New Audio Features in QuickTime 3

by Jeff Essex

Sponsored by Apple Computer, Inc.
Worldwide Developer Relations Group

New Audio Features in QuickTime 3

by Jeff Essex

Introduction

QuickTime 3 is here, and it's great news for multimedia developers who want to add high-quality audio to their content. This article will offer an overview of these new features, step-by-step tips for using them effectively, and links to sites with related information and resources. The topics covered in this article include:

- A Primer on Web file sizes
- Using MoviePlayer to compress audio
- Qualcomm's PureVoice™ compression codec for speech
- QDesign's high-quality codec for music and voice
- The QuickTime Music Architecture and new, improved Roland Instruments
- Cross-platform development issues
- Delivering media via the Web

The Past As Prologue

Before diving into the new features, let's review where QuickTime has been. This may give you a better appreciation for the QuickTime 3 update.

First, let's look at audio compression. QuickTime 2.5 largely pre-dated the explosive growth of the Web as a media delivery platform. The audio compression standards current at that time were primarily IMA-ADPCM (4:1) and μLaw (2:1). These two schemes used 16-bit audio for source material, so the resulting files were still quite large by current Web standards. For example, one minute of mono 16-bit, 22 kHz IMA audio uses 660 kBytes of storage. Streaming media wasn't supported in QuickTime 2.5, so the entire file would have to download to the user before playback could begin. By comparison, one minute of audio compressed with either RealAudio or Shockwave audio might only require 120 kBytes of storage, and thanks to streaming technology, playback could begin after just a few seconds. (Granted, this is a bit of an apples-to-oranges comparison because IMA sounds much better, but for many Web developers, quick access and short download time are more critical than high quality).

The other primary audio technology of QuickTime 2.5 is the QuickTime Music Architecture (QTMA). QTMA lets QuickTime users convert MIDI files to QuickTime data, taking advantage of MIDI's broad support and small file sizes (if you're not sure what MIDI is, hang in there until the section on QTMA in QuickTime 3). The main drawback of MIDI in QuickTime 2.5 (at least in the minds of many developers) was that the quality of the instrument sounds was marginal, and that there were some gaps in the set of available instruments.

In summary, the audio implementation in versions of QuickTime prior to 3 left room for improvement in two areas: audio compression for Web delivery, and playback quality from the QuickTime Musical Instruments synthesizer. Happily, QuickTime 3 addresses both of these issues. In fact, it sets a new standard of quality. Even better, QuickTime 3 is the first version that is fully cross-platform. Every QuickTime feature available on the Macintosh is now also available on the PC. This includes a fully cross-platform version of Movie Player, the handy utility for playing and editing QuickTime data. Which brings us to another new wrinkle in QuickTime 3. There is now a distinction between two flavors of QuickTime: 1) the standard, consumer-level version of QuickTime, and 2) QuickTime Pro 3, an enhanced version with valuable features for media professionals. Let's take a brief look at the differences.

About QuickTime 3 Pro

With the release of version 3, Apple has defined two markets for QuickTime:

- consumers who just want to play QuickTime content
- professionals who need to work with QuickTime content

If you've read this far, chances are you fall into the second category. If so, do yourself a favor and upgrade to QuickTime 3 Pro (http://www.apple.com/quicktime/upgrade/index.html). You'll need it for practically all of the applications discussed in this article. Having recently gone through the upgrade process myself, I can honestly say it was quick, painless, and –considering the many times in the past when MoviePlayer has been my tool of choice – well worth the investment. Once you have QuickTime 3 Pro installed, you'll be ready to play with the great new options in QuickTime 3.

A Brief Primer on Web File Sizes

As mentioned above, QuickTime 3 has powerful new audio compression tools designed for Web playback and slower connection speeds. The new codecs that are of particular importance to audio are the Qualcomm PureVoice™ codec and the QDesign Music codec. With the advent of the Web, the benchmarks for measuring file size are a bit different than for CD-ROM products. This is particularly true for streaming media over the Web.

Streaming audio formats are measured in thousands of bits per second (kBps). Higher rates offer more data for representing a signal, so they sound better. Considering that most sites are designed for a 28.8 kBps modem, the safest values for optimal streaming are 16-20 kBps. This leaves a bit of margin if net traffic builds up and throughput speeds are reduced. Streaming over faster connections (ISDN, T1 or intranet) usually allows for rates of 64 - 128 kbps. Different audio codecs allow for different compression rates. The Qualcomm voice codec operates at 8 kBps, while the QDesign codec supports a range of rates from 4 to 128 kBps. In comparison, Shockwave audio (a variant of MPEG II Layer 3) supports 8, 16 & 24 kBps at the low end and 128 kBps at the high end.

