Google/MSFT PWA Sync

Google/MSFT PWA Sync

This document is public.

Doc directory: Web App Link Capturing Links

PWA team stuff we're working on:

- `https://issues.chromium.org/issues?q=status:accepted%20c…`

Dec 11, 2025 | Chrome:Edge PWA sync

Attendees: dmurph@google.com kristinlee@microsoft.com Lia Hiscock rob.paveza@microsoft.com scheib@google.com ishikapatel@microsoft.com limzh@microsoft.com lu.huang@microsoft.com

Notes

- **Group policy**
  - Sync toggle never turning "back on" (until browser restart) – by design
  - Where to save the web app system state / reconcile it with the browser pref (aka policy) state -> currently done in IsWebAppInstallByUserPolicyEnabled
  -
- **Web Install**
  - Proposed order of things:
    - Installurl version (v1) - continuing dev work and moving towards the goal of OT.
      - Feature bug:: [<install> Element] Enable Declarative installation of Web Apps from Web Content [454827186] - Chromium
    - If we get a chance to ship installurl, whether manifesturl is ready or not, our plan is to go for it.
    - Continue to iterate on manifesturl design, but we don't want it to stall shipping an element version.
      - This is in consideration of the current shortcomings and extra dev work of the manifesturl: Specifying manifests vs documents · Issue #5 · WICG/install-element
  - Manifest id clarifications - Current plan is to use the same model as navigator.install. Let us know if there's any concerns.
    - `<install></install>` - Require id field defined in manifest file.
    - `<install installurl=""></install>` - Require id field defined in manifest file.
    - `<install installurl="" manifestid=""></install>` - id field not required in manifest, but provided id must match computed id.

    - FYI - PEPC currently restricts 3 elements allowed per page (currently looking into)(Required for OT) - [<install> element] Increase amount of elements allowed per page [467179136] - Chromium
    - Re-confirming we're not supporting iframes for now.
    - Check in on Google plans for install dialog redesign? Last we heard, there were plans for early 2026?

Action items

- □

Nov 13, 2025 | 📅 Chrome:Edge PWA sync

Attendees: dmurph@google.com Kristin Lee liahiscock@microsoft.com rob.paveza@microsoft.com scheib@google.com ishikapatel@microsoft.com limzh@microsoft.com lu.huang@microsoft.com

Notes

- Web Install
    -
- Follow-up from email thread on On Linux, a PWA icon gloms with the browser icon during installation [40796270] - Chromium
    - Edge PWAs no longer use separate window class - all group under main Edge icon (Wayland / GNOME) · Issue #706 · flathub/com.microsoft.Edge

Action items

- □

Oct 9, 2025 | 📅 Chrome:Edge PWA sync

Attendees: dmurph@google.com kristinlee@microsoft.com Lia Hiscock luigonza@microsoft.com scheib@google.com lingz@microsoft.com ishikapatel@microsoft.com limzh@microsoft.com lu.huang@microsoft.com zhong.ling@microsoft.com

Notes

Group Policy - Enterprises can disable the ability to install web apps via group policy [430332936] - Chromium

- Haven't heard from Mirco in 6 weeks – are there any alternative enterprise contacts we could lean on?
- Implementation Q – how should sync apps be handled?

    - Option 1: Follow the same outcomes as the SyncTypesListDisabled policy for apps, only preventing further synced apps.
    - Option 2: Modify implementation to hide all sync apps
    - Dan - if the sync types policy works already, seems fine to leverage it
        - Please make sure the right click -> install pipeline respects the AreWebAppsUserInstallable - we can hopefully grey it out if that isn't the case, and maybe fail the InstallAppLocallyCommand? InstallAppFromSyncCommand?
        - so from product perspective - probably have them still listed in chrome apps but grey out the 'install' right click option?
            - or clicking on that can show a dialog to say 'blocked by admin'?
        - chrome://apps page uses app_home.mojom
        - InstallAppLocally here could have an if statement for AreWebAppsUserInstallable, and if not, show a dialog on the current web contents, like the one for install API.
            - probably easiest
            - could also pass the web contents weakptr to the command and do it from there.... likely 'better' solution but a bit more work / changes. (would have to make that dialog shown via method in the WebAppUiManager)
- “Open in window” - Removing this would impact pre-installed app installability. Looking for product input on what is best for Chrome here
    - Our new policy does not change this behavior
    - TODO: update Enterprise Doc DONE
- SubApps CHECK(IsUserInstallable)
    - Disable sub apps altogether if possible
    - TODO: update Enterprise Doc to reference this case DONE

Web Install -

- Gating install behind installation update.
    - Ling: TPAC is coming up. We don't have plans to talk about Install API at TPAC.
    - Diego is planning on chatting about this in webapps working group
    - Mike said if we can come with PEPC explainer ready then that could get support there..... or if not done, can work on it together there.
    - Dan: no strong opinion on having permission prompt or not.
- UKMs
    - We already have the privacy reviewer’s approval. To be sure, do you know of any other approval/review we should be aware of? URL-Keyed Metrics API mentions “Before you add new metrics, you should file a proposal”
    - Monitoring UKMs - We'll be able to monitor the UKMs for Edge, but we don't have access to go/ukm . Is it possible to ask your team for help on monitoring the UKMs in go/ukm for the Chrome side?

            - We have access to chrome://ukm with the right flags, but unable to see if there were multiple hits of the same enum for the same source ID. I'm assuming this is by design?
                - Note: best way to check might be through tests. Manual test may prove difficult.
            - chrome://ukm - help with pointing us to who can help with getting App Consent enabled as a non-Googler (already tried a mini installer and using Loopback server to enable app sync) - **this is needed to see UKMs with SourceId type AppId in chrome://ukm**
                - Make sure to use start_url
                - Make sure to write tests too.
            - Note: For now, email Dan for monitoring help. Have specific questions.
            - source-id
- Notice - will be sending the CL for adding installed_by soon. Afterwards, the UKM WIP CL with questions for an early review.
    - Installed_by
        - Store a list for "invoked_by"?(maybe top 10) vs optional GURL for SB data
            - If already installed, app is just launched

Domain migration

- Microsoft is considering adopting the upstream proposal and potentially special case for 1P apps.
    - Upstream targeting ~144/145

Upstream updates

- Install UX - this quarter
- Experiments around IPH
- SB - 2026

Action items

☑ ~~Ask Chrome MEtrics Team about how MSFT can create a UKM, all docs are private etc.~~

□ kristinlee@microsoft.com - send list of metrics questions to Dan

Aug 14, 2025 | Chrome:Edge PWA sync

Attendees: Dan Murphy Kristin Lee Lia Hiscock Rob Paveza Vincent Scheib lingz@microsoft.com Ishika Patel Limin Zhu (Edge) Lu Huang zhong.ling@microsoft.com

Group Policy - [Enterprises can disable the ability to install web apps via group policy [430332936] - Chromium]()

- What's the general google process for policy work? Enterprise PM reviews, then…

        - Work with Ajay for getting the policy/documentation published? https://chromeenterprise.google/policies
            - Hope is that you would work directly with Chrome Enterprise folks / CrOS Enterprise folks like Mirco Franz to support getting this feature in the chrome documentation
                - IFF they are ok with this and Vince & I are ok with it.
                - otherwise, would have to be not enabled in Chrome at all (so documenting doesn't make sense)
- Are there any platform requirements? ie. ChromeOS, Android/mobile?
- Mirco Franz review task: https://crbug.com/438231238

Domain migration

- Microsoft M365 apps are very interested in [bit.ly/pwa-migration-prd (for external sharing) - Google Docs](). They are migrating from current domains to *.cloud.microsoft ([Unified cloud.microsoft domain for Microsoft 365 apps - Microsoft 365 Enterprise | Microsoft Learn]()).
- Given these are beyond just the eTLD+1 in the proposal, it'd be interesting to discuss path forward for cross-site. One idea is to have both old & new site include a .json file under ./well-known to indicate transition following similar approach from scope extension and related website sets.
- Marijn Kruisselbrink is owning - please contact him if you have any questions
    - explainer is going to happen within the next 3 weeks or so? for public review
- MSFT is interested. scope extensions stuff etc.
- interested in supporting cross-site scenarios as well
    - doc is only etld + 1.
    - Dan: Needs to solve how to not have malicious behavior / malicious takeover.
    - how do users understand going from origin to origin? www.foo.com -> game1.foo.com seems good & easy for user to understand.

Windows application identity

- Windows is anticipating requiring identity for "privileged" operations to be compliant with regulatory requirements ("identity" here being related to the identity of the application, not the user interacting with the application)
- Edge team has a basic implementation and is exploring how best to bring this forward for the Chromium project
- stahon@microsoft.com is looking into this.
- We have a thunking implementation. Windows - for compatibility purposes, as long as you have differente exes, one executable can have identity as part of the package. We can create process on executable, it can have identity, and that can be part of a PWA identity or part of browser identity. That will, in turn, loop in msedge.dll (or chrome.dll) in the same way that chrome.exe starts chrome.dll.
    - Any operations that require identity are brokered through mojo call, but the utility process that has identity would be making the function calls on the OS

  - might not be the best solution possible. but a lot of compatibility problems and concerns about risk
    - group policy can prevent packages from being explicitly updated, except by explicit operations
      - e.g. edge or chrome updaters cannot update certain things, so identity would break
      - So - this lets us isolate operations that can be broken.
    - Windows is well aware with our concerns & have been good about workingn with us.
- chrome windows team:
  - davidbienvenu@google.com, jessemckenna@google.com, mmentovai@google.com, robliao@google.com
  - please revisit with these folks, meeting is great if possible - to re-page this stuff into their brain and give the ultra background view here first.
  - you can add dmurph@ go meetings to help smooth communications.
-

Web install

- Status update – All planned work required for OT have landed.
- Security review follow up
  - What we think is the main point from Mike: want to avoid increasing user dissatisfaction.
  - Are there other APIs that have hit similar issues regarding SB and PEPC? How did they integrate with these Chrome's services?
    - Is this a good time to talk to the SB team - how do we go about that? Or need more discussion?
    - Suggestion
      - Put investigative hat on - what would it look like with PEPC integration? Mike said he can connect us to PEPC folks.
      - Talking to SB can help us understand what’s available for the API.
  - Is the "user success" for BIP being tracked? ie. more prompts = more installs, but are there more uninstalls too?
    - Dan: we might have the data. Pretty sure we have a ‘last install source’. Might be a UKM https://source.chromium.org/search?q=latest_install_source
- Align on next steps - What can we concretely do today to move the security review along?
  - Updating the Explainer is still in the works (Diego).
  - Look into PEPC and SB - get a better understanding of the benefit and scope.
  - Look into geolocation PEPC: PEPC/geolocation_explainer.md at main · WICG/PEPC
    - Bring up to PEPC about the worry of if PEPC is going to be shipped or not.
      - Wonder if it can be a thing turned on under OT

    - Even if we feel it may not be promising, it’s still worth talking to SB folks.
    - Dan/Mike will sync next week

- Rob: is it possible to lower the scope to minimize risk?
    - Not sure
- Can ask Mike questions and hypotheticals

Jul 10, 2025 | 📅 Chrome:Edge PWA sync

Attendees: dmurph@google.com kristinlee@microsoft.com Lia Hiscock rob.paveza@microsoft.com lingz@microsoft.com lu.huang@microsoft.com ishikapatel@microsoft.com

Notes

Web Install

- Update on Dev Trials status - Ready for Developer Testing: Web Install API
    - Are there any partners on your side that should be notified?
- Permissions model -- we're proceeding with prompting for background document installs, even if they're same origin.
    - We'll proceed with this for OT and shipping unless we receive strong feedback for otherwise.
    - We’ll be replying to the pwa-dev thread to close it.
- OT estimate - aiming for 141 but might be 142
- Our top items for pre-OT, but not everything
    - Histogram
    - ChromeOs
    - Handling the install self case
    - Incognito/guest mode dialog - We were referred to Erin Ly for Design. Waiting for a response.
        - Let Ajay know if blocked.

FYI about a icon logic change coming soon - one only icon will be used, the one the user approved.

- for translations - this will affect that / maybe make it easier.

Group Policy - Enterprises can disable the ability to install web apps via group policy [430332936] - Chromium

- Feel free to drop any questions or concerns
- Lia – add more context to the bug re. user research concerns
- Starting design work in this upcoming cycle

- Predictable Web App Updating
  - The doc is a bit ambiguous on any potential plans on supporting updates of apps that are migrating origins. Are there any thoughts/plans around this?
- PWA Migration
  - https://issues.chromium.org/380486617 PWA App Migration
    - https://bit.ly/pwa-migration-prd
    - There may be usage on the MSFT side?
- getInstalledRelatedApps - documentation

FYI about use counter and webdxfeature - make sure you implement these for your features. https://chromium.googlesource.com/chromium/src/+/HEAD/docs/use_counter_wiki.md

Action items

- ☑ ~~Talk to Vince about the enterprise policyet~~
- □ Reach out to msft partners about people who might want migration? - Lu
- ☑ ~~Write down enterprise policy use cases & reasoning in bug - Lia~~

Jun 12, 2025 | Chrome:Edge PWA sync

Attendees: Dan Murphy Kristin Lee Lia Hiscock Rob Paveza Vincent Scheib Lu Huang

Notes

- WebInstall
  - Any concerns with suppressing install permission requirement for background document installation of the current doc? [Web Install API] Handle the install(self) and install(self, self) cases [407473727] - Chromium
    - do we need to show the permission for anything that is same-origin?
      - example: the dev can switch the 'manifest link' in the page to anything that links to an app on the same origin
      - then they can call navigator.install() and it'll install that app.
    - what is the UX flow?
      - background-install-full.gif
    - [Web Install API] Handle the install(self) and install(self, self) cases [407473727] - Chromium
  - Discussion Decisions
    - Check for same origin (not exact URL)
    - skip permission dialog if same origin
    - the API should ALWAYS use the background install if a URL is given crbug.com/407473727

    - Agreed we're good to move forward with current design for OT, but changes should be applied before shipping.
  - New - [Web Install] duplicates BIP · Issue #1055 · MicrosoftEdge/MSEdgeExplainers
    - Dan convinced Marcos about requirement for id in manifest
    - Marcos: Can we have a CanInstall() api (0 param only) so developers can check before inviting a user action.
    - Marcos still thinks 0-param install is the same as BIP. Ongoing issue.
      - Option 1: yes that’s fair feedback but we still want to move forward with 0-param install because…
        - pro: Get more people setting the 'id' field
        - pro: devs say they want it (imperative api). More usable.
        - con: could be called before the manifest link is on the document
          - but seems SUPER unlikely as this is usually statically put there, and the logic can wait for it to download etc.
          - but manifest can fail to download - offline? idk
            - devs should specify an onload handler on that link?
              - then we're basically doing BIP without the protection of the install method being on the event itself
        - pro: transient activation is easier here - just call this in the event.
      - Option 2: let’s not ship 0-param install, in favor of BIP.
  - New - ChromeOS/Android https://issues.chromium.org/issues/333795265#comment17
    - In the past we did this by tracking milestones. It's ok to follow up after and do milestones.
    - Android would be a different milestone. have to ship both the no-param and param versions at the same time there.
    - ChromeOS - MSFT will check to see what was happening with the permissions on CrOS.
      - https://chromium.googlesource.com/chromium/src/+/main/docs/chromeos_build_instructions.md
  - New - incognito UX
    - planned before shipping?
    - likely will involve UX and UX writers
      - both have office hours etc
      - Ajay can organize.
- Scope extensions ship version - M138/M139?
  - Chromium Dash
  - m139 is the guidance, let's maximize eng output and not spend time on this.

    - Chromestatus entries are the source material that is fed into lots of places. Good to think about it.

Action items

- □

May 8, 2025 | 📅 Scope Extensions Chrome:Edge sync

Attendees: Howard Wolosky Dan Murphy Lu Huang

Notes

- PWA Migration
    - https://issues.chromium.org/380486617 PWA App Migration
    - Loop is interested!
        - loop.microsoft.com
        - .microsoft tld
- Scope extensions
    - SING replied and opened an issue
    - Chromium launch security review out-of-SLO
        - blocking I2S
        - Follow up with email to check on status.
    - spec url parsing
        - example: https://www.w3.org/TR/appmanifest/#ref-for-index-term-url-parser-3
    - fetch?
        - you can use algorithm: https://www.w3.org/TR/appmanifest/#ref-for-index-term-fetch-1
            - but it has a lot of arguments
            - I believe the current implementation we have CAN be specified by calling that function and only saying 'with a request with method of 'GET' and url of ... etc.
- WebInstall
    - Any concerns with suppressing install permission requirement for background document installation of the current doc?
        - reply from dan - Dan noticed a side-effect here for a feature shipping on Chrome - navigation capturing - and we need to make sure the webappinstall code doesn't change that other feature's behavior.
    - UX review for the incognito / guest / kiosk profile dialog
        - will ping ajay / yuchang / dan
- Predictable Web App Updating

  - The doc is a bit ambiguous on any potential plans on supporting updates of apps that are migrating origins. Are there any thoughts/plans around this?
    - see https://issues.chromium.org/380486617 PWA App Migration

Action items

- □

Apr 10, 2025

- Blinkon
  - good chats with May
- scope extensions
  - should we ignore manifest spec? Or add extention point / compatibility?
  - if ignore
    - then the manifest-incubations spec should cite the places of modification
      - in the "is url in scope" algorithm, change ___. do these steps first.... etc
  - if we add extension point
    - in the algorithm, add a step for an extension point -
    - do some research to see if other specs have an 'incubation' extension point.
  - **we could define 'in extended scope', and then say that all checks for 'in scope' turn into 'in extended scope'**
    - UA has a choice of when to use
    - **When the user agent is checking "is url in scope", then it MAY also consider "is url in extended scope".**
  - Kicking off chromestatus review buttons
    - Good to make sure all attached artifacts are up-to-date when emails are sent.
  - Are we changing "value" to "origin"?
    - if we did do that change, then we need to update explainer: https://github.com/WICG/manifest-incubations/blob/gh-pages/scope_extensions-explainer.md
    - task: https://issues.chromium.org/409818516
- webinstall
  -

- [*luhua@microsoft.com*] **Ability for PWAs to extend their navigation scope to associated origins** (bug)
  - [*luhua@microsoft.com*] **Ship Scope Extensions v1 (origin extensions only), on Windows/Mac/Linux/ChromeOS Classic** (bug)
- [*liahiscock@microsoft.com*] **[Web Install API] Enable Installation of Web Apps from Web Content** (bug)

