

# Reducing audio stimulus presentation latencies across studies, laboratories, and hardware and operating system configurations

Destiny L. Babjack<sup>1</sup> · Brandon Cernicky<sup>1</sup> · Andrew J. Sobotka<sup>1</sup> · Lee Basler<sup>1</sup> · Devon Struthers<sup>1</sup> · Richard Kisic<sup>1</sup> · Kimberly Barone<sup>1</sup> · Anthony P. Zuccolotto<sup>1</sup>

Published online: 14 July 2015  
© Psychonomic Society, Inc. 2015

**Abstract** Using differing computer platforms and audio output devices to deliver audio stimuli often introduces (1) substantial variability across labs and (2) variable time between the intended and actual sound delivery (the *sound onset latency*). Fast, accurate audio onset latencies are particularly important when audio stimuli need to be delivered precisely as part of studies that depend on accurate timing (e.g., electroencephalographic, event-related potential, or multimodal studies), or in multisite studies in which standardization and strict control over the computer platforms used is not feasible. This research describes the variability introduced by using differing configurations and introduces a novel approach to minimizing audio sound latency and variability. A stimulus presentation and latency assessment approach is presented using E-Prime and Chronos (a new multifunction, USB-based data presentation and collection device). The present approach reliably delivers audio stimuli with low latencies that vary by  $\leq 1$  ms, independent of hardware and Windows operating system (OS)/driver combinations. The Chronos audio subsystem adopts a buffering, aborting, querying, and remixing approach to the delivery of audio, to achieve a consistent 1-ms sound onset latency for single-sound delivery, and precise delivery of multiple sounds that achieves standard deviations of 1/10th of a millisecond without the use of advanced scripting. Chronos's sound onset latencies are small, reliable, and consistent across systems. Testing of standard audio delivery devices and configurations highlights the need for careful attention to consistency between labs, experiments, and multiple

study sites in their hardware choices, OS selections, and adoption of audio delivery systems designed to sidestep the audio latency variability issue.

**Keywords** Audio · Latency · Sound · Synchronization · Sound optimization · Sound onset latency · Known latency · Variable latency · Sound delivery · E-Prime · Sound presentation · Experimental design

In several fields of science, including psychology, researchers are becoming increasingly concerned with the replicability of timing in experiments. In neuroscience, cognitive science, and psychology, the precise and accurate timing of experimental components increases replicability and increases comparability across studies and laboratories. Some researchers have noted that some findings may simply not be replicable (Pashler & Wagenmakers, 2012). Others have noted that inaccuracies in experimental features, such as timing variability, may cause replication failures (Plant & Quinlan, 2013). In a recent attempt to identify the cortical representation of the speech envelope by linking played audio files to electroencephalography (EEG) data, the authors had to retract their article, citing failure to replicate their latency findings and reporting a need to “re-investigate the accuracy of the audio-visual synchronization used in [that] study” (Crosse & Lalor, 2014). Replication failures like these have large-scale implications for researchers, as well as a loss of forward scientific progression across disciplines. This audio latency variability may be due to several sources, but the consequences of such inconsistency in audio delivery can be important.

Sources of replication error differ and vary in their abilities to be removed systematically. One such source of timing-related errors is human error, due to coding deficiencies or lack of expertise. This can be ameliorated through sufficient

✉ Destiny L. Babjack  
drdbabjack@gmail.com

<sup>1</sup> Psychology Software Tools, Inc., Sharpsburg, PA, USA

training and the natural gain in expertise that accompanies practice within a scientific discipline. Other factors—such as choices of hardware and software, selection of presentation programs, and synchronization among equipment components—can exacerbate or reduce the factors that lead to replication failure, such as timing errors. An unlimited number of hardware and software combinations exist, and it is impossible to test them all. However, in this article we will demonstrate methods that can reduce timing variability arising from the use of differing hardware and operating system configurations to collect data, using the E-Prime experiment authoring and presentation system. Other systems can achieve low latencies through a variety of methods (e.g., MATLAB, Psychophysics Toolbox, and Presentation). Investigations of those systems remain outside the scope of the present article, though methods similar to those applied here (and others) could be used to achieve low latencies in these systems.

Computerized experimentation has changed the nature of replication within the scientific method. Computerized tools for experimentation and the units that are used to construct testing stations (e.g., sound cards, operating systems) have become commodity items, with many variations available and a near-constant cycle of updates, leading to a mass of low-cost options for equipment in scientific labs. This plethora of hardware, software, and operating systems (OSs) available for scientific experimentation continues to grow, with an infinite number of possible configurations. With that expansion comes an increasing complexity of the issues related to the consistency of experimentation that is required for rigorous replication attempts. Though many causes of inconsistencies make scientific replication vulnerable, the focus of this article is on the timing variability arising during audio stimulus presentation using E-Prime.

Timing inaccuracies can affect a variety of factors, such as response time measures and the synchronization between interconnected equipment and stimulus presentation (Plant & Quinlan, 2013; Plant, Hammond, & Turner, 2004). Updating to faster processors and new operating systems does not always minimize timing errors. In fact, it can exacerbate them. For many effects, timing inaccuracies, if small enough, can wash out over long-time-course events, such as the hemodynamic response. In the search for smaller and smaller effects that occur on faster and faster timescales (e.g., EEG), accurate and precise timing becomes more valuable. Some researchers would even argue that precise timing is more important than accurate timing, given the ability to correct for timing lags that are consistent. For example, a measurably consistent delay of 30 ms is easily corrected by adjusting the timestamps on the synced device by adding 30 ms to the event onset times, this is only possible if those delays, or latencies are consistent, or precise. In this way, precision timing, or known latency, can be more important than short yet variable latencies; their corrections are much more difficult. As multimodal synchronized methods of behavioral data collection become more popular,

the need for submillisecond accuracy and precision becomes paramount to scientific progression (e.g., Voyvodic, Glover, Greve, Gadde, & FBIRN, 2011). Timing variability on the magnitude of one millisecond can impact findings enough to warrant attention (Plant & Quinlan, 2013). Overall, reducing timing variability where possible can greatly improve scientific explorations and reduce the number of scientific errors due to timing inaccuracies (see, e.g., Crosse & Lalor, 2014).

In the present study, we examined the degree to which audio presentation variability can be reduced using off-the-shelf equipment or through minor modifications to hardware and software. Providing consistent and precise delivery of audio stimuli is a difficult process, due to the opening and closing of individual sound files, differences in the buffering and mixing processes used by different sound delivery devices (SDDs), and the various processes used to synchronize the start or end of audio presentation with other interconnected systems (Kieckhefer & Kanwal, 2000). In many studies, researchers deliver only one sound at a time, or single sounds in a sequential fashion. For these researchers, the fastest onset possible from the signaling of the presentation to the delivery of sound is most desirable. In other studies, researchers may play multiple sounds on separate channels or mix sounds. In these situations, accuracy is still desirable, but a fixed delay between an audio signal and audio delivery may be more desirable. In these cases, again, there may be a preference for a consistent, fixed latency, a *known* latency, over an inconsistent or variable delay of audio delivery, especially when those audio signals are synced to other events (e.g., behavioral data such as EEG). Audio presentation latencies can be affected, in terms of both precision and accuracy, through the use of differing hardware configurations. This is not surprising; studies have shown that varying even one small component of a testing station, such as the mouse, can impact response timing (Plant, Hammond, & Whitehouse, 2003).

The push of the federal funding dollar to multisite studies encourages, or even necessitates, the ability for results to be comparable when working across laboratories that use different equipment. Even now, most researchers are unaware of how variable their instrumentation may be. Researchers should know which application program interfaces (APIs) are used to communicate between the SDDs and the hardware drivers that operate the devices on their machines, such as speakers and microphones. Researchers may be unaware whether their API is Core Audio, DirectSound, ASIO, or something else, such as Chronos. Some researchers might not know which sound cards are installed on their machines, either as supplements or those that came as on-board audio. In the best-case scenario, each laboratory test station would be equipped identically, making their results comparable across machines. In the real world, however, it is unlikely that any given researcher's lab setup is identical to all collaborators' configurations, though comparisons between their studies are

often made. Researchers are interested to know the extent to which rather minor variations in common computing equipment can vary the timing of relatively simple events, such as audio presentation times. This will illustrate the inherent variability of audio presentation, as well as illustrate the differences by machines, OSs, and SDDs (e.g., sound cards). In this article, we seek to illustrate the nature of audio stimulus presentation variability and offer a solution for reducing timing-related variability through the selection of equipment intended for precise and accurate audio delivery.

We bench-tested a variety of common off-the-shelf configurations of hardware, OS, SDDs, and APIs in order to illustrate the variability of sound onset latencies. A standard Windows-based experiment generator (E-Prime) was used to run a simple audio stimulus delivery paradigm to play sound. To verify the timing of audio delivery, we used an external chronometry device, the Black Box Toolkit (BBTK). In the present study, five machines, six SDDs, four APIs, two Oss, and four different drivers were tested.

In the first set of analyses, a full crossing of the SDDs on all machines was not possible, due to hardware incompatibilities (e.g., sound cards could not be placed on the laptops), but when crossings of equipment were feasible, they were tested. Off-the-shelf testing provides the most realistic examination of sound delays in what would be a likely hardware/software configuration in a researcher's lab.

