

As computer based audio begins to gain popularity in the audiophile world, more and more questions arise over what happens to audio as it goes into, and comes out of a PC. This report is intended to throw some light on the various concepts that are often bandied about, how these affect the final audio output, areas to beware of, and how some popular playback schemes address the various options.

Mixers

Windows XP, Vista and OSX all contain by necessity a conceptual **mixer**. The OS (Operating System) is designed such that multiple applications can each access an audio device concurrently – for example, the user is playing some music via his media player, and he receives an e-mail. The media player has control of the audio device, but the e-mail application wants to sound an alarm bell. In this case, neither application knows about the other, so the mixer is in charge of mixing the two sounds together and supplying them to the output device. To do this, it may have to **sample rate convert** one of the sounds, and adjust the volume of both of them to prevent clipping.

For most applications, this works very well – the user playing back his MP3s hears the alarm bell mixed in when his e-mail comes in. However, to do this the mixer may make choices which may be unacceptable to an audiophile.

Firstly, most audiophile users will want to listen to **only** the music, and not be distracted by random sounds. Secondly, any volume control will alter the bits output to the audio device, possibly losing resolution. Thirdly, if the mixer decides to sample rate convert the music, rather than for instance system sounds, then his audio will be unnecessarily degraded.

Sample Rates

Digital audio is represented by a set of samples. In a single track, each sample is the same size resolution (number of bits), and the same space apart in time (sample rate). Hence, if a track is encoded at 16/44.1, this means that the amplitude of each sample is encoded as 16 bits of binary data (-32,768 to +32,767), and there are exactly 44,100 of these samples every second.

Due to the fact that a PC may contain many tracks in many different sample rates, and more than one can potentially be played at one time, the OS's must have a scheme to deal with this, which can involve **sample rate conversion**.

OSX

OSX uses a “fixed” output sample rate (set by the Audio Midi panel in “Utilities”). The user sets this, and OSX re-samples everything to match this rate.

Pros: The digital out never changes rate, and if the rate is set to the same rate as the file being played back, and all enhancements/volume controls are disabled, the output is bit perfect.

Cons: If the user has multiple sample rates, he either has to change the output sample rate manually every time the source sample rate changes, or rely on the OSX rate converter

Windows XP

XP uses a program called “K-Mixer” to handle sample rate clashes, and mixing. If the audio device is already playing a sound, and a second application attempts to play another sound, K-Mixer will resample the second sound to be the same rate as the first. Otherwise, it does not perform rate conversion, but a small volume adjustment is always in place which increases the word length to 24 bits (this is benign if the output device is 24 bit capable, but gives poor results with 16 bit devices).

Pros: If a user uses only one audio program, and has a 24 bit capable device, the output sample rate will change to meet that of the source material with very little degradation

Cons: Tricky and application specific to bypass K-Mixer for truly bit-perfect output.
See **Workarounds**.

Windows Vista

Windows Vista notionally uses a similar scheme to OSX – in the “Sounds” control panel, there is an “Output Sample Rate” setting. When using general purpose programs that use DirectSound (e.g. Windows Media Player), it works much the same as OSX – all audio is re-sampled by the OS to that set in the control panel. Similarly, the output can be bit perfect if the source rate matches the output rate. However, Microsoft have introduced a mode called “Exclusive”. In this mode, applications can talk **directly** to the sound hardware, bypassing all mixers, rate converters etc. using an API called WASAPI. The author is not aware of any fully working WASAPI implementations yet.

Pros: Vista has the potential to combine the benefits of OSX for the casual user, with the pure-audio path and auto-rate switching an audiophile requires.

Cons: Doubts as to whether WASAPI is fully ready, lack of software using this mode. Poor rate converter (see **Sample Rate Converter** measurements).

Interfaces

There are numerous interfaces from a PC that can be used to transfer audio. These are outlined below:

SPDIF

The most common digital audio interface, SPDIF works by embedding the clock and data into a single stream, at rates up to 192kS/s.

Pros: Ubiquitous, nearly all DACs have an SPDIF interface

Cons: Due to being “self-clocking”, SPDIF is prone to jitter, including data related jitter – this gets progressively worse as the sample rate increases, and as the PC is generating the timing, this will be dependent on the PC.