- Focused on changes to get install confirmation prompt working for background doc installations
    - Post to pwa-dev out in next day or two
- The place that is weird right now .... the place that the aPI need to poke at is the logic aroudn
    - "Is there an installation dialog / installation process happening right now for this browser/tab"
    - this MIGHT be implemented in a couple places - dialog coordinator? But the main place this funcionatlity is working is the ML Promotion Tracker (or something like that)
    - It would be good to.... refactor this a little to make it more deliberate
        - class CurrentInstallProcess : public WebContentUSerData - or BrowserUserData? IDK
        - inside of there we can put the ML tracker
        - IDK if we make this 'early' enough???
    - Naive possibility here is that the web install API just needs to check & make this on the initial JS call, on the browser side?
        - if the ml tracker isn't there, then create it and attach to current process (forward to command?)
        - if it IS there, then abort the install.
    - Add tests that call .install() multiple times (as well as interweaving beforeInstallPrompt()).
        - call prompt on the beforeinstallprompt event, and then install()
        - install() and then prompt on beforeinstallprompt event
        - (look at app_banner_manager browsertests as a baseline for how these might be done, but there is some terrible stuff going on in those tests that probably shouldn't be replicated)
        - not happy with this test: https://source.chromium.org/chromium/chromium/src/+/main:chrome/browser/banners/app_banner_manager_desktop_browsertest.cc
        - Dan would
            - make a page in chrome/test/data/webapps/bip_event/index.html
            - has a manifest / installable
            - in js, listen for the 'beforeinstallprompt' event, save into global variable
                - have a global promise that resolves when the above event is triggered, with the event.
                - `var beforeInstallPromptEvent = null;`
                - `var beforeInstallPromptTriggered = new Promise(...)`
            - `content::EvalJs('beforeInstallPromptTrigger`

`ed'); // good way to make native code wait for a JS promise to resolve`

- then you can call prompt on it?
  - `content::EvalJs('beforeInstallPromptEvent.prompt()");`
  - That should give you EXPECT_THAT(..., content::EvalJs::OK()); which will error, if, say, it's null.
  - take a look at ExecJs and EvalJs

- Android support
  - Treat as separate milestone post-desktop OT. This is a pretty typical practice.
  - WebAppProvider doesn’t exist on Android
    - Not sure how much background install support there is on Android
- incognito, guest, kiosk mode
  - originally - failing at random interval
  - ask from dmurph@ - dialog shown to the user.
- Chrome Apps deprecation might be prior-art for showing the dialog to the user when in incognito/guest/kiosk mode informing them it’s not supported.
  - Web Notifications: https://web-push-book.gauntface.com/demos/notification-examples/
  - Idle detection: https://idle-detection.glitch.me/
  - Howard to work with Ajay to set up a meeting with Dan, Yuchang and MS to discuss the idea of introducing this new dialog concept.
- permission strings?
  - Web Install UX review

- [*ishikapatel@microsoft.com*] **Support web apps ('webapp' platform) in getInstalledRelatedApps() on Desktop, allowing installation detection from a browser tab** (bug)
  -
- [*luhua@microsoft.com*] ♲ Strange empty header area area for PWA popup window (at about:blank?) (bug)
  - Targeting getting done in H1. *(Updated: 3/24/2025)***[old]**
- [*luhua@microsoft.com*] ♲ Web app launch from system does not work in automation tests (--enable-automation) (bug)
  - Planned for H1.
  - Partner is determining their interesting in being the public customer for the cdp protocol for automation testing, but still need this fix in the meantime. The 'long term' solution is using the cdp protocol, but that might not be applicable for all partners (and has risks).
  - Risk: Not highest priority. *(Updated: 3/24/2025)***[old]**

- PWA Subtitle
  - CRBug: Fix title bar text on installed PWAs [40857293] - Chromium
  - Chrome Status: Document Subtitle (Fix PWA app titles) - Chrome Platform Status

Feb 13, 2025
Attendees: Daniel Murphy Lu Huang howard.wolosky@microsoft.com
Notes:

- https://issues.chromium.org/40197834 Ability for PWAs to extend their navigation scope to associated origins
  - update status
  - V1: https://issues.chromium.org/issues/392657735 - This is the V1 bug.
  -
- Question about http://crbug.com/382108359 - set status?
- example: https://issues.chromium.org/issues/40144143
- --automation test infra bug (link)
  - Lu continue to own for Q1 but prioritized below scope_extensions and about:blank CCT bug
- about:blank CCT bug (link)
- Link capturing enabled-by-default soon in Chrome
  - Will also enable in Edge later after fixing bugs

Status updates (how to)

- Character Limit: Keep updates concise, as the field has a 1,000-character limit .
- Use Bullet Points: We recommend using bullet points (**3 bullets max** is recommended) to highlight key program status elements, such as:
  - **Updates**: Recent progress and accomplishments.
  - **Risks**: Current challenges and roadblocks.
  - **Escalations**: Risks requiring immediate attention from leadership
  - Please use the
    - “Me I didn't have time to write a short letter, so I wrote a long one instead.”
    - — Mark Twain
    - As a prompt for thinking critically about projects and writing status, ask the questions:
    - Where are we?
    - How did we get here?
    - Where do we need to go next?
    - Then come up with the plan to get there and express in the form of what will be done by whom and by when.
    - Here are practical tips for summarizing that plan for execs:
    - Do not bury the lead. Write what needs to be written directly and succinctly.
    - Remove jargon, acronyms, and code names so that a layperson could understand the update.
    - Put yourself in the shoes of a reader without context on your program and re-read. Does it still make sense?
    - “On track” is a valid status with no further need to elaborate if it’s true. Nothing to see here, move on.
    - For issues and risks, state the problem directly along with the next steps, their owners, and due dates.
    - Escalation requests need to be clear about what is being asked of whom, by when, along with desired outcome.
    - If the issue can be solved among the team leads, it’s ok to err on the side of listing it as an FYI with no action required

        - Less is more. Make 3 bullet points per section a goal. Links to details are encouraged since many of the leads like to dive all the way down to bug level detail.

Pending work - we look at the child bugs / tasks / etc

Nov 14, 2024

Attendees: Daniel Murphy Lu Huang howard.wolosky@microsoft.com

Notes:

- Lu / Howard / Dan BlinkOn meeting/s
    -
- Someone from MSFT could be an editor/owner of WICG github
    - Chris Wilson owns WICG stuff
    - another option is making a new WICG repo per-feature
        - https://github.com/WICG/web-app-launch/tree/main
    - ~~AI: howard.wolosky@microsoft.com to reply with who~~
    - Add Howard, Diego, Lu?
- crbug.com/295117058 - “Launch Scope Extensions” bug
    - Let’s use the Status Update and Description field here for publishing latest updates and plan to ship.
        - Partners can CC themselves and get update emails.
- web install: issues.chromium.org/issues/365555412
- AI(luhua): Scope Extensions plan
    - Part 1 - “origin” extensions
    - Part 2 - “site” extensions
    - Part 3 - Additional scope filtering syntax
    - Each part can link to a separate explainer
- Intent to ship (make Blink owners comfortable w/ shipping)
    - Get partner comment that this works
    - Explain that the changed design works (we’ve tested it, etc)
- ChromeOS Scope check failing
    - Find a better explanation
    - https://source.chromium.org/chromium/chromium/src/+/main:chrome/browser/ui/web_applications/web_app_browsertest_base.cc;l=250;drc=bc461b56c036169f465a5a04c0daff8ea32d4f80

Sep 12, 2024

Attendees: Daniel Murphy Lu Huang howard.wolosky@microsoft.com

Notes:

- Dan shared “Why is it like this? Installed Web Apps …”
    - Will share this more widely later.
- Scope Extensions
    - Early TAG review closed as “validated”.
    - Lu opened explainer PR with design changes (path filtering, response header, etc)
    - Lu will reopen TAG issue after explainer is updated.
- Lu wants to write an explainer on scope exclusions.
    - Primarily to define the concept that both main and extended scope exclusions can live in this one field at the top level of the manifest.
    - Scope extensions explainer can link here.
-

Aug 8, 2024
Attendees: Daniel Murphy Lu Huang
Notes:

- TAG status: https://github.com/w3ctag/design-reviews/issues/875#issuecomment-2272811044
    -
    - idea
        - filter syntax - behavior definitions all in manifest.json
        - first level security - the well-known file handshake
        - second level - headers on responses from origins to confirm apps that can can use it as extended scope
        - third level - cors things that Dan doesn't understand yet - hopefully never need.
- Notes - dmurph / estark / Lu - Scope Extensions & Security (tentative) - Google Docs
    -

Jul 11, 2024 | Scope Extensions Chrome:Edge sync

Attendees: Howard Wolosky Lu Huang Natasha Gaitonde Dan Murphy

Notes

- Feedback from Meet
    - would love an 'exclude' option
- proposal / idea from Dan
    - we allow an 'excludePathPrefix' option for 'origin' type. Can it be an array?
    - Then, we allow this to apply on top of the same-origin
- reason why we took this out:
    - we didn't want to rathole on "should it be url pattern, should it be paths, should it be X." etc. Wanted to avoid bikeshedding.

    - whenever we added this or talked about it - folks asked "why not Y or Z or UrlPattern".
- android
    - https://developer.android.com/guide/topics/manifest/data-element#path
    - We would have to ship w/o Android support for the exclude field
- how to make it less awkward?
    - have an include?
    - weird to have to specify the full origin if the scope field is just '/'
        - The proposal means that you are including your whole origin in your scope, but that's not what we want.
    - if you made inclusions also there
        - then you can have that match your scope, or maybe empty
    - if there is an explicit includepathprefix in the scope extension, and it is empty, then we are specifying this origin, but we are not adding anything. then we can have a new field that is exclude paths
    - other ideas
        - just a 'sameOriginPathExclusions' type?
        - type: originPaths
            - origin:
            - includes:
            - excludes:
        - type: includePrefix, excludePrefix
            - value: "https://example.com/hello/not/me/"
- conclusion
    - MAYBE we can create a proposal here that isn't crazy and allows modification of manifest scope using scope_extensions
    - **maybe we just delegate this to a future solution**
        - **scope_exclusions**
            - this happens after scope extension evaluation
    - we don't want exclusions to tie to origin, as that can be confusing because 'scope' in manifest doesn't have an origin
- We have the problem of manifest scope being set in stone - we can't change it. adding part b - extensions
- but we never had exclusions for part a - so we don't need separate exclusions - we can have the exclusions apply to both hopefully
- exclusions can apply to both
- This also solves objection - if I don't care about extensions, can I use this for manifest scope - yeah.
- conclusion again
    - scope_exclusions -> list of things to exclude. applies on 'evaluated scope', which includes manifest scope and extensions if they are there.

Action items

- ☐

Jun 6, 2024 | Scope Extensions Chrome:Edge sync

Attendees: Alan Cutter Howard Wolosky Natasha Gaitonde luhua@microsoft.com Lu Huang Dan Murphy

Notes

- Discuss open PRs to scope extensions explainer.
  - include? exclude?
  - we can say it's always OR, and use a url pattern?
    - can solve later
- partner instructions for testing?
  - [Web App Scope Extensions Origin Trial - Manual Test Instructions]
- FYI: chrome://flags/#enable-user-link-capturing-scope-extensions-pwa
- What is left?
  - add manifest parser support for new format. Registrable domain is already parse-able with blink helper function.
  - launch process?
    - still not any partner feedback
    - First party app testing in June at Google.
  - Draft spec - Diego is working on this.

Action items

- □

May 2, 2024 | Scope Extensions Chrome:Edge sync

Attendees: Alan Cutter luhua@microsoft.com Lu Huang Dan Murphy

Attendees: alancutter, dmurph, luhua

- Spoofing
  - Call out that we can re-parse the file from time to time, helps against broken / rough stuff.
- Write down that we are as specific as possible to not encourage bolting on to this.
  - Like first party sets - we don’t want that.
    - Discussion clarifying the difference between the two goals: https://github.com/WICG/first-party-sets/issues/145

    - Limit the power here
- Manifest entity concept seemed helpful
- Registrable domain vs. URLPattern for manifest entry
    - Present registrable domain as the better choice
    - URLPattern could confuse developers into thinking there are more degrees of freedom there than actually exists.
    - Algorithm: https://url.spec.whatwg.org/#host-registrable-domain

Options

- **Type-style (chosen)**
    - { type: “registrable_domain”, value: “https://www.example.com” }
        - // spec said take “value”, parse this as a url, take host and send to algorithm, get registrable domain
    - { type: “origin”, value: “https://www.example.com”}
    - Future: { type: “url_pattern”, <what should be here can be defined better>}
- Bag of stuff
    - { origin: “https://www.example.com”, treat_as_registerable: true}
    - { origin: “https://www.example.com” }
    - { url_pattern:, origin: } // what happens?
- Alternative: UrlPattern all the things
    - Don’t want to support probably
    - Would really only be ‘safe’ if we let them use all of url pattern
    - Or use css rules - use the first one that parses / ignore unparsable ones.

AI

- ☑ ~~Send email with info about plans, question about timeline, etc~~
- ☐ Update explainer with feedback from tag, adding new format example with links to spec stuff, possible spec-y thing there. When things look good send reply to tag review.

Apr 4, 2024

Attendees: alancutter, dmurph, luhua

- OT usage feedback
    - No OT renewal feedback
    - dmurph sent messages to app partners to see if they are in fact using the OT.
- Link capturing pushed back in schedule - M128 tentative
    - Need work to become consistent with existing navigation spec behavior
    - Design doc shared with Lu: https://docs.google.com/document/d/1ISz2ZnVc9Tlw367mrKM2Rnu7UbFAj2Na8lPB_Ag2430/edit?tab=t.0#heading=h.x34pvj24xq4n
- Site install and shortcut behavior in Chrome getting a refresh
- Updates on TAG review
    - https://github.com/w3ctag/design-reviews/issues/875#issuecomment-2024257259
- TAG Breakout call on 4/16 to discuss TAG review

Feb 29, 2024

Attendees: dmurph, alancutter, luhua

- In M122 which has gone stable.
- chromestatus.com stats: https://chromestatus.com/metrics/feature/timeline/popularity/4456
  - M122 released mid February, stats are every month, waiting for March 1 stats to see a rise in usage.
- No change in timeline to address remaining ship blockers (manifest parser, draft spec, etc.)
- Review scope_extensions usage next time we meet.

Feb 1, 2024

Attendees: alancutter, dmurph, luhua

- User link capturing, 1% on m121, but we found a number of bugs.
- Alan is leaving this project, handing off collaboration to dmurph@
- Testing with origin trial token
- Github issues: https://github.com/WICG/manifest-incubations/issues?q=is%3Aissue+is%3Aopen+label%3Ascope-extensions
- Issues blocking shipping after OT:
  - Need to change the wildcard syntax before shipping. Issue
  - Need to define a fetching model for association file. Issue
    - Can likely look at manifest spec for language on how icons are fetched
  - Need to write a spec draft, currently only an explainer.
- Dan wrote https://bit.ly/scope-extensions-and-third-party-auth to prevent using this for other stuff
- confident with m122 for?
- Future feature work:
  - Supporting URLPatterns.
    - We currently only scope to origin with room in the API available to add more fine grained scoping later.
- TAG review for web app `scope_extensions` · Issue #875 · w3ctag/design-reviews (github.com)
  - Concerns about web app ids; have reiterated that we are based on the origin security model.
  - App IDs are distinct because web origins are distinct
- Specification writing
  - It’s likely Diego will work on this.
  - Extension point on the manifest spec.
    - https://www.w3.org/TR/appmanifest/#dfn-processing-extension-point
  - Can live in manifest incubations.

        - launch_handler replaces an algorithm in the manifest spec for launching the web app.
            - https://wicg.github.io/web-app-launch/#launching-a-web-app-with-handling
            - Can do something similar for scope behaviour.
- Anticipated feedback
    - People will probably ask for scope_extensions to affect service worker scopes, user permissions, storage.

8 Dec 2023

Attendees: alancutter, luhua

- Launch
    - Final test pass TBD on beta M121.
    - Leadership final review after tests approved.
    - OT page is live: https://developer.chrome.com/origintrials/#/view_trial/3889984178141265921
    - Try to get token for sample app.
- Metrics
    - Will pick up for next M122.
    - Should be fine for OT, more important for shipping.
- Dev rel (Demian) handling comms for this OT launch.
    - Should explain how to force updating.
        - This should really be solved with DevTools UI.
        - Have DevTools run the update check code and have a button to force it to update if any changes found.
    - Dev feedback we're interested in?
        - Ergonomics
        - Setup
        - Does it solve the problems they have? What's missing in its capabilities? There may be other APIs that come out of this feedback.
        - Want to know if people need more specific pathing.
            - Won't ask directly, if they come to us with this need it's a much stronger signal.
        - File issues on GitHub.
- Path to shipping
    - Need spec written.
    - Need dev feedback.
    - Existing GitHub issues.
- Pause this meeting until M121 ships (~ February 2024).

Nov 9, 2023

Attendees: alancutter, luhua

- Testing doc

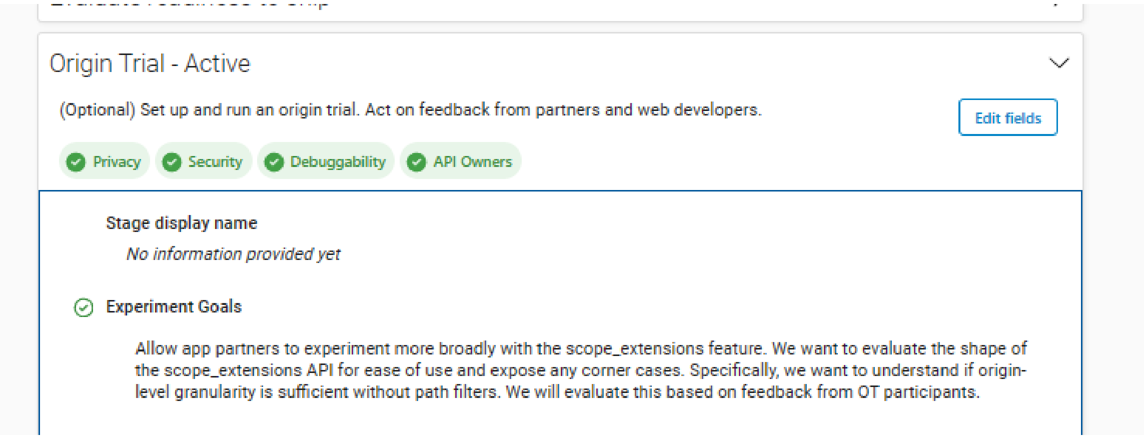
    - Web App Scope Extensions Origin Trial - Manual Test Instructions
    - Found issue in M120 with animation, should have tested with 121, updated min version in the doc.
    - Found same issue as accessibility with origin text animation into out of scope URL.
    - Need to just make that text not appear.
- Accessibility bugs filed - crbug link
    - Labels missing
        - https://bugs.chromium.org/p/chromium/issues/detail?id=1501081
        - https://bugs.chromium.org/p/chromium/issues/detail?id=1501000
    - Focus issue
        - https://bugs.chromium.org/p/chromium/issues/detail?id=1501094
    - Origin text too fast
        - https://bugs.chromium.org/p/chromium/issues/detail?id=1501046
        - Probably because the text is changing in the middle of the animation.
        - WebAppOriginText::StartFadeAnimation()
        - WebAppOriginText::DidFinishNavigation()
            - if (app_controller->ShouldShowCustomTabBar()) {
            -   **// TODO cancel the animation**
            -   return;
            - }
            - StartFadeAnimation();
            -
- label:Restrict-View-Google ?
    - cc'd Lu.
- Internal launch status:
    - Testing: in progress waiting for beta.
    - Accessibility: Found issues, blocking approval.
    - Leadership: Pending all other approvals granted before requesting.
        - Leadership approval is interested in metrics.
        - The basic usage metrics should be enough.
        - Adding the validation metrics will be better, stuff that helps us detect fishy numbers.
    - All others: Approved!
- Scope Extensions AND Isolation / Auth / Third Party Auth
- Github issues
    - Remove wildcard from scope_extensions, make a specialized entry · Issue #89 · WICG/manifest-incubations (github.com)
        - Good idea.
        - Helps our spec, avoids writing a bespoke URL parser.
        - Points us to concepts like registrable domain that we can use in spec.
    - Define the fetching model for scope extensions .well-known file · Issue #90 · WICG/manifest-incubations (github.com)

        - Alan: Big dislike towards enabling sites to .well-known any app that asks for it.
        - App has no control over this request, it's being done entirely by the user agent during install/updating.
        - Good to consider what the http-headers are in the request for spec fine grain detail.
        - Shouldn't block any launch.

