



Window Management

W3C TPAC 2023 Update (Sept 2023) - Second Screen WG

Brad Triebwasser - btrieb@google.com

Mike Wasserman - msw@google.com

Spec: github.com/w3c/window-management

Demos: github.com/michaelwasserman/window-placement-demo

Prior: [Sep 2022 Slides](#) | [May 2022 vF2F](#) | [Oct 2021 Slides](#)



Creating Fullscreen Popup Windows

[Explainer](#), [ChromeStatus](#)

Problem: Cumbersome user experience to show fullscreen content on a specific display

Goal: Permit opening a new fullscreen window on a specific display with one gesture

Proposal: `window.open()` `fullscreen` feature, gated on window-management permission

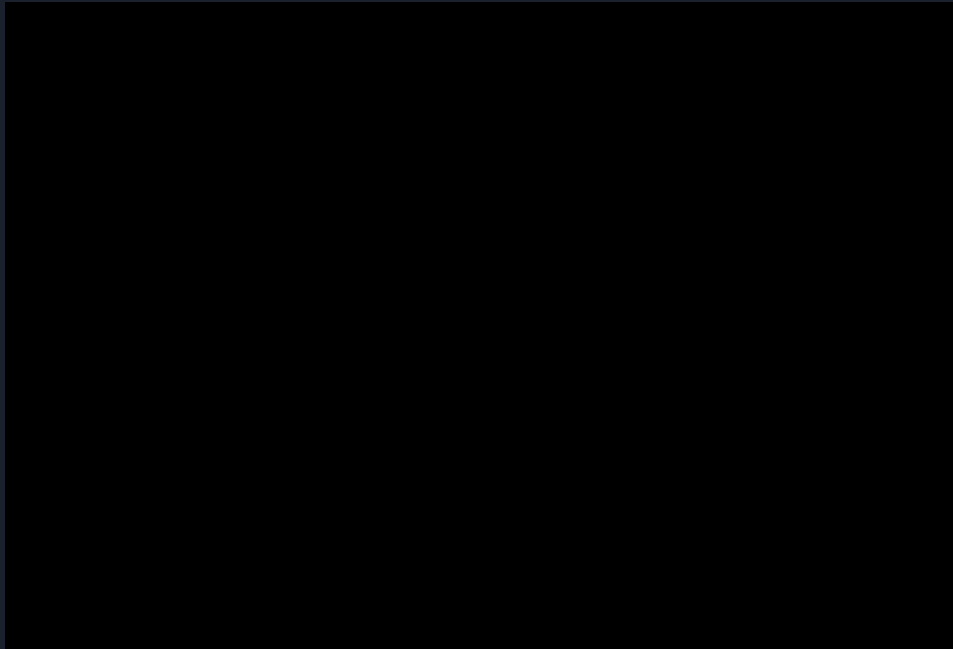


```
window.open(url, '_blank', 'popup,fullscreen');
```



Creating Fullscreen Popup Windows

[Explainer](#), [ChromeStatus](#)



Implemented in Chrome (OT
Planned M119)

Next Steps:

Addressing [TAG feedback](#), seeking
Second Screen CG/WG [feedback](#)
and incubation.

Standards Position Requests
([WebKit](#), [Mozilla](#))

[Demo](#) (Enable `chrome://flags/#fullscreen-popup-windows` in Chrome)



Many Windows, Many Screens

Explainer

Problem: Cumbersome user experience and limited flexibility to [initiate multi-screen experiences](#)

Goal: Enable script to show content on all screens from one user gesture:

- Open a popup window on each screen, potentially a new [fullscreen popup](#)
- Make an existing window fullscreen on each screen (via [Capability Delegation](#))
- Some mix thereof, e.g. one fullscreen and two popups, given three screens

