

Audiophiles (listeners seriously interested in perfect sound reproduction) have driven a small market for very high-end equipment that can play back SACD (Super Audio CD) or DVD-Audio formats written on special audio-only DVDs that require dedicated players and a system with as many as five full-frequency speakers and a subwoofer. This sound is typically sampled at a depth of 24 bits and frequency of 96 kHz. It is said by some that, while the limit of human hearing may be about 21 kHz, the unheard higher frequency harmonics (easily heard by dogs to 60 kHz, bats to 120 kHz, and dolphins to 150 kHz) “flood” the brain with pleasure-causing endorphins and lead to a fuller sensory experience.

Here are the formulas for determining the size (in bytes) of a digital recording. For a monophonic recording:

$$\text{sampling rate} * \text{duration of recording in seconds} * (\text{bit resolution} / 8) * 1$$

For a stereo recording:

$$\text{sampling rate} * \text{duration of recording in seconds} * (\text{bit resolution} / 8) * 2$$

(Remember, sampling rate is measured in kHz, or thousand samples per second, so to convert from kHz to a whole number, you must multiply by 1,000. Resolution is measured in bits per sample. Since there are 8 bits in a byte, you have to divide the bit resolution by 8.) Thus the formula for a 10-second recording at 22.05 kHz, 8-bit resolution would be

$$22050 * 10 * 8 / 8 * 1$$

which equals 220,500 bytes. A 10-second stereo recording at 44.1 kHz, 16-bit resolution (meeting the CD-quality Red Book Audio standards—an international recording standard discussed later in this chapter) would be

$$44100 * 10 * 16 / 8 * 2$$

which equals 1,764,000 bytes. A 40-second mono recording at 11 kHz, 8-bit resolution would be

$$11000 * 40 * 8 / 8 * 1$$

which equals 440,000 bytes.

Fortunately, for hard disk storage requirements at least, and for ring-tone files sent over a mobile phone, user expectations of audio quality are somewhat lower than for Grammy Award-winning recordings. (See Vaughan’s Law of Multimedia Minimums later in this chapter.)

MIDI Audio

MIDI (Musical Instrument Digital Interface) is a communications standard developed in the early 1980s for electronic musical instruments and

computers. It allows music and sound synthesizers from different manufacturers to communicate with each other by sending messages along cables connected to the devices. MIDI provides a **protocol** for **passing detailed descriptions of a musical score, such as the notes, the sequences of notes, and the instrument that will play these notes**. But MIDI data is not digitized sound; it is a **shorthand representation of music stored in numeric form**. Digital audio is a recording, MIDI is a **score**—the first depends on the **capabilities of your sound system**, the other on the **quality of your musical instruments and the capabilities of your sound system**.

A MIDI file is a list of **time-stamped commands** that are recordings of musical actions (the pressing down of a piano key or a sustain pedal, for example, or the movement of a control wheel or slider). When sent to a MIDI playback device, this results in sound. A concise MIDI message can cause a complex sound or sequence of sounds to play on an instrument or synthesizer; so MIDI files tend to be significantly **smaller** (per second of sound delivered to the user) than equivalent digitized waveform files.

Composing your own original score can be one of the most creative and rewarding aspects of building a multimedia project, and MIDI is the quickest, easiest, and most flexible tool for this task. Yet creating an original MIDI score is hard work. Knowing something about music, being able to play a keyboard, and having a lot of good ideas are just the prerequisites to building a good score; beyond that, it takes time and musical skill to work with MIDI.

Happily, you can always hire someone to do the job for you. In addition to the talented MIDI composers who charge substantial rates for their services, many young composers are also available who want to get into multimedia. With a little research, you can often find a MIDI musician to work for limited compensation. Remember, however, that you often get what you pay for.

The process of creating MIDI music is quite different from digitizing existing recorded audio. If you think of digitized audio as analogous to a bitmapped graphic image (both use sampling of the original analog medium to create a digital copy), then MIDI is analogous to structured or vector graphics (both involve instructions provided to software to be able to re-create the original on the fly). For digitized audio you simply play the audio through a computer or device that can digitally record the sound. To make MIDI scores, however, you will need **notation software** (see Figure 4-5), **sequencer software** (see Figure 4-6), and a **sound synthesizer** (typically built into the software of multimedia players in most computers and many handheld devices). A **MIDI keyboard** is also useful for simplifying the creation of musical scores.



Figure 4-5 Notation and composition software such as Sibelius provides a way for composers and musicians to create and arrange scores using MIDI instruments.



Figure 4-6 Sequencer software such as Pro Tools allows you to record, edit, and save music generated from a MIDI keyboard or instruments and blend it with digital audio.

Rather than recording the sound of a note, MIDI software creates data about each note as it is played on a MIDI keyboard (or another MIDI device)—which note it is, how much pressure was used on the keyboard to play the note, how long it was sustained, and how long it takes for the note to decay or fade away, for example. This information, when played back through a MIDI device, allows the note to be reproduced exactly. Because the quality of the playback depends upon the end user’s MIDI device rather than the recording, MIDI is **device dependent**. The sequencer software quantizes your score to adjust for timing inconsistencies (a great feature for those who can’t keep the beat), and it may also print a neatly penned copy of your score to paper.