Oct 26, 2023

Attendees: alancutter, luhua

- [Web app scope extensions - Chrome Platform Status (chromestatus.com)](https://chromestatus.com)

  Evaluate readiness to ship

  Origin Trial - Active

  (Optional) Set up and run an origin trial. Act on feedback from partners and web developers.

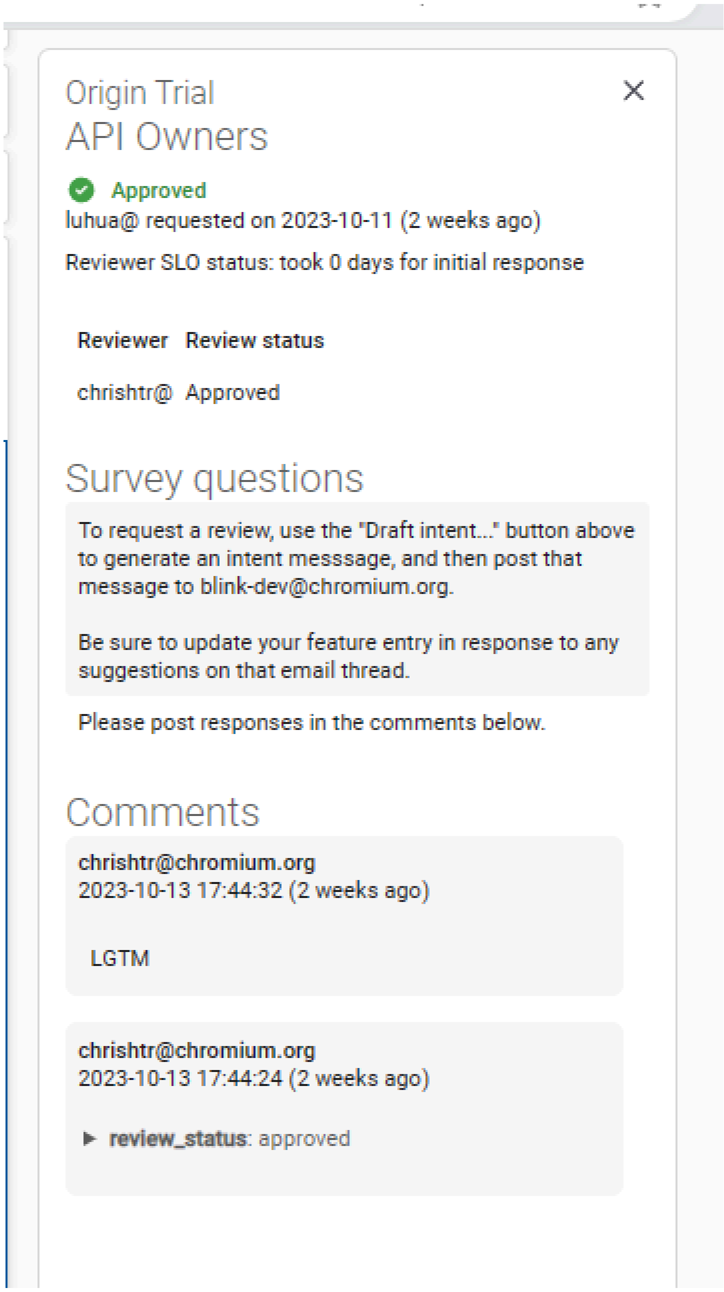
  Stage display name

  No information provided yet

  Experiment Goals

  Allow app partners to experiment more broadly with the scope_extensions feature. We want to evaluate the shape of the scope_extensions API for ease of use and expose any corner cases. Specifically, we want to understand if origin-level granularity is sufficient without path filters. We will evaluate this based on feedback from OT participants.

-

- 

Origin Trial

API Owners

✓ Approved

luhua@ requested on 2023-10-11 (2 weeks ago)

Reviewer SLO status: took 0 days for initial response

Reviewer	Review status
chrishtr@	Approved

Survey questions

To request a review, use the "Draft intent..." button above to generate an intent messsage, and then post that message to blink-dev@chromium.org.

Be sure to update your feature entry in response to any suggestions on that email thread.

Please post responses in the comments below.

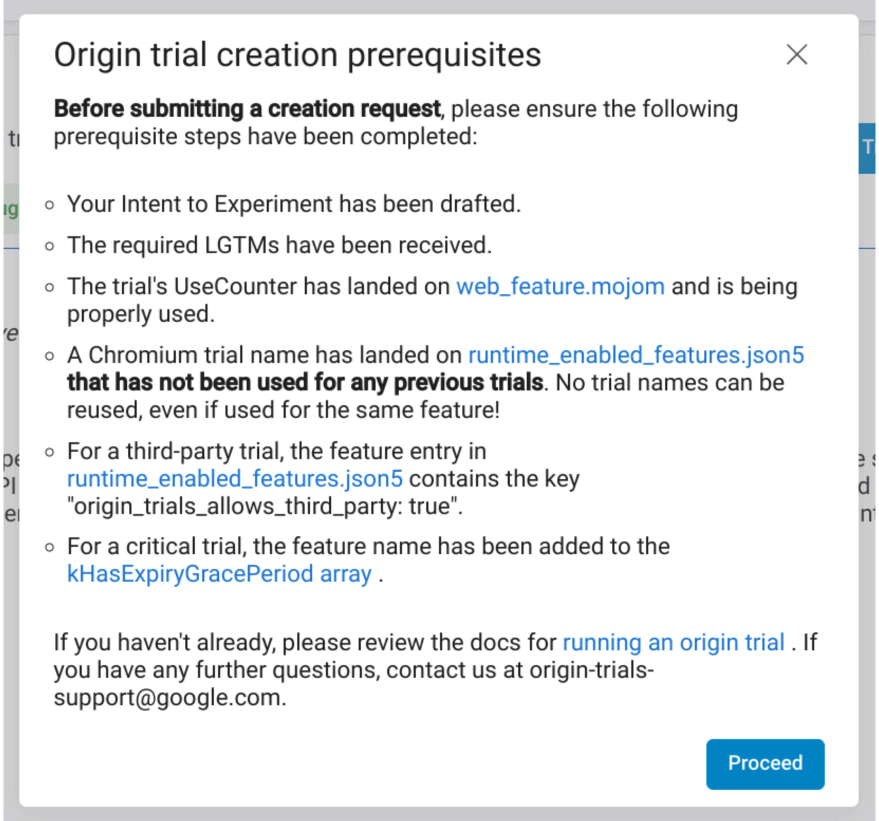
Comments

chrishtr@chromium.org
2023-10-13 17:44:32 (2 weeks ago)

LGTM

chrishtr@chromium.org
2023-10-13 17:44:24 (2 weeks ago)

▶ **review_status**: approved

- 

  Origin trial creation prerequisites

  Before submitting a creation request, please ensure the following prerequisite steps have been completed:

  - Your Intent to Experiment has been drafted.
  - The required LGTMs have been received.
  - The trial's UseCounter has landed on web_feature.mojom and is being properly used.
  - A Chromium trial name has landed on runtime_enabled_features.json5 **that has not been used for any previous trials**. No trial names can be reused, even if used for the same feature!
  - For a third-party trial, the feature entry in runtime_enabled_features.json5 contains the key "origin_trials_allows_third_party: true".
  - For a critical trial, the feature name has been added to the kHasExpiryGracePeriod array .

  If you haven't already, please review the docs for running an origin trial . If you have any further questions, contact us at origin-trials-support@google.com.

- File issue for button and comment access to non Googlers (or at least report the requirement in the UI).
- File issue for typo in “origin_trial**s**_allows_third_party”
- Submitted request to set up OT (Alan can see the button).
- TODO
  - Document test steps with screenshots for manual testing
  - Add metrics
    - For validation attempts, record an enum of all the different outcomes at the point of attempting association validation.
  - DevTools error surfacing
    - Needs a button to attempt validation and report error/success.
    - Backlog task, wouldn't block shipping.
- Relatedish spec: web app tab strips
  - https://github.com/WICG/manifest-incubations/blob/gh-pages/tabbed-mode-explainer.md
  - Uses URL Patterns, is setting precedence in the manifest.
- Link capturing complex scopes
  - Maybe have the scope spec not care about OS link capturing specificity capabilities and just over capture and "manually" in the browser open the OS browser at that URL if it doesn't lie in the fancy web app scope.
- Update(10/27): chromestatus.com issues filed
  - Typo. error in "Origin trial creation prerequisites" dialog · Issue #3459 · GoogleChrome/chromium-dashboard (github.com)

    - Ability to see “Request Trial Creation” button · Issue #3454 · GoogleChrome/chromium-dashboard (github.com)
    - Ability to comment on Origin Trial Privacy, Security, Debuggability, API Owners reviews · Issue #3453 · GoogleChrome/chromium-dashboard (github.com)

Oct 12, 2023

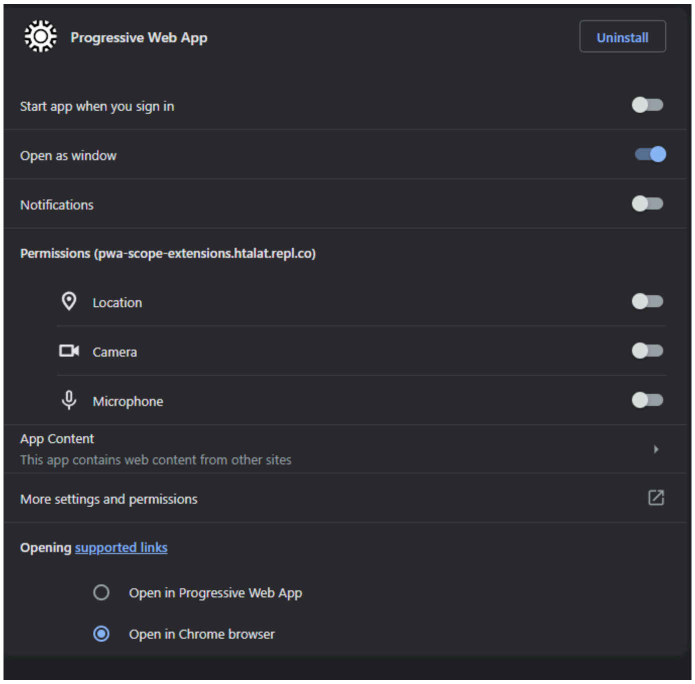
Attendees: alancutter, luhuang

- I2E sent: https://groups.google.com/a/chromium.org/g/blink-dev/c/rGcbrUBB-5E/m/USDt4CmDGQAJ
    - New UI to request privacy, security, debuggability review
        - Security+Privacy: manifest-incubations/scope_extensions-security-privacy-questionnaire.md at gh-pages · WICG/manifest-incubations (github.com)
        - Debuggability: answers in I2E thread
        - alancutter copied answers from thread over into chromestatus.com entry.
- Missing a OT feature name in runtime_enabled_features.json5
    - Looks like this is needed by OT browser test token generation
- Emailed origin-trial-support@chromium.org
- Internal launch update
    - Security review approved.
- WebUI cl merged
- Origin text cl
    - Test All/WebAppNonClientFrameViewAshTest.ShowManagePasswordsIcon/0 failed.
    - Cl to replace RunUntilIdle
- ChromiumOS build: https://chromium.googlesource.com/chromium/src/+/HEAD/docs/chromeos_build_instructions.md
- 120 feature freeze is Monday (Oct 16 ‘23), so we have to push to M121. Add a note in I2E.
- Metrics
    - Record use of scope_extensions in ManifestManager::RecordMetrics().
    - Rename kWebAppEnableScopeExtensions to kWebAppScopeExtensions to reflect the site using the feature rather than it being enabled.
    - Can add some more histograms e.g. validation rates, navigation into extended scope etc. Nice to haves not blocking launch.
- DevTools support
    - Would be good to expose failure messages in DevTools so developers can debug issues in their site e.g. failure to fetch/parse the association file.
- Origin trial length
    - Initial can be 6 milestones.
    - https://www.chromium.org/blink/launching-features/#origin-trials

29 Sept 2023

Attendees: alancutter, luhuang

- Out of scope CHECKs in launch
  - SWAs may hit the CHECK in WebAppLaunchQueue.
  - Considering cases where the launch URL may be out of scope of the web app.
- Nit changes being made on title bar flash CL: https://chromium-review.googlesource.com/c/chromium/src/+/4602916
  - Optional nits optional, can ignore.
- TAG review security question.
  - Response made: https://github.com/w3ctag/design-reviews/issues/875#issuecomment-1740122112
  - Should avoid talking about Chromium, instead add the general behaviour of informing the user about origin traversal.
  - Explainer cruft
    - We're still talking about authorizing link capturing.
    - https://github.com/WICG/manifest-incubations/blob/gh-pages/scope_extensions-explainer.md#declarative-link-capturing-events
    - We should replace that with talking about informing the user when they switch origins within the extended scope.
  - "JavaScript events and queue data such as launchParams should not be shared with origins in extended scope."
    - Should change to say "out of the extended scope" or delete.
- Link capturing
  - Launch review
    - Security seems happy with link capturing behaviour.
    - A bit slow to mark as approved, poking them.
  -
- Scope Extensions UI (4898483) · Gerrit Code Review (googlesource.com)
  - Needs screenshots (add to a new crbug for this UI)

    - 

Progressive Web App

Uninstall

Start app when you sign in

Open as window

Notifications

Permissions (pwa-scope-extensions.htalat.repl.co)

Location

Camera

Microphone

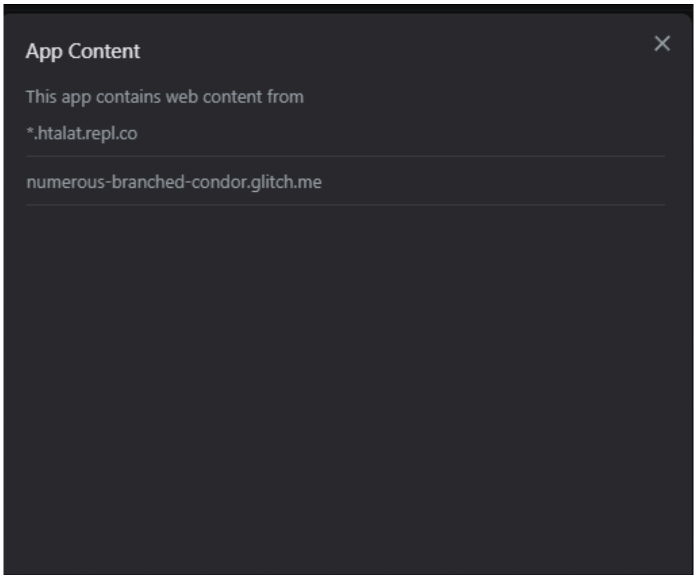
App Content
This app contains web content from other sites

More settings and permissions

Opening supported links

○ Open in Progressive Web App

● Open in Chrome browser



App Content

This app contains web content from

*.htalat.repl.co

numerous-branched-condor.glitch.me

    - Chrome OS
        - App management lives in Ash (the OS) while web apps live in Lacros (separate browser binary/process).
        - Ash cannot access WebAppProvider (it has one but it's for system web apps temporarily while we migrate them out).
        - Chrome OS already surfaces the scope extension URLs via link capturing info which uses the validated scope extension origins. No need to add to Chrome OS UI.
    - WML
        - Link capturing UI could surface scope_extensions like CrOS but doesn't currently so we need a separate place to show it.

        - Can fold it in later when scope_extension link capturing is supported on WML.
    - Who to review changes?
        - Cargo cult off dibyapal's link capturing changes.
            - https://bugs.chromium.org/p/chromium/issues/detail?id=1412856
            - https://chromium.googlesource.com/chromium/src/+log/0e40c97d095e58a561876e82eec61e6304e40539/chrome/browser/ui/webui/app_management/app_management_page_handler.cc
            - tsergeant appears to be the main app-management reviewer.
- Origin trial
    - Must test because manifest OTs are special.
        - Cargo cult off: https://source.chromium.org/chromium/chromium/src/+/main:chrome/browser/ui/views/web_apps/web_app_tab_strip_browsertest.cc;l=1255?q=f:browsertest%20f:web_app%20origintrial&ss=chromium
    - Actually get OT
        - https://www.chromium.org/blink/launching-features/#:~:text=Step%205%20(Optional)%3A%20Origin%20Trial
        - Need to email origin-trials-support@google.com to get it set up.
        - Needs I2E LGTM'd.
            - Only have I2T: https://groups.google.com/a/chromium.org/g/blink-dev/c/OBZBZwS6qr0/m/VRySwd40AQAJ?pli=1
- blinkon
    - Need to record a video showcasing scope_extensions.

15 Sept 2023

Attendees: alancutter, luhuang

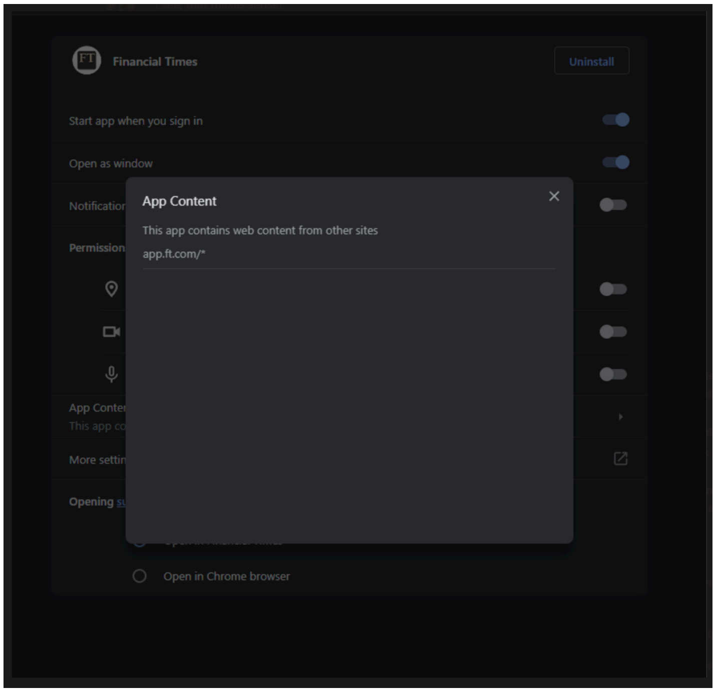
- Lu at TPAC
    - MiniApps
        - Comparison with IWAs.
        - MiniApp Addressing (w3.org)
        - MiniApp URI Scheme · Issue #478 · w3ctag/design-reviews · GitHub
        - Should try to solve same problems once not twice.
    - Permission elements
        - Page Embedded Permission Control (Permission Element) · Issue #35 · w3c/tpac2023-breakouts · GitHub
        - To try and let sites expose permission settings on the page rather than using browser UI.
        - Browser permission UI is clunky and generally not well perceived by users.
    - launch_handler
        - Apple Mozilla not happy with LaunchQueue.
        - Would like to use events instead.