The second set of testing was aimed at optimizing performance in specific configurations, or reducing sound onset latencies. In that set of optimization tests, we examined performing subset of configurations. More savvy researchers with some expertise in audio delivery might optimize their hardware and software to achieve better sound onset latencies. Three methods were explored to optimize sound playback and reduce sound onset latency. Another type of SDD, an ASIO-based sound card, was added; some researchers have reported success with these types of cards. Psychology Software Tools advises against the use of ASIO with E-Prime due to poorer timing issues (Psychology Software Tools, 2015). Another approach to optimization was to switch from the manufacturer's driver to HD Audio when the API was Core Audio; HD Audio drivers tend to outperform this manufacturer's driver with regard to audio playback speed. Finally, an approach recommended by Presentation (Neurobehavioral System, Inc., 2015) was adopted: the use of cards from Creative Labs (e.g., X-Fi Titanium) in "bit-matched" playback mode. Bit-matched playback mode removes several preadjustments to the sound data, thus removing the delay caused by those adjustments. Many cards from Creative Labs (X-Fi) have a "Bit-Matched Playback" or a "Stereo Direct/Bit Accurate Playback" mode, intended to reduce the delays between presenting and playing a sound. These methods for reducing sound onset latency might be comparable to, or even better than, the off-the-shelf methods. The optimized samples were compared against Chronos in its off-the-shelf status.

There are three approaches to minimizing audio latency. First, we tested off-the-shelf options. Researchers that desire to do so could "plug and play" using these hardware and software combinations and be confident of knowing their latencies. A second option is adopt the use of systems, like Chronos, that are committed to achieving accurate and precise sound onset latencies. The final sound latency reduction option requires a little more effort: Certain custom configurations may be able to reduce sound onset latencies considerably. We will be address advantages and cautions regarding each of these approaches.

In the present study, we bench-tested a variety of common configurations, as well as configurations with optimization adjustments. Next, we illustrate the variability in sound onset latencies *when playing two sounds*. At first this might not appear to be innovative, yet many SDDs are not capable of playing multiple sounds simultaneously and cannot mix and play these sounds consistently. Researchers who desire to play multiple sounds or to play sounds simultaneously might be forced to use expensive, although reliable, equipment that is capable of *real-time* audio processing (e.g., the Psychoacoustics Workstation from Davis Technologies). Real-time audio processing considerably reduces sound onset latencies. Data are then presented that compare some common configurations with a new approach to delivering multiple sounds using the Chronos audio subsystem. Chronos adopts a novel approach to the mixing and buffering of multiple sounds that reduces onset latencies when two sounds are played together. The strengths and limitations to these approaches are also discussed.

## Method

### Hardware

A series of three hardware tests using common personal computer (PC) hardware configurations of multiple performance classes were conducted to examine sound onset latencies. Some configurations were not possible, given compatibility constraints of the hardware components, which prevented a fully crossed design.

The first test used sound onset latencies that occurred when using E-Prime to play one sound using the hardware as it comes *off the shelf*, or without any modifications to the device settings. We believe most users adopt their products and use them in their *off-the-shelf* state, because modifications require a high degree of specialized knowledge to implement, and sometimes require a considerable time commitment for setup and proper bench-testing of these custom configurations. In this test, the most comprehensive crossing possible was adopted, given the selected off-the-shelf hardware and software. The hardware and software components were selected for this test on the basis of known configurations adopted by E-Prime users, and they meet

the E-Prime minimum configuration recommendations. As is detailed in Table 1, the hardware combinations were tested using options from five computers, six SDDs, two Windows operating systems, and four APIs, including Chronos as its own API. At the time of testing, Chronos was only compatible with E-Prime; thus, other presentation software packages were not examined in combination with Chronos.

In the second test, we attempted to optimize the performance of some of the off-the-shelf devices by performing recommended modifications and examining additional equipment that has been recommended by other researchers (e.g., ASIO API cards). We conducted all tests on the Dell Precision WorkStation Q6600 NumProc2 2.4-GHz computer with the Windows 7 operating system. We chose to keep the computer and OS static in order to reduce the burden of an exorbitant number of independent variables. We selected the Dell because it was a PC, which is more common than laptops in research lab setups, and because in the previous sets of tests, SDDs had performed more equivalently on Dell than on HP PCs. We added a new SDD, the AudioBox 22VSL, that used an ASIO API. After some difficulty testing the AudioBox 22VSL in its off-the-shelf configuration, we set its buffer to 10 ms, which greatly improved its performance. In some tests, we switched the SDD's driver from the manufacturer driver to HD Audio, in an attempt to reduce sound onset latencies. The SoundBlaster X-Fi Titanium PCIe was also tested in bit-matched mode, as is recommended by Presentation to achieve optimal sound latencies (see Table 2 for the components tested in the optimization assessment).

**Table 1** Independent variables examined in one-sound onset latency testing

Computer with Model/Processor/Speed
HP Pavilion Q6600 NumProc2 2.4 GHz
HP Pavilion AMD A8-5500 Trinity 3.2 GHz
Dell Precision WorkStation Q6600 NumProc2 2.4 GHz
HP Pavilion dv8 Intel Core-i7820QM 1.73 GHz
HP Pavilion dv7 Intel Core-2 Duo 2.0 GHz
Sound Delivery Device (SDD)
Chronos (onset priority mode aka Mix Mode 1)
On-board
SIIG DP SoundWave 5.1 PCIe
SoundBlaster X-Fi Titanium PCIe
SoundBlaster Recon SoundCore 3D
OS
Windows 7 × 64
Windows 8.1 update 11, 2 × 64
API
Core audio
DirectSound
Chronos (its own API)

In the last test, we compared the sound onset latencies incurred when playing two sounds using E-Prime. Again, we chose to keep the computer and OS static in order to reduce the burden of an exorbitant number of independent variables. We tested Chronos in fixed-onset mode with its configurable buffer set to a purposeful 6-ms delay (i.e., the buffer is user-specified). The ASIO-based card had to be configured with a 10-ms buffer in order to prevent sound-skipping errors. Only one tested card, the SoundBlaster X-Fi Titanium, was capable of entering the proprietary bit-matched playback mode. See Table 3 for the components tested for latencies of playing two sounds.

## Chronos

Chronos is a USB-based multifunction response and stimulus device that collects response data and delivers multiple forms of stimuli under the Windows OS. Chronos provides accurate collection and verification of tactile, auditory, visual, and analog responses, and also provides a precise source of audio and generic analog output timing. Chronos

- features millisecond accuracy, microsecond precision, and consistent sound output latencies across machines,
- uses the Cypress FX2 microcontroller (1 kB), a field programmable gate array (FPGA, 2 kB), and a stereo codec, totaling approximately 3 kB of memory in the audio data path,
- uses the FX2 USB 2.0 microcontroller to communicate over its USB,
- has 16 digital inputs and 16 digital outputs, eliminating the need for a parallel port,
- has an audio jack for speakers and headphones, and
- connects to a PC using a vendor-defined USB interface bound to the WinUSB driver.

**Table 2** One-sound play latency optimization testing device characteristics

Sound delivery device (SDD)
Chronos (onset priority mode, a.k.a. Mix Mode 1)
SoundBlaster X-Fi Titanium PCIe
AudioBox 22VSL
API
Core audio
Chronos (its own API)
ASIO (ASIO4ALL)
Driver
Manufacturer
HD audio
Mode
Bit-matched
Not bit-matched

**Table 3** Two-sound play latency testing device characteristics

Sound delivery device (SDD)
Chronos (fixed-onset mode, a.k.a. Mix Mode 2)
SoundBlaster X-Fi Titanium PCIe
SoundBlaster Recon SoundCore 3D
API
Core audio
Chronos (its own API)
Driver
Manufacturer
HD audio
Mode
Bit-matched
Not bit-matched

All responses collected are synchronized to the E-Prime time domain. Multiple Chronos devices can be connected to a single PC (with E-Prime 2.0 Professional), so as to facilitate multiparticipant experiments in which stimulus presentation and response collection are controlled by a common host.

For the delivery of audio stimuli, Chronos uses its audio subsystem. All volume and panning changes are governed by the host application, in these examples E-Prime. The host application controls how many sound files can be played at a time and how long sounds can be played; Chronos has no impact on the number of sound files that can be played or their durations. This system has two mixing modes that govern how sound is delivered, one optimized for fastest single sound delivery and one optimized for precision mixing of multiple sounds with consistent sound onset latencies. In the former mix mode, Mix Mode 1, also called *onset priority*, Chronos offers a blend of accuracy and precision. In the latter mix mode, Mix Mode 2, also called *fixed-onset*, Chronos offers consistent, precise sound onset latencies well-suited for multiple-sound mixing, with a fixed delay. The fixed-onset mode buffer is user-customizable from 4 to 10 ms. In the present study, the fixed mixed mode buffer was configured to 6 ms. Each mix mode was tested separately against other common SDDs across multiple performance classes. See Fig. 1 for a broad representation of how Chronos delivers sound.

