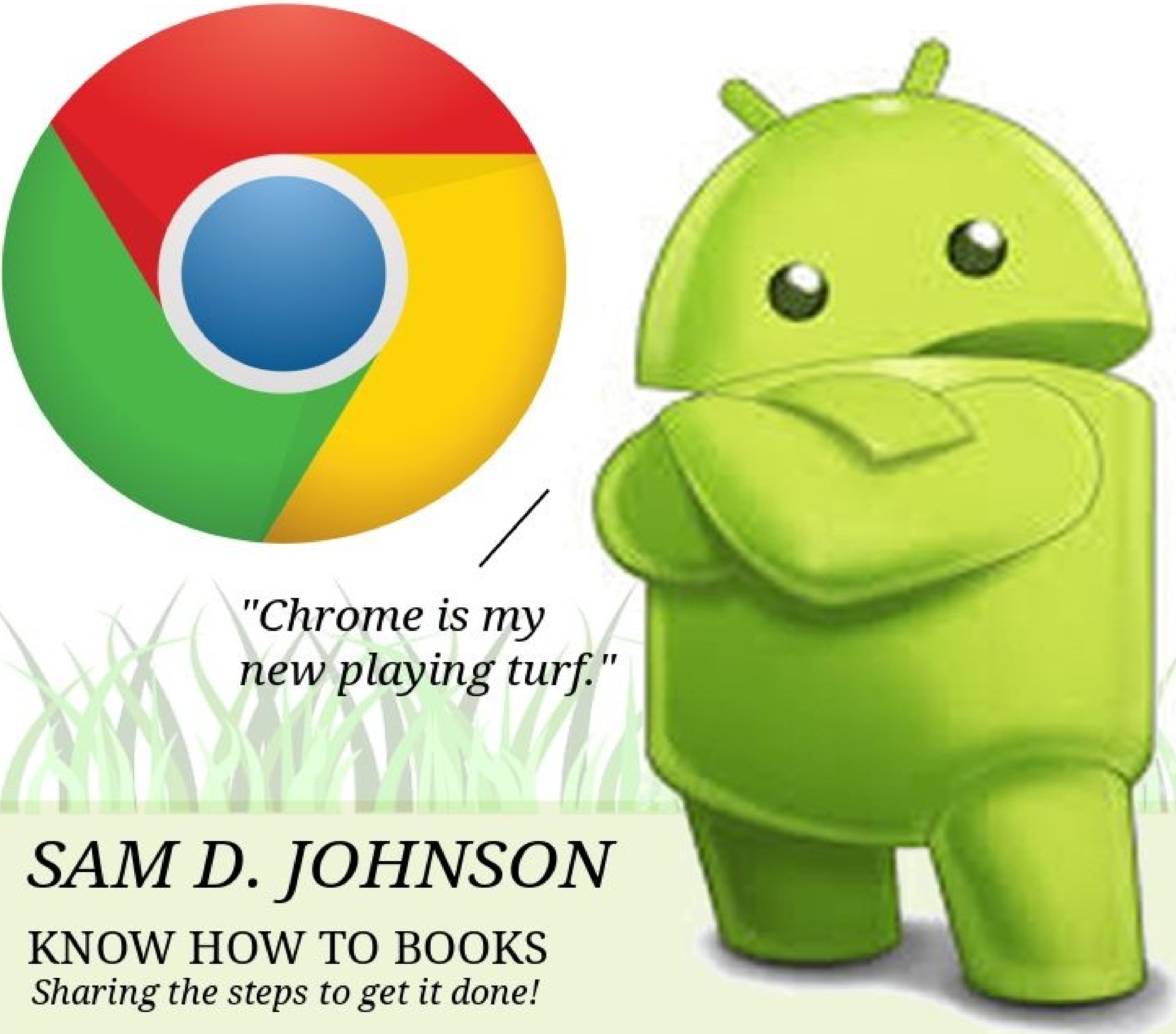
How to
Run Android™ Apps
in Chrome™ Browser

An *Easy-to-Follow* Comprehensive Step-by-Step Book



SAM D. JOHNSON

KNOW HOW TO BOOKS

Sharing the steps to get it done!