        - Discussion about process with Chromium proposing and semi-shipping things.
    - scope_extensions
        - Brought up in the context of link capturing.
        - Mainly name dropped rather than discussed extensively.
    - First-Party Sets
        - Renamed to Related Website Sets.
        - https://github.com/WICG/first-party-sets
- Scope_extensions launch process
    - Launch_handler leaking info to other origin - waiting on response from security on whether this is okay to launch with.
- Origin title flash
    - Reverting back to DidFinishNavigation
    - Too complex to detect CCT during navigation commit.
    - Worked out what to do to stop the flakes.
- App settings UI
    - https://chromium-review.googlesource.com/c/chromium/src/+/4851340?forceReload=true
        - Needs more webui backend logic.
- Fine to check in code up to M119 branch. Freeze is meant for feature quality - we shouldn't enable any flags between freeze and branch.
- Talk to Dibya about nested scope + scope_extensions for link handling.
    - The way of demarcating scope with nested apps may not work for extended scope.
    - Experience from working on disambiguation and user pref from url_handlers could be useful.

1 Sept 2023

Attendees: alancutter

- Progress on UX CL:



Financial Times

Uninstall

Start app when you sign in

Open as window

Notification

Permission

App Content

This app contains web content from other sites

app.ft.com/*

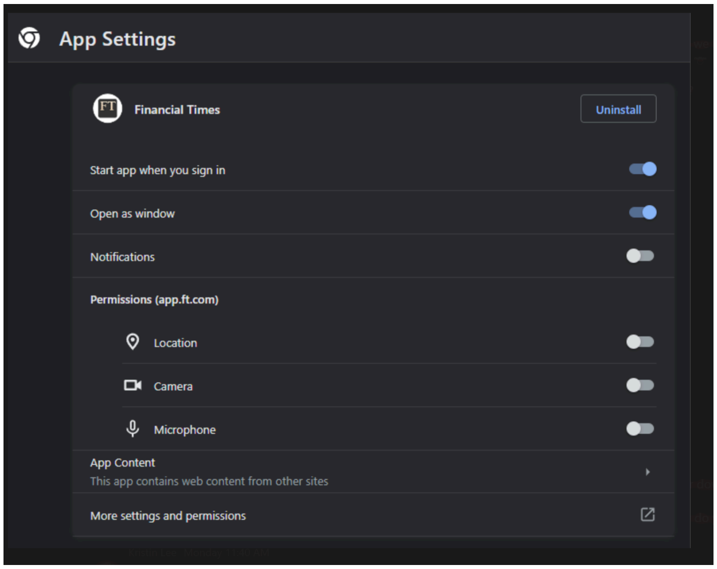
App Conte

This app co

More settin

Opening

Open in Chrome browser



App Settings

Financial Times

Uninstall

Start app when you sign in

Open as window

Notifications

Permissions (app.ft.com)

Location

Camera

Microphone

App Content

This app contains web content from other sites

More settings and permissions

17 Aug 2023

Attendees: alancutter, luhua

- Nearly landing link capturing on CrOS: https://chromium-review.googlesource.com/c/chromium/src/+/4659957
  - Screen recording: https://drive.google.com/file/d/10RkMtKhXCLYI56azMTUKqqIJofoLQiP6/view?usp=drive_link
  - Test site: https://noon-stream-music.glitch.me/
- Include link capturing as part of scope_extensions?

    - Put link capturing for focus-existing behind a different flag as it's security sensitive (user click URL gets leaked onto existing page).
    - Keep the rest of link capturing in as part of scope_extensions.
    - Make sure UX review is up to speed on this inclusion.
    - Easy to put behind a separate flag if need be.
- Showing scope_extensions origins in "supported links" UI.
    - Tied to link capturing, if link capturing is turned off for CrOS then they won't show in supported links.
    - Will need to hack supported links to show them anyway.
    - Should we show non-validated scope_extensions?
        - Yes
        - This will need work.
- AppService port matching fixed
    - This should mean multiple options in intent picker also fixed.
- Origin text CL
    - Changing to ReadyToCommitNavigation works.
- UX
    - ajayrahatekar@, yuchanghu@ (will review impl.)
- Remaining work
    - Title bar flash
    - WML supported links modal dialog
    - Make CrOS supported links UI show non-validated origins too.
- Back burner work
    - Add support in manifest parser for file handlers etc. to use extended scope
    - WML app picker
    - WML link capturing

Aug 3, 2023

Attendees: alancutter, luhua

- WML link capturing implementation underway.
    - https://chromium-review.googlesource.com/c/chromium/src/+/4727423
    - https://chromium-review.googlesource.com/c/chromium/src/+/4745079
- CrOS scope_extensions link capturing
    - Found bug in App Service link capturing where port was ignored.
    - Fixing bug: https://chromium-review.googlesource.com/c/chromium/src/+/4736858
- Scope overlap
    - When the user preference is set all other apps that have overlapping scope are unset.
    - This is the intended behaviour on CrOS and intended for WML as well.
- UX review
    - Agreed to implement design in scope_extensions UX - Google Slides
- Targeting code complete in M118. May fall back to M119.

- Origin text CL
  - https://chromium-review.googlesource.com/c/chromium/src/+/4602916
  - Stuck on flaky test failure.
    - Try wrapping navigation in theme colour await.
    - Otherwise revert back to static HTML, not worth the trouble.
  - When out-of-scope bar is visible and origin text animates (theme color change), maintain current behavior of showing start_url origin. Discuss with Security separately whether that is desirable.
  - https://source.chromium.org/chromium/chromium/src/+/main:content/public/test/browser_test_utils.h;l=165?q=NavigateToURL&ss=chromium seems to wait for the URL to be committed.

21 Jul 2023

Attendees: alancutter, luhua

- TAG review: https://github.com/w3ctag/design-reviews/issues/875
  - Security self review: https://github.com/WICG/manifest-incubations/blob/luhua/scope_extensions-security-privacy-questionnaire.md
    - Create scope_extensions-security-privacy-questionnaire.md by LuHuangMSFT · Pull Request #81 · WICG/manifest-incubations (github.com)
    - First question:
      - What information might this feature expose to Web sites or other parties, and for what purposes is that exposure necessary?
The feature does not expose any information to web sites or other parties.
      - We actually do in the case of cross-origin link capturing when launch_handler: { client_mode: "focus-existing" } is set. Cross origin user navigations are sent to the page as a launch event.
      - We can cause these to just navigate if needed to avoid the cross origin URL leak without blocking launch but the current behaviour needs to be seen by TAG and security review.
- UX slides: scope_extensions UX
- Editing app management layout is hard.
  - Want row like in: chrome://settings/privacy
  - mxcai and tsergeant have worked on app management UI before.
- Web App Scope Extensions - Security Review
- Link capturing on ChromeOS still WIP, got stuck on testing.
  - Created end-to-end association fetch mocking: https://chromium-review.googlesource.com/c/chromium/src/+/4659957/3/chrome/browser/web_applications/web_app_scope_extensions_browsertest.cc#124
- Review title bar flash CL.
  - No need to write end-to-end test.
  - Should flash current page origin instead of start_url, this was what was agreed.

        - The original title bar flash on theme colour change was using start_url even if you're out of scope.
        - Need to start a doc+thread with security UX on what to do here.
        - Let's show page origin when in extended scope and start_url when out of scope to preserve existing behaviour.
    - DidFinishNavigation waits for page load to complete which is probably too long, ReadyToCommitNavigation is probably better.

modified: **../../chrome/browser/resources/app_settings/app.html**
modified: ../../chrome/browser/ui/webui/app_settings/web_app_settings_ui.cc

7 Jul 2023

Attendees: alancutter, luhua

- TAG review not yet filed. Will check with Diego.
- Explainer updates
    - WAOA syntax changes
    - Move `scope` mechanism to a future section
    - Move basing manifest URL references on extended scope to a future section
    - Move scope reduction concept to a future section
    - https://github.com/WICG/manifest-incubations/pull/79
- CL to add origin text flash on origin change.
    - Concerned about coverage of origin association file fetching.
    - Asking to use EmbeddedTestServer request handler instead to intercept the fetch.
    - Maybe instead replace the association fetcher mock with mocking the url_loader_ in the real fetcher.
- Security UX changes
    - Showing the list of associated origins is already supported in Chrome OS link capturing settings UI and it should happen for free as part of app service integration.
    - Discuss with Penny whether the list of associated origins is needed for WML given the lack of link capturing at the moment. Not doing it, assuming security review is fine with it, could reduce launch delays. Double check with Penny.

Jun 8, 2023

Attendees: alancutter, luhua

- Vendor signals might get some response from WebKit given Apple's recent web app push and the use of scopes for how to open links in web apps.
- Lu currently working on title bar reflash when navigating origins.
    - Text is created at construction, needs to wait for WebContents to attach instead.

    - PwaInstallView is an example of a persistent UI button that's constantly watching for when the active tab changes.
- Web app scope_extensions - Impact on permissions UI - Google Docs
    - App settings page shows a handful of permission states (Location, Camera, Microphone) but doesn't state which origin they're for.
    - Could add the primary origin as a subtext under the "Permissions" heading.
    - The user can access the other origin permissions while on those pages in the web app, this is probably fine.
    - Considering showing the scope extension rules behind an ellipsis in app settings.
        - Should they link out to the corresponding permission pages?
            - What do we do for *?
            - Could link to a search page within settings for it.
            - Let's not.
        - Should we show validated extensions or requested extensions?
            - Show requested extensions, the intention of the developer is probably more important.
- alancutter may work on adding link capturing for scope_extensions (ChromeOS)
    - Will need to add the UI changes to link capturing dialogs and settings.
    - Saw some interest internally for handle_links, asked whether it got shipped in Edge.
        - Was not shipped because link capturing hasn't been implemented in Edge.
- Recapping security UX pieces
    - Origin title flash on navigation
    - App settings page (show primary origin, show extended origins)
    - Link capturing dialogs and settings (show extended origins).
- scope_extensions affecting manifest URLs
    - mgiuca thought it would be best to leave it as a later addition.
    - Also considered it would probably be fine to not validate these extended URLs, if you open the app in an out of scope URL that should be fine? We show the CCT and everything.

May 25, 2023

Attendees: alancutter, luhua

- Web App Scope Extensions - Security Review
    - Action items
        - Show origin text prompt in titlebar when the origin changes.
        - Documenting permission UX doesn't change meaning. (Done.)
            - Can look at the UX around permission.site.
    - Associated origins don't necessarily need to provide origin trial keys since their association file can be seen as an opt in.
        - If we need associated origins to provide an OT key they can deliver it as part of the HTTP headers however every origin needs a separate OT key, it would be quite burdensome.

    - Non-goal
        - Messing with the existing origin permission model
    - Future goal for some other project
        - Messing with the existing origin permission model.
        - Have permissions tied to web app identity rather than origins.
        - Permissions granted to a web app do not bleed over to the drive by web or other web apps with overlapping scopes.
- Path to Origin Trial
    - Process: https://www.chromium.org/blink/launching-features/#origin-trials
    - Satisfy security, UX
    - Implement OT keys.
    - Spec. draft (Diego can help with this).
    - Tag review request
        - Might ask if we're changing: https://www.w3.org/TR/appmanifest/#dfn-within-scope-0
        - Explainer currently says start_url will be expanded to associated scopes: https://github.com/WICG/manifest-incubations/blob/gh-pages/scope_extensions-explainer.md#background
        - What happens if validation fails??
            - Maybe we should focus on including this in the OT.
            - Maybe just use the primary scope as a fallback and see how it goes.
    - Scope origin trial down to just CCT, exclude link capturing and manifest URL constraint extension from this.
    - Request comment from other engines.

12 May 2023

Attendees: alancutter, luhua

- Web App Scope Extensions - Security Review - Google Docs
    - dPWA scope_extensions Design - Google Docs
    - New UI proposal: Show when origins are being switched somehow.
        - Maybe show the origin flash in the titlebar like when the app is opened.
        - Maybe have a dismissable UI similar to the CCT that the user clicks away once per origin per app.
        - Intended to address points raised by mkwst last time.
            - Just because an app publisher trusts a set of origins doesn't mean the user also does, we should keep the user informed of any origin traversals.
    - GIF of user experience, Web app scope extensions - YouTube
    - Comment from mkwst previously about not blurring origin boundary: https://docs.google.com/document/d/1-idhx8heajbPYI3cdXFVCjpIuf96cRa_DrRk6147ELI/edit?disco=AAAAlro2P2Y

- alancutter to set up security review meeting.
    - Prefer to have lu alan and mike together but if not then just lu and mike.
    - Give dogfood instructions.
        - CL landed in 115.0.5760.0
        - https://pwa-scope-extensions.htalat.repl.co/
- luhua will try to create demos of link cap. and titlebar scroll before review meeting.
    - Origin text animation: code search
- UI treatment for link capturing with scope extensions
    - Currently link capturing can be enabled from a browser tab in scope with a "pop out" icon that opens an app picker with a "remember my choice" option.
    - The "remember my choice" option can have a "more details" option that opens the app's settings page with the ability to show the full set of scope extension origins.
    - UX may want to change the design, we want to know from security where the limits are.

Apr 27, 2023

Attendees: alancutter, luhua

- Current CLs
    - https://chromium-review.googlesource.com/c/chromium/src/+/4396426
    - https://chromium-review.googlesource.com/c/chromium/src/+/4421489
- Lu working on new section in doc with info
    - Sample app
    - Screen caps of experience
- First Party Sets
    - alancutter filed an issue asking about "what if web apps uses this?".
    - https://github.com/WICG/first-party-sets/issues/145
    - alancutter concluded not a good fit.
- Aim for Mike West review
    - Should try to reveal any security surprises earlier rather than later.
    - Last meeting notes: Google-MSFT PWA Sync on Scope Extensions (Notes)
    - Can make a screen recording for Mike to see the actual behaviour intended.
    - CCT
        - Already have a CL that does this.
    - Link capturing
        - alancutter made a hack CL that gets link capturing working for WML, can be used to create screen recording.
    - Should be prepared to talk about First Party Sets overlap and why not use it.
        - https://github.com/WICG/first-party-sets/issues/145#issuecomment-1515765927
    - Be clear that it's the same fundamentally as url_handlers.
        - Same update mechanism as before.

        - Did not go beyond OT due to API shape, not due to lack of need or security concerns.
    - What needs this?
        - Chrome OS interventions for Microsoft 365.
        - chat.google.com / mail.google.com/chat
        - Country specific TLDs.
    - luhua to write up agenda doc and ping alancutter when ready to set up security overview meeting.
- Blink launch process
    - https://www.chromium.org/blink/launching-features/
    - "dev trial" is when the prototype implementation is functional enough that you can tell people on blink-dev to start trying it out.
- Link capturing
    - On Chrome OS
        - alancutter can probably implement this by converting validated scopes to intent filters published to app service.
        - https://pwa-scope-extensions.htalat.repl.co/ test site that uses wildcard.
    - On Edge
        - There's a start up flow that does link capturing on the command line URL (like how url_handlers worked).
- [Allow WebAppInstallInfo to be copy constructed (4450666) · Gerrit Code Review (googlesource.com)]()
    - Suggestion from alancutter: try to std::move WebAppInstallInfo into FinalizeInstall and FinalizeUpdate.

Apr 13, 2023

Attendees: alancutter, luhua

- Ok to do manifest update browser tests in a new file
    - Probably better as you can set up your own third party origin test infra.
- Follow up on whether const WebAppInstallInfo& signature in FinalizeInstall/FinalizeUpdate can be pass by value (instead of cloning within).
- Concerns about validated_scope_extensions in WebAppInstallInfo and what that means for FinalizeInstall/FinalizeUpdate's API.
    - Put CHECK in FinalizeUpdate that it's always set as a change detecter for if the callsite changes.
    - Confusing and unfortunate.
- First Party Sets
    - It's a centralised list! https://github.com/GoogleChrome/first-party-sets/blob/main/first_party_sets.JSON
        - Actually it's a per-browser list.

    - Probably like this to enable data sharing between multi origins but also avoid user tracking across the internet.
    - Not appropriate for scope_extensions, we don't do any data sharing and a centralised list is nope.

Mar 30, 2023

Attendees: alancutter, luhua

- Installing scope_extensions validation data
    - Putting validation in WebAppInstallFinalizer
        - Concern about what happens when finalizer is turned into a command: https://bugs.chromium.org/p/chromium/issues/detail?id=1300011
    - Changes to ManifestUpdateCheckCommand
        - After scope_extensions has updated in install info (so as to validate with updated origins)
        - Add WAOA validation result to the diff
            - If any changes, set UpdateNeeded state
        - Add validation result to new web app install info to be committed later
- Removing CCT for extended scopes
    - Existing functionality for this: https://source.chromium.org/chromium/chromium/src/+/main:chrome/browser/web_applications/chromeos_web_app_experiments.h;l=45?q=chrome%20os%20web%20app%20experiment&ss=chromium
    - Add Microsoft Office web app experiment for ChromeOS (3912642) · Gerrit Code Review (googlesource.com)
    - Leverage and generalize WebAppRegistrar::GetUrlInAppScopeScore hatalat@microsoft.com
    - alancutter recommends:
        - Update ChromeOsWebAppExperiments to use ScopeExtensionInfo values.
        - Turn ChromeOsWebAppExperiments implementation into the real scope_extensions implementation.
        - Reduce the injected scope extensions done by ChromeOsWebAppExperiments to be a small footprint on WebAppRegistrar that just injects hard coded values for the MS365 web app.
- WML link capturing
    - https://bugs.chromium.org/p/chromium/issues/detail?id=1280448
    - Still need to chat with dibyapal.
        - Get on the same page and make sure what's expected to be implemented is actually what is.
    - Not sure why some bugs are internal

Mar 16, 2023

Attendees: alancutter, luhua

- CLs are open - good progress there.
- Minor change: GenerateAppIdUnhashed should return GURL.
- Happy with web_app_identity naming.
- Add Daniel to CL review to get opinion on changes to install sequence.
- Validate WAOA after dialog shows and closes and gives a positive result to continue install.
- Can use TestFuture instead of RunLoop.
- Install flow refactor idea:
  - Similar to how InstallFinalizer is at the end of every install flow.
  - What about something that does the download of the resources referenced by WebAppInstallInfo?
  - This would make it easier for associations fetching to have a single place to live.
  - Pull out fetching WebAppIntallInfo resources as a common action.
  - WebAppDataFetcher should be able to fetch and populate a WebAppInstallInfo.
  - AI(alancutter): Advocate this idea with dPWA team.
- Is registrar_unsafe() temporary?
  - No, it's needed for synchronous access by UI.
- We currently have a semi-dead lock with manifest identity updating.
  - Manifest updating takes a lock on the app.
  - The user is given a choice to uninstall the app.
  - The uninstall command wants a lock on the entire system.
  - It gets blocked on the currently running manifest update command and the user sees nothing happen.
  - We need a way to schedule commands in the context of an already held lock and be able to "reuse" what's already locked in order to compose commands inside each other.
  - AI(alancutter): Raise this with dPWA team.