To summarize, the Chronos audio data path delivers sound in two ways. Onset priority mode is optimized for single-sound delivery with minimal sound onset latency. Fixed-onset mode is optimized for multiple-sound play and mixing, and adopts a novel approach to sound delivery. To explain the differences, we provide examples (see Fig. 2). In both cases, E-Prime generates the audio data as binary data, and the USB host controller serializes those binary data into electrical, serialized USB signals. Those digital electrical, serialized signals are carried over a wire from the USB to the Chronos

device. Once inside Chronos, the digital audio signal follows similar paths for one-sound and multiple-sound delivery, with a few diversions for multiple-sound delivery.

The general path for all data inside Chronos is the same. Inside Chronos, the FX2 microcontroller converts the serialized data back into binary data. Next those data travel to the FPGA, which reads bulk data from the FX2 and then stores them in an internal variable-size buffer; this buffer is always kept full if data are available from the FX2. Then the FPGA separately serializes one sample at a time from the internal buffer for transmission to the audio codec. If the buffer is empty, “silence” is played. The codec converts the serialized digital sample from the FPGA into an analog sample that can be played through a 1/8-in. stereo jack. The data on the wire are transported from the codec to the speakers using a continuous analog voltage. The speaker converts the analog sample value into a mechanical displacement of the cone, reproducing the sound.

The novel approach that Chronos adopts is in its fixed-onset mode. Chronos can play multiple sounds precisely by adopting a mixing, buffering, and remixing approach to delivering multiple sounds. The ways that Chronos plays one audio file (onset priority mode) versus two audio files (fixed-onset mode) are nearly identical. In both cases, a series of steps occur along the audio path: (1) No sound is playing; (2) an E-Prime SoundOut subobject on a Slide object plays a sound; (3) volume and panning are applied to that sound; (4) samples begin playing nearly immediately (onset priority mode), or querying, mixing, buffering, remixing, and playing occur (fixed-onset mode); (5) the sound ends; and (6) the audio thread transfers no data (onset priority mode) or a buffer of full silence is sent to Chronos (fixed-onset mode).

In fixed-onset mode, when playing multiple sounds, the buffering, mixing, remixing, and playing process allows for higher precision in multiple-sound play. For example, when playing two sounds, Chronos begins in fixed-onset mode by continuously playing a buffer of silence. After the Slide SoundOut plays the first sound, an event is sent to wake up the audio thread. Volume and panning are then applied to that sound, and the currently playing buffer of silence is aborted. Chronos is then queried to discard the remaining samples in the FX2 buffer and to report the sum of committed samples (i.e., the samples played plus those samples in the FPGA buffer). The FPGA buffer ensures a constant delay before the new samples play. At another time (i.e., even microseconds later), another sound is played, which sets a second event, waking a thread. The existing transfer is then aborted, and Chronos is queried to discard the FX2 buffer sample and report back the sum of the committed samples. Samples from the first sound that were discarded from the FX2 buffer and samples that had not been transferred are then recalled from a history buffer to be mixed with the second sound. The buffer of mixed sound is sent to Chronos with the constant delay. At some point, the



**Fig. 1** Chronos audio subsystem approach. The Chronos audio subsystem adopts a buffering, aborting, querying, and remixing approach to the delivery of audio in one of two modes: onset priority

mode (Mix Mode 1) and fixed-onset mode (Mix Mode 2). E-Prime signals sounds that are then delivered via the Chronos audio subsystem

shorter sound ends and an event is set again. Chronos is queried to discard the FX2 samples and report back the sum of committed samples. Samples from the first sound that were uncommitted are recalled from the history buffer and sent to Chronos again, and the buffer is sent to Chronos with the constant delay.

### Design and procedure

In brief, a waveform file was played using E-Prime version 2.0.10.353 SP1 and delivered via Chronos, on-board audio, or a sound card (i.e., an SDD) across multiple performance classes of PC, including towers and laptops. The selected machines and hardware are known to be used by researchers in

laboratories to collect behavioral data through interaction with E-Prime users via customer support and networking. The difference between initiating the sound signal and playing the sound was recorded using BBTK version 2.0; this is termed the *sound onset latency*, or SOL. This measurement of SOL is consistent with most researchers' interpretation of latencies; we refer to the *latency* as the time elapsed between the sending of an audio signal and its actual play time.

Often, the term *sound latency* can be used to refer to the time elapsed between the *intended* delay (e.g., buffer size, delay from bench tests) and the *actual* delay of sound presentation. That definition of is different from the latency reported in this article. Defined in the former manner, *latency* refers to a *relative difference from the already expected delay*, a common



**Fig. 2** The Chronos audio data path. E-Prime generates the audio data as binary data, which are then transmitted as serialized data via USB. Then, inside Chronos, the FX2 microcontroller reverses the serialized data back to binary. The field programmable gate array (FPGA) reads bulk data in from the FX2 and stores them in a buffer, which is always kept full if data

are available. Then the FPGA produces a serializing sample for the audio codec. If the buffer is empty, “silence” is played. The codec converts the serialized digital sample from the FPGA into an analog sample that can be played through a 1/8-in. stereo jack

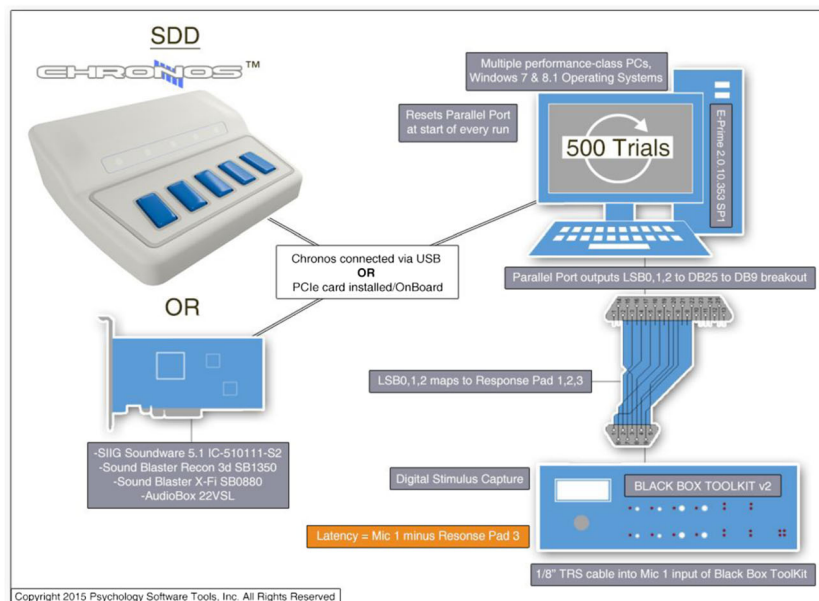
way of defining latencies in hardware and software studies. We will call this the *relative latency*. In this article, we examine latency in terms more akin to the way that researchers view latency, as the *absolute difference between the time a sound file is transmitted and the time the sound actually plays*. We will call this the *absolute latency*. Thus, sound latencies using different definitions of latency, such as the relative and the absolute latency, are not comparable, and close attention should be paid when interpreting latency reports. The relative latency, by definition, is much smaller than the absolute latency, because it factors in a buffer of intended delay. Relative latencies might be more accurately conceptualized as deviations from what we call *absolute latencies*. For example, the Presentation software sometimes treats sound latency as the time elapsed from an intended delay, or the relative latency. In other words, the software tracks the time from when a sound is sent, delayed for an expected period of time, and then measures the delay that occurs *after the intended delay and report it as the “relative” sound latency*. Differing definitions of latency make comparisons between white papers, journal articles, and websites difficult, even when the same equipment is tested. Here, we test the *absolute latency*, the time elapsed between the signaling of a sound to be play and when it is played. From this point forward, we refer to the *absolute latency* as the *SOL*.

The procedure to collect each SOL sample follows. For all tests, sound files were played, from the start of the waveform, through the currently selected E-Prime Sound Device API (Core Audio, DirectSound, or Chronos). See Fig. 3 for a visual

illustration of the SOL recording procedure. In short, a SDD connected via USB or installed/on-board on computers played a waveform file through E-Prime. Each computer’s parallel port was used for transistor-to-transistor logic (TTL) signaling, combined with digital stimulus capture sessions on the BBTK. This allowed for recording and comparisons of the waveform signal edges on the BBTKs response pads (i.e., 1/2/3) and the Microphone 1 input channels. Data bit 0 of the parallel port mapped to Response Pads 1–3, one after the other. All data bits were toggled via an E-Prime task event. TTL pulses traveled through the parallel ports to a breakout that was then read into the response pad input at the back of BBTK, where the onset and offset of the E-Prime task events occurred (when sounds were intended to play). The SDD was connected to a 1/8-in. TRS cable that recorded the actual delivery of the sound produced from playing the sound file through the E-Prime task event.

SOLs were calculated by subtracting the timestamp registered at Response Pad 3 from the timestamp at the microphone using BBTK. The BBTK sampled every 250  $\mu\text{s}$ , which created millisecond statistics accurate to the tenth decimal place. Mean latency statistics are reported in milliseconds to the tenth position for this reason. See Fig. 3 for a representation of the data collection procedure.

E-Prime display settings included 32-bit color depth, flipping enabled, 25 % refresh alignment, and 640  $\times$  480 resolution (unless the graphics adapter forced a minimum resolution higher than 640  $\times$  480, at which time the next lowest resolution crossing was used).