USB

The most common PC interface, USB works as a packet-based protocol. There are assorted implementations of audio transfer, and is gaining traction as a popular interface, especially as it is capable of being encapsulated over Ethernet.

Pros: Ubiquitous on PCs. If audio device uses “Asynchronous” mode, then the audio device is in absolute control of the timing, and hence can eliminate jitter. No drivers are needed.

Cons: Many implementations use “Adaptive” mode, which will usually give far worse jitter performance than SPDIF. Many implementations are limited to 16/48, although some now accept 24/96.

1394 / Firewire

A rival to USB, typically used for DV applications. Like USB, 1394 is packet based. Losing popularity, as it is not featured on the latest set of MacBooks, and is not common on PC laptops.

Pros: 1394 can have sufficient bandwidth to do higher sample rates, and use a similar technique to the USB “Asynchronous” mode.

Cons: Not natively supported as audio by Microsoft or Apple. Apple seem to be losing interest in 1394. Requires drivers, which places a question mark over future support and reliability.

Ethernet

Audio can be streamed over TCP-IP, allowing existing network infrastructure to act as an audio bridge.

Pros: Network Infrastructure, routability of Ethernet.

Cons: Ethernet has no globally agreed audio format. No guarantee of bandwidth. Not natively supported in any OS. Can be troublesome connecting to a home network.

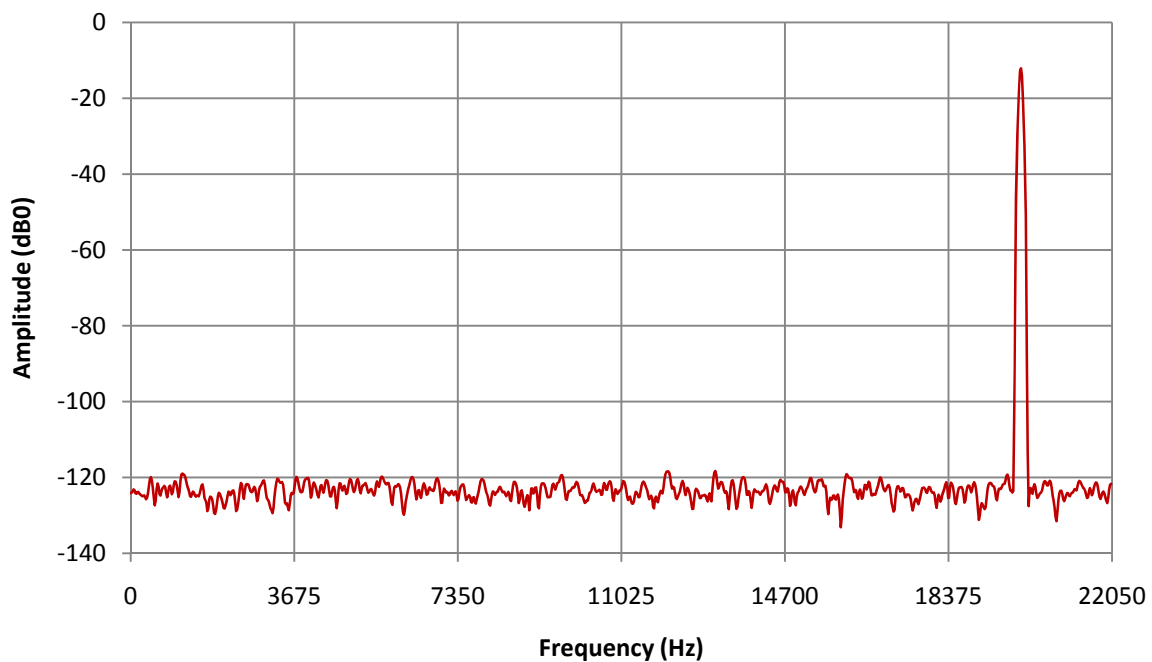
Sample Rate Conversion

As has been mentioned previously, part of the job of the audio subsystem in an OS is to track at any sample rate to a device, which may be running at a different rate (e.g. in OSX, the output is set at 96kS/s, the source is 44.1kS/s).