Proposal: Enhance [User Activation](#), gated on window-management permission

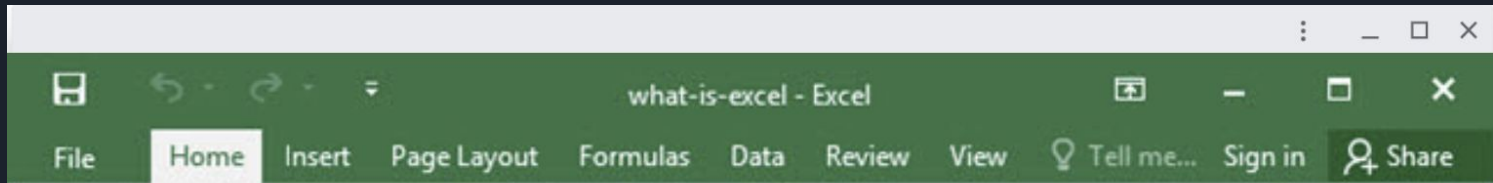
- Transient user activation can permit N=Screens window-management consumptions
- More flexible and canonical than [Fullscreen Companion Window](#) internal slots

Design is in early stages! Feedback welcome: msh@chromium.org, road to incubation?

Additional Windowing Controls

[Explainer](#), [ChromeStatus](#)

Problem: VDI web clients cannot integrate remote app windows with local desktops



Goal: Enable new capabilities, gated on **window-management** permission

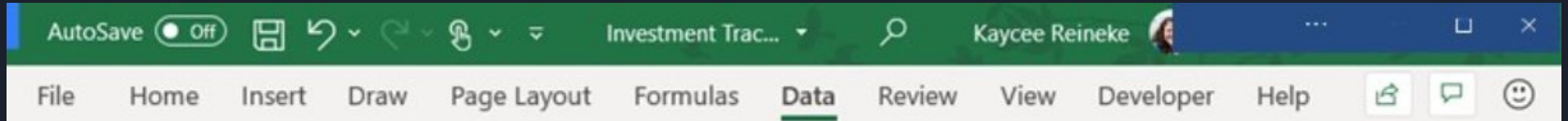
- **await window.minimize()/maximize()/restore()** changes the window display state
- **display-state** CSS media feature yields the current window display state
- **window.onmove** event is fired when the window moves
- **await window.setResizable(bool)** sets whether the window is user resizable
- **resizable** CSS media feature yields whether the window is user resizable

Addressing [Webkit feedback](#) (e.g. gate by display-mode, etc.). [More feedback](#) is welcome!

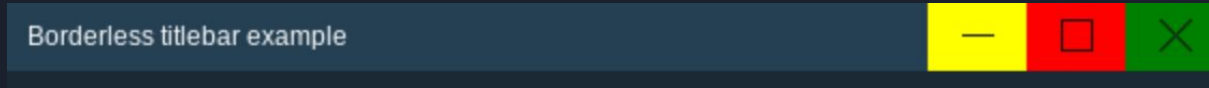
Borderless mode

[Explainer](#), [ChromeStatus](#)

Problem: Web app title bar customization is insufficient (even with [window-controls-overlay](#))



Goal: Enable full control over the appearance of the title bar area



Proposal: Web App Manifest WICG incubation: { "[display_override](#)": ["borderless"] }

Only available to [Isolated Web Apps](#), and gated on **window-management** permission.

Addressing [TAG feedback](#) and [more feedback](#) is welcome!



Virtual Display Testing

[Chromium test framework support for managing virtual displays at the OS level](#)

Functional automated coverage on ChromeOS and macOS

Progress towards adding support for Windows OS, driver complexity

Exploratory work for Linux

Apple Feedback “Virtual display support for automated testing” (CoreGraphics Suggestion):

<https://feedbackassistant.apple.com/feedback/12349099>



Thanks!



Window Management

W3C TPAC 2023 Update (Sept 2023) - Second Screen WG

Brad Triebwasser - btrieb@google.com

Mike Wasserman - msw@google.com

Spec: github.com/w3c/window-management

Demos: github.com/michaelwasserman/window-placement-demo

Prior: [Sep 2022 Slides](#) | [May 2022 vF2F](#) | [Oct 2021 Slides](#)