Mar 2, 2023

Attendees: alancutter, luhua

- Manifest update CL merged
  - https://chromium-review.googlesource.com/c/chromium/src/+/4190417
  - Lu working on followup to add origin association validation back to manifest update manager.
  - Considering observing manifest update checking to trigger association revalidation.
  - Or just add another state enum value for triggering revalidation.
- Format changes to associations.

  - Dropping origin sub scoping for associations, entire origin to entire origin associations.
    - Scenario
      - Web app associates with *.site.com
      - How can site.com restrict which subdomains the web app associates with?
      - Maybe don't provide this capability.
      - Maybe let site.com specify include and exclude subdomains?
  - This makes it very similar to first party sets.
    - https://github.com/WICG/first-party-sets#first-party-sets
  - Desires for origin associations
    - Be able to exclude specific subdomains e.g. documentation.origin.com.
    - Be able to exclude subdomains from each other.
  - Past issue raised about subdomains: https://github.com/WICG/first-party-sets/issues/46
  - Drop the name "web app"?
    - Probably shouldn't be generic about origins being associated as First Party Sets already does this.
    - Probably should be specific to web apps to avoid overlap in purpose.
    - Drop "web"?
      - There are web related schemes for associations with other app platforms e.g. verified links.
      - Should probably keep "web app" naming as it's for the web app platform.
  - Maybe association file should be a list of objects rather than a dict of app id keys.
    - Much more like scope_extensions' format.

    -
      ```
      "scope_extensions": [
        {"origin": "*.example.com"},
        {"origin": "example.co.uk"},
        {"origin": "*.example.co.uk"}
      ]
      ```

    -
      ```
      "web_apps": [
        {"app_id": "https://example.com"},
        {"app_id": "https://other-site.com"}
      ]
      ```
  - Maybe provide no sub scoping at all initially and wait for feedback to give an indication which side is best to decide.
  - Crazy idea: "scope" is removed, sites are always entire origins, maybe scope_extensions should be origin_extensions.
- Link capturing is being worked on by dPWA team!
  - But they are going to re-implement link capturing entirely.
  - WML inclined to not depend on App Service. Unfortunately means duplicating its link capturing implementation.
  - alancutter to find crbugs to follow this work.

        - Yes but not public?? https://bugs.chromium.org/p/chromium/issues/detail?id=1412856#c2
- Takeaways for Lu
    - Change association file format to list of objects. App_ids keying empty objects seem silly if scope is removed.
    - Manifest `scope_extensions` format to stay the same.
    - Config of scope/paths can be added later to either side after studying requests from real usage.
    - Still possible to demo link capturing with appservice but will have to rely on new W/M/L work long term.

17 Feb 2023

Attendees: alancutter, luhua

- Need to clean up old url_handler prefs because this went to origin trial.
    - [dPWAs] Deprecate url_handlers local state prefs (4226384) · Gerrit Code Review (googlesource.com)
- Next set of changes: validating origin associations
- Manifest update refactor: https://chromium-review.googlesource.com/c/chromium/src/+/4190417
- Updating origin association validations
    - Trigger at the end of manifest updating after we're completed either because we're up to date or an update has been applied.
    - Inject a step before ManifestUpdateManager::OnUpdateStopped().
        - Watch out for early exits if no update is needed. Turn apply update step into a no-op if no updates are needed and flow onto the next step.
- Questions about WebAppLaunchManager
    - [dPWA] Refactoring BrowserAppLaunch to use async launch (4111472) · Gerrit Code Review (googlesource.com)
    - BrowserAppLaunch uses WebAppCommandScheduler::LaunchApp which keeps the browser alive while the web app is launching.
    - WebApps::Launch() calls directly into web_app_launch_manager_->OpenApplication(std::move(params)); without any keepalives and has a whole bunch of discrepancies in the prelude code.
- AppLaunchParams could be created earlier in the start up launch process.
    - StartupBrowserCreator::ProcessCmdLineImpl -> StartupWebAppCreator::MaybeHandleWebAppLaunch -> WebAppCommandScheduler::LaunchApp (which creates AppLaunchParams)
    - Instead, StartupWebAppCreator could create AppLaunchParams and use WebAppCommandScheduler::LaunchAppWithCustomParams instead.
        - Any additional info from StartupBrowserCreator could be set in StartupWebAppCreator and then set in AppLaunchParams and be available through the rest of the launch.

3 Feb 2023

Attendees: alancutter, luhua

- Abhinav moving.
  - Lu picking up implementation work again. Expect CLs to be lined up soon.
- Thinking of listing all the UI surfaces that may need modifications.
  - Will be good to present to trust/safety/security to collaborate on as part of their review.
- Code changes involved in the behind-a-flag implementation
  - Manifest parser
  - Updating
  - Storage on WebApp.
  - Integration with out-of-scope bar.
  - Integration with link capturing (Chrome OS only currently, can poke at enabling it for WML).
- TODO(Lu) Experiment with W/M/L link handling. Could be a good demonstration of scope extensions if low effort.
  - https://source.chromium.org/chromium/chromium/src/+/main:chrome/browser/apps/intent_helper/common_apps_navigation_throttle.h;l=29?q=commonappsnavigationthrottle&ss=chromium
  - Try removing the ifdef here.

20 Jan 2023

Attendees: Natasha, alancutter

- Interest in scope extensions for multi app purposes
  - Probably better to seek support from sub app APIs.
    - Talk to SF dPWA team (Dan Murphy, Penny McLachlan)
  - Web app isolation may break this use case.
    - Thinking of making every web app stop sharing the same local storage like they do now to be more like native apps.
    - An iframe'd origin in one web app won't be able to talk to itself in another web app.
- Link capture could capture the information that you clicked a link to somewhere else. Needs to be rationalized when combined with Launch Handler APIs (that cause link capture launches from other origins to fire an event in the app with the URL).
- Manifest update for origin association verification
  - It stores its data in local state.
  - Having this used at start up is a nice to have we can add on later, we don't need it now.
  - We can move the storage into the web app database like all the other fields.
  - We can make it update just like all the other manifest networked data.

Jan 5, 2023

Attendees: alancutter, luhua

- Recap: Dec 12 call with dmurph et al. Google-MSFT PWA Sync on Scope Extensions (Notes) - Google Docs
  - Had contact with Mike West
  - Takeaway, needs a prototype to better understand the intent.
  - Show user experience so they can think about security implications.
- alancutter been OOO most of December.
- Luhua was OOO Oct/Nov. May need more OOO time in Jan.
- Abhinav working on implementation
  - https://chromium-review.googlesource.com/c/chromium/src/+/4118114
  - Prototype: https://chromium-review.googlesource.com/c/chromium/src/+/4034807
  - Chrome status: https://chromestatus.com/feature/574653795611443
- Existing meeting is quite stale now, let's make a new one with alancutter as owner.
  - Add howard.wolosky@microsoft.com nagaiton@microsoft.com akasamsetty@microsoft.com to new meeting invite.
- Design doc: dPWA scope_extensions Design Doc
  - Luhua to remove “permission” language. It’s not the right wording to describe the authorization concept. Also authorizations can be added later.
- Prototype
  - Draft CL Web App Scope Extensions Prototype (DO NOT COMPLETE) (I59412664) · Gerrit Code Review (googlesource.com)
    - alancutter to try this out and review
    - IsUrlInAppExtendedScope() is key thing to look at.
    - Doesn't yet integrate with app service intent filters which would provide link capturing scope extension.
  - Try to implement ChromeOS link capturing in the prototype to illustrate UX
    - W/M/L would also work if
    - WebAppPublisherHelper::CreateIntentFiltersForWebApp()
- Web App Origin Association Design (URL Handlers) - Google Docs (utility process for fetching/parsing association files. FYI.)
  - Idea: Would be very cool to have a Blink utility process to run manifest/media query/CSS parsing in without the weight of a full web page attached.

16 Sept 2022

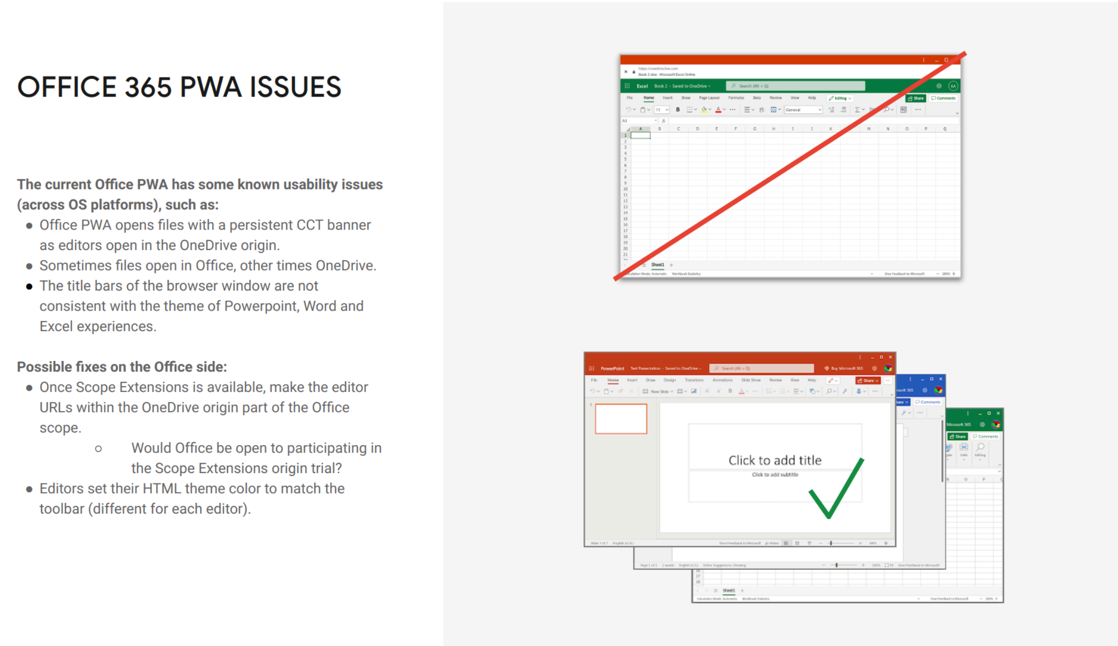
Attendees: alancutter, luhua

- TPAC
  - Isolated apps
    - To enable APIs not available on the general web.
    - Similar to Electron.

    - Has more security guarantees.
    - reilyg, pjmclachlan
  - Fugu presentation
  - Web app WG
    - Talking about multi-origin scopes.
    - Known issue to be addressed for web apps.
    - Do permissions get lumped together?
    - Needs request for position from WebKit.
  - Icon updating
    - Had good discussions.
    - Token syntax too undiscoverable.
    - Use image URL diff if token not set.
    - Let token: null mean user agent decided. Can have image URL diffing be the recommended algorithm.
  - Media Queries and Overrides
    - Lots of debate both ways.
- Office PWA
  - Found PM owner, OOO for a while.
  - Internal chat before starting an email thread.
  - Found more specific owner.
  - Things to cover
    - Understanding the multi-origin architecture better. Deliberate or for technical reasons?
    - Interest in using scope_extensions?
      - To remove CCT when editors are opened in Office PWA.
      - To enable link capturing so opening documents don't bounce out to the browser.
      - Would need to link capture https://1drv.ms/w/s!ABzlSdJE68Q5fg?login_hint=&ct=1662074610244&wdOrigin=OFFICECOM-PWA.MAIN.MRU URLs as well.
    - Having the editors set their <meta name="theme_color" content="<color>"> theme colors.
- Scope extensions
  - Spec might have assumptions about scope being single origin.
  - Probably need to audit it as part of multi origin.
- launch_handler syntax updated:
  - https://developer.chrome.com/docs/web-platform/launch-handler/
  - launch_handler: { client_mode: 'focus-existing' }

September 1, 2022

Attendees: alancutter, mgiuca, luhua, mjackson

- 

OFFICE 365 PWA ISSUES

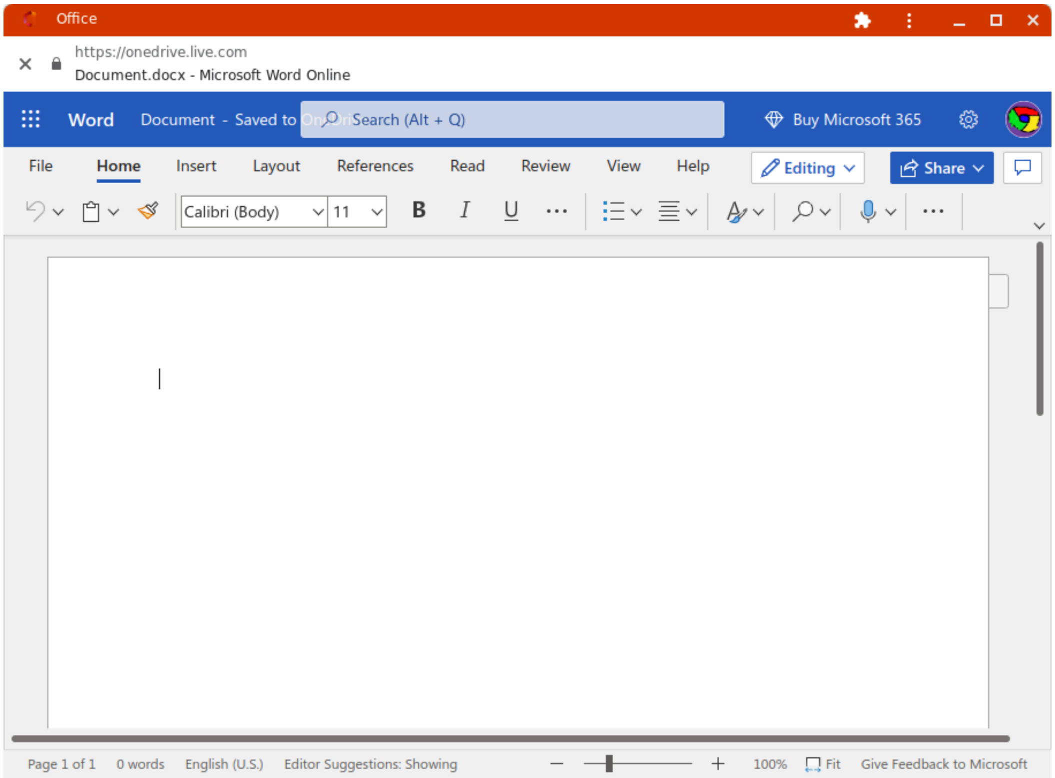
The current Office PWA has some known usability issues (across OS platforms), such as:

- Office PWA opens files with a persistent CCT banner as editors open in the OneDrive origin.
- Sometimes files open in Office, other times OneDrive.
- The title bars of the browser window are not consistent with the theme of Powerpoint, Word and Excel experiences.

Possible fixes on the Office side:

- Once Scope Extensions is available, make the editor URLs within the OneDrive origin part of the Office scope.
    - Would Office be open to participating in the Scope Extensions origin trial?
- Editors set their HTML theme color to match the toolbar (different for each editor).

- Office PWA lives at office.com
- OneDrive PWA lives at onedrive.live.com
- Office editors (Word, Excel, PPT) are hosted under onedrive.live.com
    - This is causing UI issues when using the Office PWA.
- Repro steps:
    - Log into and install office.com
    - Create new Word document in PWA window.
    - CCT shows and the theme colour doesn't match Word.

    - 

    - Also if you open a document from office.com it will bounce to the browser.
- (AI:luhua) Find Office contact
    - cc alancutter@google.com and mgiuca@google.com
- ChromeOS is getting feedback about Office experience
    - Office doesn't feel like an app when docs open in the browser.
- Ideal experience:
    - Either the editors take place within the Office app, or
    - There is a dedicated Word/Excel/PPT app separate to Office, and when you open the editor, it bounces back to the browser and you get a prompt to install those separate PWAs. (Also fine, but there doesn't seem to be any editor-specfic PWAs.)
- scope_extensions could extend Office's scope to include these editors and treat them as part of the app.
    - SE project has been tracking this use case for general cloud storage solutions like OneDrive but also others.
- Not a ChromeOS-specific issue, but we're reporting it because we don't have native editors here (it generally affects the Office PWA).
- Document URLs look like:
    - https://onedrive.live.com/edit.aspx?resid=39C4EB44D249E51C!126&cid=39c4eb44d249e51c&CT=1662074393938&OR=ItemsView
    - Maybe Office could scope_extend to "https://onedrive.live.com/edit.aspx" to capture all the OneDrive editors.
    - Not sure if that is reasonable, needs discussion with Office team.

    - Also redirects via a short URL which might also need to be considered part of the scope.
        - https://1drv.ms/w/s!ABzlSdJE68Q5fg?login_hint=&ct=1662074610244&wdOrigin=OFFICECOM-PWA.MAIN.MRU
- Help with Scope Extensions
    - Biggest help that Google can provide would be helping to get through the release process. (Really helps to have justification for it.) AI(mgiuca): Follow up with MTV folks to prioritize and help get through the process.
- TPAC
    - Matt emailed Marcos to try to give scope_extensions + launch_handler a 45 minute slot. Awaiting response.
    - Mike and Lu from MS will be there.
    - Eric and Louise from Google Sydney will be there.
- Thinking about dev trials, origin trials. How do we justify this?
    - Are there clients that would actually use this?
        - Office, Zoom (sub-domains) could be interested
    - 1. It affects CCTs everywhere.
    - 2. It affects link capturing on ChromeOS.
    - 3. It affects OS link capturing on Windows+Edge.
    - 4. Hopefully someday Chrome Win/Mac/Lin will support link capturing and Edge will support in-browser link capturing.
- In browser link capturing code: https://source.chromium.org/chromium/chromium/src/+/main:chrome/browser/chrome_content_browser_client.cc;l=4490
- Should there be multiple feature flags governing which features are affected by scope extensions?
    - No, just one flag to extend the scope, and all features respond to it the same (modulo bugs).

August 4, 2022

Attendees: alancutter, luhua

- scope_extensions going slow.
- Still looking for a champion.
- UX considerations
    - When we show the origin we need to show more than one?
        - During installation.
    - How do site permissions work with multi origin apps?
        - App management currently surfaces site permissions for the primary origin, what about other origins?
    - What would the UX be if permissions were different?
    - Should the user be informed of origin changes during manifest update?
        - We have an update prompt now to inform the user of sensitive changes.