Sample Rate conversion is a mathematically challenging operation, and is not a priority to an OS designer. Hence, if possible an audiophile will want the bits from his original source to reach his audio equipment untouched by the PC. Below are some spectra showing what can happen inside a PC or Mac – the reference tone chosen is almost worst-case for 44.1kS/s data – the trade-off in rate conversion is the slope of the filter, and where it starts rolling off. An ideal filter is flat to 20kHz, then rolls off to be as attenuated as possible by 22.05kHz. We are using a -12dB0 level to avoid ripple in the filter causing overloads.

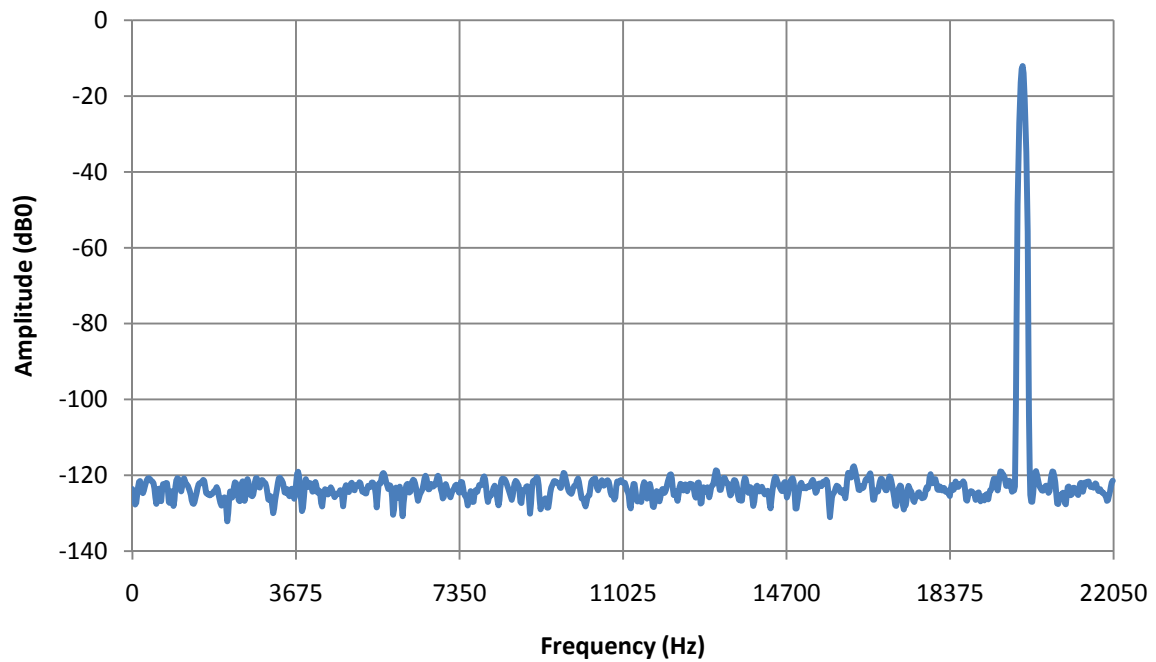
Reference : A -12dB0, 20kHz sine wave – reproduced by Foobar using ASIO4All on Windows XP. Checking the raw data reveals it to be a bit-perfect reproduction of the WAV file.

