MULTI-WINDOW AND YOUR APP

ANDROID 7.0 AND CHROME OS

FORMS OF MULTI-WINDOW

- Split-screen
- Picture-in-picture: mostly for Android TV
- Freeform
- Chrome OS: freeform kinda sorta but not

GETTING YOUR APP IN MULTI-WINDOW

- Generally, do nothing unusual
- Your app winds up in windows... whether you like it or not
- Older `targetSdkVersion` may give a warning `Toast` to the user

GETTING YOUR APP OUT OF MULTI-WINDOW

THE THEORY

- Use `android:resizeableActivity="false"` on `<activity>`
- Older `targetSdkVersion` and `android:screenOrientation`

GETTING YOUR APP OUT OF MULTI-WINDOW

THE REALITY

- Root activity of the task determines the window behavior
- Exported activities attempting to break out of multi-window need to ensure they run in a task other than default
- Changes in task behavior may impact other things (BACK behavior, overview screen entries)

ABOUT THE SIZING

- User can change the size of the window (e.g., drag split bar in split-screen)
- Your window might be as small as 220dp in either direction
- Bonus: if you can handle this, you can handle `small`-rated devices
- Can specify a minimum size, but then your UI is cropped

AND NOW, A WORD FROM ARS TECHNICA

WITH REGARDS TO ANDROID 7.0 MULTI-WINDOW

[Sometimes], you'll find that apps don't actually refresh themselves unless the app has focus—tap away from your messaging app to work in Word or browse in Chrome, for example, and you may stop seeing new messages until you tab back over to your messaging app.

LIFECYCLE METHODS

PAUSE FOR A MOMENT AND THINK ABOUT `onPause()`

- Paused = visible but not receiving user input
- Stopped = no longer visible
- Outside of multi-window, being only paused is unusual
- With multi-window, being only paused is very common
- Are you doing the right thing when you are paused?

CONFIGURATION CHANGES

- By default, activity may be destroyed and recreated due to multi-window
  - User enters multi-window with your activity visible
  - User resizes your window (possibly)
  - User exits multi-window with your activity focused
  - Changes occur if size deemed large enough delta, may vary by orientation
- If fast enough for orientation change, probably fast enough here
- `android:configChanges` now "out of the doghouse"

ANOTHER WORD FROM ARS TECHNICA

Sometimes you'll want to open two instances of the same app side-by-side... Whether or not you can do this depends on the app. Chrome seems to be the only app that really supports being opened twice: just enter split screen mode, press Chrome's menu button, and pick "Move to other window." Chrome then takes over both sides of the screen.

ANOTHER WORD FROM ARS TECHNICA

(CONTINUED, BECAUSE THEY ARE WORDY)

> *For other apps, the general rule seems to be "if it makes more than one thumbnail in Recent Apps, you open it twice in split screen." Just pick one thumbnail, trigger split screen, and then pick the other. This feels more like a trick than an intended feature though.*

LAUNCHING ANOTHER WINDOW

- Add `FLAG_ACTIVITY_LAUNCH_ADJACENT`, along with appropriate task flags, to `Intent`
- Ignored if device is not in multi-window mode
- "This flag is only used in split-screen multi-window mode" (???)

DETECTING MULTI-WINDOW

WHAT PROBABLY WORKS: `onMultiWindowModeChanged()`

- Override on your `Activity` or `Fragment`
- Do something useful (e.g., enable/disable "launch adjacent" action bar item)

DETECTING MULTI-WINDOW

THEN, THERE IS `isInMultiWindowMode()`

- User exits multi-window mode; your activity may be destroyed and recreated
- In `onCreate()`, `isInMultiWindowMode()` returns `true` (???) for a while (?!?) after which it returns `false`, even if you never left the main application thread (?!?!?)
- IOW, a race condition ... but one that is "working as intended"

DRAGGIN' AND DROPPIN'

- `startDrag()`, `OnDragListener`, and kin
- Ill-used, but around since API Level 11
- Works across activities, if both are visible
- Cross-app behavior
  - Opt-in to initiate a drag event that can be dropped in another app's window
  - Cannot avoid cross-app drop events, must detect "local state" and ignore if missing