        - Should we be informing about changes that weren't mentioned at install time? Probably not.
- What if web apps are separated from browser state?
    - Separate cookies, local storage, user permissions.
    - Might be confusing for users given web apps aren't fully distinct as different from sites.
    - Electron apps are like this already.
    - What about the stuff that's still shared? Password management, extensions?
- CCT not showing on associated domains is a big goal.
    - Could we ship this part without dealing with the site permissions part?
        - App management links out to other origins but doesn't surface their settings.
        - Permissions granted to single origin at a time (no change).
        - Mostly annoying for the user in these cases but not broken.
        - Very complex to solve.
- Implementation plan.
    - Re-use:
        - Validation.
        - Some include path code.
    - Push logic into scope checks in the registrar.
- Manifest updating stuff that alancutter's doing.
    - https://chromium-review.googlesource.com/c/chromium/src/+/3759058/comment/a2787656_7e2391f3/
- Different TLDs
    - Pinterest has different TLD sites that are all their own unique web app IDs.
    - Origin associations could enable cross origin installation so multi-TLD sites install just one app ID.
- Pinterest is using url_handlers still.
    - https://www.pinterest.com.au/.well-known/web-app-origin-association/
    - https://www.pinterest.com.au/manifest.json
- New link capturing UX work on Chrome OS.
    - chrome://flags/#link-capturing-ui-update
    - chrome://flags/#link-capturing-infobar
    - chrome://flags/#intent-chip-skips-intent-picker
    - chrome://flags/#intent-chip-app-icon

July 7, 2022

Attendees: alancutter, luhua

- Url handling about:flags expiring. Remove from about:flags but keep the feature flag in place for testing until enough code has been reused by scope extensions.
- Discussion about icon updating
    - Name change update prompting is launching in M104.
    - It happens as part of manifest update

        - This happens for browser tabs. Showing web app related prompts while using the drive by web is weird, we should consider changing that.
    - Apply updates immediately?
        - Web app windows would have to deal with that.
        - They already deal with theme colour changing dynamically: soft-puppy.glitch.me
- Link capturing
    - On W/M/L - currently unowned.
        - https://bugs.chromium.org/p/chromium/issues/detail?id=1280448
    - On Chrome OS - actively having its UI updated.
        - chrome://flags/#link-capturing-ui-update
        - chrome://flags/#link-capturing-infobar
        - chrome://flags/#intent-chip-skips-intent-picker
        - chrome://flags/#intent-chip-app-icon
- Web app tab strip actively being worked on.
    - https://bugs.chromium.org/p/chromium/issues/detail?id=897314
    - Captured links could open new tabs.
- Scope extensions
    - Looking for Chromium champion from dPWA team (same timezone and ability to speak to scope_extension needs in the dPWA team).

June 9, 2022

Attendees: alancutter, luhua

- Scope extensions work might be starting soon.
- Split into two parts, removals and additions.
    - Main-origin filtering (include_paths, exclude_paths)
    - Additional origins
    - Same syntax for filtering can be used in manifest, and in association files.
- We could land code behind flag to observe the effects of extended scope
    - Document effects on various other features
    - Intent to prototype already sent so unblocked on landing code.
- Questions from the explainer.
    - Are wildcards for sub-origins necessary?
        - Necessary for dynamic sub-origins such as tenant names
        - Assoc. File would have to be served from origin 1 level up. TODO: check with security reviewers that this is appropriate.
        - Alternative: Have association file at every subdomain?
            - Breaks guarantee of being able to download association file at install time.
            - Bottleneck to navigation.
            - Fails offline.
    - Do we need include paths and exclude paths in the association file at the start?
        - No.

        - Try to implement without include_paths (i.e. default is /*). TODO: clear this with security after prototyping.)
        - Security may be concerned about /* being the default.
    - Instead of `include_paths`, we could also reuse existing `scope` syntax in association files.
        - Allows the default to be no-URLs-included.
        - Doesn't introduce new syntax.
- Include and removal as separate manifest fields?
    - Liking the split form between [same origin and associated origins](#).
    - Using same include/exclude syntax between manifest and association files is an ergonomics plus.
    - Same scope removal can be isolated syntactically from origin association.

Mar 31, 2022

Attendees: alancutter, luhua

- Any organizational updates on scope modifications?
    - MS candidate to work on next after link capturing but not set in stone.
    - alancutter priorities for Q2 still being decided.
- OS level link capturing
    - alancutter: sgtm
    - User preference source of truth will be hard
        - In the OS?
        - In the browser?
    - Can apps query their own OS preference?
        - They can plumb that state back to Chrome so in-browser link capturing is consistent with OS link capturing.
- Any updates on launch_handler?
    - Which milestone will have format changes?
        - https://github.com/WICG/sw-launch/pull/53
        - M102 (targeting)
        - Remove old values?
            - Will do that when shipping.
- App partner will use launch_handler Origin Trial
    - What should clients do with the new syntax?
        - Use both:

        -
          ```
          {
            "route_to": ["existing-client", "existing-client-retain"],
            "navigate_existing_client": "never",
          }
          ```
    - No negative feedback or requests from Edge clients.

Mar 3, 2022

Attendees: luhua, hatalat, alancutter

- alancutter back on link capturing work.
- MS partners interested in launch_handler.
    - OT currently available (to 102), extension will be requested soon. Works on all platforms.
    - https://developer.chrome.com/origintrials/#/view_trial/2978005253598740481
    - Docs: https://web.dev/launch-handler/
    - Demo site: Musicr 2.0 (launch-handler.glitch.me)
    - alancutter's demo site: https://sulky-lopsided-bean.glitch.me/
    - Expect “navigate_existing_client” to change soon: https://github.com/WICG/sw-launch/pull/53
- MS currently working on handle_links in Edge currently taking most of the time.
- MS okay with Google working on primary scope modifications as a separate project to cross origin scope extensions.
- Cross origin scope extensions can reuse the include/exclude URL syntax that the primary scope modifications spec defines.
- Should browser link capturing perform more complex scope algorithms than the host platform?
    - Would be difficult to present to app developers

Feb 3, 2022

Attendees: luhua, mjackson, alancutter, estade, hatalat

- Admin: add hatalat@microsoft.com, akasamsetty@microsoft.com to meeting invite (5 min)
    - Hassan and Abhinav are devs who will help with this effort.
- Link capturing from WebContents and OS early implementation.
    - App picker showing at start up.
    - May not be compatible with Chrome’s UI plans for link capturing.
        - Want to unblock the navigation, just open in the browser by default and offer a low friction UI to open in app window and persist the choice.
        - Chrome OS strongly going in this UI direction, WML likely to follow suite.
        - Chrome wants to play it safe with the behaviour initially and stick to browser navigation and let the user choose to deviate.
    - No commitment to make use of the handle_links signal from Chrome yet but not ruled out for future iterations to the UI.
    - Should `handle_links` exist in Chromium repo or Edge?
        - Given no current plans from Chrome product managers to make use of `handle_links` (although possible in the future)
        - alancutter thinks it’s reasonable to exist in Chromium as an experimental feature we may adopt later but also okay if it’s removed to live in Edgium.

        - The startup app picker UX is unlikely to be used in Chrome and should probably be removed.
        - Keep URL Handlers for now to be able to exercise the origin associations code, remove when scope_extensions is further developed.
- Demo startup app launch (10 min)
    - Feature flag?
- What’s the status of app service for desktop? (5 min)
- Partner feedback (10 min)
- scope_extensions format (doc link)
    - Preliminary decision?
    - Use two crbugs/chrome features to manage if we’re splitting scope_extensions and “scope_extensions_associated_origins”.
-

Jan 20, 2022

Attendees: luhua, alancutter, estade, mjackson

- Scope_extensions already in I2P. Start prototyping to flush out issues.
    - Maintaining a list of issues here
- Privacy and security
    - Have had a thread with harrisonsean@google.com about scope extensions, they are on Chrome privacy review. Also ravjit@google.com.
- Should we decouple scope_extensions and scope_exclusions implementations?
    - Scope_exclusions is likely to have fewer security/privacy concerns
    - Create examples with different formats and share out for comment
- Demo possible usage of handle_links
    -
- Any more feedback from partners on how they might use handle_links or scope_extensions?
- `.well-known/web-app-origin-association.json` proposal: https://github.com/WICG/manifest-incubations/blob/gh-pages/scope_extensions-explainer.md#proposal
- Is it okay for app associations to get very out of date if the user doesn’t enter the extended app scope for years?
    - Was discussed and has a doc section somewhere.
    - What if a.com primary origin includes b.com which turns bad and a.com’s manifest can no longer be reached? b.com may start impersonating a.com’s identity maliciously.
- API design sketch playground