**Fig. 3** Audio latency data collection procedure. The sound onset latency sampling procedure is pictured above. A sound delivery device (SDD) was connected to multiple performance-class PCs operating on Windows 7 and Windows 8.1. The parallel port on the PC was reset at the start of every run. The PC used E-Prime to send a sound through parallel-port

output to response pads plugged into the Black Box Toolkit (BBTK). The sound onset latency was calculated as the time (in milliseconds) from registering the sound on Response Pad 3 to the time the sound registered on the microphone

Common PC hardware configurations of multiple performance classes were made, each containing one SDD (e.g., Chronos, SoundBlaster, or SIIG). Chronos was tested in both onset priority mode and fixed-onset mode, and descriptive data were calculated for each. For the most comprehensive comparison across SDDs and various other factors, Chronos onset priority mode was used. Analyses of variance (ANOVAs) were used to compare the SOLs of the different hardware configurations. When the assumption of homogeneity of variances was violated, follow-up one-way ANOVAs using the Welch method were used to confirm any effects (Welch, 1951). Monte Carlo studies have shown that the Welch method provides power superior to that of the Brown–Forsythe method, and better control of Type I errors, when extreme means are paired with small variances (Tomarken & Serlin, 1986), the case that was most frequently observed here. The same SDDs were repeatedly tested on differing computer hardware, different operating systems, and different APIs, with Chronos as its own API.

A total of 500 samples of SOLs were collected in each unique configuration. Previous testing established that 500 samples were sufficient to detect all effect sizes. When samples were contaminated through a physical bump to the testing station, an erroneous setup of the testing station, or a SOL was obtained that was outside the possible range of the observation, the case was deleted, which resulted in some configurations with fewer than 500 observations.

For all tests comparing one-sound SOLs, an E-Prime experiment played one sound file. The experiment included a fixation followed by a single Slide object that played a sound file. Two SlideText objects were displayed for 50 ms total, while simultaneously (i.e., 0-ms offset) a 1,000-ms total duration waveform file was played that contained a 10-ms lead of white noise. The waveform file contained 10 ms of white noise on the right channel, from times 0 to 10 ms, followed by 990 ms of silence. The file was a two-channel, uncompressed file, cycling 48,000 samples per second with a byte rate of 192,000 (16 bits per sample). A single sound was played in this procedure. This procedure was used to test Chronos in onset priority mode, as well as every other SDD.

When testing SDDs playing two sounds, Chronos was tested in fixed-onset mode. The procedure for playing audio and subsequent recording of the SOL differed from the procedure for one-sound testing. In this mode, the goal was to mix sounds; thus, two consecutive Slide objects played one sound each, resulting in two sounds playing per trial. The first Slide object playing the first sound set parallel port Data Bit 1 to high (+2.4–5v) at the onset of the Slide. This Slide displayed two SlideText subobjects for 50 ms total, for diagnostic readouts, while simultaneously (0-ms offset) playing a 750-ms waveform sound file containing 750 ms of noise on the left channel. As in the prior procedure, two SlideText objects played sounds for 50 ms total while simultaneously (0-ms

offset) playing the aforementioned waveform file, which contained 10 ms of white noise on the right channel, from times 0 to 10 ms, followed by 990 ms of silence. This resulted in the playing of two waveform files, such that each was played in isolation, with overlap during only a portion of their playtimes. This procedure mixed the delivery of multiple sound files, delivering simultaneous and precise sound play easily. This is a difficult task for many sound cards, and mixing sounds as was done here often requires special configurations; Chronos requires no special configuration, due to its fixed-onset mode.

## Results

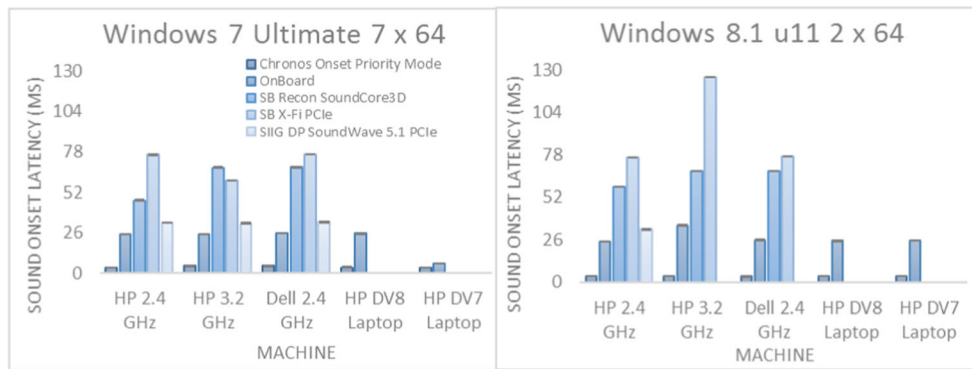
### Off-the-shelf SOL testing for playing one audio file

The effect of switching OS was dependent on SDD, machine, and API. The four-way interaction between OS, SDD, machine, and API was significant, as were all three-way interactions and the follow-up robust tests of equality of means for main effects using the Welch method for calculating  $F$ . The largest observed effect in which OS was involved was the three-way interaction between OS, device, and machine,  $F(4, 37935) = 2166.37, p = .000, \eta_p^2 = .19$ .

As can be seen in Fig. 4, the three-way interaction between OS, SDD, and machine was significant. Across machines and OSs, both Chronos and on-board SDDs performed consistently. Chronos continued to provide the lowest, consistent sound onset latencies. Under the Windows 7 Ultimate OS, Chronos varied only 1.0 ms across different machines, and it varied only 0.3 ms across different machines under the Windows 8 update 11 (u11) OS. Across all configurations, Chronos latencies varied only 1.0 ms. On-board sound varied approximately 19 ms across machines running under the Windows 7 OS, and only 10 ms under the Windows 8.1 u11 OS. Across all configurations, on-board sound latencies varied only 28 ms.

Traditional sounds cards that perform well in one machine and version of Windows OS do not necessarily perform consistently well in another OS, hence the three-way interaction. For example, the SoundBlaster Recon SoundCore 3D card offered consistent sound delivery under Windows 7 with an average SOL of about 68 ms on the HP 3.2-GHz machine as well as the Dell 2.4-GHz machine. However, it performed faster on the HP 2.4-GHz machine with an average SOL of 47 ms. Under the Windows 8.1 u11 OS, this same card (SB Recon SoundCore 3D) performed equivalently on the HP 3.2 GHz machine and the Dell 2.4-GHz machine with an average SOL of 68 ms. However, this card performs slower on the HP 2.4-GHz machine under the updated OS by approximately 11 ms, with an average SOL of 58 ms. Also, the SoundBlaster, X-Fi PCIe, performed relatively consistently under Windows 7 and Windows 8.1 u11 on both the HP and





**Fig. 4** Sound onset latencies across machine and sound delivery devices (SDDs) by operating system (OS). The three-way interaction between OS, SDD, and machine was significant. Across machines and OSs, both

Chronos and on-board SDDs performed relatively consistently, with Chronos offering the most stable performance. Error bars reflect  $\pm 1$  standard error of the mean

Dell 2.4-GHz machines. The SOLs were around 76 ms in each of these configurations. Updating to Windows 8.1 u11 delayed SOLs ( $m = 125.8, SD = 0.2$ ) by about 66 ms when

running on the HP 3.2-GHz PC, as compared to running under Windows 7 ( $M = 59.3, SD = 0.1$ ). See Table 4 for the SOLs by OS, machine, and SDD.

**Table 4** Sound onset latencies by OS, machine, and sound delivery device (SDD; in milliseconds)

SDD		HP 2.4 GHz	HP 3.2 GHz	Dell 2.4 GHz	HP DV8 Laptop	HP DV7 Laptop
Windows 7 Ultimate 7 × 64						
Chronos	Mean (SD)	0.6 (0.1)	0.5(0.1)	0.5(0.1)	0.6(0.1)	0.5(0.1)
	N	500	500	499	500	500
	Min–Max	0.3–1.0	0.3–0.6	0.3–0.6	0.5–1.0	0.5–0.8
On-Board	Mean (SD)	25.0(0.1)	25.0(0.1)	25.6(0.1)	25.4(0.1)	6.3(0.2)
	N	1,000	1000	1,000	1,499	500
	Min–Max	4.3–52.8	4.3–72.5	5.0–60.0	4.8–53.8	2.8–8.3
SB Recon SoundCore3D	Mean (SD)	46.9 (0.1)	67.7(0.1)	67.9(0.1)	–	–
	N	1,500	999	1,000	–	–
	Min–Max	5.0–85.5	45.0–94.0	45.0–108.3	–	–
SB X-Fi PCIe	Mean (SD)	76.0(0.1)	59.3(0.2)	76.5(0.1)	–	–
	N	1,000	1,000	1,000	–	–
	Min–Max	21.5–168.0	21.5–355.8	21.5–162.3	–	–
SIIG DP SoundWave 5.1 PCIe	Mean (SD)	32.3(0.1)	32.1 (0.1)	32.8(0.1)	–	–
	N	1,000	1,000	1,000	–	–
	Min–Max	0.3–72.8	0.3–70.8	0.3–93.5	–	–
Windows 8.1 u11 2 × 64						
Chronos	Mean (SD)	0.5 (0.1)	0.5(0.1)	0.5(0.1)	0.7(0.1)	0.6 (0.1)
	N	500	500	500	500	500
	Min–Max	0.2–0.8	0.2–0.8	0.3–0.6	0.2–1.0	0.5–1.0
On-Board	Mean (SD)	25.1 (0.1)	34.8 (0.1)	25.8 (0.1)	25.4 (0.1)	25.5 (0.1)
	N	1,499	1,000	1,000	1,500	1,498
	Min–Max	4.3–53.8	4.3–100.8	5.0–84.5	4.8–90.3	4.3–54.0
SB Recon SoundCore3D	Mean (SD)	58.1 (0.1)	67.9(0.1)	68.0(0.1)	–	–
	N	1,000	1,000	1,000	–	–
	Min–Max	21.5–122.5	45.0–94.0	45.0–107.8	–	–
SB X-Fi PCIe	Mean in ms (SD)	76.3(0.1)	125.8(0.2)	76.8(0.1)	–	–
	N	1,000	500	1,000	–	–
	Min–Max	21.5–140.5	112.3–141.5	0.3–141.0	–	–
SIIG DP SoundWave 5.1 PCIe	Mean (SD)	32.4(0.1)	–	–	–	–
	N	1,000	–	–	–	–
	Min–Max	0.3–70.8	–	–	–	–