LEGACY MULTI-WINDOW

THREE FLAVORS

- No code changes required (e.g., Jide's Remix OS)
- Manifest changes required (e.g., Samsung, LG)
- Proprietary SDK and custom code required (e.g., SONY)

LEGACY MULTI-WINDOW

LG

```
<meta-data
  android:name="com.lge.support.SPLIT_WINDOW"
  android:value="true" />
```

LEGACY MULTI-WINDOW

SAMSUNG

```
<uses-library
  android:name="com.sec.android.app.multiwindow"
  android:required="false" />
<meta-data
  android:name="com.sec.android.support.multiwindow"
  android:value="true" />
<category
  android:name="android.intent.category.MULTIWINDOW_LAUNCHER" />
```

HEY, WHAT ABOUT FREEFORM?

Manufacturers of larger devices can choose to enable freeform mode, in which the user can freely resize each activity. If the manufacturer enables this feature, the device offers freeform mode in addition to split-screen mode.

HEY, WHAT ABOUT FREEFORM?

PAY NO ATTENTION TO THAT MULTI-WINDOW MODE BEHIND THE CURTAIN

- Docs say that you should test on freeform
- Official Testing Approaches
  - 0 additional windows: test normally
  - 1 additional window: test in split-screen mode
  - N additional windows: ¯_(ツ)_/¯

ANDROID ON CHROME OS

WHAT WE WERE TOLD ORIGINALLY

- Developer builds would support this on three devices in June
- Rollout to ordinary users in the fall of 2016 for a long list of devices (but still a subset of total Chrome OS devices)

ANDROID ON CHROME OS

WHAT WE ARE BEING TOLD NOW

- June: Developer builds availble for the ASUS Chromebook Flip
- July: Developer builds for the Acer Chromebook R11/C738T and the 2015 edition of the Chromebook Pixel
- Rollout to ordinary users "later in 2016/2017"

CHROME OS AND YOUR APP

- By default, most apps will ship to Chrome OS devices that support touchscreens
  - Unclear what percentage of Chrome OS market has touchscreens
  - Other app features might still block availability on these Chromebooks
- By default, most apps will *not* ship to Chrome OS devices that lack touchscreens