An advantage of structured data such as MIDI is the ease with which you can edit the data. Let’s say you have a piece of music being played on a honky-tonk piano, but your client decides he wants the sound of a soprano saxophone instead. If you had the music in digitized audio, you would have to re-record and redigitize the music. When it is in MIDI data, however, there is a value that designates the instrument to be used for playing back the music. To change instruments, you just change that value. Instruments that you can synthesize are identified by a **General MIDI** numbering system that ranges from 0 to 127 (see Table 4-3). Until this system came along, there was always a risk that a MIDI file originally composed with, say, piano, electric guitar, and bass, might be played back with piccolo, tambourine, and glockenspiel if the ID numbers were not precisely mapped to match the original hardware setup. This was usually the case when you played a MIDI file on a MIDI configuration different from the one that recorded the file.

ID	Sound	ID	Sound
0	Acoustic grand piano	16	Hammond organ
1	Bright acoustic piano	17	Percussive organ
2	Electric grand piano	18	Rock organ
3	Honky-tonk piano	19	Church organ
4	Rhodes piano	20	Reed organ
5	Chorused piano	21	Accordion
6	Harpsichord	22	Harmonica
7	Clarinet	23	Tango accordion
8	Celesta	24	Acoustic guitar (nylon)
9	Glockenspiel	25	Acoustic guitar (steel)
10	Music box	26	Electric guitar (jazz)
11	Vibraphone	27	Electric guitar (clean)
12	Marimba	28	Electric guitar (muted)
13	Xylophone	29	Overdriven guitar
14	Tubular bells	30	Distortion guitar
15	Dulcimer	31	Guitar harmonics

Table 4-3 General MIDI Instrument Sounds

ID	Sound	ID	Sound	ID	Sound
32	Acoustic bass	80	Lead 1 (Square)	Percussion Keys	
33	Electric bass (finger)	81	Lead 2 (Sawtooth)	35	Acoustic bass drum
34	Electric bass (pick)	82	Lead 3 (Calliope lead)	36	Bass drum 1
35	Fretless bass	83	Lead 4 (Chiff lead)	37	Side stick
36	Slap bass 1	84	Lead 5 (Charang)	38	Acoustic snare
37	Slap bass 2	85	Lead 6 (Voice)	39	Hand clap
38	Synth bass 1	86	Lead 7 (Fifths)	40	Electric snare
39	Synth bass 2	87	Lead 8 (Bass + lead)	41	Low-floor tom
40	Violin	88	Pad 1 (New Age)	42	Closed high-hat
41	Viola	89	Pad 2 (Warm)	43	High-floor tom
42	Cello	90	Pad 3 (Polysynth)	44	Pedal high-hat
43	Contrabass	91	Pad 4 (Choir)	45	Low tom
44	Tremolo strings	92	Pad 5 (Bowed)	46	Open high-hat
45	Pizzicato strings	93	Pad 6 (Metallic)	47	Low-mid tom
46	Orchestral harp	94	Pad 7 (Halo)	48	High-mid tom
47	Timpani	95	Pad 8 (Sweep)	49	Crash cymbal 1
48	String ensemble 1	96	FX 1 (Rain)	50	High tom
49	String ensemble 2	97	FX 2 (Soundtrack)	51	Ride cymbal 1
50	SynthStrings 1	98	FX 3 (Crystal)	52	Chinese cymbal
51	SynthStrings 2	99	FX 4 (Atmosphere)	53	Ride bell
52	Choir aahs	100	FX 5 (Brightness)	54	Tambourine
53	Voice oohs	101	FX 6 (Goblins)	55	Splash cymbal
54	Synth voice	102	FX 7 (Echoes)	56	Cowbell
55	Orchestra hit	103	FX 8 (Sci-Fi)	57	Crash cymbal 2
56	Trumpet	104	Sitar	58	Vibraslap
57	Trombone	105	Banjo	59	Ride cymbal 2
58	Tuba	106	Shamisen	60	High bongo
59	Muted trumpet	107	Koto	61	Low bongo
60	French horn	108	Kalimba	62	Mute high conga
61	Brass section	109	Bagpipe	63	Open high conga
62	Synth brass 1	110	Fiddle	64	Low conga
63	Synth brass 2	111	Shanai	65	High timbale
64	Soprano saxophone	112	Tinkle bell	66	Low timbale
65	Alto saxophone	113	Agogo	67	High agogo
66	Tenor saxophone	114	Steel drums	68	Low agogo
67	Baritone saxophone	115	Wood block	69	Cabasa
68	Oboe	116	Taiko drum	70	Maracas
69	English horn	117	Melodic tom	71	Short whistle
70	Bassoon	118	Synth drum	72	Long whistle
71	Clarinet	119	Reverse cymbal	73	Short guiro
72	Piccolo	120	Guitar fret noise	74	Long guiro
73	Flute	121	Breath noise	75	Claves
74	Recorder	122	Seashore	76	High wood block
75	Pan flute	123	Bird tweet	77	Low wood block
76	Bottle blow	124	Telephone ring	78	Mute cuica
77	Shakuhachi	125	Helicopter	79	Open cuica
78	Whistle	126	Applause	80	Mute triangle
79	Ocarina	127	Gunshot	81	Open triangle

Table 4-3 General MIDI Instrument Sounds (Continued)

TIP Making MIDI files is as complex as recording good sampled files; so it often pays to find someone already set up with the equipment and skills to create your score, rather than investing in the hardware, software, and the learning curve. Once you have gathered your audio material, you will need to edit it to precisely fit your multimedia project. As you edit, you will continue to make creative decisions. Because it is so easy to edit MIDI data, you can make many fine adjustments to your music as you go along.

Since MIDI is device dependent and the quality of consumer MIDI playback hardware varies greatly, MIDI's true place in multimedia work may be as a **production tool** rather than a delivery medium. MIDI is by far the best way to create original music, so use MIDI to get the flexibility and creative control you want. Then, once your music is completed and fits your project, lock it down for delivery by turning it into digital audio data.