Sample sizes varied because machines were tested loaded with varying SDDs and running different APIs. A full crossing of all hardware and software to every possible combination was not possible due to hardware incompatibilities

The use of traditional sound cards and on-board audio (i.e., excluding Chronos) on multiple machines and OSs can impact SOL variability with a magnitude of 120 ms. This variability excludes the inconsistency introduced by using different APIs, which only exacerbates the magnitude of variability. Adoption of sound delivery systems designed to reduce SOL variability can reduce the variability on multiple machines and OSs to approximately 1 ms.

### Variations in audio latencies by device

In the present study, the tested SDDs included sound cards and Chronos. Sound cards used in modern computers suffer from a startup latency, which results in a delay between when the sound is requested to be played and the time at which it can be detected as being played through speakers, earphones, or recording devices. Startup latencies typically range from a few milliseconds to several seconds and can vary dramatically among different SDDs. In Mix Mode 1 (i.e., onset priority mode), Chronos was designed to minimize such latencies through its unique approach to preloading and buffering audio to produce small, consistent SOLs. In Mix Mode 2 (fixed-onset mode), Chronos was optimized to produce a fixed latency with limited variability, for researchers who prioritize precision over speed. See Table 5 for the SOLs observed in both mix modes of the Chronos audio subsystem across all configurations.

We examined SOL variability across different SDDs. It was predicted that Chronos would have the smallest timing delay across all SDDs. For playing one sound and measuring SOL variability, the fullest spectrum possible of crossing API, SDD, machine, and OS was tested. A range of SOLs from less than 1 to 77.7 ms was observed across the five tested SDDs. The best-performing SDD was Chronos, followed by on-board SDDs. Chronos delivered sound less than 1 ms after the sound was requested, and the variability was extremely narrow ( $SD = 0.1$  ms). As we predicted, Chronos delivered the fastest and most consistent signal of all SDDs tested. A one-way ANOVA using the Welch method (due to a homogeneity-of-variances violation) confirmed this large

effect,  $F(4, 32490) = 6,875.37, p = .000, \eta_p^2 = .46$ . Bonferroni post-hoc comparisons were used to reduce the Type I error and confirmed that Chronos delivered sounds faster than every other SDD. Chronos delivered sounds approximately 21 ms to 77 ms closer to the intended delivery time than did the other sound SDDs. As is shown in Table 6, Chronos delivered sound accurately (within 1 ms) and precisely (0.1-ms variability).

### Variations in audio latencies by API

APIs make it possible to move information between programs. Sound cards rely on APIs to govern communication between the applications, such as drivers and programs, that actually send a sound signal, and those that play the sounds, such as E-Prime. As such, different APIs can impact the performance of SDDs when APIs are required. Chronos does not require an API and communicates with E-Prime directly. Therefore, Chronos is treated as its own API. Two common APIs, DirectSound and Core Audio, were compared using the same SDDs to illustrate the impact of varying APIs. Given that DirectSound has often been criticized for its long latencies and high variability, it was not surprising to find that Core Audio outperformed DirectSound on every SDD with which it was tested; DirectSound lagged in delivering sound by up to 92 ms relative to Core Audio. At its best, Core Audio delivered sound within 3 ms of the sound request. However, Chronos still significantly outperformed Core Audio, by delivering sounds 2.4 ms faster on the best-performing sound card (SIIG DP SoundWave 5.1 PCIe). The effect of device variability on SOLs was dependent on API, such that the use of Core Audio could lower SOLs and SOL variability,  $F(3, 32486) = 14,877.31, p = .000, \eta_p^2 = .56$ . Better yet, the use of Chronos resulted in the smallest SOLs and SOL variability, indicating the most accurate and precise audio delivery method. Bonferroni post-hoc comparisons showed that Chronos delivered sounds faster than Core Audio by 15.8 ms ( $SD = 0.01, p = .000$ ), and faster than DirectSound by 75.7 ms ( $SD = 0.01, p = .000$ ). Given the homogeneity-of-variances violation, the  $F$  for the main effect of API was calculated using the Welch method. The effect remained robust,  $F_{\text{Welch}}(2, 18181.40) = 46,708.86, p = .000$ . See Table 7 for SOLs by SDD and API.

As is shown in Fig. 5, SOLs varied dramatically by SDD and API configurations. The effect of SDD variability was dependent on API. Core Audio produced lower SOLs across all classes of sound cards when excluding Chronos. Therefore, Core Audio produced more precise audio stimulus delivery than DirectSound. However, Core Audio still had less accurate audio stimulus delivery than did Chronos, by approximately 16 ms. The difference in SOLs moving from Core Audio to DirectSound in traditional sound cards ranged from approximately 40 to 85 ms. Even the fastest traditional SDD,

**Table 5** Sound onset latencies by sound delivery device across all configurations playing one sound

Mix mode	M (ms)	Min (ms)	Sound onset latencies		
			Max (ms)	SD	<i>n</i>
Chronos onset priority mode (one-sound)	0.5	0.2	1.0	0.1	4999
Chronos fixed-onset mode (two-sound)	6.9	6.3	9.5	0.6	5500

The customizable sound buffer was set to 6 ms for fixed-onset mode

**Table 6** Sound onset latencies by sound delivery device playing one sound

Sound Delivery Devices (SDDs)	Sound onset latencies					Post-hocs SOL increase from Chronos (ms)
	M (ms)	Min (ms)	Max (ms)	SD	<i>n</i> *	
Chronos (onset priority mode)	0.5	0.2	1.0	0.1	4,999	–
On-Board	22.0	2.8	100.8	20.1	11,496	+21.5**
SIIG DP SoundWave 5.1 PCIe	32.4	0.3	93.5	29.7	4,000	+31.9**
SoundBlaster X-Fi PCIe	77.8	21.5	355.8	47.7	5,501	+77.2**
SoundBlaster Recon SoundCore 30D	59.4	5.0	122.5	27.6	6,499	+58.9**

\* Sample sizes varied because SDDs were tested on multiple machines (e.g., PCs and laptops) and each SDD was compatible with a different number of hardware devices. \*\*  $p = .000$

on-board devices, was slower than Chronos, though only by about 6 ms, but these still showed greater variability than Chronos. This suggests that SOLs using off-the-shelf equipment in different combinations (SDDs and APIs) can range from about 6 ms (on-board using Core Audio) to 112 ms (SoundBlaster X-Fi PCIe), a range of over 100 ms. Using Chronos with any existing system can sidestep the variability caused by differing APIs on devices, by being its own API and by delivering sounds within 1 ms of the audio stimulus request and with microsecond precision.

### Variations in audio latencies by machine

To fully explore the nature of the impact of varying hardware and OS configurations on SOLs, two additional factors were explored: machine (PCs and laptops) and OS (Windows 7,  $7 \times 64$ , or Windows 8.1 u11,  $2 \times 64$ ). Researchers rarely ensure that all machines in a laboratory are identical. Multisite studies that operate using identical machines for data collection are not frequent.