One question I often hear from clients is "I want to play a sound that's ___ seconds long. How big will that be? How long will it take to download?" The answer, of course, is "It depends." As with most aspects of digital media production, it's a trade-off between file size and quality. But once you know the data rate of a file, it's easy to calculate file sizes and download times. Let's assume I have one minute of music compressed at 16 kBps. Remembering that there are 8 bits in one byte:

60 sec. @ 16 kBps = 60 sec. @ 2 kBytes/second = 120 kBytes file size

This file could easily stream to a user with a 28.8 modem. But let's say you're the no-compromise type and want to trade "streamability" for quality. Taking the same one minute file at 64 kBps, we have:

60 sec. @ 64 kBps = 60 sec. @ 8 kBytes/sec = 480 kBytes file size

Again assuming a 28.8 modem connection with an average speed of 20 kBps, this would work out as follows:

60 sec. @ 64 kBps = 3840 kBits 3840 kBits / 20 kBps = 192 seconds

Rather than listening to the file in real time via streaming, the user would have to wait more than three minutes before the file could be played. Keep in mind that this is still superior to the requirements for a 16 bit, 22 kHz file with IMA compression, where one minute of data would require 660 kBytes of storage, or roughly five and a half minutes at 20 kBps.

Another issue to note with the new QuickTime audio codecs is that their approach to file compression is a bit different than in previous versions. The two main types of compression supported in QuickTime 2.x were IMA-ADPCM and µLaw, which provided for 4:1 and 2:1 compression respectively. Basically, these approaches just took a linear stream of sample data and did some fancy mathematical tricks to reduce the number of samples.

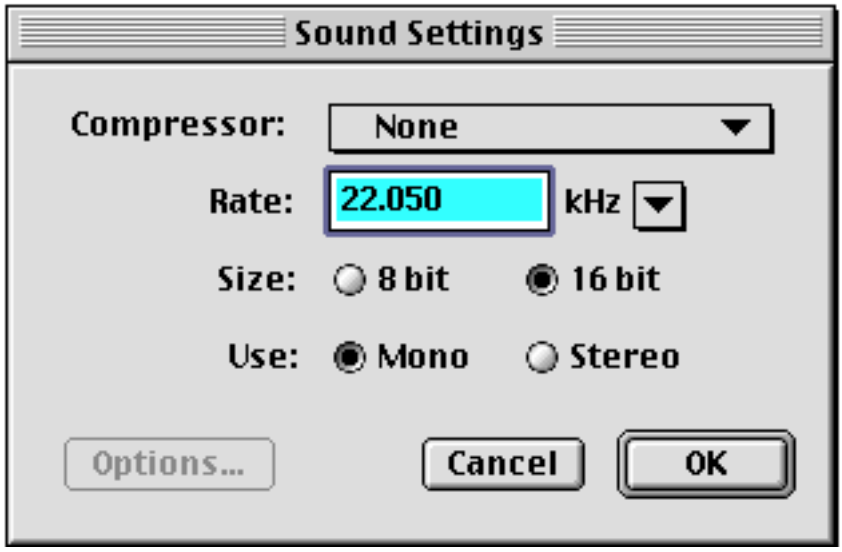
The new QDesign and Qualcomm codecs, on the other hand, achieve much smaller file sizes through *perceptual encoding.* Perceptual encoding works from mathematical models of how we actually perceive sounds. For example, imagine a sample of a bass drum, snare drum and cymbal being struck simultaneously. This sample would have broad band of frequencies, from the low thump of the bass to the high sizzle of the cymbal. The snare would be smack in the middle of this range. Consider that the highest frequencies of the bass drum (i.e. the "thwack" of the mallet on the drum head) woud be in the same range as the low end of the snare. Also, the low end of the cymbal might also be at the high end of the snare, masking the rattle and buzz characteristic of snare drums. In audio parlance, the frequencies from the bass drum and cymbal are *masking* frequencies from the snare drum.