Thank You

Hi SDJ Reader!

I am excited you have found an interest in what I have to offer. So, thank you for becoming a reader of my work! My writing is about sharing what has proven to work for me by giving you a detailed and straightforward account of how I did it. There is no fluff (filler, useless, and irrelevant content) to get lost in. So, when and if you choose to follow the steps that I do, I hope you are able to successfully replicate what works for me.

All the best,

Sam D. Johnson

Author

Know How To Books

Share Feedback	Free Updates
If you have feedback you would like to share, please send any of your questions and/or comments to me by using my email link below: SDJ Reader: Feedback (*Use this title as the subject line, too, if it doesn't automatically*	If you want to be kept in the loop of my how-to works and any free updates I do, please reach out to me by using my email link below: SDJ Reader: Updates (*Use this title as the subject line, too, if it doesn't automatically*

populate.) Note: Clicking on the link should open your OS email client or your mobile will ask you which email client to use.	*populate*.) Note: Clicking on the link should open your OS email client or your mobile will ask you which email client to use.

How to Run Android™ Apps in Chrome™ Browser

An Easy-to-Follow Comprehensive Step-by-Step Guide

SAM D. JOHNSON

KNOW HOW TO BOOKS

Sharing the steps to get it done!

This publication is part of a series of products and publications.

This book is not intended to provide exact details or advice. This book merely shares a detailed account of what has proven to work for the author after spending hours of researching, testing, and troubleshooting. This book is for informational purposes only. Author reserves the right to make any changes necessary to maintain the integrity of the information held within. This book is not presented as legal or accounting advice.

author.

NOTICE OF LIABILITY. In no event shall the author or the publisher be responsible or liable for any loss of profits or other commercial or personal damages, including but not limited to special incidental, consequential, or any other damages, in connection with or arising out of furnishing, performance or use of this book. By continuing, you are agreeing to this and understand and accept the rights and notice of liability.

————-

DISCLAIMERS: NO AFFILIATION, JUST REFERENCED. © 2012 Google Inc. All rights reserved. Google and the Google logo are registered trademarks of Google, Inc., used with permission. Use of the following trademarks are subject to Google Permissions: Google Chrome, Chrome, Google Chrome Experiments, Google Chrome Extensions, Google Drive, Google Play, and Android. In addition, the Android robot is reproduced or modified from work created and shared by Google and used according to terms

A Quick History Snapshot

****

The day I was given a Pandigital Novel™ eReader was the day I became an Android user. The Pandigital eReader ran Cupcake, which just wasn’t sufficient for me. I wanted much more and since I am into repurposing all kinds of things, I searched Google™ for ways to upgrade its software instead of chucking it somewhere.

I quickly learned that I could turn the Pandigital eReader into an Android tablet. And, I got real excited - Google Play™ is loaded with awesome games and office apps! So, after finding a couple of resources I deemed as highly credible, I followed the steps (had to piece some together) that were provided and successfully converted it into an Android tablet that ran Eclair. It was awesome! Quite naturally, I wanted Jelly Bean next, but that would really be pushing it on the Pandigital. Eventually, a Jelly Bean phone was gifted to me, which I also rooted—bloatware is ridiculous—and repurposed it. I’m actually quite satisfied with Jelly Bean, but will one day upgrade it to the latest.

Meanwhile, I had also got my mind and hands into running Android natively on computers (desktop and laptop). That was pretty cool, too, to say the least. Ever since I came into all this wonderful knowledge and learned more about open-source resources (including OS), I started to strongly consider getting rid of Windows® OS more than I ever had in the past. However, I never got around to actually doing it, though, until recently.

After more than two decades of being in an arranged virtual marriage—uh, no, relationship—with Windows OS, I have to say that I do not for a second regret my decision to cut the cord. Yes, I am telling you that I have finally and totally dumped Windows for Linux® OS! No straddling the fence with dual booting; just me giving Windows two boots in the behind and moving Linux into its new empty mansion on my desktop computer, or PC.