We tested five relatively similar machines: three PCs and two laptops (refer back to Table 1 in the Method section for the specifications), including two HP Pavilion PCs, one Dell PC,

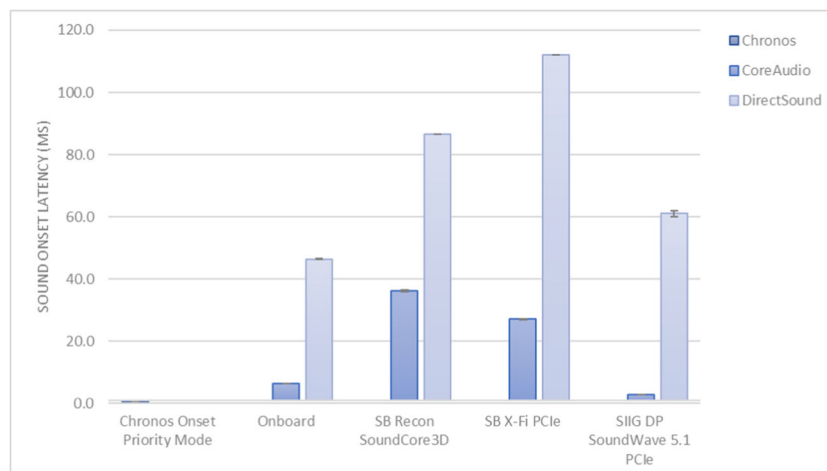
and two HP laptops. The processor speeds varied from 1.73 to 3.2 GHz. Across all SDDs and APIs, we found significant variability across these five machines,  $F_{\text{Welch}}(4, 17979.32) = 2,579.14$ ,  $p = .000$ ,  $\eta_p^2 = .06$ . The machine that delivered audio fastest and with least variability was the HP Pavilion dv7 Intel Core-2 Duo 2.0-GHz laptop, with a mean SOL of 9.8 ms ( $SD = 13.4$ ). The slowest machine was the Dell Precision WorkStation Q6600 NumProc2 2.4-GHz PC, with a mean SOL of 42.4 ms ( $SD = 40.6$ ). SOLs varied by machine up to 32 ms. Comparing two PCs with similar processors and speeds, yet different models (the HP Pavilion Q6600 NumProc2 2.4-GHz and Dell Precision WorkStation Q6600 NumProc2 2.4-GHz vary only by model) results in SOL differences of approximately 7 ms (see Table 8 for the SOLs by machine).

Exploring the impact of machine selection on SOL variability further, we found that the effect of machine on SOL variability was influenced by SDD selection. The interaction of machine with device was significant,  $F(3, 37962) = 1, 989.72$ ,  $p = .00$ ,  $\eta_p^2 = .24$ . However, the homogeneity-of-variances assumption was violated. Follow-up one-way ANOVAs found that each main effect was robust against the variance violation. The effect of machine was significant, with

**Table 7** Sound onset latencies by SDD and API

Sound Delivery Devices (SDDs)	Descriptive statistics									
	M (ms)	SD	<i>n</i> *	Min	Max					
Chronos (onset priority mode)	0.5	0.1	4,999	0.2	1.0					
	Core audio					Direct sound				
	M (ms)	SD	<i>n</i> *	Min	Max	M (ms)	SD	<i>n</i> *	Min	Max
On-Board	6.3	1.0	6,999	2.8	8.5	46.4	7.4	4,497	34.5	100.8
SIIG DP SoundWave 5.1 PCIe	2.86	1.5	2,000	0.3	5.5	61.0	44.1	2,000	51.3	93.5
SoundBlaster X-Fi Titanium PCIe	27.1	8.1	2,502	21.5	355.8	112.0	1.4	2,999	76.5	168.0
SoundBlaster Recon SoundCore 3D	36.2	14.2	3,500	5.0	48.5	86.6	6.4	2,999	64.0	122.3

\* Sample sizes varied because SDDs were tested on multiple machines (e.g., PCs and laptops) using every API possible. A full crossing of all factors was not possible due to hardware incompatibilities



**Fig. 5** Sound onset latencies (SOLs) across sound delivery devices (SDDs) and API combinations. Chronos delivered the most accurate and precise audio stimuli. The effect of the SDD variable was dependent on the application program interface (API), such that Core Audio

produced lower SOLs across all classes of sound cards. However, some sound cards were less affected by varying API (e.g., the SB Recon SoundCore 3D, which had smaller variability), yet delivered audio imprecisely. Error bars reflect  $\pm 1$  standard error of the mean

$F_{Welch}(4, 33575.92) = 1,438.33, p = .000$ . The effect of device was also significant, with  $F_{Welch}(4, 15066.46) = 12,769.98, p = .000$ .

Data showed that some SDDs can combat the variability introduced by using different machines. This highlights the need for diligence when selecting machines and SDDs when precise and accurate audio timing is required. As can be seen in Fig. 6, the lowest SOLs are produced when using Chronos. Chronos varied only 0.6 ms across all machines tested, and ranged from 3.6 ms ( $SD = 0.1$ ) to 4.2 ms ( $SD = 0.1$ ) on the 2.4-GHz and 3.2-GHz HPs, respectively. The next best option was a matter of preference. If lower SOL were desirable, it would be best to use the on-board SDDs, which produced SOLs 25.1 ms ( $SD = 0.1$ ) to 29.9 ( $SD = 0.1$ ) in magnitude; again the 2.4-GHz HP performed fastest, and the 3.2-GHz HP performed slowest. If precision were more desirable than accuracy, the next best-tested option after Chronos was the SIIG SP SoundWave 5.1 PCIe sound card. The SoundWave card produced SOLs that ranged from 32.1 ( $SD = 0.2$ ) to 32.8

( $SD = 0.2$ ) ms, providing consistent SOLs across all three tested PCs. Chronos remained the best option, since it most accurately and precisely delivered audio across all machines.

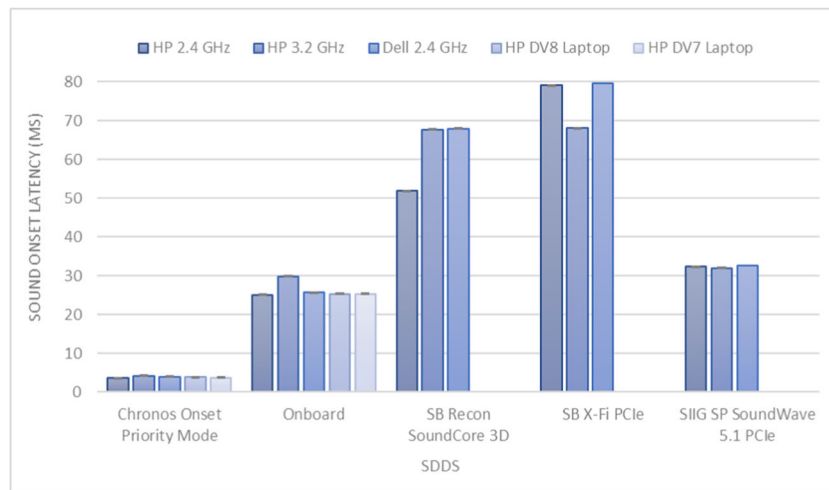
**Variations in audio latencies by OS**

OS variability can also impact SOLs. The rates at which different departments and universities make updates to the newer versions of OSs following releases may differ, impacting SOL variability. It is not unlikely to have different OSs on different computers within a single laboratory. Perhaps surprisingly, updates to an OS may not always result in lower SOLs for every API and SDD. The following data illustrate the impact of varying machines, OS, SDDs, and APIs on SOLs. Overall, holding all other factors constant, we found that updating from Windows 7 Ultimate ( $M = 31.5, SD = 35.2$ ) to Windows 8.1 u11 ( $M = 33.7, SD = 38.6$ ) slowed the average SOL only by about 2 ms,  $F_{Welch}(1, 36577.83) = 32.35, p = .000, \eta_p^2 = .07$ .

**Table 8** Sound onset latencies by machine

Computer with Model/Processor/Speed	Sound onset latencies (ms)				
	<i>M</i>	<i>SD</i>	<i>n</i> *	Min	Max
HP Pavilion Q6600 NumProc2 2.4 GHz	35.7	38.2	10,999	0.3	168.0
HP Pavilion AMD A8-5500 Trinity 3.2 GHz	40.0	38.4	9,000	0.3	355.8
Dell Precision WorkStation Q6600 NumProc2 2.4 GHz	42.4	40.6	8,999	0.3	162.3
HP Pavilion dv8 Intel Core-i7820QM 1.73 GHz	13.0	16.0	4,999	0.2	90.3
HP Pavilion dv7 Intel Core-2 Duo 2.0 GHz	9.8	13.4	3,998	0.3	54.0

\* Sample sizes varied because machines were tested with varying SDDs and APIs. A full crossing of all hardware and software was not possible due to incompatibilities



**Fig. 6** Sound onset latencies across machine and sound delivery devices (SDDs). Chronos delivered the most accurate and precise audio stimuli. The effect of machine variability on sound delivery was dependent on the SDD. Error bars reflect  $\pm 1$  standard error of the mean

### SOL optimization tests for playing one-audio file

For those with a little ingenuity and understanding of how audio is delivered, optimization of sound latencies is possible. Recommended configurations were compared against Chronos in its off-the-shelf state in fixed-onset mode. In fixed-onset mode, the buffer must be configured. In the present study, the buffer was configured to 6 ms. All tests were performed on a Dell Precision WorkStation Q6600 NumProc2 with a 2.4-GHz processor using Windows 7. The results could vary if different hardware and software configurations were used. We investigated three feasible adjustments for lowering SOLs. First, the driver used by the SDD from the manufacturer's driver was adjusted to HD Audio, which typically outperforms the manufacturer's driver. Second, as is recommended by the makers of Presentation (Neurobehavioral Systems, 2015), we tested the recommended X-Fi Titanium SDD in bit-matched playback mode. Bit-matched playback mode removes several preadjustments to the sound data, removing the delay caused by those adjustments. Many cards from Creative Labs (X-Fi) have a "Bit-Matched Playback" mode or a "Stereo Direct/Bit Accurate Playback" mode, intended to reduce the delays between presenting and playing a sound. Finally, we tested an ASIO-based SDD, the AudioBox 22VSL, that has been reported to perform well by other researchers. We predicted significant differences across these configurations, with Chronos achieving the lowest SOL.