    -
      ```
      "scope_extensions": [{
          "include_origins": ["*.app.com", "*.app.co.uk"]
        }]
      ```

  -
    ```
    "scope": "/app/scope",
    "scope_extensions": [{
        "exclude_patterns": ["a/*/a", "b/*/b"],
        "include_origins": ["*.app.com"]
      }, {
        "include_paths": ["/c", "/d"],
        "exclude_paths": ["/a", "/b"],
        "include_origins": ["*.app.com", "*.app.co.uk"]
      }, {
        "exclude_paths": ["/a", "/b"]
      }]
    ```

Jan 6, 2022

Attendees: luhua, alancutter, mjackson, estade, ibo

- Comment on Slack: https://chromium.slack.com/archives/C02ARGE1Z97/p1639668793053800
  - Is app settings a blocker for link capturing?
    - Yes but may not be the last blocker.
  - Post-nav UI
    - Icon UI that appears after app window launches
    - Button in app window to persist link capturing preference.
    - Blocking WML link capturing launch.
    - Did not block CrOS link capturing launch (already shipping).
- Scope_exclusions
  - Partners have URLs in app scope that are just regular pages e.g. price lists that should open in a browser rather than the app window..
  - They want to exclude those links from the app scope.
    - Are they blocking adding a manifest on scope exclusions?
      - If so then making handle_links: none remove the UI promotion for link capturing could be a fast stop gap.
- Scope_extensions
  - Getting handle_links into OT before scope_extensions implementation work.
  - Handle_links OT may be M100.
  - Scope_extensions has a lot of design discussion to work through, design can happen in parallel with handle_links.
  - Unsolved issues to document and address:
    - Settings UI, how do we show multi scopes?
    - What happens when users set notifications on multi scoped web apps?
    - Do all places in the code that check scope work with this extended/excluded scope?
    - How do we handle extended scopes becoming invalid?

  - What happens to file handlers on scope extension invalidation that become out of scope?
- Uninstallation now offers to clear data and permissions from all origins?
  - But what about past extended origins?
  - At the moment permissions don't even get cleared from the main origin (although they should). Since they don't and the world's not on fire, maybe worrying about past extended origins isn't super important?

Dec 9, 2021

Attendees: luhua, alancutter, mjackson

- ChromeOS: default to opening links in the browser for all app types and letting the user opt into app link capturing (@alancutter)
  - Does this refer to the "prompt and opt-in" behavior?
    - Yes
  - Does this apply to out of browser link capturing?
    - The intention is for it to, yes.
    - In the design we don't want to block navigation.
  - Default behaviour:
    - Start in "ask" mode.
    - Open links in browser tabs.
    - Show the bubble after navigation to prompt the user to pick always/never open in app window.
  - Possible alternate default behaviour:
    - Open in app window and ask whether to open in browser instead.
    - Possibly do this only when `handle_links: handle` is set in the manifest.
  - What if the browser isn't open and link capturing is enabled?
    - Open in the app window directly.
  - Should W/M/L deviate from this?
    - They intend to be the same as Chrome OS by default and make deliberate product decisions to deviate as necessary during their implementation.
  - Workaround for UI before app settings page is available
    - Same UI treatment as "run on os login" etc?
    - Ask Evan on Slack.
  - Launch handler is in M98 OT for all platforms
    - Screen cast: https://youtu.be/EZFnGuIUfrc
    - Test site: https://sulky-lopsided-bean.glitch.me/
- Handle_links parsing comment re optional manifest field: https://chromium-review.googlesource.com/c/chromium/src/+/3290595/comment/8c3d114a_844bf819/

    - Mojom lacking the ability to make primitive fields optional: https://bugs.chromium.org/p/chromium/issues/detail?id=657632

Nov 11, 2021

Attendees: luhua, alancutter, estade, mjackson

- User controlled link capturing is shipped on Chrome OS today.
    - Screen cast of prompt and opt in: https://drive.google.com/file/d/1c3ArZhiRTfYWGM6i0M7c_BDyB5l6mAmk/view?usp=sharing
    - The UI will change in the near future but the setting is done and will remain unchanged.
    - The prompt UI is changing but semantics will remain the same.
    - WML intends to ship the updated UI when it's done. Also blocked on having a settings page.
    - WML app settings page could ship Q1 2022.
- What happens when a new web app is installed with an overlapping scope with the user preference?
    - The preference is not reset.
- Android link handling UI
- Have agreement to use "turn off other overlapping scopes" simple approach for first shipped implementation.
    - Can make this more complex later if we find cases that don't work well.
    - Introduction of scope_extensions is more complicated. Can be analyzed separately.
- Chrome will allow the user's setting to override developer's `handle_links` input. Spec. can word `handle_links` as a hint. This information can be used to control if browser prompts user, for eg.
    - Which of "open/don't open/ask" is the default can be informed by the developer's hint. For eg. if handle_links is "none", default setting is "don't open" and browser does not match links or prompt. User can find the app setting and override.
- Consider replacing "all"/"none" with terms that work better as hints from the developer? Eg. "preferred"/"not-preferred". "all"/"none" were chosen to contrast with additional config to define "some".
- Is "handle_links" still a good name to represent a hint API? "link_handlers" is also possible but suggests a list.

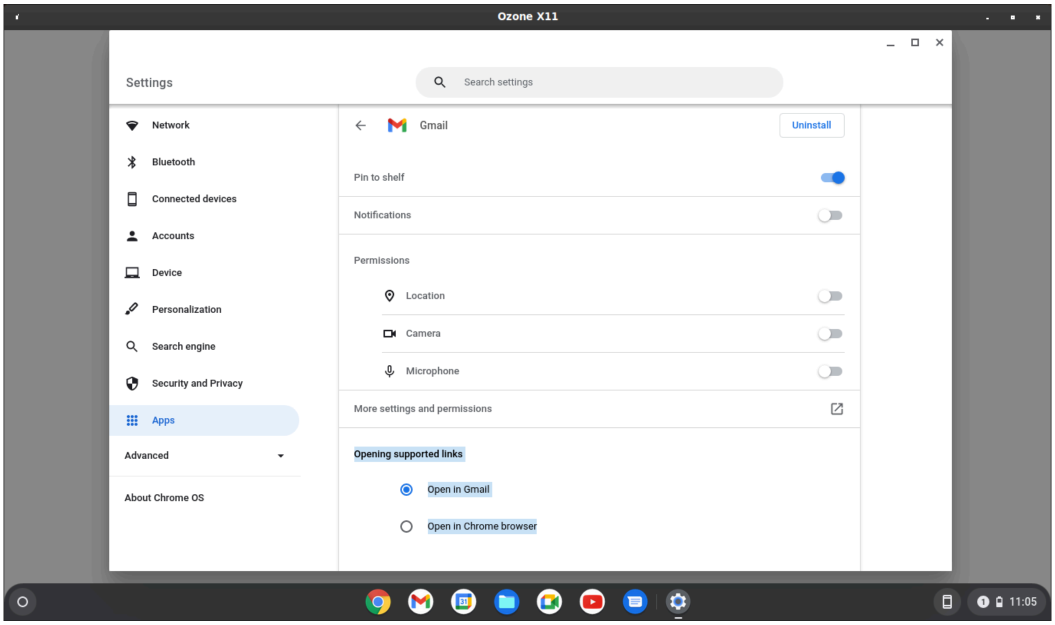
Nov 4, 2021

Attendees: luhua, mandy.chen, mek, mjackson, alancutter, estade

- Add mjackson@microsoft to meeting invite
- OT admin
    - https://developer.chrome.com/origintrials/#/view_trial/-5376172055173529599
    - Ending M97, Jan 26 2022.

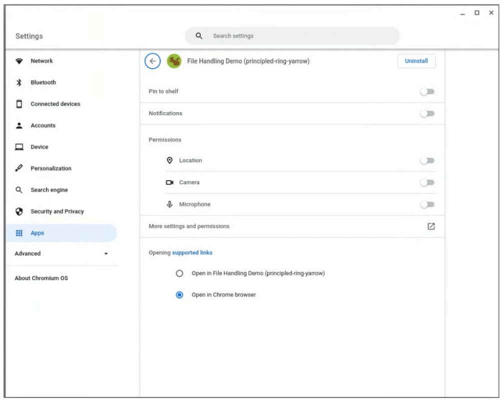
    - Prep. an intent to extend, to extend OT past M97/Jan 22.
- New links (design docs WIP):
    - dPWA Scope Extensions Project Document (Public) - Google Docs
    - PWA - `scope_extensions` design - Google Docs
    - dPWA Handle Links Project Document (Public) - Google Docs
    - PWA - `handle_links` design - Google Docs
- Alan Cutter What are we happy to include in the MVP for scope_extensions?
    - Aiming for what's currently in the explainer.
    - exclude_paths not supported on some platforms - e.g. ChromeOS
    - Can we ensure all path syntaxes are "extensible"? The scope_extensions field and origin association file formats.
    - Allow a format that caters to basic platform capabilities e.g. only able to support include_paths.
    - Is exclude_paths a blocker for initial launch?
        - Don't expect so. Will double check. (Lu)
- Alan Cutter Is handle_links: none necessary?
    - CrOS product isn't comfortable with dev opting out link capturing for their entire scope.
    - Two different user preference models at play:
        - User preference overriding manifest.
        - Manifest gating whether there's a user preference at all.
            - In this scenario handle_links: none have more meaning.
    - Discussions about handle_links extensions to exclude paths
        - We would prefer to exclude from the actual scope instead.
        - Let's pursue this in spec before extending handling_links to be able to exclude.
    - Scope exclusions: as scope extensions already requires auditing of all the places that check scope, this would not require more work
        - Potential inline specifications, but would be another proposal
- Settings UI plan (URL Handling API)
    - No user preference at the moment because code for the settings UI was not checked in.
    - Necessary for sticky preferences, users are being prompted every time currently.
    - Massive web UI refactor happening for web app settings. Delayed the implementation for URL Handler settings UI.

- Some settings landed for link capturing on CrOS



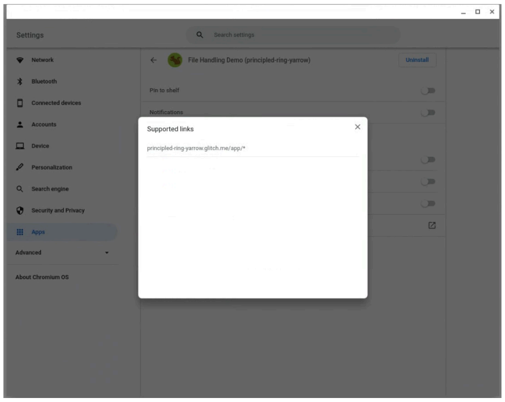
Pin to shelf: on
Notifications: off
Permissions
Location: off
Camera: off
Microphone: off
More settings and permissions

Opening supported links
● Open in Gmail
○ Open in Chrome browser



Pin to shelf: off
Notifications: off
Permissions
Location: off
Camera: off
Microphone: off
More settings and permissions

Opening supported links
○ Open in File Handling Demo (principled-ring-yarrow)
● Open in Chrome browser



Supported links
principled-ring-yarrow.glitch.me/app/*

- Main concern: user settings land and it's changed again - we should look ahead and aim to land the final UI
- Current OS settings are displayed as a list (e.g. X link's default handler is Y app), but that doesn't necessarily translate to per app settings
- AI(estade): share link capturing UX slides?
  - No but existing functionality which is available on Chrome OS is ~equivalent to future plans

Oct 14, 2021

Attendees: Lu Huang (luhua@microsoft.com), Alan Cutter (alancutter@chromium.org), Mandy Chen (mandy.chen@microsoft.com), Mike Jackson (mjackson@microsoft.com)

- Invite Evan for a meeting
- WML link capturing user settings
  - What will the UI be?

        - Current direction seems to be to reuse CrOS app management settings UI.
    - User preference model
        - CrOS has one in use already for web apps (shipping M96): https://source.chromium.org/chromium/chromium/src/+/master:components/services/app_service/public/cpp/preferred_apps_converter.h
        - https://source.chromium.org/chromium/chromium/src/+/master:components/services/app_service/public/mojom/types.mojom?q=enum%5C%20ConditionType
            - Lu should eval this for link handling settings requirements and report back.
        - Reusing the CrOS settings UI will make it easier to reuse this permission data/storage model.
- URL Handling Settings UX
    - Standalone page that lists and allows editing of user chosen default handlers
- App Settings over site permissions
    - https://bugs.chromium.org/p/chromium/issues/detail?id=1255194
    - App settings does not have infrastructure for policy control or UI widgets (site permissions does).
    - dpwas: Implement WebAppSettings policy (Ieee30c68) · Gerrit Code Review (googlesource.com)
    -

Sept 30, 2021

Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com), Alan Cutter (alancutter@chromium.org)

- Any feedback from Chrome OT that can be shared?
    - AI(mek): Check with origin trial team for current sign ups
- Any new crbugs? Query
    - 1225380 - [Mac] No way to determine which profile to use when launching a PWA - chromium fyi
- Explainer published for `handle_links` manifest member
- I2P posted for `scope_extensions` and `handle_links`
    - Mandy and Lu start working on design docs
- App service intent handling preference storage: https://source.chromium.org/chromium/chromium/src/+/master:components/services/app_service/public/cpp/preferred_apps_converter.h
    - Lu: see if we are able to reuse this code
- Scope extensions explainer: accept multiple scope values to allow configuration for different platforms · Issue #37 · WICG/manifest-incubations (github.com)
- JSON format compat. discussion resulting in some changes to the format we want to propose.

- Current `scope_extension` in explainer:
    - ~~"scope_extensions": [~~
    - ~~{"origin": "*.example.com"},~~
    - ~~{"origin": "example.co.uk"},~~
    - ~~{"origin": "*.example.co.uk"}~~
    - ~~]~~
- ~~“Scope_extensions” : [[...], [...]]   // not possible~~
- “Scope_extensions” : { “origins” : [ ] } 👍
- “Scope_extensions” : [{ “origins” : [ ] }] 👍
- ~~"handle_links: {~~
- ~~"mode" : "auto" | "all" | "none"~~
- ~~} "~~
- “Handle_links” : “all” 👍
- “Handle_links” : [👍
        { mode: "matching: “patterns”, include_patterns: [...], exclude_patterns: [...] },
        { mode: “patterns”, include_patterns: [...] },
        { mode: “paths”, include_paths: [...], exclude_paths: [...] },
        { mode: “paths”, include_paths: [...] },
        { mode: “none” }
    ]
- Green options need to be implemented now instead of later for compatibility.
- AI(lu): Create github issues/PRs with the above format changes
- Launch handler way of handling enum list:
    - https://github.com/WICG/sw-launch/blob/main/launch_handler.md#launch_handler-manifest-member
    - ManifestParser::ParseFirstValidEnum()
    - Both `route_to` and `navigate_client` also accept a list of values, the first valid value will be used. This is to allow new values to be added to the spec without breaking backwards compatibility with old implementations by using them.
    - For example if `"matching-url-client"` were added sites would specify `"route_to": ["matching-url-client", "existing-client"]` to continue to control the behaviour of older browsers that didn't support `"matching-url-client"`.

Sept 16, 2021

Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com), Mandy Chen (mandy.chen@microsoft.com), Alan Cutter (alancutter@chromium.org)

- `url_handlers` origin trial is now active
- Meeting logistics

    - Three general areas worth discussing:
        - settings ui related things - should involve estade@
        - scope_extensions and spec side - should involve alancutter@
        - origin trial/launching stuff - no activity currently
    - Might not both need to be the same meeting/frequency, ad hoc meetings should be fine for settings UI
    - scope_extensions champ TBD
- Settings UI
- scope_extensions
    - https://www.chromestatus.com/feature/5746537956114432
    - https://discourse.wicg.io/t/proposal-web-app-scope-extensions/5584
    - I2P to be sent soon
- handle_links
    - Explainer not public yet
    - No changes since last time
- Excluding links
    - Exclude from scope? (*)
        - Hard to express in manifest API in a way that's consistent with scope_extensions cross origin excluding.
    - Exclude from link handling?
        - Easy to express in the API e.g. handle_links: {exclude_paths: [“./exclude”]}
        - Probably hard to execute in implementation, set intersection with scope?
    - Do developers just want to be able to install from outside their scope?
        - Currently they can install from anywhere on their origin.
        - There could be better API support for site controlled install initiation e.g. installing web apps from other origins, creating a web app store front.
    - Can we do nothing?
        - Maybe sites can just architect their paths to match the current scope behaviour.
- Manifest scope field could be either a value or a list
    - This allows format improvements over time without hurting compatibility
        - Developers can specialize the configuration for different platforms to take advantage of improvements while providing fallback configuration if they are not available
    - This is a mechanism that is useful elsewhere as well

Sept 9, 2021

Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com), Mandy Chen (mandy.chen@microsoft.com), Alan Cutter (alancutter@chromium.org)

- M94 OT approved. All bits flipped to positive
- Waiting for OT enable request to be processed

- User settings
    - There were some questions from before about the right level of control for users
    - Should there be sticky user settings when `url_handlers` will be replaced
    - Set up a meeting with Evan to review existing URL handlers user setting design
        - App settings tracking bug
        - Collect questions and feedback, then manage in a doc
        - Can be an on-going meeting until there is a firm plan to ship settings UI
    - New UI needs to go through the launch process again
- Handle_links manifest field proposal
    - Not enough evidence for a need for include/exclude paths in the presence of scope_extensions yet.
    - Will aim for just auto/all/none for first launch.
    - Considering JSON format wrt forward compatibility.
        - handle_links: [{include_paths: [...]}, 'all']
        - handle_links := <handle_links_item> | [<handle_links_item>]
        - handle_links_item : <enum string> | {include_paths: [<path>], … }
- Scope_extensions
    - Omit_navigation_warning_ui
        - Security review might find that the app itself needs to give explicit permission to remove warning ui when top level content is off the main origin. Let's include explicit permission only if necessary.
        - The spoofing can go both ways e.g. if the user is commonly on the other origin the main origin could spoof it one day.
            - We will probably need to indicate the current origin whenever it changes. Ideally in a non-intrusive way
                - Text in titlebar is problematic because of title overlay
                - Bubble over "..." menu might be the solution
                - Let's include this only if necessary.
- WebAppBrowserController::IsUrlInAppScope() needs fixing to check the extended scope.
    - This will fix the CURRENT_TAB disposition hack in WebAppLaunchManager.
    - This will also fix middle-click link navigations that traverse the extended scope in tabbed windows, at the moment they will just open in the browser window.

Sept 2, 2021

Attendees: Lu Huang (luhua@microsoft.com), Alan Cutter (alancutter@chromium.org)

- capture_links (later renamed to handle_links)
    - auto (default behavior), all, none
    - Can be extended with paths in the future if there is demand from developers
    -

Aug 26, 2021

Attendees: Lu Huang (luhua@microsoft.com), Mandy Chen (mandy.chen@microsoft.com)

- Status of security review
- A11Y fix verified in M95
  - TODO: Merge fix to M94 (luhua@microsoft.com)
- Adjourned (no quorum today)

Aug 19, 2021

Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com), Mandy Chen (mandy.chen@microsoft.com), Alan Cutter (alancutter@chromium.org)

- Check with Security if they can give an estimate for when they're able to review (mek@chromium.org)
- Test signed off
- UX's issue with [app picker dialog not being anchored to a browser window] is not resolved yet (Lu: now resolved ✅)
- Accessibility fix ready, CL needs to be reviewed and merged back to M94
  - Target OT now starts in 94 and go until 97 (4 milestones)
  - Ping intent to experiment thread about delay (luhua@microsoft.com, ✅)
- Settings
  - Evan wants us to not add user settings since URL Handling will be replaced by Scope Extensions
  - Main concern: how complex should the settings be?
  - Lu to propose a new settings design (luhua@microsoft.com, ❓)
  - Interested in implementing settings without 4 to 8 milestones of OT to persist user choices
  - Should decide how temporary OT implementation can be, since now settings design is blocked by future feature plan of scope extensions
    - Potentially wipe user perfs with a new implementation - PM decision
    - Starting from a simpler model can allow additional changes
- "scope_extensions"
  - Not much progress in the past week
  - GitHub issue: Avoid adding web to names
  - Should let partners know that we intend for the URL handling behavior to be available continuously as we try to transition from "url_handlers" to "scope_extensions". (Ideally with an overlapping period so "url_handlers" can be deprecated in usage)
    - Redirect to scope extensions explainer to that of url handlers'
  - URLPatterns can be nice
    - Origin based settings can be a lot simpler - maybe don't present paths to users, like Windows Apps for Websites settings
    - https://github.com/WICG/sw-launch/issues/30
  - Unique app ID instead of manifest URL

    - CCT:
        - Consider “hide-CCT” being an opt-in origin provided permission.
        - Is there any case where an origin is captured by an app but shouldn’t be within the scope of that app?
            - With scope extensions, if an app can directly navigate to a link, it should be within app scope and not show the CCT.

Aug 12, 2021

Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com), Alan Cutter (alancutter@chromium.org)

- Mandy has changes to fix the a11y bug in the app picker
    - Lu is testing the changes and creating CL as Mandy is OOF
    - Since 8/12 is branch date, cherry-picker change back to 93 might be challenging
- Test plan created
    - Should be getting sign-off soon
- UX reviewer has questions about app picker (pending)
    - Lu will get in touch and resolve any issues
- Security reviewer has some questions (pending)
    - alancutter@, dmurph@, dominickng@ can speak to implementation details
- Future work
    - Alan will sync with Diego (luigonza@microsoft.com)
    - The scope_extensions explainer has a section towards the end that depends on DLC. Could this depend on the browser having “link handling” behavior instead? Will open issue in repo.
    - https://github.com/WICG/pwa-url-handler/blob/main/explainer.md#web-app-origin-association-file
    - https://github.com/WICG/manifest-incubations/blob/gh-pages/scope_extensions-explainer.md#proposal
    - Scope_extensions uses an object as its web_apps value while URL Handlers uses a list of objects.
    - Scope_extensions should use a list instead and sites should provide both URL Handler and scope_extensions objects. The respective parsers should ignore any objects they don’t understand.
- Discussion of “deep linking” concept
    - “Deep linking” can be a part of the browser’s “link handling” behavior.
    - No need for standardization.
    - All URLs within app scope are eligible for “deep linking” because they are implicitly verified.
    - App scope can be adjusted now or in the future with more expressive syntax (for eg. with URLPattern) and whatever the app scope is, deep linking will respect.
    - It’s not allowed to make inclusions/exclusions any other way.

    - If “scope_extensions” is used, browser implementation may want to disable deep linking.

Aug 5, 2021

Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com), Mandy Chen (mandy.chen@microsoft.com), Alan Cutter (alancutter@chromium.org)

- Launch bug status:
    - Security still pending, no reply in over a week. mek@ pinged the bug again, last update was some questions around revalidation frequence that were answered
    - Accessibility:
      “*I was able to get this to work on Windows. I found the same issue on Windows that I found on Mac. I also found that the list buttons lacked a state (I.e. selected/unselected).*
      This needs to be remediated for me to approve this launch.”
        - **High priority**, fix in main first and figure out if we can get it back-ported
    - Test: no updates on the launch bug, but work on writing test plan is in progress, no blockers yet.
    - UX review: requested last week
- Settings
    - Original plan to ship without settings UI in M93, and then implement some UI in M94. However large thread/discussions around what the UI should or should not look like have happened. So unclear what UI should now be implemented. M94 branches next week, so it doesn't seem likely that the settings UI discussion will reach resolution by then.
    - Not a high business priority as of right now - haven’t had any partners request this.
- Ask for OT extension now if necessary.
    - Settings, scope extensions, etc. are on the radar
    - Per https://sites.google.com/a/chromium.org/dev/blink/origin-trials/running-an-origin-trial#TOC-How-long-do-Origin-Trials-typically-last-, origin trials are typically 4 milestones, with a max of 9 milestones without good reasons to go past that.
- Scope extensions
    - Settings required? Need security and privacy review to determine
    - Right now similar to URL handling without association
- Launch handlers: existing client behavior
- Diego will start helping out as PM from the MS side, but tricky with timezones
- We should try to move email discussions to documents and Slack channel (#pwas-link-handling) instead. Easy to get lost in large email threads.
- **AI: figure out timeline for OT, official launch, scope extension, etc.**

Jul 29, 2021

- Cancelled
- Test, A11y, Security reviews still in progress; mek@ will kick off UX review as well

Jul 22, 2021

Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com), Mandy Chen (mandy.chen@microsoft.com)

- I2E approved
- Looking through Test and Accessibility questionnaires.
  - Base all answers on M93
- AI(luhua): will link a doc that has steps, descriptions, screenshots for testing M93 build.
  - Capture any differences between Mac/Win/Linux.
  - (Done.) Web App URL Handling UX Description - M93 - Google Docs
- Request OT registration
  - https://www.chromium.org/blink/origin-trials/running-an-origin-trial
  - AI(mek): Request origin trial to be setup
- Launch bug needs remaining sign off from
  - Security (by Beta 93)
  - A11Y, Test, UX, Enterprise, Leadership (after all other approvals) (By Stable 93)
- Recent email threads confirm that DLC and URL Handlers concepts are being reorganized into Scope Extensions, Launch Behaviors, and Link Capturing (first 2 are spec concerns)
  - Scope Extensions and Launch Behaviors specs can be drafted in Manifest Spec Incubations repo)
  - Link capturing is not a spec concern. Can collaborate in chromium docs.
- Handler permissions discussion
  - Some current URL Handling settings are maybe too complicated than needed, but we won't know without OT.
- Remaining WebUI work target M94
  - Go ahead with CLs until decisions around settings/permissions change
  - Hide behind feature flag in M94 until reviews give permission to remove flag check
  - Settings UI visibility will not depend on runtime feature flag (OT token)

Jul 1, 2021

Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com), Mandy Chen (mandy.chen@microsoft.com)

- CL for site settings webui backend up for review
- Is the settings page required for OT?
  - Protocol Handling is going to OT without it.
  - There's no security & privacy concerns without settings.

    - Intent to experiment
- Meeting next Tuesday to discuss site details options and transitions
- Handlers
    - Don't do any renaming for now as long as URL Handlers is in OT
- No meetings week at Google next two weeks, skip this meeting.
- Settings discussion
    - Ask/Allow, block, manage
    - Granularity is per origin then per app, does not dive into specific paths under the origin.
    - User choice is currently path specific. Perhaps it should be less granular and just ask permission to capture for all paths under the origin.
        - This could be implemented without losing the granularity of the current implementation by simply saving the choice for all paths that apply to the app in the origin.

June 24 2021

Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com), Mandy Chen (mandy.chen@microsoft.com), Alan Cutter (alancutter@chromium.org)

- Took a survey of everything that still needs to be done. Questioning if it all can be done for M93.
- Starting the process of breaking up the settings changes into smaller CLs.
- All current bugs in the UI>Browser>WebAppInstalls>UrlHandling component are either being worked on or not blocking.
- DLC bugs that end up in WebAppInstalls>UrlHandling should be redirected to WebAppInstalls>ChromeOS
- Site permissions design revisions are being worked on - will get broken up also
- Out-of-scope URL launch in app window: dPWA: Allow URL handling feature to launch app with out-of-scope URLs (I74d9abfd) · Gerrit Code Review (googlesource.com)
    - If considered in extended scope, this new behavior is fine
    - Bridge the gap between the current impl to public docs
    - Lu will ping reviewers to make sure this is expected
- Proposal for launch_handler.
    - Lets the site customise the behaviour of OpenApplication().
- All the ways web apps are launched is too UI specific to live in W3C specs. We should keep our own Chromium "specs" (READMEs) for these behaviours that we decide for ourselves. In particular any UI surface (like notifications) that could open a browser or a web app we should state the conditions for this.
- Alan is starting to work on eventing design. Lu and Mandy can provide feedback.
    - Eventing should be available to different features
- Should preferences be available for both OS and in-browser launches? (Alan)

    - Probably yes. Eg: a user could want to disable content navigation link capturing because they want browser windows to open predictably (A11Y concern), but do expect OS link activations to open browser windows (Lu)
    - Aim for efficiency in code and UI/UX reuse. Avoid duplication for DLC preferences.
    - ChromeOS will have a different prefs store but could potentially share data model

June 17 2021

Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com), Mandy Chen (mandy.chen@microsoft.com), Alan Cutter (alancutter@chromium.org)

- UX doc feedback
    - Settings update in UX doc
    - Settings CL is open for review with WIP bits - potentially chop into multiple CLs
    - Permissions CL WIP
- Scope extensions
    - URL Patterns - potentially unify, but the current syntax is not forwards compatible
    - App ID - can be backwards compatible with manifest_url, just need app_id as well
    - Android doesn't support exclude_paths, is this okay spec wise?
        - Treat as a feature that may or may not exist on the platform.
        - exclude_paths is a convenience for not listing every individual include_path.
        - Could use computed manifest API for feature detecting this.
        - Same requirements for ChromeOS in Android context.
        - Should Chrome OS treat Chrome browser exclude_paths the same as in Android contexts?
    - What syntax for matching?
        - Same as service worker plans to use url patterns.
    - Browser can translate into the app shim, no need to be compatible with the OS
        - On Windows, the browser is required for validation
    - CCT options
        - To still have browser level CCT-like UI to go back to the app but not look like a warning.

June 10 2021

Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com), Mandy Chen (mandy.chen@microsoft.com), Alan Cutter (alancutter@chromium.org)

- about://settings/urlHandlers cl: dPWA: Implement chrome://settings/urlHandlers WebUI page (Id0b69e46) · Gerrit Code Review (googlesource.com)
    - Can be built and tested

    - Some TODOs remaining
- Site details settings page: dPWA: Add web app URL handling entry to site details settings page (I998c6481) · Gerrit Code Review (googlesource.com)
    - Wip
- New requirement added 6/7/2021: show URL handling in site details settings page
    - Make a work item to check profile prefs in matching code once profiles are loaded - site details permission could prevent apps from participating in match (luhua@)
- Add scenarios w/ description to UX doc with accompanying screenshots/gifs (luhua@)
- Legal review approved
- Separation of concerns
    - “scope_extensions”: expanding the set of origins for capture
    - “capture_links”: link capturing behaviours
    - 1.) OS vs. 2.) in-browser navigation can have separate user preferences
- In a future with ‘scope_extensions’, ‘capture_links’, and “web app link capturing” browser feature:
    - Browser can capture links to web apps as a browser feature even in the absence of ‘scope_extensions’ or ‘capture_links’
        - There is a default launch behavior (eg. always open new client)
            - Default launch behavior could be different for desktop vs. mobile platforms
        - Browser link capturing could be off by default, enabled by app-specific user settings
        - Could also work for apps without manifests
    - URL handling scope extensions depends on browser web app link capturing concept and default behavior
    - “capture_links” adds more capture behaviors to the default behavior
    - URL handling does not depend on ‘capture_links’
    - ‘capture_links’ has no user settings
    - ‘scope_extensions’ does not have user settings but could change the appearance and data of the corresponding feature
        - Eg. URL handling scope extension adds to web app link capturing user settings
    - The current URL handling implementation can be used almost directly as the browser web app link capturing implementation
        - Except it needs to support in browser navigation events also
        - And the browser web app link capturing concept needs to be formalized and documented
    - What would the URL handling and DLC specs look like if they depend on a browser feature (web app link capturing)?
        - “If the browser implements link capturing to web apps, the scope can be extended…” ?
        - “If the browser implements link capturing, the default behavior can be augmented with more behaviors…” ?

June 3 2021

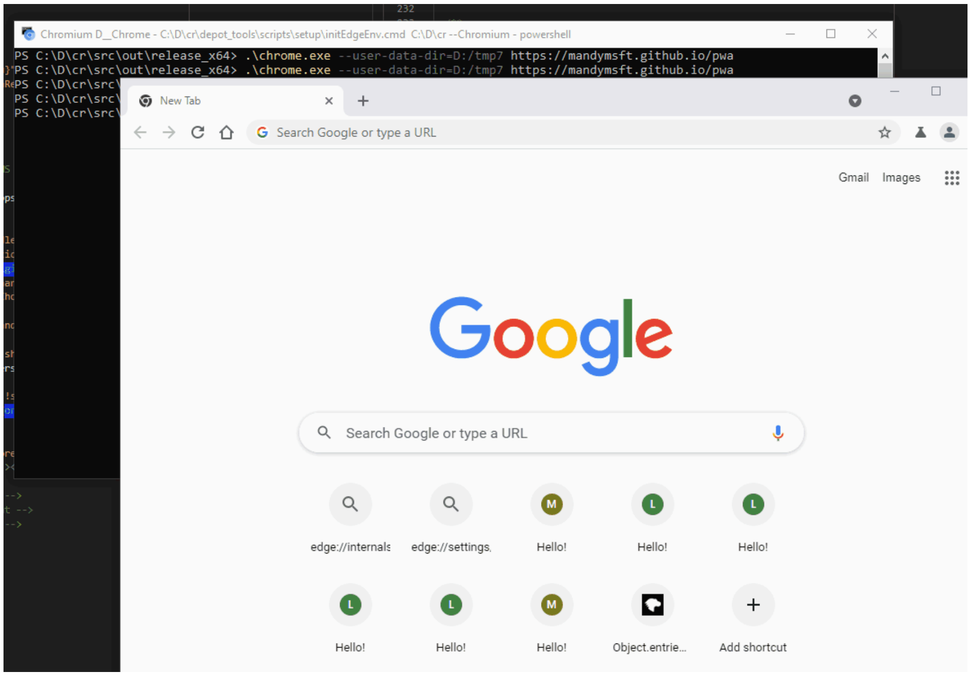
Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com), Mandy Chen (mandy.chen@microsoft.com), Alan Cutter (alancutter@chromium.org)

- More design feedback from MS side on settings menu, going back and forth on what to do with the don't handle section.
- Feedback from security review:
    - We have one request for now. We would also like to have an entry in the Site Details page for Url Handlers, with the drop down having 3 items: **Allow, Block and Manage.** (Idea by @Evan Stade from this WIP Slide Deck here). We ask this to have an easier way for the users to view and revoke their decisions, with the journey starting from within the PWA.
    - **Manage** will take the user to the chrome://setttings/urlHandlers page implemented by Lu.
    - **Block** will disallow url handling by PWAs from that origin.
    - **Allow** should be the default state. It basically represents the current status of the feature, i.e show a prompt whenever an eligible URL is handled, unless the choice was remembered from a previous prompt. In a way it works as an "Ask" state, but I think the word "Allow" is better. Especially for cases when there are some saved associations, it won't seem correct to show "Ask" even though there are some saved associations and no prompts will be shown while handling those URLs.
- July 5-16 are Chrome no meetings weeks, pause the release schedule
- web.dev article on url handling went live
- Generalization of manifest and association file
    - Client event won’t be ready for OT -- V2, will need to go through launch process again
    - Developers should use both OT for URL Handling & DLC
    - DLC for startup link capturing would be a separate codepath from the rest of DLC. DLC for in browser navigation needs to be ported to win/mac/linux, while startup capturing needs to be implemented separately. Could just do the startup capturing part to get client events for url handling working.
    - Right now you don't need DLC to need link capturing for URL handling. Might be weird to need DLC for events. URL handling exposes the user preference side, while DLC exposes the web developer side of it, letting a web developer say how they want things captured.
    - Scope extension meeting June 9th

May 27 2021

Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com), Alan Cutter (alancutter@chromium.org), Mandy Chen (mandy.chen@microsoft.com)

- Show gif of current UX implementation (luhua@)

- 

Chromium D__Chrome - C:\D\cr\depot_tools\scripts\setup\initEdgeEnv.cmd C:\D\cr --Chromium - powershell