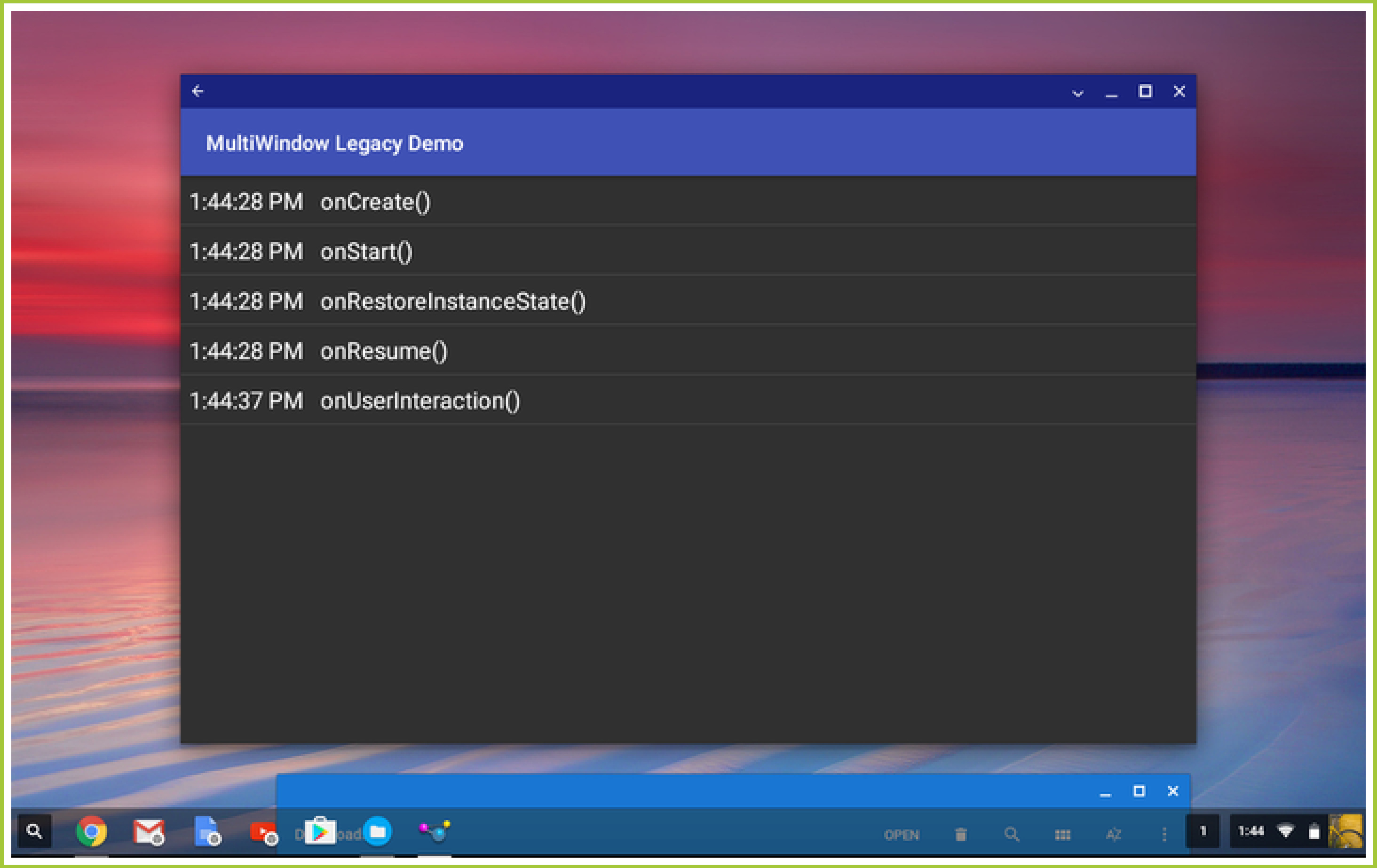
WHAT'S DIFFERENT ON CHROME OS

WINDOWS

- Landscape
- Portrait
- Full-screen
- Future: N freeform?