A variety of methods can be used to successfully reduce onset latencies. One option is to adopt a system, such as Chronos, that can be used directly out of the box to achieve the lowest observed SOL and smallest variability. As we predicted, we found differences in sound onset latency by configuration, with Chronos achieving the smallest SOL,  $F_{\text{Welch}}(4, 1005.49) = 88,702.77$ ,  $p = .000$ ,  $\eta_p^2 = .97$ . Chronos delivered sound within a millisecond with virtually

no variability. Other configurations achieved small onset latencies, as well. Bonferroni post-hoc comparisons showed that each adjustment resulted in significantly smaller onset latencies.

The slowest-responding SDD tested was the AudioBox 22VSL card paired with the ASIO4ALL driver (see Table 9). Its average SOL was 30.8 ms. A 30-ms lag is reasonable, especially when paired with a reasonable degree of variability ( $SD = 1.2$  ms) and small range (5.5 ms). The SoundBlaster X-Fi Titanium (X-Fi) using the Core Audio API performed faster than the AudioBox 22VSL, but not necessarily better. Using the manufacturer's driver and its out-of-the box configuration, the X-Fi performed faster, with an average SOL of 26.4 ms, but with a higher degree of variability ( $SD = 3.1$ ) and a wider range at 10.8 ms. Entering bit-matched playback mode again reduced the average SOL to 16.6 ms but still maintained high degrees of variability ( $SD = 3.0$ ) and dispersion (range = 10.7). Moving out of bit-matched playback mode and using the X-Fi card with HD Audio drivers (and the Core Audio API) resulted in another drop in both SOL and variability. After Chronos, the X-Fi card using an HD Audio driver was the next best option for accuracy ( $M_{\text{SOL}} = 3.2$  ms) and precision ( $SD = 1.0$ , range = 3.3). Chronos in onset priority mode still delivered sound most accurately ( $M_{\text{SOL}} = 0.5$  ms) and most precisely ( $SD = 0.1$ , range = 0.3) in its out-of-the box state. No configuration adjustments were required to achieve this degree of accuracy and precision with the E-Prime system.

### SOL optimization tests for playing two audio files

To examine SOLs for playing two sounds, or *multiple-sound onset latencies*, we selected Chronos and the two best-performing Creative SDDs, the SoundBlaster Recon SoundCore 3D and the SoundBlaster X-Fi Titanium PCIe.

**Table 9** Single sound onset latencies by configuration: optimization test

Configuration	<i>M</i>	<i>SD</i>	Min	Max	Range	<i>n</i> *
AudioBox 22VSL	30.8	1.2	28.0	33.5	5.5	500
SoundBlaster X-Fi Titanium PCIe–Core Audio Using Manufacturer’s Driver	26.4	3.1	21.5	32.3	10.8	500
SoundBlaster X-Fi Titanium PCIe–Core Audio in Bit-Matched Playback Mode	16.6	3.0	11.3	22.0	10.7	500
SoundBlaster X-Fi Titanium PCIe–Core Audio HD Audio Driver	3.2	1.0	1.5	4.8	3.3	500
Chronos (onset priority mode, or Mix Mode 1)	0.5	0.1	0.3	0.6	0.3	499

The effect of SDD on single sound latencies was significant,  $F_{\text{Welch}}(4, 1005.49) = 88,702.77, p = .000$ . Bonferroni post-hoc comparisons showed a significant difference between every comparison above

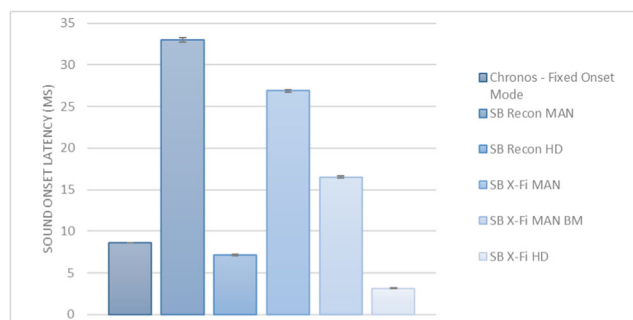
Both of these SDDs are ideal for game-playing, given their increased ability to handle multiple sounds. The X-Fi Titanium card’s setting can be modified and entered into bit-matched playback mode, as is recommended by Neurobehavioral Systems to reduce SOLs. We tested bit-matched playback mode and modified both Creative SDDs’ settings to use HD Audio drivers to further reduce SOLs. All data were collected on a Dell Precision WorkStation Q6600 NumProc2 with a 2.4-GHz processor using Windows 7.

Figure 7 shows that sound onset latencies varied by configuration. The fastest SDD was the SoundBlaster X-Fi Titanium using an HD Audio driver,  $F_{\text{Welch}}(4, 1170.78) = 12,074.95, p = .000, \eta_p^2 = .94$ . Bonferroni post-hoc comparisons showed significant differences between the tested configurations. The next-best-performing SDD after the sound cards using HD Audio drivers was Chronos.

Table 10 shows the descriptive statistics for all tested configurations of SDDs, modes, and drivers playing two sounds. Of those tested, the SoundBlaster Recon SoundCore 3D paired with the manufacturer’s driver in its off-the-shelf status performed the worst, in terms of both SOL ( $M = 33.0$  ms) and variance ( $SD = 5.4$ , range = 19.8). The fastest configuration was the SoundBlaster X-Fi Titanium PCIe paired with HD Audio drivers, which achieved an average 3.2-ms delay with

acceptable variance ( $SD = 1.0$ , range = 3.3). Chronos in fixed-onset mode was the next fastest option after Recon with the HD driver for delivering multiple sounds, mostly because its buffer was set to 6 ms, inflating the SOL. The Recon card performed faster than Chronos but with greater variability. Chronos achieved the smallest variability, with a 1/10th-ms variability and range of 0.3 ms.

Each of the tested configurations varied significantly from the others, indicating a wide range of variation by configurations. For the Creative SDDs tested, using HD Audio in lieu of the manufacturer’s own drivers reduced multiple SOLs and narrowed the standard deviation and the observation range of delays. Entering Creative’s proprietary bit-matched mode also produced an improvement from the manufacturer’s setting on the X-Fi Titanium card, but it does not fare as well as using HD Audio drivers; these two adjustments to the X-Fi card are mutually exclusive and cannot be combined. Thus, adjustments to SDDs can be made to reduce the latencies observed when playing multiple sounds. If users desire the fastest onset of multiple-sound delivery, the SoundBlaster X-Fi Titanium PCIe using HD Audio is the correct solution. For users that prefer known and consistent variability over inconsistent variability, Chronos may be a better choice.



**Fig. 7** Multiple-sound onset latencies by configuration. When delivering multiple sound files, Chronos delivered the most precision, whereas the SoundBlaster X-Fi Titanium PCIe sound card using the HD Audio driver delivered the most accurate, or fastest, but not the most precise audio play. With the Creative cards tested, HD Audio drivers delivered both faster and more precise audio under the Core Audio API. Error bars reflect  $\pm 1$  standard error of the mean

## Discussion

A number of factors were tested to demonstrate the amount of SOL variability that can be introduced into research studies through differing configurations of hardware, operating systems, and SDDs. Even when testing relatively common machines, APIs, and OSs, the present study showed that a great deal of variability was introduced by making small shifts in configurations. The main effects for all tested components were significant (e.g., SDD, API, machine, OS), but these were overshadowed by interactive effects that showed that Chronos reduced the variability introduced by other components.

Of the tested components that varied in these configurations, some were bigger offenders than others. Changes in OS when updating from Windows 7 to Windows 8.1 u11 had a minimal impact on SOLs. This is consistent with the

**Table 10** Multiple SOLs by configuration (playing two sounds)

Configuration	<i>M</i>	<i>SD</i>	Min	Max	Range	<i>n</i> *
Chronos (fixed-onset mode, a.k.a. Mix Mode 2)	8.6	0.1	8.5	8.8	0.3	500
SoundBlaster Recon SoundCore 3D–Core Audio Manufacturer’s Driver	33.0	5.4	28.0	47.8	19.8	499
SoundBlaster Recon SoundCore 3D–Core Audio HD Audio Driver	7.2	1.3	5.3	27.0	21.7	501
SoundBlaster X-Fi Titanium PCIe–Core Audio Bit-Matched Playback Mode	16.5	2.8	11.0	21.0	10.0	500
SoundBlaster X-Fi Titanium PCIe–Core Audio Manufacturer’s Driver (Not Bit-Matched)	26.9	2.9	20.0	31.3	11.3	500
SoundBlaster X-Fi Titanium PCIe–Core Audio HD Audio Driver	3.2	1.0	1.5	4.8	3.3	500

\* The effect of SDD on multiple sound latencies was significant,  $F_{\text{Welch}}(5, 1174.89) = 10015.32, p = .000$ . Bonferroni post-hoc comparisons showed a significant difference between every comparison above

findings reported by Plant and Turner (2009) that updates to new OSs resulted in less accurate and precise timing. Another factor that impacted SOL was API. Chronos performed faster than on-board audio, which performed faster than both Core Audio and DirectSound cards, in that order. The API selection could impact sound latencies by 100 ms. Three APIs are common in addition to Chronos, including Core Audio, DirectSound, and ASIO. In our tests, Core Audio tended to perform better than the DirectSound or ASIO. However, a new sound card hitting the shelves tomorrow might perform fastest, and especially when paired with ASIO drivers. Sound card companies are not likely concerned with achieving accuracy and precision of audio delivery that would meet academic standards when developing their products. Researchers are generally not their target user audience, and they are probably unaware of the difficulties experienced when trying to sync fast-time-course physiological data to audio events.