Reference -12dB0 20kHz Sine Wave 16/44.1



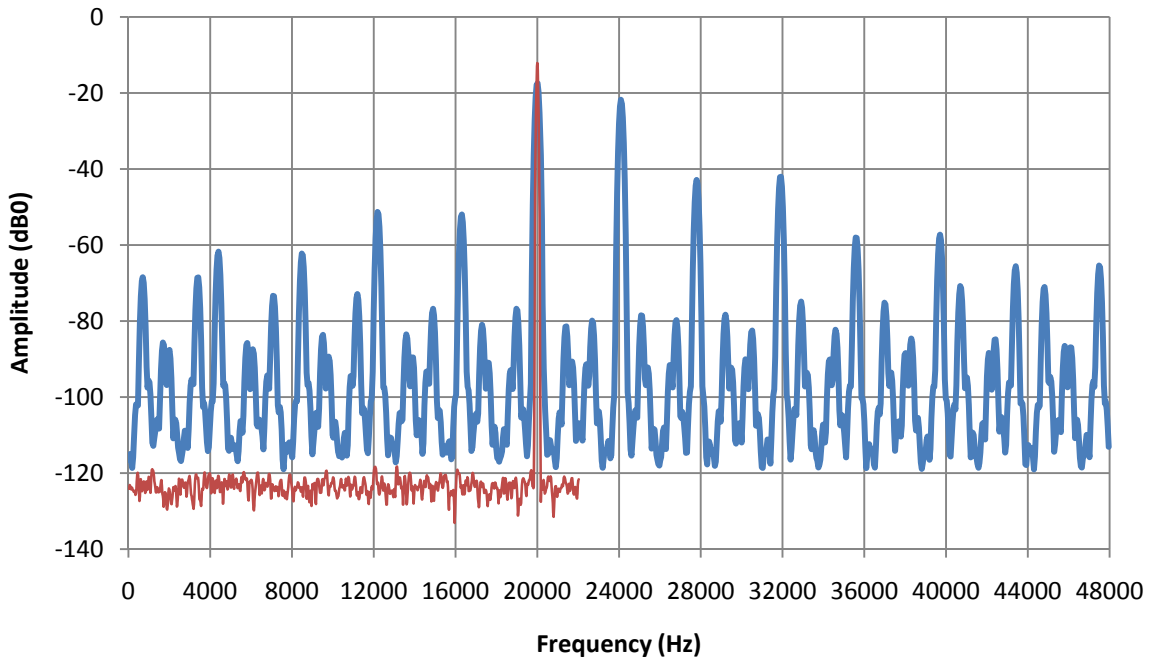
Below, we will show the same file being reproduced by the different operating systems using their default settings and playback software, into an audio device that has a maximum capability of 24/96 over USB. We will then show the best performance when setup properly.

Windows Media Player 11, Windows XP



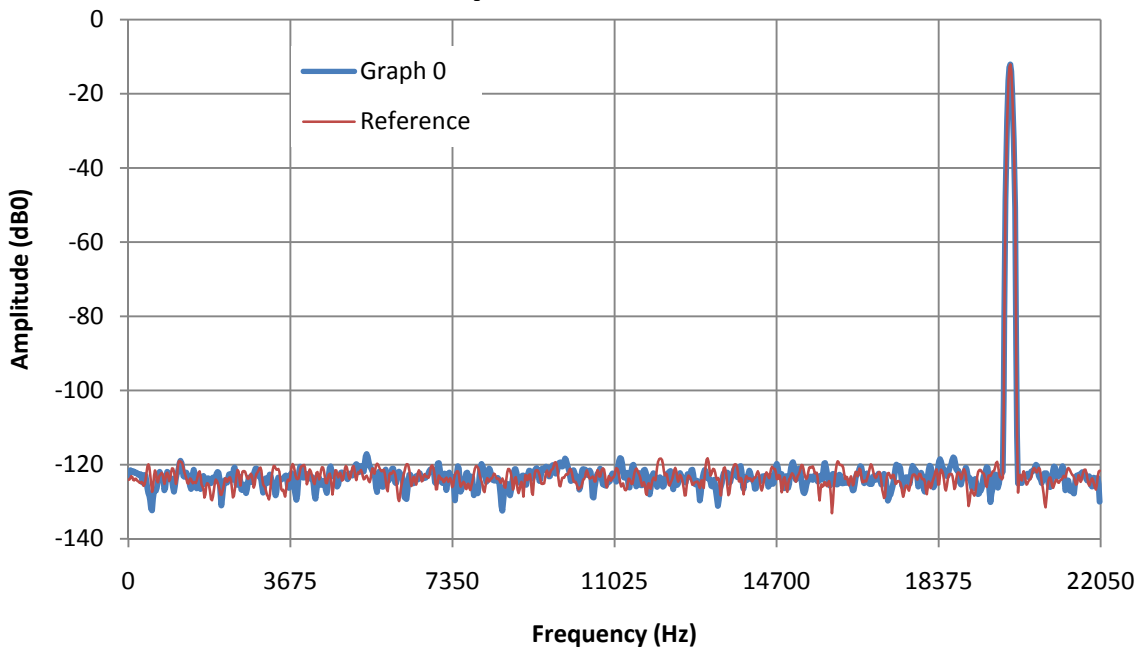
As you can see, with its default settings, WMP11 on XP can be spectrally excellent. Examination of the raw data indicates that there is a miniscule gain adjustment, which causes the 16 bit source data to lengthen to 24 bits. This will be benign with the 24 bit output path, and is completely acceptable.