In this situation, much of the snare drum information could be subtracted from the signal, and the listener would barely notice. Our brains do a pretty good job of completing the aural picture with just a rough sketch of the outline. Removing the extra frequency information through perceptual encoding algorithms makes the file size much smaller. To save even more space, the remaining data can be reduced even further using traditional file compression techniques. Of course, keep in mind that your audio will sound increasingly less natural as higher levels of compression are applied. This is the type of tradeoff that most multimedia producers have faced before.

Choosing Codecs in MoviePlayer

If you've never used compressors from within MoviePlayer (or any sound program that uses the standard QuickTime dialogs), this section will show you how to apply a compressor to your audio. If you're starting with an external AIFF or WAV file, start by opening the file in MoviePlayer. It's always best to start with high quality source data (16-bit 22.050 kHz or better). If possible, use an audio editing program like SoundEdit 16 (http://www.macromedia.com/software/sound/), Bias Peak (http://www.bias-inc.com/) or SoundHack (http://shoko.calarts.edu/~tre/) to *normalize* the file. This brings the file to the highest possible volume without clipping or distorting the sound data. If possible, use plug-ins like those available from Waves (http://www.waves.com) to slightly compress the dynamic range of the signal.

Once the file is open in MoviePlayer, choose Export from the File menu, then click the Options... button to bring up the Movie Settings dialog. Click on the Settings... button in the Sound section (the lower half of the dialog) to bring up the Sound Settings window. Now you can click on the pop-up menu to select a compressor. Some compressors (like Qualcomm PureVoice™ and QDesign Music) have additional settings that can be accessed by clicking the Options... button in the lower left corner of the window.



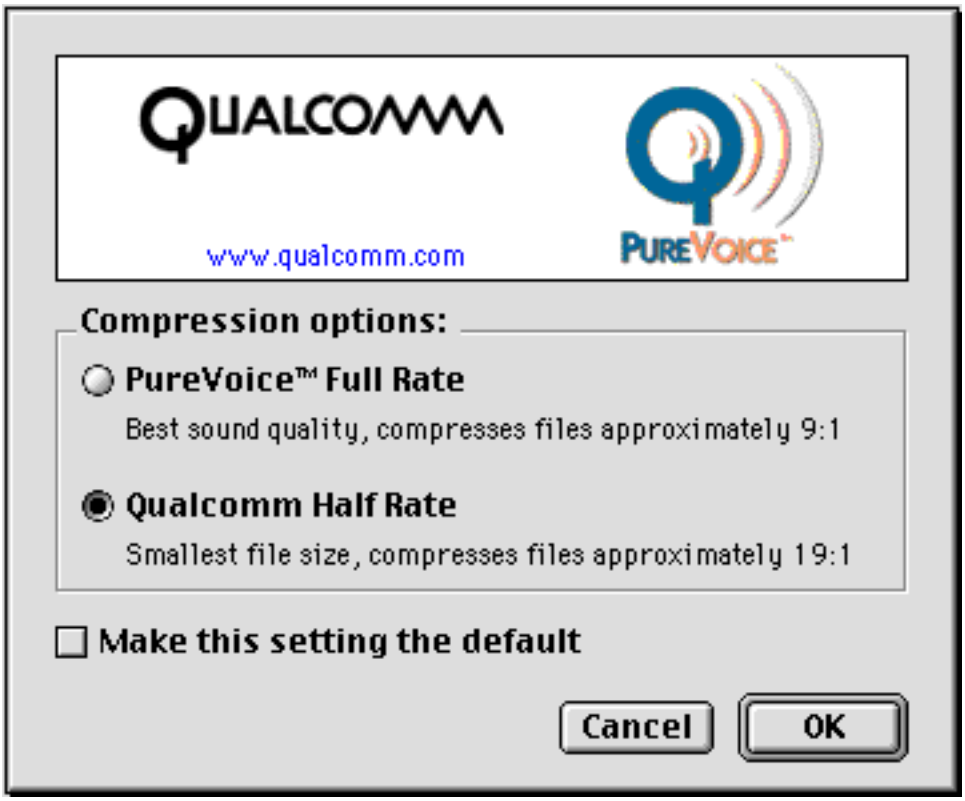
The Sound Settings Dialog lets you set the compressor, sample rate and bit depth for sounds before they are exported.

After you've chosen your settings, click the OK button in each window until you arrive back at the original File Export dialog. Enter a name for the file, export it, and presto - you have a compressed audio file.