```
PS C:\D\cr\src\out\release_x64> .\chrome.exe --user-data-dir=D:/tmp7 https://mandymsft.github.io/pwa
PS C:\D\cr\src\out\release_x64> .\chrome.exe --user-data-dir=D:/tmp7 https://mandymsft.github.io/pwa
PS C:\D\cr\src\
PS C:\D\cr\src\
PS C:\D\cr\src\
```

  - Some more TODOs but will ping UX reviewers with this and update often.
  - Will also share CL asap.
- Origin-centric settings vs app-centric settings
  - Multi-origin apps (either multi-origin in scope or in scope extension) would not work well if settings are limited to a single origin view
  - Even within a single origin, multiple apps need different settings
  - Could be important for enterprise
- scope_extension security
  - Does associating with a web app imply you trust them not to spoof you?
  - Web apps that capture links of other sites could impersonate them.
  - E.g.: NewsAggregator app pretends to be ReputableNews site and steal money through fake subscription pages.
  - Alan: Should there be a low trust tier of association that is guaranteed not to be spoofable?
  - Lu: The browser should be protecting the user from these attacks in its UI. We should continue to have *-client-event with general association.
  - Example: Hybrid apps (web + native) with url handling for other associated origins. One day the app could change owners and turn into malware.
  - Al(alancutter): Add a “permissions:” [“intercept-links”] entry to the web app association file in explainer. Without this permission *-client-events wouldn’t be allowed to provide harder protection against app-site spoofing.

May 20 2021

Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com), Alan Cutter (alancutter@chromium.org),

- Trying to get a branch/CL made that has all of the outstanding changes, to hopefully help getting approval.
- Privacy approval received, pinged security review.
- Pushing OT to M93
- Discussion about site settings
    - General case just show origin being captured with list of apps
    - What about the negative case? Ie. when a user has picked to open a particular URL in chrome rather than a PWA. Does it make sense to show apps in that case?
    - Is the choice "never open URLs with any app?", no it currently only excludes the specific set of apps that matched the app.
    - Do we need to show the list of apps that aren't picked for particular URL?
    - Show all the URLs for which the browser was picked under "never handle links from", with no further app details, analogous to how the other section doesn't show all the apps that aren't picked either.
- Had a bug where manifest update would lose all preferences, had to somehow merge manifest changes with previously saved preferences.
- Long term having a per-app view for permissions makes sense as well, although that doesn't really exist today so not blocking for launching.
    - Alan is still pushing on getting some kind of app management page on non-chrome os desktop platforms. Probably doesn't make sense to share the chrome os implementation though.
    - Would possibly have saved months of work for URL handling if something like that already existed.
- Optimizing for the multi-profile case might be premature since most users have only one profile. Although enough users have two profiles that that seems something to worry about after all.
- Scope extension stuff
    - Is capturing of links outside "self" scope a use case? Yes, capturing cross origin URLs is definitely a use case, such as short URLs.
    - How are those captured? Resulting in client events or something like that. Not necessarily existing client events but some kind of event.
    - Shortened URLs would redirect and end up in a self scope? Redirects aren't desired here, if the browser isn't already running.
    - Is there a difference between self scope and the URLs a web app might want to capture? I.e. is there a difference between URL handling and a "extend the app scope" feature. Are there URLs that you don't consider part of yourself, but should still be captured?
    - We want OS link capturing for other origins, but not necessarily have those origins be rendered in the app window. I.e. we want the event handling behavior, where the apps javascript can decide what to do with the captured URLs.
    - That difference is mostly DLC. Which lets you specify if self-scope URLs should either navigate or fire an event.

  - DLC only has existing-client-event, not new-client-event. For URL handling we might want "open start url" and fire existing-client-event in that, rather than just open the URL being handled directly.
  - Not specified/figured out yet exactly what page should be loaded for out-of-scope URLs being handled.
  - For certain URLs you might not want the URL to be opened, but instead open a router page that decides what to do. Which could be a DLC feature even if these URLs just expend the self-scope of a PWA.
  - If we can actually expand the scope of PWAs with these possibly cross-origin URLs, DLC with enough flags could be all that is needed.
  - Different apps want different things for URL handling, either want the top-level page to navigate or fire an event (which is what DLC gives).
  - Use case where you do recognize other site, and don't want CCT warning, but don't have full control over the other site. In that case getting back to the app's main origin/scope is hard. Might be enough of a boundary case not to worry about? I.e. you want some of the UI CCT gives the user without actually having the CCT UI.
  - Example: You're a news app with an agreement with news agencies to display their content. You don't have people working over there changing how they write their website, but might still have agreement to capture their URLs.
  - What about windows control overlay on those sites? Have "main" app provide control overlay, while content is the cross-origin page. Kind of radical since you'd end up with two pages being rendered at the same time. Might be a better solution for this use case? In the same class of solutions as using iframes to load the embedded content.
  - DLC could open a fixed URL, fire event and page could load real URL in an iframe. Seems less radical than trying to do something with window controls overlay.
  - TL;DR. There might be a place for scope extensions for this kind of "external content" that aren't entirely self-scope extensions. TBD what the best API for these kinds of use cases is.
  - Don't want to "force" everyone to use iframes.
  - On the other hand, if we can do "true" app scope extension that would solve most problems. We should strive for simplicity, start with just self-scope extension and only if that doesn't solve everything try to come up with more complicated things.
  - Alan will write an explainer for this kind of self-scope extension.

May 13 2021

Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com), Mandy Chen (mandy.chen@microsoft.com)

- DLC only on Chrome OS initially
- Still a couple of large changes out that would be needed for OT, including various UI bits
- OT most likely M93 rather than M92, as branch point is next week

May 6 2021

Attendees: Lu Huang (luhua@microsoft.com), Alan Cutter (alancutter@chromium.org), Mandy Chen (mandy.chen@microsoft.com)

- “url_handlers” extend scope in a specific way. Similarly, web apps could benefit from other ways of extending scope.
    - The manifest field could be generalized as “scope_extensions” with “url_handling” being one of the behaviors applied to a set of paths
    - Eg. usage: another behavior could be to allow out-of-scope content to not trigger the CCT warning UI
    - Web-app-origin-association file format could be generalized as well to serve different features (eg. getInstalledRelatedApp() supports PWAs but uses assetlinks.json for the handshake).
    - Let’s move discussion to pwa-dev sync meeting and mail group and look for support/interest there
    - Can be proposed separately from URL handling
- Discussed some use cases for:
    - Including multiple paths from same origin in a scope extension
    - Including multiple origins in a scope extension
    - Including multiple sites in a scope extension

April 29 2021

Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com), Alan Cutter (alancutter@chromium.org), Mandy Chen (mandy.chen@microsoft.com)

- Comments on privacy review
- URL Handling vs DLC
    - Fine to keep them separate for origin trial, but should figure out better integration before shipping.
    - Table of behaviours
    - Certain apps only want associations, not extending the scope. Some want to extend the scope.
    - No DLC until app intent handling system is ported to WML
    - AI(alancutter): Get notes for how to port app service intent handling to WML.

April 23 2021

Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com)

- UX Review
    - AI(mek): Sync up with dmurph@ to figure out how this relates to protocol handler launch approvals etc.
- Security & Privacy Review
    - No updates

- Is only showing apps installed in the last used profile a blocking issue?
    - AI(luhua): Add a section to UX doc with intermediate solution.
- Do we need to separate OS navigation as a browser feature?
    - https://docs.google.com/document/d/1xzu4otVNFRfF4IcPE4LLHD3FbRYnlo_h0dsMVPMfuag/edit?usp=sharing
    - Can we invite alancutter@ to next week's sync or make use of PWA dev sync?
- Discuss "existing-client-event" status next sync.

April 2 2021

Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com)

- Security+Privacy review kicked off today https://crbug.com/1173758
- Working on CLs to save users choice in intent picker
- Tested on Windows and Linux, works. But Mac seems to work differently. Fix for Mac won't be in 91.
- Manifest member UMA
    - Are there any internal dashboards/tools re metrics?

March 19 2021

Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com), Mandy Chen (mandy.chen@microsoft.com)

- UX review doc: PWA as URL Handlers UX - Google Docs
    - AI(mek): Kick off review process
- Shorter design doc WIP, ETA next week
- Basic end-to-end experience in master: https://i.imgur.com/J55sdKe.gif
- Origin Trial approval for web platforms is a public process - intent to experiment
- March 26 - Google Holiday, no meeting

March 12 2021

Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com), Mandy Chen (mandy.chen@microsoft.com)

- CLs landed, things generally should work behind a flag
- No UI yet, working on that
- Working on a CL that actually navigates to the URL
- Document event work is ongoing, ask Alan
- Site settings
    - Current UI is per-origin, no UI that is scoped per app if multiple apps exist for one origin
    - Put it in chrome://settings/handlers instead, with a subpage & display saved user preferences for default rules
    - chrome://settings/content permission settings can have wildcards (test with permission.site)

March 2 2021

Attendees: Marijn Kruisselbrink (mek@chromium.org), Lu Huang (luhua@microsoft.com), Mandy Chen (mandy.chen@microsoft.com)

- Code:
    - Most of logic in, still need to tie in to UI changes (intent picker)
    - Working on chrome settings page
    - Still need metrics
    - Dev tools? Probably good already.
- Shipping origin trial first
    - Developers will control which set of users to ship to
    - Need to migrate user data after any changes
    - OT for ~3 milestone
    - OT requirements are roughly the same as official shipping
        - For blink, OT only requires an explainer, but official shipping needs to consult other renderers
- Launch process requirements
    - Privacy (beta), security (beta), UX (stable), accessibility reviews - all are unrequested right now
        - Privacy & security: separate, shorter doc to show user-facing aspects and sum up the feature, and make sure all the doc are up-to-date
    - Tracking reviews: ask Marijn
    - Timeline unknown for UX review
- Integration with DLC
    - Check for DLC flag at browser startup. If URL in-scope, launch
    - Use in-browser navigations for our-of-scope URLs
    - Need to think over how out-of-scope URL navigation will be handled
- Timeline
    - Dev trial in 91
    - Origin trial earliest in 92

Other Meeting Notes

Sep 29, 2025 | 📅 Alexander / Dibya - Manifest Localization Chat

Attendees: Alexander Kyereboah Dan Murphy Dibyajyoti Pal

Notes

-
- Testing?
- UI surfaces?
- name / short_name stuff
- What are we 'shipping' here at least the first milestone
    - We can't change security sensitive fields w/o user confirmation.
    - full 'everything done' implementation
        - store everything offline
        - even during offline use, we can present update UX for these things to change if the user changes the system language
        - lots of work, not worth first milestone
    - first milestone - all non-security-sensitive stuff updates on language change immediately / quickly
- 📄 PWA Manifest Localization
- part 1 - Using the 'localized' manifest when populating manifest -> web app install info transformation
    - This can be done via the job Dibyajyoti Pal wrote - ManifestToWebAppInstallInfoJob
    - questions?
        -
- part 2 - dynamically... updating... the non-security stuff when system language changes
    - future - maybe introduce pending update for sec sensitive
    -
- data storage
    - do we invest more in direct mirroring of the manifest?
        - and then choose what gets stored in the base (title/name or icons fields?)
    - or other stuff
- GetTitle for web app title - gets that from the registry API
    - planning to expose the localized name
        - thus dynamically changing.
    - how to make it not dynamically changed?
    - Dan - we don't want to make that. We shouldn't provide an API here?
    - Two different states - What is in the web app database with all the localization information, and what is our actual app state based off of language
- Dan - the web app system should be the one deciding what the current app looks like - name / icon / shortcuts / etc

  - surfaces don't need localized information, it can just get the current info from the app system
  - There should be a storage separation here between manifest data and 'what is the app state right now'
- Dibya - so what about when syncing? Do we sync the french name to the other computer that is in english
  - Dan - yes
  - And then when the manifest is loaded, english mode, then it populated with english names, and manifest update occurs with UX
- Probably we should save the 'dir' member from the manifest: https://www.w3.org/TR/appmanifest/#dir-member
  - https://source.chromium.org/chromium/chromium/src/+/main:third_party/blink/public/mojom/manifest/manifest.mojom;l=38?q=manifest.mojom&ss=chromium
- Advice for project execution
  - Milestone 1 - update the manifest -> webappinstallinfo code to populate using the system language
    - Please also store the TextDirection data in the web_app.proto & add registrar method to read that - so we can use that
      - annoyingly we'll need to incorporate this in comparison / update - future
    - Also store the language used when doing manifest -> webappinstallinfo transformation
  - Milestone 2 - saving data so we can update offline - only for non-security-sensitive
    - offline: means that we can change the app dynamically if the user's language changes
    - Dan needs to help with database design here.
    - listening to system language changes - maybe just on startup?
  - Milestone 3 - having offline update of sec-sensitive
  - 2 and 3 might not be necessary - dev can just serve manifest offline, and the manifest -> install info translation should register update just fine.
    - I guess we can't do it when the app isn't launched, but with milestone 1 impl, apps can be localized fully if dev has offline support. and if not, still localized fully, the next time we are online to load a manifest.
- Launch process here?
  - easy because we already standardized
  - just blink launch process
  - Don't need origin trial - already agreed upon standard.
    - Need to document agreement / (link to the agreements we've made over the years - TPAC with addison input, manifest bug, etc) - new spec editor…
  - https://www.chromium.org/blink/launching-features/
- Artifacts
  - spec'd already: https://www.w3.org/TR/appmanifest/#x_localized-members

    - agreement across vendors already - Apple and Google
      - File a request from Firefox if necessary
  - pull request with issue links & discussion: https://github.com/w3c/manifest/issues/1087
  - Core manifest spec issue link: https://github.com/w3c/manifest/issues/676
  - tpac discussions:
    - https://docs.google.com/document/d/1RIaeXT_-j8n4iGPI3odZyJTqtHCKqe3oWZTO-j21z9U/edit?tab=t.0#heading=h.5i48hfw584ma
    - Also talked about as 'overrides'
  - TPAC 2024
    - WebApps WG TPAC 2024 Meeting (Day 1 & 2 & 3)
  - TPAC 2023
    - WebApps WG TPAC 2023 Meeting (Day 1)
    - WebApps WG TPAC 2023 Meeting (Day 2)
  - TPAC 2022
    - https://www.w3.org/2022/09/12-webapps-minutes.html
- Storing direction?
  - `enum TextDirection { TEXT_DIRECTION_UNSPECIFIED, TEXT_DIRECTION_LTR, TEXT_DIRECTION_RTL }`
  - Enum from manifest.mojom: manifest.mojom - Chromium Code Search

Action items

□ Update DD
□ Send I2P from blink launch process (consolidate artifacts to reference them)
□ Start sending CLs