In addition to describing the instrument and the note, MIDI data can also describe the **envelope** of the sound: the **attack** (how quickly a sound's volume increases), the **sustain** (how long the sound continues), and the **decay** (how quickly the sound fades away).

TIP Test your MIDI files thoroughly by playing them back on a variety of hardware devices or with different MIDI players before you incorporate them into your multimedia project. Windows Media Player and QuickTime will play MIDI on your computer.

MIDI vs. Digital Audio

In contrast to MIDI data, digital audio data is the actual representation of a sound, stored in the form of thousands of individual numbers (*samples*). The digital data represents the instantaneous amplitude (or loudness) of a sound at discrete slices of time. MIDI data is to digital audio data what vector or drawn graphics are to bitmapped graphics. That is, MIDI data is device dependent; digital data is not. Just as the appearance of vector graphics differs depending on the printer device or display screen, the sounds produced by MIDI music files depend on the particular MIDI device used for playback. Similarly, a roll of perforated player-piano score played on a concert grand would sound different than if played on a honky-tonk piano. Digital data, on the other hand, produces sounds that are more or less identical regardless of the playback system. The MIDI standard lets instruments communicate in a well-understood language.

MIDI has several advantages over digital audio and two huge disadvantages. First, the advantages:

- MIDI files are much more **compact** than digital audio files, and the size of a MIDI file is **completely independent of playback quality**. In

general, MIDI files will be 200 to 1,000 times smaller than CD-quality digital audio files. Because MIDI files are small, they don't take up as much memory, disk space, or bandwidth.

- Because they are small, MIDI files embedded in web pages **load** and **play** more **quickly** than their digital equivalents.
- In some cases, if the MIDI sound source you are using is of high quality, MIDI files may **sound better** than digital audio files.
- You can **change the length** of a MIDI file (by varying its tempo) without changing the pitch of the music or degrading the audio quality. MIDI data is completely **editable**—right down to the level of an individual note. You can manipulate the smallest detail of a MIDI composition (often with submillisecond accuracy) in ways that are impossible with digital audio.
- Because they represent the pitch and length of notes, MIDI files can generally be converted to musical notation, and vice versa. This is useful when you need a printed score; in reverse, you can scan a printed score and convert it to MIDI for tweaking and editing.

Now for MIDI's disadvantages:

- Because MIDI data **does not represent sound but musical instruments**, you can be certain that playback will be accurate only if the MIDI playback device is identical to the device used for production. Imagine the emotional humming chorus from *Madame Butterfly* sung by a chorus of Budweiser frogs—same score, wrong instrument. Even with the General MIDI standard (see the General MIDI table of instrument sounds in Table 4-3), the sound of a MIDI instrument varies according to the electronics of the playback device and the sound generation method it uses.
- Also, MIDI **cannot easily be used to play back spoken dialog**, although expensive and technically tricky digital samplers are available.

In general, **use MIDI** in the following circumstances:

- **Digital audio won't work** because you don't have enough memory or bandwidth.
- You have a **high-quality MIDI sound source**.
- You have **complete control over the machines** on which your program will be delivered, so you know that your users will have high-quality MIDI playback hardware.
- You don't need **spoken dialog**.

The most important advantage of digital audio is its consistent playback quality, but this is where MIDI is the least reliable! With digital audio you can be more confident that the audio track for your multimedia project will sound as good in the end as it did in the beginning when you created it. For this reason, it's no surprise that digital audio is used far more frequently than MIDI data for multimedia sound delivery.

There are two additional and often more compelling reasons to work with digital audio:

- A wider selection of application software and system support for digital audio is available for both the Macintosh and Windows platforms.
- The preparation and programming required for creating digital audio do not demand knowledge of music theory, while working with MIDI data usually does require a modicum of familiarity with musical scores, keyboards, and notation, as well as audio production.

In general, use digital audio in the following circumstances:

- You don't have control over the playback hardware.
- You have the computing resources and bandwidth to handle digital files.
- You need spoken dialog.

Multimedia System Sounds

You can use sound right off the bat on your computer because beeps and warning sounds are available as soon as you install the operating system. Open the Sound Control Panel to listen to your system sounds, change them, or make a new, custom sound (see Figure 4-7).

In Windows, system sounds are WAV files, and they reside in the Windows\Media subdirectory. System event sounds include start.wav, chimes.wav, chord.wav, ding.wav, logoff.wav, notify.wav, recycle.wav, tada.wav, and the Microsoft sound.wav that typically plays when Windows starts up.

As you can see in Figure 4-7, you can assign these sounds to system events such as Windows startup, warnings from other applications, or clicks outside of an open dialog box (which causes the default beep in Windows). And you can create schemes of sounds and select a particular scheme according to your mood. You can also add your own sound files and install them so they play when system events occur: place the WAV sound files into your ~\Windows\Media directory and use the Sound Control Panel to select them.

In OS X on a Macintosh, you can only change your system alert sound. Put your custom sound file (in AIF format) into ~/System/Library/Sounds, then select it in the Sound preference pane.

TIP If you are new to computers, your first multimedia sound experience might be simply finding one of these system sounds in the Sound Control Panel and testing it.

