chrome.app.window alwaysOnTop property - Chrome API Proposal

Proposal Date
2013-10-03
Edited 2013-12-06

Who are the primary Eng and PM contacts for this API? (email addresses)
Eng: tmdiep@chromium.org
PM: saroop@chromium.org, kmess@chromium.org

Which team will be responsible for this API? (team email address)
chrome-apps-eng-syd@google.com

Do you know anyone else, internal or external, that is also interested in this API?
See: http://crbug.com/171597

What's a reasonable Chrome milestone target for landing most of the code on Trunk?
M32 with first-party apps whitelisted
Mitigation of security issues targeted for M33

Overview
Apps may want to have a window which stays on top of other windows. The chrome.app.window API will be extended to include:

- A new `alwaysOnTop` (boolean) option for the chrome.app.window.create() function. Defaults to false.
- An `AppWindow.isAlwaysOnTop()` function to query the state of this property.
- An `AppWindow.setAlwaysOnTop(boolean)` function to change this property after creation of the window.

Use cases

- Media players in "minimized mode" that sit on top of other windows.
- Chat/video chat notifications and popups that let users quickly go back to the "video chat window" if they clicked on other windows.

How would you implement your desired features if this API didn't exist?
Not possible.

Can these use cases be addressed by leveraging the standard web platform?

No.

If not, is it something that could/should be part of the web platform?
Probably not.

If not, could these use cases be addressed by extending the functionality of an existing chrome.* API?
This is a proposal for extending the existing chrome.app.window API.

Does this API expose any functionality to the web?
No.

Do you expect this API to be fairly stable? How might it be extended or changed in the future?
Yes. It is consistent with existing properties of app windows, such as isMinimized, isMaximized, isFullscreen, etc.

If multiple apps/extensions used this API at the same time, could they conflict with each other? If so, how do you propose to mitigate this problem?
No. The property is specific to an app window.

List every UI surface belonging to or potentially affected by your API:
This change affects the behavior of Chrome app windows.

Actions taken with app/extension APIs should be obviously attributable to an app/extension. Will users be able to tell when this new API is being used? How? Can it be spoofed?
The app window will stay above other windows in the desktop. There may or may not be a visual indication (see security UI or other mitigations).

Does this API impose any requirements on the Chrome Web Store ?
No.

Does this API have any interaction with other Chrome APIs ? Does it impose any restrictions on other APIs that can be used in conjunction ?
No.

How could this API be abused?
If app developers put their windows on top unnecessarily. However, this option has been available to desktop applications on many operating systems and there is no widespread abuse. Applications which have windows that stay on top generally provide an option to disable this behavior.

Imagine you're Dr. Evil Extension Writer, list the three worst evil deeds you could commit with your API (if you've got good ones, feel free to add more):

1. Create an always on top window which fills the screen and cannot be resized due to minimum size constraints. It could be a problem if combined with a transparent frameless window, but this has been mitigated in the Shaped and Translucent Windows API proposal.
2. Some users find windows which stay on top a nuisance and want the option to switch it off.

What security UI or other mitigations do you propose to limit evilness made possible by this new API?

Some options:

- Chrome could provide a way for the user to demote a specific window from always being on top if it is a nuisance, such as an item in the window's context menu or a "pin" button on the window title bar.
- The ability to disable this feature for a specific app.
- Use permissions and display warnings on app install.
- A Chrome setting to disable this feature globally.
- etc...

[Edit 2014-01-23]

Main functionality landed in M32, but was restricted to Dev channel only and whitelisted for some first-party apps.

After a security review, the following mitigations have been implemented and complete by M34:

- An app needs to declare the "alwaysOnTopWindows" permission in its manifest. This permission will not appear as a warning in the install prompt, but will enable us to track potential abusers of this API.
- Fullscreen app windows cannot have always-on-top enabled. The always-on-top property will be temporarily switched off while the app window is in fullscreen mode. This allows the user to still use Alt+Tab in order to activate another window.
- Always-on-top windows cannot cover the Windows taskbar. This was achieved by temporarily switching off the always-on-top property when the app window intersected (even in part) with the Windows taskbar. On Mac, Linux and ChromeOS, always-on-top windows are always ordered below the dock/shelf, etc.

Could a consumer of your API cause any permanent change to the user's system using your API that would not be reversed when that consumer is removed from the system?

No.

Draft Manifest Changes

An app needs to declare the “alwaysOnTopWindows” permission in its manifest in order to enable always-on-top app windows.

Draft API spec
Patch:
https://codereview.chromium.org/25745006

API Doc:
https://chrome-apps-doc.appspot.com/_patch/25745006/apps/app_window.html

Open questions
1. Which operating systems support the always on top property?
This feature has been prototyped and confirmed to work on the following so far:

- Windows 7
- ChromeOS
- Linux (GTK)
- Mac OS X 10
    - Mac allows several window levels. We will use the floating window level.

2. Is an onAlwaysOnTopChanged notification necessary?
The always on top property would change as a result of a call to chrome.app.window.setAlwaysOnTop() from the app itself. Unless the change can originate from Chrome (see security UI or other mitigations), I don’t see a need for the notification.