Windows Media Player 11, Windows Vista (default)



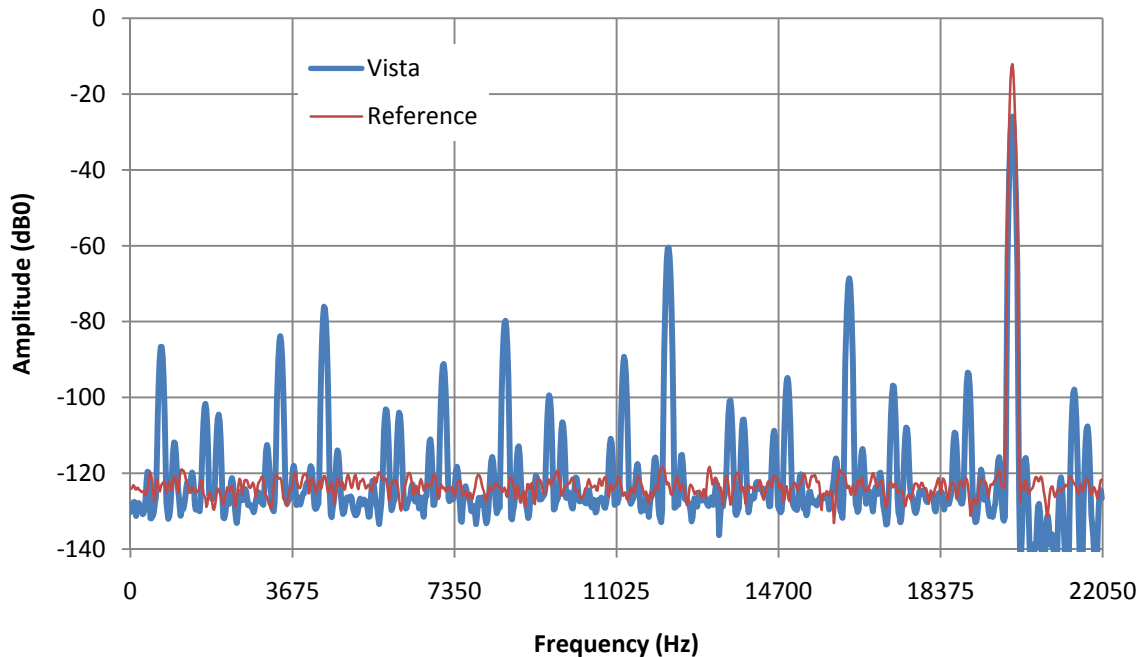
Look at this graph. The first thing to note is that the frequency axis has changed – Vista by **default** outputs at the **highest** rate that the audio device supports. Hence, it is **sample rate converting** everything to 96kS/s. The second thing to note is that Vista is doing this extremely poorly. We have added the “Reference” as a thinner line, so you can see what has happened. The original 20kHz sine wave is still there, but is substantially lower in level, demonstrating early roll-off. Additionally, there are numerous aliasing artefacts, many of which are in the audio band. The aliases outside of the audio band are very high in level and may well present problems for analogue circuitry downstream of any connected DAC.