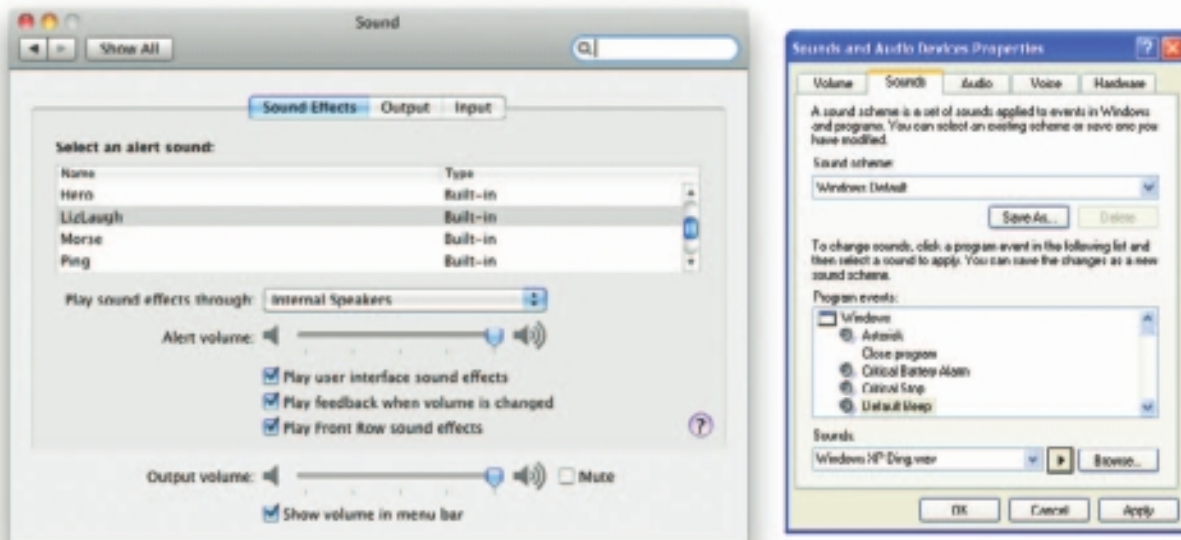


Figure 4-7 Sound Control Panels for Macintosh and Windows

Audio File Formats

When you create multimedia, it is likely that you will deal with file formats and translators for text, sounds, images, animations, or digital video clips. A sound file's format is simply a recognized methodology for organizing and (usually) compressing the digitized sound's data bits and bytes into a data file. The structure of the file must be known, of course, before the data can be saved or later loaded into a computer to be edited and/or played as sound. **The file name extension identifies which method of storage is used.**

There are many ways to store the bits and bytes that describe a sampled waveform sound. The method used for consumer-grade music CDs is **Linear Pulse Code Modulation (LPCM)**, often shortened to PCM. An audio CD provides up to 80 minutes of playing time, which is enough for a slow-tempo rendition of Beethoven's Ninth Symphony. Incidentally, being able to contain Beethoven's Ninth is reported to have been Philips's and Sony's actual size criterion during early research and development for determining the length of the sectors and ultimately the physical size of the compact disc format itself. The **CD-ROM/XA (extended architecture)** format for reading and writing CDs was developed later so you could put several recording sessions of music or data onto a single CD-R (recordable) disc. LPCM tracks from an audio CD are usually converted and stored on a computer in uncompressed **AIFF** (Audio Interchange File Format) or **wave format (WAV)** files when copied from the CD.

AIFF is historically used for Macintosh sound files. The WAV format was introduced by Microsoft when Windows was first released. Both formats contain uncompressed sound data. But there are huge numbers of sound file formats and “multimedia containers” that store sound data (more than three hundred different file name extensions are used for sound files), and often a converter is required to read or write sound files in the format you need. Hoo Technologies (www.hootech.com) offers MP3 to SWF, SWF/FLV to MP3, AIFF to MP3, MIDI to MP3, WMA to MP3, WAV to MP3, and OGG to MP3 converters. Their AIFF to MP3 converter will read the following formats: 3G2, 3GP, 3GP2, 3GPP, 4XM, AAC, AC3, ADX, AFC, AIF, AIFC, AIFF, ALAW, AMR, AMV, APE, ASF, AU, AVI, AWB, CAF, CDA, CDATA, DIF, DIVX, DTS, DV, DVD, DVR-MS, DXA, FLAC, FLC, FLI, FLIC, FLV, FLX, GSM, GXF, H261, H263, H263+, H264, IT, KAR, M1A, M1V, M2A, M2TS, M2V, M4A, M4B, M4V, MID, MIDI, MJ2, MJPEG, MJPG, MKA, MKV, MLP, MLV, MMF, MO3, MOD, MOV, MP+, MP1, MP2, MP3, MP4, MPA, MPC, MPE, MPEG, MPG, MPGA, MPP, MPV, MTM, MTS, MTV, MVI, MXF, NSA, NSV, NUT, NUV, OGA, OGG, OGM, OGV, OGX, OMA, PSP, PSX, PVA, QCP, QT, RA, RAM, RM, RMI, RMVB, ROQ, RPL, S3M, SDP, SHN, SMK, SND, SOL, SPX, STR, SWF, TS, TTA, UMX, VFW, VID, VMD, VOB, VOC, VQF, W64, WAV, WAVE64, WM, WMA, WMD, WMV, WV, XA, XM, XVID, XWMV, and YUV. And it will output to MP3, WAV, WMA, AAC, MP4, M4A (MPEG-4 Audio), M4B (MPEG-4 AudioBook), OGG, AMR, and AWB formats. But rest easy—you will likely only work with a handful of sound file types.

