Web Audio Changes in m36

By Chris Wilson
Chris is a contributor to Web**Fundamentals**

Web Audio changes

At Google, we love standards. We're on a mission to build out the standards-defined Web platform. One of the small warts on that for some time has been the webkit- prefixed implementation of the Web Audio API (notably the webkitAudioContext object), and some of the deprecated bits of Web Audio that we've continued to support.

It was originally planned that m36 would remove support for the prefixed webkitAudioContext, since we had begun supporting the unprefixed AudioContext object. This turned out to be more troublesome than expected, so m36 supports both unprefixed and prefixed - however, even in the reintroduced webkitAudioContext, several legacy methods and attributes like createGainNode and createJavaScriptNode have been removed. In short, in m36 webkitAudioContext and AudioContext are aliases of each other; there is no difference in functionality between the two.

We will remove the support for the prefix completely after m36, likely in a couple of releases. We'll make an announcement here when the change is imminent, and we are continuing to reach out to authors to fix their Web Audio applications.

Why have we done this, rather than reverting to the previous implementation? Well, in part, we've been reticent to move too far backwards; we've already removed those APIs, and as a nice side-effect to this aliasing, applications can then work nicely on Firefox, which has never supported a prefixed AudioContext object (and quite right, too!) in their Web Audio support initially released last fall.

The rest of this update provides a guide to fixing things that may be broken in your code due to this change. The great thing about fixing these problems is that your code is then quite likely to just work in Firefox, too! (I'd thought for a long time that my Vocoder application was broken due to Firefox's implementation, but it turned out to be one of these problems!)

If you just want to get up and running, you may want to take a look at a monkey-patch library I wrote for applications that were written to the old Web Audio code - this can help you get up and running in a minimum amount of time, as it will alias the objects and methods appropriately. Indeed, the patches the library lists is a good guide to the things that have changed.

First and foremost:

Any references to `window.webkitAudioContext` should be made to `window.AudioContext` instead. Frequently, this has been fixed with a simple:

```
window.AudioContext = window.AudioContext || window.webkitAudioContext;
```

If your app is responding with something like “Unfortunately, your browser does not support Web Audio. Please use Chrome or Safari.” - it’s quite likely explicitly looking for `webkitAudioContext`. Bad developer! You could have been supporting Firefox for months!

But there are a few other, more subtle code removals, some of which can be less obvious.

- The BiquadFilter enumerated type constants for the `.type` attribute (which is now a string) no longer appear on the `BiquadFilterNode` object, and we do not support them on the `.type` attribute. So you don’t use `.LOWPASS` (or 0) anymore - you set it to “lowpass”.
- Also, the `Oscillator.type` attribute is similarly now a string enumerated type - no more `.SAWTOOTH`.
- `PannerNode.type` is also now a string enumerated type.
- `PannerNode.distanceModel` is also now a string enumerated type.
- `createGainNode` was renamed to `createGain`
- `createDelayNode` was renamed to `createDelay`
- `createJavaScriptNode` was renamed to `createScriptProcessor`
- `AudioBufferSourceNode.noteOn()` is now replaced by `start()`
- `AudioBufferSourceNode.noteGrainOn()` is also now replaced by `start()`
- `AudioBufferSourceNode.noteOff()` is renamed to `stop()`
- `OscillatorNode.noteOn()` is renamed to `start()`
- `OscillatorNode.noteOff()` is renamed to `stop()`
- `AudioParam.setTargetValueAtTime()` is renamed to `setTargetAtTime()`
- `AudioContext.createWaveTable()` and `OscillatorNode.setWaveTable()` are now renamed `createPeriodicWave() and`setPeriodicWave()`.

- `AudioBufferSourceNode.looping` was removed, in favor of `.loop`
- `AudioContext.createBuffer(ArrayBuffer, boolean)` to synchronously decode a blob of encoded audio data has been removed. Synchronous calls that take a long time to complete are poor coding practice; use the asynchronous decodeAudioData call instead. This is one of the more challenging changes - you need to actually change logic flow - but a far better practice. Mozilla's Ehsan Angkari wrote a nice example of how to do this in their post on converting to standard Web Audio.

Many of these (like the renaming of createGainNode, and the removal of the synchronous decoding in createBuffer) will obviously show up in the developer tools console as an error - however, some others, like this usage:

```
myFilterNode.type = myFilterNode.BANDPASS;
```

will not show up at all, and silently fail (myFilterNode.BANDPASS will now resolve to undefined, and the attempt to set .type to undefined will simply fail to produce any effect. This, by the way, was what was causing the vocoder to fail.) Likewise, just assigning the filter.type to a number used to work:

```
myFilterNode.type = 2;
```

But now, you need to use the string enumeration:

```
myFilterNode.type = “bandpass”;
```

So, you may wish to grep your code for the following terms:

- `webkitAudioContext`
- `.LOWPASS`
- `.HIGHPASS`
- `.BANDPASS`
- `.LOWSHELF`
- `.HIGHSHELF`
- `.PEAKING`
- `.NOTCH`
- `.ALLPASS`
- `.SINE`
- `.SQUARE`

- `.SAWTOOTH`
- `.TRIANGLE`
- `.noteOn`
- `.noteGrainOn`
- `.noteOff`
- `.setWaveTable`
- `.createWaveTable`
- `.looping`
- `.EQUALPOWER`
- `.HRTF`
- `.LINEAR`
- `.INVERSE`
- `.EXPONENTIAL`
- `createGainNode`
- `createDelayNode`
- `.type` (yes, this will have lots of false positives - but it’s the only way to catch the last example above!)

Once more, if you're in a hurry and want to get up and running, just grab a copy of my monkeypatch webkitAudioContext library and include it in your application. Happy Audio Hacking!

Last updated July 2, 2018.