Choices of SDDs heavily impacted SOLs, with Chronos providing the fastest and most consistent audio delivery. The range of SOLs across SDDs was approximately 75 ms. Data showed that the use of the Chronos audio subsystem guarded against the SOL variability introduced by other components. Across different classes of machines and OSs, Chronos provided low-latency audio file delivery with limited variability. In fact, across configurations, Chronos was able to provide sounds with SOLs less than or equal to 1 ms with variances of approximately a tenth of a millisecond. Thus, the adoption of such a system or similar device that can provide submillisecond accuracy and precision would help researchers capture data more accurately.

In lieu of adopting a system like Chronos, performing lag tests and timing corrections can reduce the impact of varying SOLs on replicability and comparability. Assessing each assembled configuration (e.g., machine, OS, API, and SDD) for its SOL using an external chronometry device such as the BBTK can provide the average latency. This average latency can then be used to move the actual event in analysis forward in time, correcting for the SOL lag. This correction is

essential, especially when synchronizing multiple types of data to the same time-logged sound event.

The increased variability introduced by the ever-increasing availability of such low-cost commodities as off-brand PCs and SDDs will continue. Though not exhaustive, the data presented here show that even among commonly used PCs, a great deal of SOL variability exists. These data show that newer is sometimes better, but not always. We observed that Core Audio universally outperformed its predecessor, DirectSound, and that the new Chronos audio subsystem outperformed all predecessors. But in some cases—for example, when updating to Windows 8.1 u11 from Windows 7 on some devices—there was an increase in SOL. It will continue to be difficult to determine which PC, API, traditional sound card, and operating system are best in a market in which updates and new products hit shelves daily. The products and updates released tomorrow may indeed be improvements on the current contenders. Continued rigorous timing testing will be required to increase the consistency between studies and laboratories and between interconnected devices and systems.

As scientists ask questions that push them to explore on tighter and tighter timelines, systems that can offer submillisecond accuracy and or precision become necessary. The present approach offered by the Chronos audio system reliably delivers and/or timestamps audio stimuli with low latencies between sound signal and sound presentation that vary by  $\leq 1$  ms, independent of hardware and Windows OS/driver combinations. The Chronos audio subsystem adopts a buffering, aborting, querying, and remixing approach to the delivery of audio, using E-Prime as the presentation system and standard E-Objects (e.g., Slide) to achieve consistent 1-ms audio onset latencies without the use of advanced scripting. The resulting audio onset latencies are small, reliable, and consistent across systems. Additional testing of standard audio delivery devices and configurations highlights the need for careful attention to consistency between labs, experiments, and sites in their hardware choices and OS selection. Also, attention is equally important to the adoption of audio delivery

systems designed to sidestep the audio latency variability issue.

We observed what some researchers might consider “reasonable” sound onset latencies when playing one sound using the on-board SDDs on our test machines. Though the observed *mean* SOLs were relatively low (approximately 25–35 ms on PCs and 6–25 ms on laptops), the ranges varied widely. The largest observed range in an on-board configuration was approximately 4–100 ms. It may be troublesome to attempt to sync physiological data that occur on a timeframe smaller than this range (e.g., in EEG). The SOL for time-marked sound events could vary enough to cause overlap, making the syncing of unique events from the presentation software or timestamp to the physiological data extremely problematic. Minor modifications can greatly improve sound onset latencies for researchers with a little ingenuity and diligence. A few modifications made to a relatively inexpensive and quality SDD, the SoundBlaster X-Fi Titanium PCIe, mean that SOLs and their observed range can drop dramatically. Changing the X-Fi playback mode dropped mean SOLs from its off-the-shelf performance of 26 ms to 16 ms, and using an HD Audio to drive the card reduced SOLs to 3 ms. These adjustments also narrowed the SoundBlaster X-Fi’s variability. Thus, a few quick adjustments could go a long way toward reducing both the SOL and its variability. Furthermore, the “reasonable” on-board SOLs observed during bench-testing in the present article are not likely to be observed on machines in laboratories. Some of the machines tested here are used primarily for bench-testing purposes and rarely for other purposes; that is, they are infrequently used. SOLs in laboratories using machines that are used more frequently, and perhaps that are configured using less optimal settings, are likely to be larger.

In this testing, the Core Audio API surpassed DirectSound performance. This is not a green light to universally select Core Audio as the API for SDDs, because the interaction between a sound card and API can greatly impact performance. We also observed that HD Audio drivers outperformed the manufacturer’s drivers that we tested, but again, the interaction between API and driver can greatly impact variability. Overall, the multitude of hardware and software configurations can greatly impact latencies.

There are two solid approaches to minimizing SOL and increasing the accuracy and precision of sound delivery. The first option involves ensuring that hardware and software configurations do not impact SOLs greatly or differently. This can be achieved in a variety of ways. One approach is the “do-it-yourself” method. This involves equipping each testing station in a study (or a discipline) with identical hardware and software configurations, conducting bench tests to ascertain the best software settings on that hardware (e.g., SDDs, APIs, and drivers), using external chronometry devices to test the SOLs for one-sound and multiple-sound delivery from each

testing station, and repeating this after making software or hardware changes (e.g., updates to a new OS) in order to maintain equivalent performance across devices. This is obviously a large time commitment, and for a number of reasons may not be feasible for all researchers who work with audio data. The second approach is to hire a timing specialist to perform these evaluations for you using an external chronometry device (e.g., a BBTK). This method will provide the bench-testing required to assure the most accurate and precise timing possible in your own laboratory. A third approach would be to use on-the-market devices, like Chronos, committed to providing accurate and precise audio delivery across hardware and software configurations.

**Author note** All of the authors work for Psychology Software Tools, Inc., in Sharpsburg, Pennsylvania, a research and innovation company.

## References

- Crosse, M. J., & Lalor, E. C. (2014). The cortical representation of the speech envelope is earlier for audiovisual speech than audio speech. *Journal of Neurophysiology*, *111*, 1400–1408. doi:10.1152/jn.00690.2013 (Retraction published 2014: *Journal of Neurophysiology*, *112*, 2667. doi:10.1152/jn.z9k-2710-retr.2014)
- Kieckhefer, E., & Kanwal, J. S. (2000). A computer program for sequencing and presenting complex sounds for auditory neuroimaging studies. *Journal of Neuroscience Methods*, *101*, 43–48. doi:10.1016/S165-0279(00)00249-1
- Neurobehavioral Systems, Inc. (2015). Presentation system recommendations: sounds cards. Retrieved online on April 29, 2015, from [https://www.neurobs.com/menu\\_presentation/menu\\_hardware/system\\_configuration](https://www.neurobs.com/menu_presentation/menu_hardware/system_configuration)
- Pashler, H., & Wagenmakers, E.-J. (2012). Editors’ introduction to the special section on replicability in psychological science: A crisis of confidence? *Perspectives on Psychological Science*, *7*, 528–530. doi:10.1177/1745691612465253
- Plant, R. R., Hammond, N., & Turner, G. (2004). Self-validating presentation and response timing in cognitive paradigms: How and why? *Behavior Research Methods, Instruments, & Computers*, *36*, 291–303. doi:10.3758/s12315-013-0166-8
- Plant, R. R., Hammond, N., & Whitehouse, T. (2003). How choice of mouse may affect response timing in psychological studies. *Behavior Research Methods, Instruments, & Computers*, *35*, 276–284. doi:10.3758/BF03202553
- Plant, R. R., & Quinlan, T. (2013). Could millisecond timing errors in commonly used equipment be a cause of replication failure in some neuroscience studies? *Cognitive, Affective, & Behavioral Neuroscience*, *13*, 598–614. doi:10.3758/s13415-013-0166-6
- Plant, R. R., & Turner, G. (2009). Millisecond precision psychological research in a world of commodity computers: New hardware, new problems? *Behavior Research Methods*, *41*, 598–614. doi:10.3758/BRM.41.3.598
- Psychology Software Tools, Inc. (2015). E-Prime knowledge base: 1307 – INFO: Sound latency – Not all sound cards provide optimal millisecond timing. Retrieved April 29, 2015, from [www.psnet.net/support/kb.asp?TopicID=1307](http://www.psnet.net/support/kb.asp?TopicID=1307)
- Tomarken, A. J., & Serlin, R. C. (1986). Comparison of ANOVA alternatives under variance heterogeneity and heterogeneity and specific



- noncentrality structures. *Psychological Bulletin*, 99, 90–99. doi:10.1037/0033-2909.99.1.90
- Voyvodic, J. T., Glover, G. H., Greve, D., Gadde, S., & FBIRN. (2011). Automated real-time behavioral and physiological data acquired display integrated with stimulus presentation for fMRI. *Frontiers in Neuroinformatics*, 5, 27. doi:10.3389/fninf.2011.00027
- Welch, B. L. (1951). On the comparison of several mean values: An alternative approach. *Biometrika*, 38, 330–336.