Deprecations and Removals in Chrome 58

By Joseph Medley
Technical Writer

In nearly every version of Chrome, we see a significant number of updates and improvements to the product, its performance, and also capabilities of the Web Platform. This article describes the deprecations and removals in Chrome 58, which is in beta as of March 16. This list is subject to change at any time.

Mouse on Android stops firing TouchEvents

Until Chrome 57, Android low-level mouse events in Chrome primarily followed an event path designed for touch interactions. For example, a mouse drag motion occurring while a mouse button is pressed generates `MotionEvents`, delivered through `View.onTouchEvent`.

But since touch events cannot support hover, hovering mousemoves followed a separate path. The design had many side-effects including mouse interactions firing `TouchEvents`, all mouse buttons appearing as *left* mouse buttons, and `MouseEvents` being suppressed by `TouchEvents`.

Starting with Chrome 58, a mouse on Android M or later will:

- No longer fire `TouchEvents`.
- Fire a consistent sequence of `MouseEvents` with appropriate buttons and other properties.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Remove case-insensitive matching for usemap attribute

The `usemap` attribute was formerly defined as caseless. Unfortunately implementing this was complicated enough that no browsers implemented it correctly. Research suggested that such a complicated algorithm is unnecessary, and even ASCII case-insensitive matching is unnecessary.

Consequently, the specification was updated so that case-sensitive matching is applied. The old behavior was deprecated in Chrome 57, and is now removed.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Remove content-initiated top frame navigations to data URLs

Because of their unfamiliarity to non-technical browser users, we're increasingly seeing the `data:` scheme being used in spoofing and phishing attacks. To prevent this, we're blocking web pages from loading `data:` URLs in the top frame. This applies to `&lt;a&gt;` tags, `window.open`, `window.location` and similar mechanisms. The `data:` scheme will still work for resources loaded below by a page.

This feature will be removed in Chrome 60.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Remove deprecated names for motion path properties

Motion path CSS properties allow authors to animate any graphical object along an author-specified path. In compliance with the spec, several properties were implemented in Chrome 45. The names of these properties were changed in the spec in mid 2016. Chrome implemented the new names in Chrome 55 and Chrome 56. Console deprecation warnings were also implemented.

In Chrome 58, the old property names are being removed. The affected properties and their new names are shown below.

Removed Property	Current Name
motion-path	offset-path
motion-offset	offset-distance
motion-rotation	offset-rotate
motion	offset

Intent to Remove

Remove EME from non-secure contexts

Some usages of Encrypted Media Extensions (EME) expose digital rights management implementations that are not open source, involve access to persistent unique identifiers, and/or run unsandboxed or with privileged access. Security risks are increased for sites exposed via non-secure HTTP because they can be attacked by anyone on the channel. Additionally, when user consent is required, acceptance persisted for a non-secure HTTP site can be exploited by such an attacker.

Support for non-secure contexts was removed from the EME version 1 spec and is not supported in the proposed recommendation nor anticipated in the subsequent final. will not be in the upcoming proposed recommendation or subsequent final recommendation. The API has been showing a deprecation message on non-secure origins since Chrome 44 (May 2015). In Chrome 58, it is now removed. This change is part of our broader effort to remove powerful features from unsecure origins.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Remove legacy caller for HTMLEmbedElement and HTMLObjectElement

That an interface has a legacy caller means that an instance can be called as a function. Currently, `HTMLEmbedElement` and `HTMLObjectElement` support this functionality. In Chrome 57 this ability was deprecated. Starting in Chrome 58, calling throws an exception.

This change brings Chrome in line with recent spec changes. The legacy behavior is not supported in Edge or Safari, and it is being removed from Firefox.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Remove pre-standard ChaCha20-Poly1305 ciphers

In 2013, Chrome 31 deployed new TLS cipher suites based on Prof. Dan Bernstein's ChaCha20 and Poly1305 algorithms. These was later standardized, with small tweaks, at the IETF as RFC 7539 and RFC 7905. We shipped the standardized variant early in 2016 with Chrome 49. We are now removing the pre-standard variants.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Remove support for commonName matching in certificates

RFC 2818 describes two methods to match a domain name against a certificate: using the available names within the `subjectAlternativeName` extension, or, in the absence of a SAN extension, falling back to the `commonName`. The fallback to the `commonName` was deprecated in RFC 2818 (published in 2000), but support remains in a number of TLS clients, often incorrectly.