Now that you have a better understanding of how the new audio compressors work, let's look at each of them in more detail.

The Qualcomm PureVoice™ Codec

The Qualcomm PureVoice™ codec uses technology licensed from Qualcomm, the makers of the popular Eudora e-mail program. Qualcomm originally developed voice compression algorithms for use with cellular phones. It turned out that these compression schemes were also useful for computer applications such as recording and attaching voice messages to e-mail documents.



The Qualcomm PureVoice™ options dialog lets you choose between higher quality or smaller file size.

The Qualcomm codec operates at rates as low as 8 kbits per second. You can select from a range of sample rates (8, 11 or 22 kHz) to get the best combination of file size vs. quality for your particular application, but the compressor is optimized for 8 kHz. While this rate might not be suitable for music or other broad bandwidth content, it covers most of the frequencies found in human speech.

The codec also offers a choice between "full" and "half" quality compression. Full quality sounds better, but creates larger files. As an example, compressing a 30 second clip of 16-bit, 44.1 kHz voice down to 8 kHz Half quality dropped the file size from 2.7 megabytes to just 27 kBytes! While there is a

noticeable difference in quality, the speech is quite intelligible considering that it was compressed by about 100:1. For more information visit their site: http://www.eudora.com/purevoice/quicktime.html.

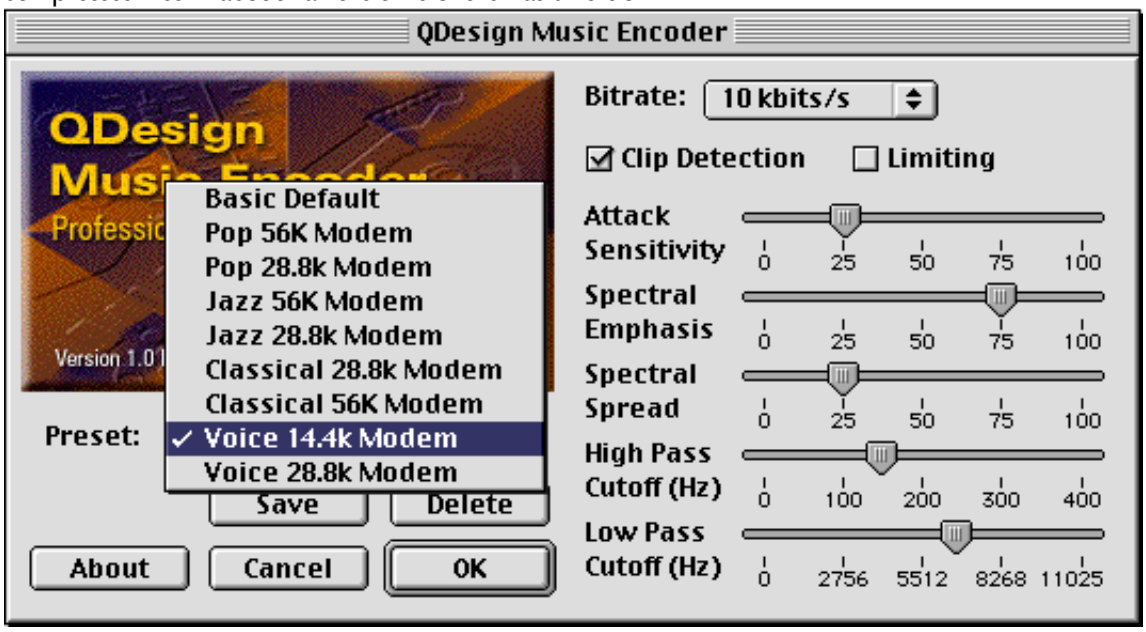
To achieve the best results, experiment by using an equalizer to boost the signal energy around the 4-5 kHz range, and cut frequencies below 70-80 Hz. This will increase the "presence" of the voice, and reduce any "booming" in the low end.

The QDesign Music Codec

The QDesign codec is a great new option for full-bandwidth music delivery over the Web. It offers compression levels and sound quality that rival or surpass any of the current popular online music players, and it's easily integrated into other QuickTime media.

One benefit of the QDesign codec over similar technologies like MPEG is that it works very well in stereo as well as mono. The added separation and ambience makes it easier to hear finer details in the music, and listeners are used to hearing stereo in most other music playback settings.