The MP3 format was developed by the Moving Picture Experts Group (**MPEG**) and evolved during the 1990s into the most common method for storing consumer audio. It incorporates a “lossy” compression algorithm to save space. An audio CD, for example, may hold an hour or so of uncompressed LPCM sound. That same CD, using MP3 compression, can store almost seven hours of the same music, but with a slight loss of quality. WMA (Windows Media Audio) is a proprietary Microsoft format developed to improve MP3. OGG was developed as an open-source and royalty-free “container” for sound compressed using Vorbis algorithms similar to MP3—because the Vorbis sound data resides within an Ogg container, these audio files are normally called “Ogg Vorbis.”

MP4 is a format based on Apple’s **QuickTime movie (.mov)** “container” model and is similar to the MOV format, which stores various types of media, particularly time-based streams such as audio and video. The mp4 extension is used when the file streams audio and video together. The m4a extension is used when the file contains only audio data. M4p files contain only audio, but are encrypted for Digital Rights Management (DRM). M4r files are used for ringtones on Apple’s iPhone. Other GSM

(Global System for Mobile Communications) mobile phones use 3gp files for their ringtones, a format also based on the MPG container model.

The AAC (Advanced Audio Coding) format, which is part of the MP4 model, was adopted by Apple's iTunes store, and many music files are commercially available in this format. AAC is the default format for iPod, iPhone, PlayStation, Wii, Dsi, and many mobile phones including Motorola, Nokia, Philips, Samsung, Siemens, and Sony Ericsson. The SWF format is a container for vector-based graphics and animations, text, video, and sound delivered over the Internet. Typically created using Adobe's Flash, SWF files require a plug-in or player be installed in the user's browser. Adobe claims that the Flash Player is installed in more than 98 percent of Web users' browsers and in more than 800 million handsets and mobile devices. Flash video files (FLV) contain both a video stream and an audio stream, and the FLV format has been adopted by YouTube, Google, Yahoo, Reuters.com, BBC.com, CNN.com, and other news providers for Internet delivery of content.

TIP *The most common sound formats you might use are wav, aif, aac, flv, mp3, mp4, mov, swf, wma, ogg, or for ringtones, m4r, aac, midi, mmf, 3g2, 3gp, 3gp2, and 3gpp. Be sure your audio software can read and write the formats you need.*

A **codec** (compressor-decompressor) is software that compresses a stream of audio or video data for storage or transmission, then decompresses it for playback. There are many codecs that do this with special attention to the quality of music or voice after decompression. Some are “lossy” and trade quality for significantly reduced file size and transmission speed; some are “lossless,” so original data is never altered. While editing your audio files, be sure to save your files using a lossless format or codec—with repetitive saves in a lossy format, you will notice a quality degradation each time. A container format such as MP4, MOV, or OGG may encapsulate data streams that use one of many codecs available in that container.

Vaughan's Law of Multimedia Minimums

A classic physical anthropology law (Liebig's Law of the Minimum) proposes that the evolution of eyesight, locomotor speed, sense of smell, or any other species trait will cease when that trait becomes sufficiently adequate to meet the survival requirements of the competitive environment. If the trait is good enough, the organism expends no more effort improving it. Thus, if consumer-grade electronics and

Vaughan's Law of Multimedia Minimums

There is an acceptable minimum level of adequacy that will satisfy the audience, even when that level may not be the best that technology, money, or time and effort can buy.

a handheld microphone are good enough for making your sound, and if you, your client, and your audience are all satisfied with the results, conserve your energy and money and avoid any more expenditure. And keep this Law of Minimums in mind when you make all your trade-off decisions involving other areas of high technology and multimedia, too.

Adding Sound to Your Multimedia Project

The original 128K Macintosh, released in January 1984, was technically a multimedia-capable machine. It displayed bitmapped graphics (albeit in black and white) and, more significantly, boasted 8-bit digital audio capability right on the motherboard. In fact, the very first Macintosh actually introduced itself by voice when it was unveiled by Steve Jobs.

Here's a little history: In order to use the Apple moniker, the original founders of Apple Computer, Inc., worked out an arrangement with the Beatles (yes, *those* Beatles). One part of that agreement stipulated that Apple Computer, Inc., would never venture into the music business. To Steve Jobs and Steve Wozniak, working out of their garage in the late 1970s on a machine that could barely manage a convincing system beep, that clause probably seemed a harmless one. Little did they know that years later their computer and the Apple itself would become the most popular provider of music in the world through its iTunes facility. The company did finally pay representatives of the Beatles about \$30 million to settle the issue once and for all.

Whether you're working on a Macintosh or in Windows, you will need to follow certain steps to bring an audio recording into your multimedia project. Here is a **brief overview** of the process:

1. Determine the file formats that are compatible with your multimedia authoring software and the delivery medium(s) you will be using (for file storage and bandwidth capacity).
2. Determine the sound playback capabilities (codecs and plug-ins) that the end user's system offers.
3. Decide what kind of sound is needed (such as background music, special sound effects, and spoken dialog). Decide where these audio events will occur in the flow of your project. Fit the sound cues into your storyboard (see Chapter 10), or make up a cue sheet.
4. Decide where and when you want to use either digital audio or MIDI data.
5. Acquire source material by creating it from scratch or purchasing it.

6. Edit the sounds to fit your project.
7. Test the sounds to be sure they are timed properly with the project's images. This may involve repeating steps 1 through 4 until everything is in sync.

When it's time to import your compiled and edited sounds into your project, you'll need to know how your particular multimedia software environment handles sound data. Each multimedia authoring program or web browser handles sound a bit differently, but the process is usually fairly straightforward: just tell your software which file you want to play and when to play it. This is usually handled by an importing or "linking" process during which you identify the files to play.