WHAT'S DIFFERENT ON CHROME OS

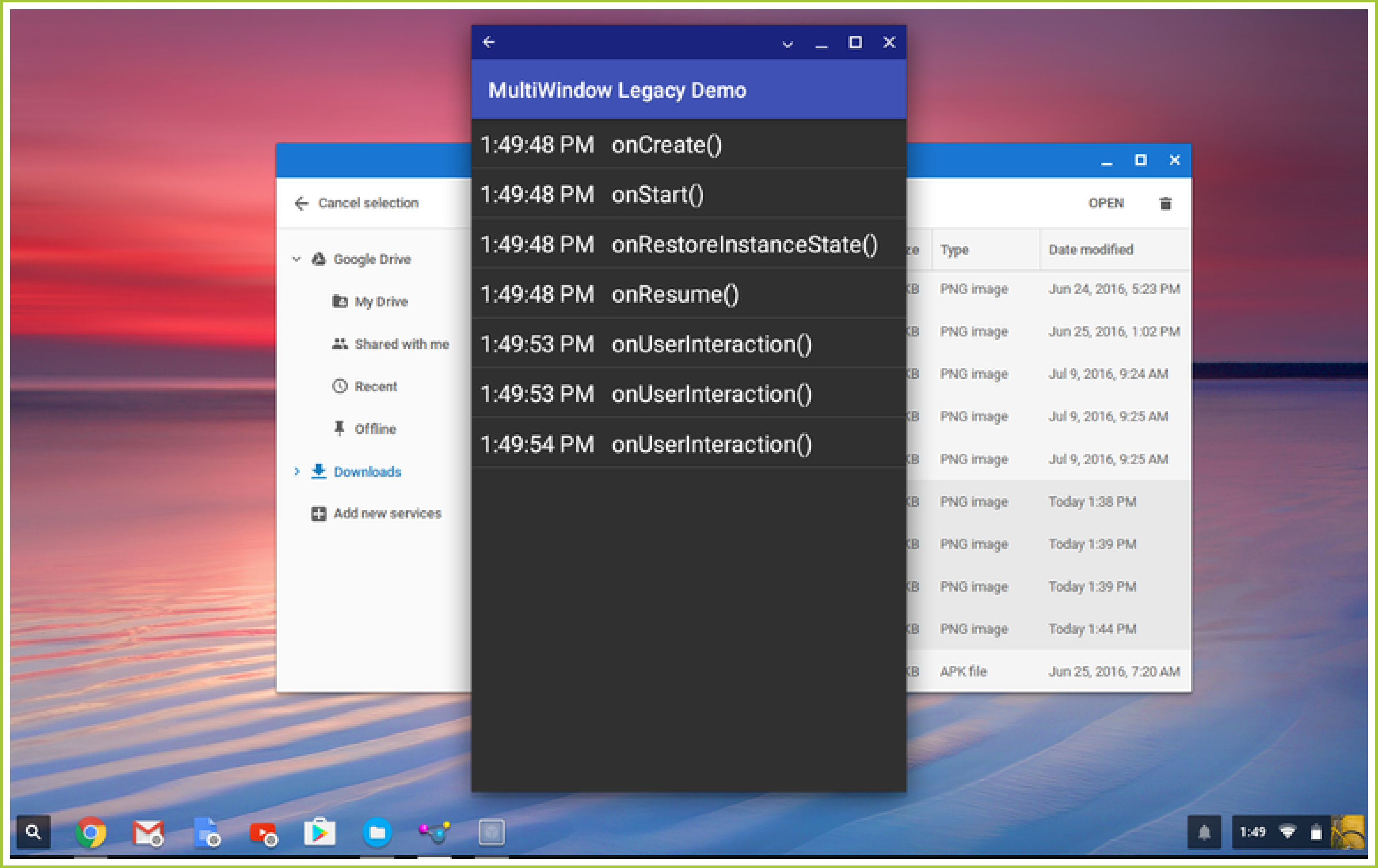
LANDSCAPE



```
1:44:28 PM   onCreate()
1:44:28 PM   onStart()
1:44:28 PM   onRestoreInstanceState()
1:44:28 PM   onResume()
1:44:37 PM   onUserInteraction()
```

WHAT'S DIFFERENT ON CHROME OS

PORTRAIT



MultiWindow Legacy Demo

```
1:49:48 PM   onCreate()
1:49:48 PM   onStart()
1:49:48 PM   onRestoreInstanceState()
1:49:48 PM   onResume()
1:49:53 PM   onUserInteraction()
1:49:53 PM   onUserInteraction()
1:49:54 PM   onUserInteraction()
```

Type	Date modified
PNG image	Jun 24, 2016, 5:23 PM
PNG image	Jun 25, 2016, 1:02 PM
PNG image	Jul 9, 2016, 9:24 AM
PNG image	Jul 9, 2016, 9:25 AM
PNG image	Jul 9, 2016, 9:25 AM
PNG image	Today 1:38 PM
PNG image	Today 1:39 PM
PNG image	Today 1:39 PM
PNG image	Today 1:44 PM
APK file	Jun 25, 2016, 7:20 AM

WHAT'S DIFFERENT ON CHROME OS

STUFF CHROME OS DOESN'T LIKE

- `android.software.input_methods`: for implementing your own IME
- `android.software.app_widget`: for apps whose primary purpose is to publish an app widget
- `android.software.live_wallpaper`: for apps whose primary purpose is to publish some live wallpaper
- `android.software.home_screen`: for home screen replacement apps

WHAT'S DIFFERENT ON CHROME OS

NOT ALWAYS A TOUCHSCREEN

- Many Chromebooks, all Chromeboxes/Chromebits lack touchscreens
- Need to say that touchscreen is not required in the manifest to be distributed to such devices
- To do *that*, need to adequately test your app with keyboard/mouse combination
- Do not assume a trackpad!