There are two flavors of the codec. The Basic Edition is included with QuickTime Pro, and provides encoding bitrates between 8 - 48 kbits per second. The Professional Edition gives you control of a wide range of encoding options, including bit rates from 4 - 128 kbps, low- and high-pass filters, fine-tuning of spectral performance, menus of presets, the ability to save edited settings, and clip detection. It also compresses files in about half the time of the Basic Edition.



QDesign's Professional Encoder gives audio professionals precise control of encoding parameters.

The ability to dial in custom settings is a big plus for developers. In a sense, it lets you focus the compression algorithm's attention on the most important elements of the music, so more computing power is applied to the critical program material rather than the whole frequency range. For more details, visit the QDesign site – http://www.qdesign.com/newsite/products/qdmcspec.htm

QuickTime and MIDI

With QuickTime 3, the promise of MIDI has finally arrived. First, let's briefly review what MIDI is, and why it's an important tool for musical content.

What is MIDI?

MIDI (the Musical Instrument Digital Interface) is a widely used and accepted format among musicians. The most common analogy for explaining MIDI is the player piano. The piano roll contains all the data needed to recreate a specific performance. MIDI is the high-tech equivalent of the player piano. It's a communication protocol that lets musical instruments, computers and other compatible devices transmit and receive data so that performances can be recorded, edited and replayed. Like the holes in a player piano roll, MIDI describes the information needed to recreate a performance: what notes were played, how long the keys were held down, etc. This performance data can be conveyed in just a few bytes, so a MIDI sequence requires far less storage than a section of digitally recorded audio. For example, a 2-3 minute MIDI sequence might only require 30kBytes of storage; to store the same piece as CD-quality audio (16-bit, 44.1 kHz stereo) would require 20-30 megabytes! This makes MIDI ideal for music playback in Web applications.

The QuickTime Music Architecture (QTMA)

As discussed earlier, QTMA is Apple's technology for integrating MIDI files with QuickTime data streams, along with the synthesis engine for playing MIDI files. QuickTime imports Standard MIDI files, a cross-platform format supported by practically every MIDI authoring/creation program (usually called *sequencers* because they're used to create and edit sequences of MIDI data). Once a file has been imported, QuickTime stores it as a Music track. Generally speaking, there are four things you can do with the Pro version of MoviePlayer and a Music track:

- play it back
- cut, copy and paste Music track data between movies
- export the track back out as a Standard MIDI file
- export the track as an AIFF file, thereby "rendering" MIDI to a traditional digital audio file.

These features were already available in QuickTime 2.x. The main improvement in QuickTime 3 is in the quality of the sound output. First, the samples in QuickTime 3 are higher quality (16-bit for 3.0 vs. 12-bit compressed in 2.x). Second, QuickTime 2.x only had a subset of the full General MIDI instrument set. For example, there was one trumpet sample; if you selected tuba, trombone or French horn, QuickTime 2.x automatically substituted the trumpet sample. This made it a bit more difficult to create rich horn arrangements (as in a symphony orchestra). When all the horns sound the same, there's less definition between instruments and the result is muddy and poorly defined.

Going one step further than the standard General MIDI instrument set, QuickTime 3 uses Roland Corporation's GS instruments, an extended version of General MIDI that includes 250 instruments instead of the standard 128, plus nine percussion sets (like orchestral, electronic and jazz) along with the standard General MIDI percussion set. Composers can choose from a much broader range of instrument sounds.

Best of all, the QuickTime instrument set sounds remarkably close to the actual Roland SoundCanvas hardware synthesizer, which is *the* standard device for composing General MIDI files. This means that composers can finally be assured that what they hear as they compose will be very close to what the listener experiences (I call this WILTIWYG - What I Listen To Is What You Get). This is valuable because it's still not possible to accurately play the QuickTime music sounds directly from a MIDI keyboard or controller. There is noticeable latency between the time a key is struck and the sound is actually generated by the synthesizer. Fine control over nuances of rhythm and phrasing are difficult to achieve in real time, but QuickTime's strict adherance to the performance of Roland's GS sound engine makes composition more predictable and much easier.

Finally, QuickTime 3 has enhanced support for external hardware synthesizers via the Open Music System (OMS) created by Opcode Systems (http://www.opcode.com). You can create customized setups from within the QuickTime Settings control panel, making it easier to switch between the internal QuickTime synthesizer and external MIDI devices.