The use of the `subjectAlternativeName` fields leaves it unambiguous whether a certificate is expressing a binding to an IP address or a domain name, and is fully defined in terms of its interaction with Name Constraints. However, the `commonName` is ambiguous, and because of this, support for it has been a source of security bugs in Chrome, the libraries it uses, and within the TLS ecosystem at large.

The compatibility risk for removing `commonName` is low. RFC 2818 has deprecated this for nearly two decades, and the baseline requirements (which all publicly trusted certificate authorities must abide by) has required the presence of a `subjectAltName` since 2012. Firefox already requires the `subjectAltName` for any newly issued publicly trusted certificates since Firefox 48.

Note: Enterprises that need to support such certificates for internal purposes may set the `EnableCommonNameFallbackForLocalAnchors` Enterprise policy.

Intent to Remove | Chromestatus Tracker | Chromium Bug

VTTRegion-related bits of TextTrack

The interface elements `regions`, `addRegion()` and `removeRegion()`, have been removed from the WebVTT spec and are removed in Chrome 58 to comply with the latest spec. We expect little impact from this removal since the feature was never enabled by default (meaning it was behind a flag). Those needing an alternative can use the `VTTCue.region` property which is being added in Chrome 58.

Chromestatus Tracker | Chromium Bug

WebAudio: remove AudioSourceNode interface

The `AudioSourceNode` interface is not part of the Web Audio specification, is not constructible, and has no attributes so it basically has no developer- accessible functionality. Therefore it is being removed.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Remove webkitdropzone global attribute

The `dropzone` global attribute was introduced by the HTML5 drag and drop specification as a declarative method for specifying an HTML element's willingness to be the target of a drag-and-drop operation, the content types that can be dropped onto the element, and the drag-and-drop operation (copy/move/link).

The attribute failed to gain traction among browser vendors. Blink and WebKit only implement a prefixed form of the attribute, `webkitdropzone`. Because the `dropzone` attribute was removed from the spec in early March 2017 the prefixed version is being removed from Chrome.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Deprecate insecure usage of notifications

Notifications are a powerful feature as they allow websites to invoke a system UI to transmit either private information itself or a signal that private information has been changed. Attackers may sniff or steal any information sent through a notification over an insecure connection. Web push requires a secure origin, so this change will align non-push notifications with push notifications. This change is part of our broader effort to remove powerful features from unsecure origins.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Deprecate usage of notifications from insecure iframes

Permission requests from iframes can confuse users since it is difficult to distinguish between the containing page's origin and the origin of the iframe that is making the request. When the requests scope is unclear, it is difficult for users to judge whether to grant or deny permission.

Disallowing notifications in iframes will also align the requirements for notification permission with that of push notifications, easing friction for developers.

Developers who need this functionality can open a new window to request notification permission.

Removal is in Chrome 62.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Remove indexedDB.webkitGetDatabaseNames()

We added this feature when Indexed DB was relatively new in Chrome and prefixing was all the rage. The API asynchronously returns a list of existing database names in an origin, which seemed sensible enough.

Unfortunately, the design is flawed, in that the results may be obsolete as soon as they are returned, so it can really only be used for logging, not serious application logic. The github issue tracks/links to previous discussion on alternatives, which would require a different approach. While there's been on-and-off interest by developers, given the lack of cross-browser progress the problem has been worked around by library authors.

Developers needing this functionality need to develop their own solution. Libraries like Dexie.js for example use a global table which is itself another database to track the names of databases.

This feature is removed in Chrome 60.

Intent to Deprecate | Chromestatus Tracker | Chromium Bug

Deprecation policy

To keep the platform healthy, we sometimes remove APIs from the Web Platform which have run their course. There can be many reasons why we would remove an API, such as:

- They are superseded by newer APIs.
- They are updated to reflect changes to specifications to bring alignment and consistency with other browsers.
- They are early experiments that never came to fruition in other browsers and thus can increase the burden of support for web developers.

Some of these changes will have an effect on a very small number of sites. To mitigate issues ahead of time, we try to give developers advanced notice so they can make the required changes to keep their sites running.

Chrome currently has a process for deprecations and removals of API's, essentially:

- Announce on the blink-dev mailing list.
- Set warnings and give time scales in the Chrome DevTools Console when usage is detected on the page.
- Wait, monitor, and then remove the feature as usage drops.

You can find a list of all deprecated features on chromestatus.com using the deprecated filter and removed features by applying the removed filter. We will also try to summarize some of the changes, reasoning, and migration paths in these posts.

Last updated July 2, 2018.