WHAT'S DIFFERENT ON CHROME OS

OTHER FIDDLY BITS

- Notifications: basics work, but get converted into Chrome OS-style notifications
- `Theme.Translucent`: still gets a window
- External displays: owned by Chrome OS, no `Presentation`

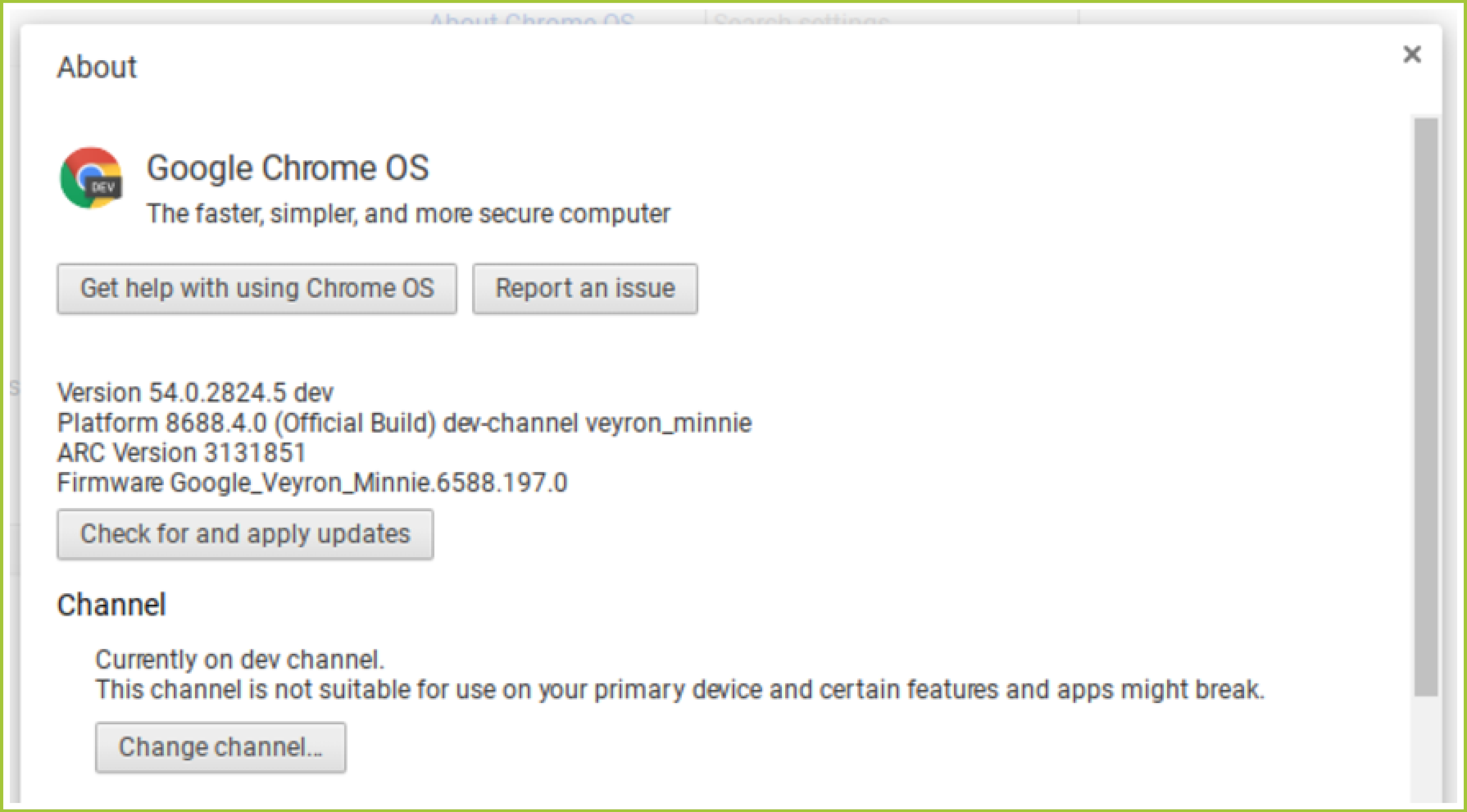
TESTING ON CHROME OS

BASIC SETUP

- Get a test device
- Switch device to the "dev channel" via Settings > About Chrome OS
- Check the Settings checkbox to enable Android apps
- Agree to Play terms of service and sign in