Customizing QTMA Instruments

If the sounds in the standard Roland GS bank aren't enough for your needs, you can add your instrument sounds to QuickTime Music tracks. There are two basic ways to achieve this:

- Use Apple's Atomic Editor utility to create your own custom instrument extensions that can be loaded at startup
- Drag and Drop new sounds onto the Instrument list to embed new instrument samples into specific movies or Music tracks.

For a great walk through of using Drag and Drop, refer to Judy and Robert's Little QuickTime Page (http://www.bmug.org/quicktime/). Briefly, here are the steps they describe:

1. Create a new custom sound using a program like Bias Peak or Macromedia's SoundEdit 16
2. Save it to the Desktop as a System 7 sound (.snd resource)
3. Open a movie with a Music track
4. Select Get Info for the movie, and choose the Music track from the pop-up menu at the top left side of the window
5. Select "Instruments" from the pop-up menu at the top right of the window
6. At this point you should see a list of instruments for that file. Click the System 7 sound file on the Desktop, then drag and drop it onto one of the instruments on the list. The instrument will be replaced by the new sound file, and the sound data will be stored in the QuickTime movie.

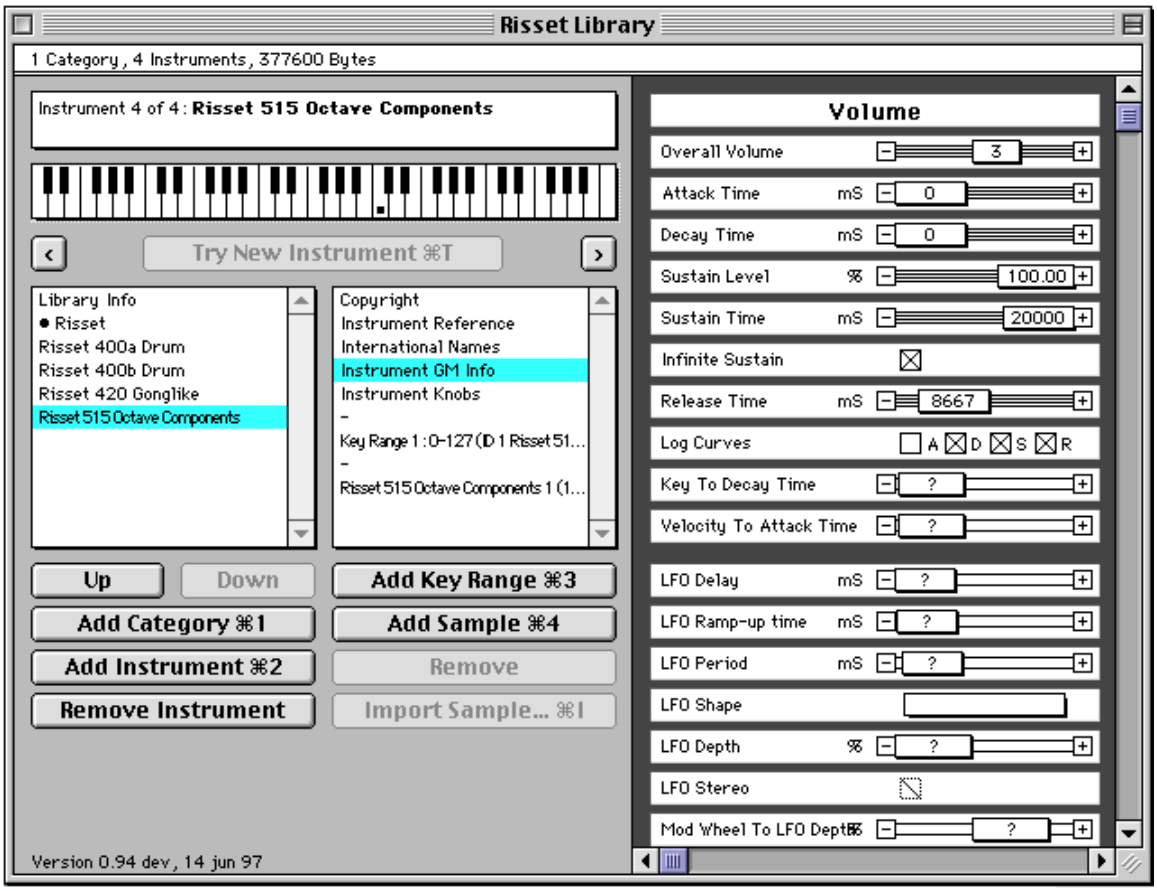
This approach is great for adding sound effects and percussive sounds to Music tracks. But there are a few drawbacks to this method if you're trying to substitute melodic instrument samples. One is that it's difficult to create instruments that are properly pitched relative to other instruments in the Roland GS sound banks (samples should match the pitch of C on a keyboard or instrument tuner to play in tune with the Roland instruments).