Windows Media Player 11, Windows Vista Output Corrected



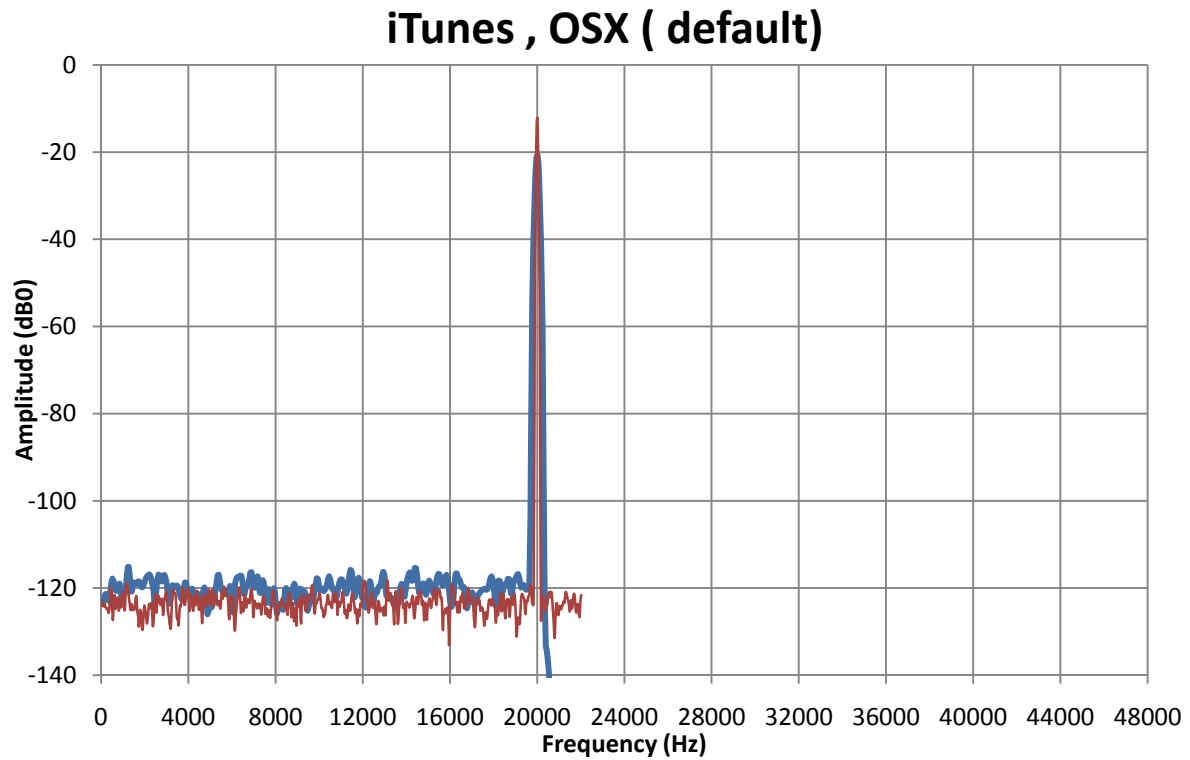
Here, we have gone into the Vista Control panel, selected “Sounds”, “Audio Device”, “Properties”, changed the Sample rate to 44.1kS/s 24 bit, and repeated the data acquisition. We can now see that Vista is now not corrupting the data in the same way. Closer examination reveals that using WMP11, with the system sample rate set to the same as the file, it is now bit-perfect – the small differences in the noise floor are just capturing the data at a different point in the file.