Scripting languages such as revTalk (RunRev), Lingo (Director), and ActionScript (Flash) provide a greater level of control over audio playback, but you'll need to know about the programming language and environment. In multimedia authoring environments, it is usually a simple matter to play a sound when the user clicks a button, but this may not be enough. If the user changes screens while a long file is playing, for example, you may need to program the sound to stop before leaving the current screen. If the file to be played cannot be found, you may need to code an entire section for error handling and file location. Sample code is generally provided in both printed and online documentation for software that includes sound playback. For web pages, you will need to embed a player and point to your sound file using HTML code.

Space Considerations

The substantial amount of digital information required for high-quality sound takes up a lot of storage space, especially when the quantity is doubled for two-channel stereo. It takes about 1.94MB to store 11 seconds of uncompressed stereo sound.

If monaural sound is adequate for your project, you can cut your storage space requirement in half or get double the playing time in the same memory space. With compression codecs, you might be able to store the sound in one-eighth the space, but you will lose some fidelity. Further, to conserve space you can downsample, or reduce the number of sample slices you take in a second. Many multimedia developers use 8-bit sample sizes at 22.05 kHz sampling rates because they consider the sound to be good enough (about the quality of AM radio), and they save immense amounts of digital real estate.

The following formula will help you estimate your storage needs. If you are using two channels for stereo, double the result.

$$(\text{sampling rate} * \text{bits per sample}) / 8 = \text{bytes per second}$$

If you prefer to solve for kilobytes (KB), not bytes, then try:

$$\text{sample rate} * \text{sample size} / 8 * \# \text{ seconds} * 2 \text{ (if stereo)} = \text{file size in KB}$$

For example, 60 seconds of stereo in Red Book Audio:

$$44.1 * 16 / 8 * 60 * 2 = 10,584\text{KB} \approx 10.59\text{MB}$$

This is an approximate result using 1000 instead of 1024 bytes per KB, but yielding the quick handy answer “...about ten and a half megabytes.”

You face important trade-offs when deciding how to manage digitized sound in your multimedia project. How much sound quality can you sacrifice in order to reduce storage? What compression techniques make sense? Will compressed sound work in your authoring platform? What is good enough but not amateurish? Can you get away with 8 bits at 11.025 kHz for voice mail, product testimonials, and voice-overs and then switch to higher sampling rates for music?

Many people feel that MP3s files sampled at 128 Kbps provide decent audio quality for music, especially when played through small speakers. For better quality, sample your music at 192 Kbps. Because the human voice does not use a wide range of frequencies, you can sample speech or voice at 96 Kbps or even 64 Kbps.

TIP *The sound of the human voice comes from one point (the mouth), so there is not much to gain by recording (or playing) it in stereo.*

Audio Recording

If your project requires CD-quality digitized sound at 44.1 kHz and 16 bits, you should hire a sound studio. High-fidelity sound recording is a specialized craft, a skill learned in great part by trial and error, much like photography. If you do decide to do it yourself at CD-quality levels, be prepared to invest in an acoustically treated room, high-end amplifiers and recording equipment, and expensive microphones.

As already stated, there are many trade-offs involved in making multimedia. For example, if you are satisfied with 22.05 kHz in your project or are constrained to this rate by storage considerations, any consumer-grade digital or analog recorder of reasonable quality will do fine. This, of course, also applies to conversations recorded from the telephone, where a sampling rate of 11.025 kHz is adequate. Noise reduction circuits and metal

With the collaboration of composer Dave Soldier, Komar & Melamid's Most Wanted Painting project (www.diacenter.org/km/index.html) was extended into the realm of music. A poll, written by Dave Soldier, was conducted on Dia's web site (www.diacenter.org). Approximately 500 visitors took the survey. Dave Soldier and Nina Mankin used the survey results to write music and lyrics for the Most Wanted and Most Unwanted songs.

A Note from the Composer

This survey confirms the hypothesis that popular music indeed provides an accurate estimate of the wishes of the vox populi. The most favored ensemble, determined from a rating by participants of their favorite instruments in combination, comprises a moderately sized group (three to ten instruments) consisting of guitar, piano, saxophone, bass, drums, violin, cello, synthesizer, with low male and female vocals singing in rock/r&b style. The favorite lyrics narrate a love story, and the favorite listening circumstance is at home. The only feature in lyric subjects that occurs in both most wanted and unwanted categories is "intellectual stimulation." Most participants desire music of moderate duration (approximately 5 minutes), moderate pitch range, moderate tempo, and moderate to loud volume, and display a profound dislike of the alternatives. If the survey provides an accurate analysis of these factors for the population, and assuming that the preference for each factor follows a Gaussian (bell-curve) distribution, the combination of these qualities, even to the point of sensory overload and stylistic dis cohesion, will result in a musical work that will be unavoidably and uncontrollably "liked" by 72 plus or minus 12 percent (standard deviation; Kolmogorov-Smirnov statistic) of listeners. The most unwanted music is over 25 minutes long, veers wildly between loud and quiet sections, between fast and slow tempos, and features timbres of extremely high and low pitch, with each dichotomy presented in abrupt transition. The most unwanted orchestra was determined to be large and features the accordion and bagpipe (which tie at 13 percent as the most unwanted instrument), banjo, flute, tuba, harp, organ, synthesizer (the only instrument that appears in both the most wanted and most unwanted ensembles). An operatic soprano raps and sings atonal music, advertising jingles, political slogans, and "elevator" music, and a children's choir sings jingles and holiday songs. The most unwanted subjects for lyrics are cowboys and holidays, and the most unwanted listening circumstances are involuntary exposure to commercials and elevator music. Therefore, it can be shown that if there is no covariance—someone who dislikes bagpipes is as likely to hate elevator music as someone who despises the organ, for example—fewer than 200 individuals of the world's total population would enjoy this piece.