The other limitation is that looped samples aren't supported via standard Drag and Drop as described above. Looped samples are very important for many instrument types. It's impractical to create a 10 second sample to assure that a note doesn't cut off if it's played for an extended length of time. Most synthesizers and sample playback devices take advantage of a basic principle of sound propagation: many sustaining sounds (like piano, flute or violin) have a brief but harmonically complex attack portion (the initial start of the note), then settle into a relatively regular and predictable waveform. In the case of the piano, for instance, there's a lot of complex, chaotic activity that occurs when a felt hammer strikes a string. After a few hundred milliseconds, however, the string settles down and vibrates at a fixed frequency. It's likely that you could create a realistic piano sound with just a one second sample, where the last half of the sample (the regular waveform) loops until the key is released. For a sample playback device, especially one running in software like QuickTime, this is a much more efficient approach.

The good news is that you can get the best of both worlds using Apple's Atomic Editor utility, available free at the QuickTime FAQ (http://www.quicktimefaq.org/software/). This utility is unsupported, and has a rather non-standard interface, so you'll need to be a bit adventurous. Nonetheless, it's essential for anyone interested in making the most of QuickTime Music tracks.

Using Atomic Editor

The functions of Atomic Editor are pretty well described in the .pdf document that accompanies the software. But here's a broad stroke of some of the main issues you should be aware of as you use the program. I won't attempt to describe all of them in detail. I'm also assuming anyone interested in this level of control will have some knowledge of what these functions imply.



Atomic Editor has a wealth of options for customizing the QuickTime synthesizer.

1. Atomic Editor has an extremely powerful set of editable synthesis parameters, including keyboard mapping, velocity scaling and switching, filter and pitch envelopes, resonant filters (!) and flexible LFO/modulation routing.

2. Adding your own samples isn't exactly intuitive if you just open the program and start using it without reading the documentation. If you absolutely can't wait to play with it:
    - Open one of the sample files provided with the Editor
    - Click Add Instrument to create a placeholder for your sample
    - Click Add Key Range to be sure it's mapped on the keyboard
    - Click Add Sample to add a placeholder sample of a plucked string. Notice that it's highlighted at the loop points
    - If you want to experiment, click "Try Instrument" then play the plucked string sample, by clicking on the piano keyboard graphic
    - Click "Import Sample" to replace the placeholder with your custom sample. This option will only be available when the sample is selected, showing the sample waveform in the righthand window (ideally, your sample should be a 16-bit AIFF file at either 22.050 or 44.1 kHz, with a looping end section)
    - Click "Try Instrument" and play your new sample as described above

3. If you're looking at the right side of the Editor window on a small monitor (like my Powerbook 540c) you may be wondering how to access other editing controls. They're in the same window, but since there's no "zoombox" icon in the upper right corner of the window, the scroll bar may be stuck below the bottom of your screen. If you mousedown on any of the black areas of the editor page, however, your cursor will turn into a hand, allowing you to grab that page and slide it to the left. Keep scrolling through the windows this way until you've seen all eight of the columns (the last is "Effects").
4. Here comes the neat part! Once you've created your sample, you can Drag and Drop it onto the Instrument list in the Get Info dialog of Movie Player, as described earlier. Instead of dragging a System 7 sound from the Desktop, however, click and drag the instrument info window displayed above the keyboard graphic (it contains text describing the sound, for example "Instrument 1 of 1, Untitiled Instrument 1"). This sound will be embedded into the movie document with its loop points intact, along with any other parameters that have been assigned to it in Atomic Editor.

Why Embed Instruments into Movies?

As noted earlier, there are two ways to distribute custom sounds with QuickTime Music tracks: create a custom extension using Atomic Editor, or embed sounds into the file using Drag and Drop either from Atomic Editor or the Desktop. Consider your specific needs carefully before selecting one method or the other. I suspect that most people will opt to package sounds right into the movies. This guarantees that the correct sounds will always be available when needed. It also means the user doesn't have to install a new extension and reboot their machine before they'll be able to hear the music. But there may be other situations where creating an extension is preferred. If you were creating an episodic Web site, for example, you'll probably be using the same instruments and sound effects over and over. It might make more sense to have users install a custom instrument extension, rather than download the same custom samples every time a new file is played.