iTunes , Windows Vista - Vista 44.1, iTunes 96

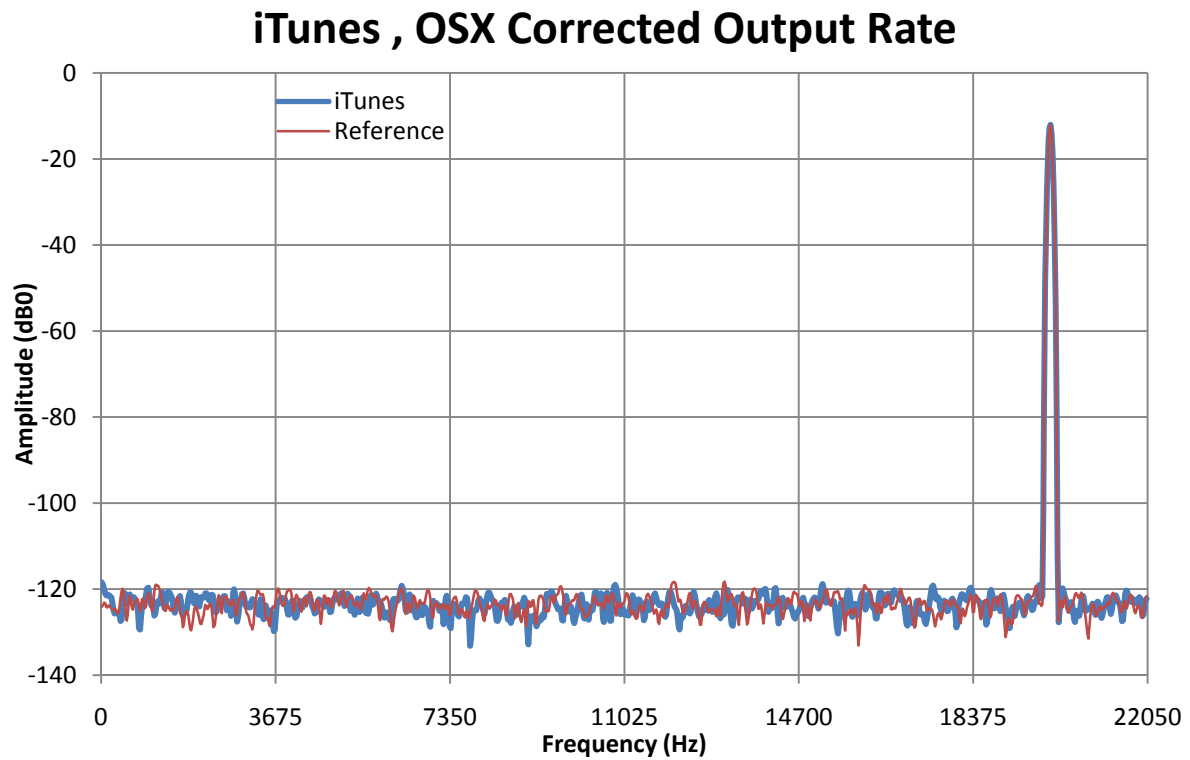


Here, we have kept the Vista sampling rate at 44.1kS/s, and played the same file with iTunes v8.0.0.35. In this case, the test equipment indicated that the sample rate met the source, but something had happened to the data. Spectral analysis provides the above plot – you can see that some fairly serious corruption has gone on here. Further investigation reveals that iTunes has its **own** sample rate control – set in the “QuickTime” item in control panel, which defaults to 96kS/s. In this case, iTunes is rate-converting to 96kS/s, and providing this audio to Vista. Vista is then rate-converting **back** to 44.1kS/s! The effects, as you can see, are terrible, and will almost certainly be audible on material with high frequency content.

Setting BOTH the QuickTime sample rate and the Vista audio to equal that of the source (44.1kS/s) restored the output to being bit perfect.



Here is the result of playing the reference tone through a Mac Mini running OS X 10.5.5. As you can see, by default the output sample rate is 96kS/s, so the Mac is doing some sample-rate conversion. You can see that it does a much better job than Vista, although it is still not audiophile-grade – notice how the tone is already 8dB down at 20kHz.



Going to Finder->Applications->Audio Midi, and setting the output sample rate to 44.1S/s gives us bit-perfect output again.

Workarounds

Freeware available on the internet allows ASIO playback using Windows Media Player & Foobar, using both Windows XP and Vista. Using these “workarounds” lets the application pick the sample rate, and guarantees that XP & Vista can be bit perfect. The author is unaware of any software that allows iTunes to play back multiple sample rates **without** the SRC being used for source rates that do not match the output rate.

Summary

It is perfectly feasible to achieve bit perfect output from Windows XP, Windows Vista and OS-X. Choosing the interface carefully, together with being aware of what can be going on “behind the scenes” can result in a PC/Mac source that meets every criteria (jitter performance, etc.) needed to be a “True High-End Source”, which combined with the convenience and flexibility make PC based audio almost irresistible.

This document is intended to illustrate that audiophiles **cannot** just assume that the PC will generate bit-perfect output without checking the settings, and that setting up the PC incorrectly can result in **severe** audio distortion. The most extreme example of this is playing back audio using iTunes on Vista. If the user does not check the sample rate being used, performance can be extremely compromised with little indication of the extra processing being applied.

Windows Vista is the most problematic at the moment (it is very easy to have its sample rate converter kick-in with disastrous results), but it does seem to have the architecture in place to allow future software to be bit-perfect at multiple sample rates with very little effort. We await keenly more media player software that utilises WASAPI in Exclusive mode to this end.

Windows XP, although the oldest in this test, is the most straight-forward, although the longevity has to be in question.

OSX has a **much** better rate converter than Vista, although it seems deeply entrenched in the OS. This could prove to be a problem as future media may have a wider variety of sample rates. OSX is still very much a supported OS, so future iterations may address this problem.