Dave Soldier, composer and musician, who provides the Most Wanted Song and the Most Unwanted Song on a CD at www.diacenter.org/km/musiccd.html

tapes are helpful to remove hiss, but at a sampling rate of 22.05 kHz you are only going to be digitizing audio frequencies as high as about 11 kHz, anyway. Both the high and low ends of the audio hearing spectrum are therefore less important to you, and that is OK, because those areas are precisely the add-value focus of very elaborate and expensive consumer equipment.

Digital audio tape (DAT) systems provide a tape-based 44.1 kHz, 16-bit record and playback capability. You may, however, find that DAT is high-fidelity overkill for your needs, because the recordings are too accurate, precisely recording glitches, background noises, microphone pops, and coughs from the next room. A good editor can help reduce the impact of these noises, but at the cost of time and money.

Mobile phones can often record audio (and video), and applications and hardware attachments are available to manage external microphones and file transfer. USB and flash memory recorders range in quality, some suitable for voice only, some generating compressed MP3 files, and some recording in CD-quality stereo. Recordings can be directly downloaded as digital files using a USB cable or flash memory card reader.



Keeping Track of Your Sounds

In an elaborate project with many sounds, it is important to maintain a good database, keeping a physical track of your original material—just in case you need to revert to it when your disk drive crashes or you accidentally delete the work file. A database is particularly important because you may need to give your sound files such unhelpful names as `janesEyesOpenWide.aiff` or `Chapter11inSpanish03.wav`; these names contain some clues about the files' actual content, but you may need a more descriptive cross-reference. You don't want to have to load and play many sound files just to find the one you need.

Audio CDs

The method for digitally encoding the high-quality stereo of the consumer CD music market is an international standard, called ISO 10149. This is

also known as the **Red Book Audio** standard (derived simply from the color of the standard's book jacket). Developers of this standard claim that the digital audio sample size and sampling rate of Red Book Audio (16 bits at 44.1 kHz) allow accurate reproduction of all the sounds that humans can hear. Until recently, dedicated professional sound-studio equipment was used for this high-fidelity recording; today most off-the-shelf computers will record and play 16-bit sampled sound at 44.1 kHz and at 48 kHz. Converter and burning software such as Toast and CD-Creator from Roxio can translate the digital files of Red Book Audio found on consumer compact discs directly into a digital sound file formats such as MP3 or WAV.

Unlike DVDs, audio CDs do not contain information about artists, titles, or tracklists of songs. But player software such as Apple iTunes and AOL Winamp will automatically link to a database on the Internet when you insert a music CD. The precise length of your CD's Table of Contents (TOC) is then matched against the known TOC length for more than five million CDs containing more than 60 million songs. When it finds a match, the database service sends back what it knows about the CD you inserted. The database, formerly known as the Compact Disc Database or CDDb, was built up over the years by fans from all over the world submitting information about their favorite CDs. The database is currently maintained by Gracenote Media Recognition Service (www.gracenote.com).

Sound for Your Mobile

Ringtones are perhaps the most widely- and often-heard sounds in today's world. Unlike plain old telephones, where a pulsating 90-volt signal is sent down copper wires to energize a hammer that klangs a bell, there is no bell in a digital mobile telephone. When the mobile receives a notice that someone is calling, the unit's software takes over and, depending on the programmed options, plays the user's choice of ringtone—either generated by internal MIDI software or played from a stored sound file. Ringtones play on a very small speaker and often compete in a noisy environment. Perhaps an urban myth, it is reported that an inventive sales executive recorded herself coughing and sent that awful sound to her phone as a ringtone. When she received an incoming call during a meeting, she would quickly cover her mouth, continue the cough, and excuse herself from the room in order to take the call. Ringtones aren't the end of it. Into the daily lexicon have entered answertones, ringbacktones, truetones, realtones, singtones, videotones, and “ringles.” Most are for sale from enterprising small and large businesses all over the world. MP3 files will play on most mobiles; check your phone's manual to be sure.

Sound for the Internet

There are several methods for playing digital or MIDI sound from a web page. The sound is actually not part of the web page but is a separate file with its own address on the Internet, which is “embedded” in the page. Web browsers associate files with applications and plug-ins: Figure 4-8 shows the Preferences panel from Firefox, which lets you specify what to do when the browser downloads a particular file type. The simplest way to embed a sound file in a web page is to call it from an inline HTML anchor:

```
<a href="mysound.wav"> Click here to play MySound! </a>
```

As an HTML anchor, the text “Click here to play MySound!” will usually be underlined, and when that link is clicked, the browser will find the file `mysound.wav` (in this case, in the same directory as the web page), will download it, and, depending on how the user has instructed the browser to manage WAV files, will open a player and play the sound.

Media players are designed to play files as soon as enough of the data is cached in your computer’s **buffer** (a place where data is stored temporarily). The downloading continues to fill the buffer faster than you empty it by playing the sound file, allowing the sound file to stream into your computer in the background, keeping ahead of what has already been played so the playback doesn’t pause or break up. **Streaming** files are dependent upon connection speed: you must wait longer (**streaming latency**) before the streamed sound begins to play when using a dial-up modem (low bandwidth) than when using a high-speed DSL connection (high bandwidth).