Other New File Options

QuickTime 3 adds support for high-resolution audio file formats (24-, 32- and 64-bit) that are becoming increasingly important for professional recording applications. While 16-bit audio has long been the standard for CD audio playback, most systems perform audio calculations at either 24- or 32 bits of resolution, then convert back to a 16-bit result. Over the course of many file operations, this can increase quantization noise. The vast majority of users will rarely need this level of precision, but audio professionals will appreciate that QuickTime can now accomodate higher quality formats.

Cross Platform & Web Issues

The good news about QuickTime 3 just keeps getting better! For the first time, QuickTime and MoviePlayer are completely cross-platform. All of MoviePlayer's powerful editing functions and codecs are supported equally on Windows and the Mac. If DirectSound is installed on the PC, QuickTime's audio outyut will take advantage of DirectSound's mixing capabilities. Even if DirectSound isn't present, QuickTime 3 adds the ability to mix multiple QuickTime audio tracks into one stream. Never again will you have to worry about audio channels being "stolen" during QuickTime playback.

For Web applications, QuickTime 3's ability to stream audio is an invaluable enhancement. QuickTime 3 also gives Web authors the ability to create customized sets of media for users with different connection speeds. Home users running at 28.8 and corporate users running on T1 can be seamlessly connected to media that is optimized for their particular bandwidths. Once a user has specified a default connection speed in the QuickTime Settings control panel, QuickTime has the intelligence to request the best file for that connection speed.

MIDI Music tracks on Windows play through the QuickTime synthesizer rather than the MIDI device on the PC's sound card. This is a blessing, because it assures consistent playback every time. If you're planning on adding custom samples to your Music tracks, be sure the sounds are embedded into the file (as described above in the QTMA section) so that they'll be available for playback on PCs. At the present time, there doesn't seem to be a way to channel all the MIDI output from any PC application to the QuickTime synthesizer. But you can take advantage of the QuickTime synthesizer for any MIDI that you encounter through a Web browser. Just set the QuickTime Plug-In to be the default player for MIDI files.

Summary

With QuickTime 3, Apple has really delivered the goods for audio content creators. The new web-savvy codecs and Roland Instrument samples put QuickTime way out in front of other system-level technologies for audio playback. Cross-platform compatibility assures that developers will be able to address the broadest possible audience with a single set of media assets. Now that better tools are in our hands, we can spend more time focusing on what's really important: creating audio that engages the listener and enriches the interactive experience.

Resources

Here is a complete list of the online resources discussed in this article.

Audio Editing & Processing

SoundEdit 16	http://www.macromedia.com/software/sound/
Bias Peak	http://www.bias-inc.com/
SoundHack	http://shoko.calarts.edu/~tre/
Waves Plug-ins	http://www.waves.com

QuickTime 3 Tools

QuickTime 3 Pro	http://www.apple.com/quicktime/upgrade/index.html
Qualcomm PureVoice™	http://www.eudora.com/purevoice/quicktime.html
QDesign	http://www.qdesign.com/newsite/products/qdmcspec.htm
Little QuickTime Page	http://www.bmug.org/quicktime/

QuickTime Music Architecture

Opcode Systems, OMS	http://www.opcode.com
The QuickTime FAQ	http://www.quicktimefaq.org/software/

About the Author

Jeff Essex, Creative Director of audiosyncrasy, specializes in the use of digital audio and MIDI to create music, sound effects and voiceover for multimedia. In addition to his creative and audio engineering skills, he is intimately familiar with multimedia tools and technology. A veteran of MacroMind and Macromedia, he served two years as Technical Support Lead for Macromedia Director and SoundEdit. He is credited on over 40 CD-ROM titles, including products from Virgin Sound and Vision, Corbis Productions, Mindscape, 3DO, Disney Interactive and Pulse Entertainment. For the past year he has worked as the primary composer and audio consultant for the leading children's interactive web site. His book, Multimedia Sound and Music Studio, (Random House/Apple New Media Library,1996) is the definitive guide to multimedia audio production. It won the Computer Press Award for Best Advanced How-To Book of 1996.