- Prepare for newer firmware updates to possibly wipe out your Android environment

TESTING ON CHROME OS

CHANNELS



Google Chrome OS

The faster, simpler, and more secure computer

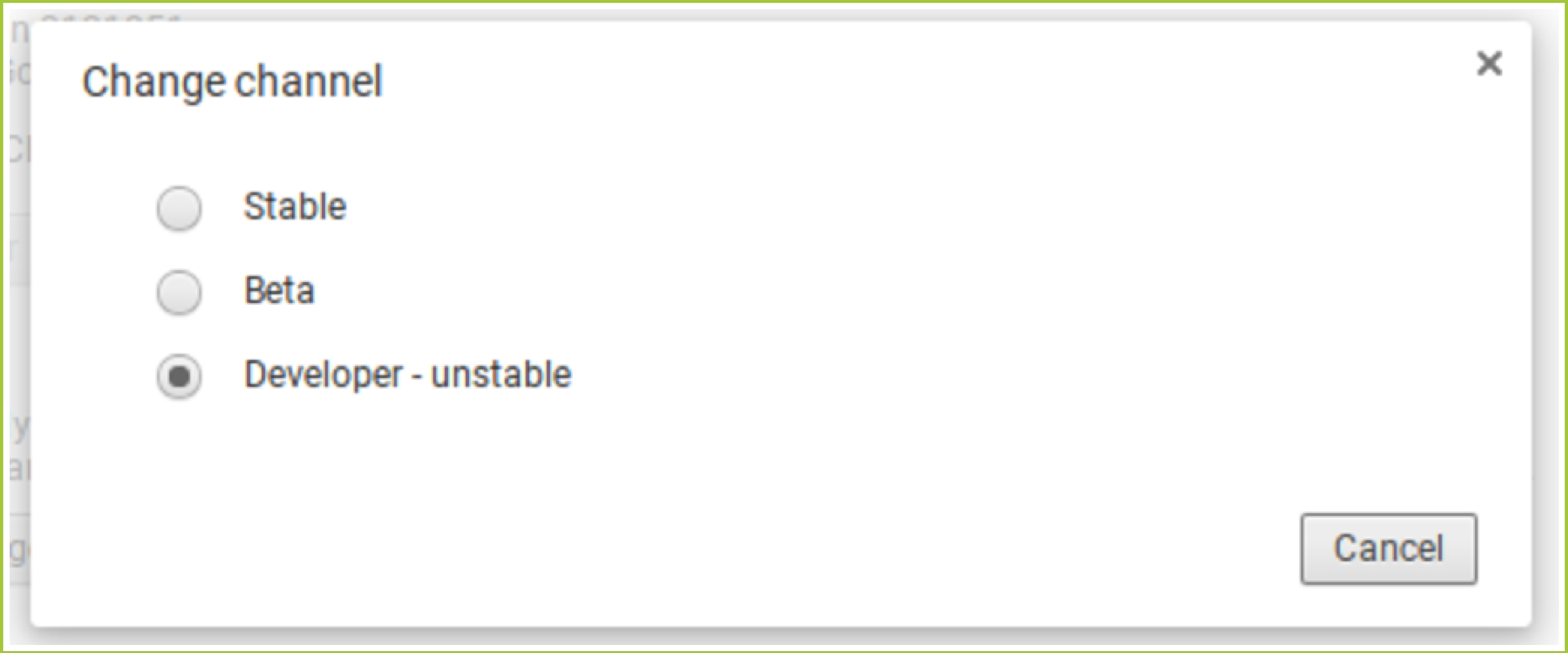
Version 54.0.2824.5 dev
Platform 8688.4.0 (Official Build) dev-channel veyron_minnie
ARC Version 3131851
Firmware Google_Veyron_Minnie.6588.197.0

Channel

Currently on dev channel.
This channel is not suitable for use on your primary device and certain features and apps might break.