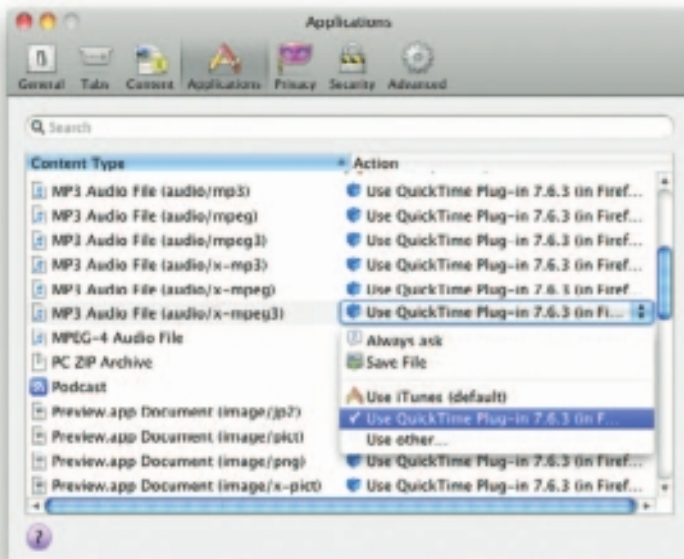


Figure 4-8 Web browsers must be told what to do when they download file types.

TIP See Chapter 13 for more about the HTML5 `<AUDIO>` tag, which can be used to play sound on a Web page without requiring a special player or helper application.

Adobe’s Flash allows you to integrate the sound tracks that you have made using a sound editor into a Web-based multimedia presentation, including both event sounds like button clicks and streaming sounds like background music. Because it can read and save MP3 files, Flash offers web designers serious and powerful options for solving the quality conundrum of high-quality (big) files and slow downloads versus low-quality (small) files and speedy delivery—with nice results. Because it must break

a sound into “frames” so it plays in sync with the timeline, Flash resamples the audio track if you ask it to “stream” in a movie clip; for the best quality, import an uncompressed audio clip into the Flash library and let Flash do the compression.

Testing and Evaluation

Putting everything together can be tough, but testing and evaluating what you’ve done can be even tougher—especially if your project involves a complicated live presentation, or if you’re shipping a commercial multimedia application. Unless you plan ahead, problems will not emerge until you begin testing.

TIP *During editing and authoring, regularly test the sound-and-image synchronization of your project. If you are delivering your sound on the Web, test it with different browsers and different connection speeds.*

Don’t forget to evaluate your sound storage medium. How much RAM does your project need to run effectively? Some authoring and delivery packages will stream sound directly off the hard disk or CD-ROM; others require the sound to be loaded into memory from the hard disk before they play. Sometimes you will need to break a sound or a music file into smaller parts. And MIDI files that sound terrific with expensive General MIDI during development will not have the same quality on a low-end FM-synthesis device at the end user’s site or on a handheld.

In the world of professional film and video production, sound is incorporated during **post-production**, or a **post-session**, after all the film and video footage has been assembled. Just so with multimedia—and don’t give it short shrift because of time or budget constraints. The sound track can make or break your project!

Copyright Issues

Ownership rights are significant issues for multimedia producers who would love to use a few bars of Beyonce’s latest hit or a nostalgic background of Bach suites played by Pablo Casals. Producers may rightfully fret about copyrights and permissions. Most developers play it safe by always making their own custom music from scratch in a sound studio, or with synthesizers, or by using sounds that have a clear and paid-for ownership and permission trail. Others simply take a risk and break the law.

WARNING *You are breaking the law if you record and use copyrighted material without first securing the appropriate rights from the owner or publisher of the material.*

As more and more multimedia is produced by more and more developers who are hungry for sound content, the copyright of sounds and images has become a major issue—not so much about who owns something, but how much of it they own. Because it is so easy to manipulate and edit a sound, just how much of someone’s original work do you have to change before it then becomes your own? There are separate licensing issues for use of a musical composition (even if you create a MIDI performance of it yourself) and for use of a particular recording of a musical composition (as when you make a copy of a song downloaded from iTunes). Different licensing arrangements may be required, depending upon exactly how you use the music in different types of multimedia programs—from a presentation you create for a client’s annual stockholders meeting, to a musical foundation beneath a commercial application. As this suggests, music licensing is a specialized and complicated area, so you should make sure you have cleared all the necessary rights before using any music in a product. The Harry Fox Agency (www.harryfox.com), for example, represents more than 27,000 music publishers and is the premier licensing resource for the mechanical use of music reproduced in all formats and media. Copyright issues and methods of securing permission for use (equally relevant for sounds, still images, and motion video) are discussed in more detail in Chapter 11.

A number of software vendors have entered the multimedia marketplace by selling digitized clip sounds with an unlimited-use, royalty-free license. Some of these products include musical clips, and some just include sound effects (doors closing, dogs barking, and water dripping). Other products have a mixture of both. But beware of sources claiming to be public domain that offer clips of Bella Swan and Edward in the *Twilight* Saga, or one-liners from Humphrey Bogart movies, because these sounds have likely been used without permission. Also, carefully read the licensing terms that come with any collection you purchase. Although the box may claim that the sounds are “unlimited-use, royalty free,” the fine print inside most likely limits their use to your personal machine and does not include the right to use them in any commercial use or republication in a form that would allow others to obtain them (such as using them on a web site).

WARNING *Taking a camera or tape or video recorder to some public events may be illegal without proper permission.*