TESTING ON CHROME OS

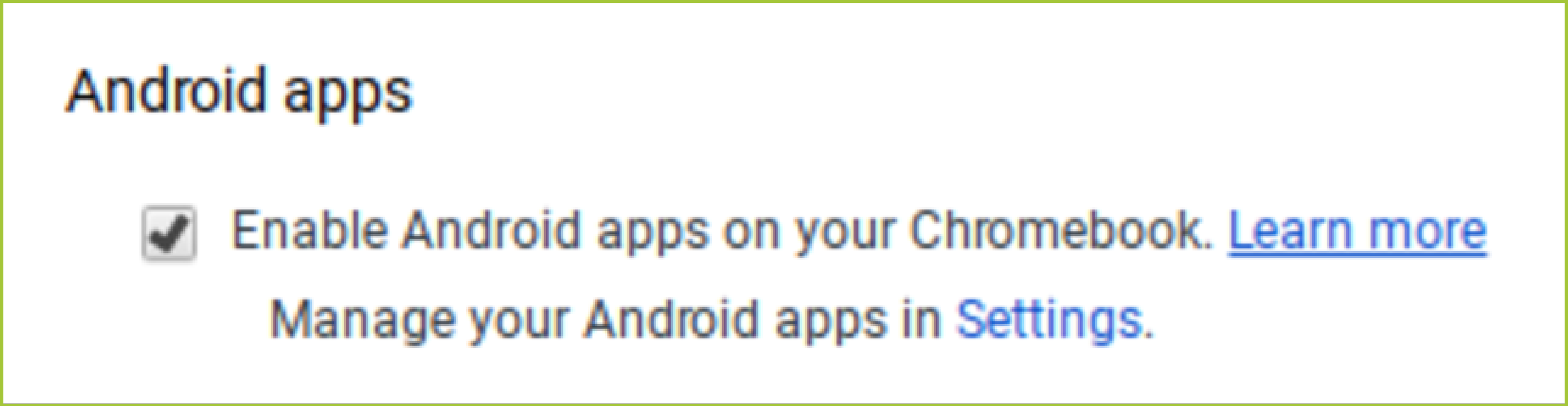
CHANGING THE CHANNEL


Change channel

Stable [✓ Developer - unstable] Beta

TESTING ON CHROME OS

I CAN HAZ DROID?



Android apps

✓ Enable Android apps on your Chromebook. Learn more

Manage your Android apps in Settings.

TESTING ON CHROME OS

ENABLING SIDE-LOADING

- Move your device into "developer mode" (details vary by device)
- Prepare to press Ctrl-D on every reboot (or wait 30 seconds)
- Enable "unknown sources" in Android's Settings app

TESTING ON CHROME OS

adb

- Seriously nasty set of instructions, affecting Chrome OS device and your development machine
- End result: use `adb connect [IP]:22` to connect to Chrome OS device, run apps from Android Studio

WHAT YOU SHOULD WORRY ABOUT NOW

ANDROID 7.0 MULTI-WINDOW

- Does your app fit within a small window?
- Does your opt-out strategy really work?
- Are you handling `onPause()` properly?
- Do you need to optimize configuration changes?

WHAT YOU SHOULD WORRY ABOUT NOW

ANDROID ON CHROME OS

- Grab a developer device, do some light testing
- Nothing much more, unless Chrome OS support is strategic
- A "run-and-jump" for niche with uncertain delivery schedule makes little sense otherwise
- Consider hacky opt-out (e.g., require `android.software.app_widget`) if seeing problems or wish to avoid lots of testing work

QUESTIONS?