Now, I have tried a handful of many Linux OS distros but I am beyond happy with Linux Mint™ 17. So, that is the environment in which I use to get Android apps running in Chrome.

Also, in order for me to stay truthful, I will only write about what I use to make Android apps run in Chrome browser. Fair enough? Okay, without further ado, here is the detailed account—in real time—of what has proven to work for me.

****

If you want to know more about what I stated above, just contact me to let me know by using my email links on the '*Thank You'* page. I can also put together other step-by-step book(s) detailing how I successfully repurposed each type of technology.

Table of Contents

Run, Android!: Method No. 1

Where **Twerk** Meets **Vladikoff ARChon Runtime**

Prerequisite: Folder Creation

Get Organized: Folder Creation Shortcut

1. Download the pre-packaged folder setup from Google Drive™.
2. There should only be 5 folders contained in the .zip file and nothing else (no files).

Get Organized: Manual Folder Creation

1. On the Linux Mint taskbar, I click on the default green folder icon (when I hover over it, it reads ‘Files’).
2. A ‘Home’ windows opens up.
3. I navigate to the left gray sidebar and under ‘My Computer’, I click on ‘Documents’.
4. In the now ‘Documents’ window, near the top and to the right, I click on the icon that is a folder with a + sign on it (the hover message reads ‘Create a new folder’)
5. I name the folder ‘Run Android Apps in Chrome’.
6. Double click it to open it.
7. I click on the icon that is a folder with a + sign on it to create another folder.
8. Name the folder ‘0 - Setup’.
9. Click on the icon that is a folder with a + sign on it to create another folder.
10. Name the folder ‘1 - APK Downloader Downloads’.
11. Click on the icon that is a folder with a + sign on it to create another folder.

2. Name the folder ‘2 - Twerk APK Conversions’.
3. That’s all for now, unless you want to add sub folders to folders 1 and 2 to represent the different categories of apps out there.

Requirements: Downloads

Google Chrome™ Browser for Linux

- This is the site from where I will be getting Chrome Browser: https://www.google.com/chrome/browser/

Vladikoff ARChon Runtime for Chrome

- This is the site from where I will be getting ARChon Runtime: https://github.com/vladikoff/chromeos-apk/blob/master/archon.md

Google Play Store (not a download)

- This is the site from where I will be getting APK Downloader: https://play.google.com/store/apps?hl=en

APK Downloader

- This is the site from where I will be getting APK Downloader: http://apps.evozi.com/apk-downloader/

Twerk

- This is the site from where I will be getting Twerk: https://chrome.google.com/webstore/detail/twe
hl=en

Step-By-Step: Google Chrome Browser

Download: Google Chrome Browser

1. Using Firefox browser, I go to the Google Chrome Browser download page.
2. I take notice of 'Debian/Ubuntu/Fedora/openSUSE' Linux OS distros listed under the 'Download Chrome' blue button. According to LinuxMint.com, Linux Mint is "based on Debian and Ubuntu…"
3. Click on the 'Download Chrome' blue button.
4. A 'Download Chrome for Linux' window/lightbox pops up.
5. I have a 32-bit system and a Debian/Ubuntu based distro, so I select '32 bit .deb (For Debian/Ubuntu)'.
6. I actually didn't read the 'Google Chrome Terms of Service' literature, but don't be like me and not read it.
7. Next, I click on the 'Accept and Install' blue button to download the package to my 'Downloads' folder.

Install: Google Chrome Browser

1. After it downloads, I click on the blue download arrow icon on the Firefox browser to quickly access my downloads.
2. I click on ‘google-chrome-stable_current_i386.deb’ file.
3. Package Installer opens up.
4. To install Chrome, which is currently version 39.1.2171.99, I click on the ‘Install Package’ button.
5. When done, I click on the ‘Close’ button to exit out of Package Installer.

Step-By-Step: Vladikoff ARChon Runtime

Download: Vladikoff ARChon Runtime

1. Using Chrome browser, I go to Github to download Vladikoff ARChon Runtime.
2. I scroll down a little and see a download chart section.
3. The runtime that is appropriate for my 32-bit system is the second option: ARChon 1.2 - Intel x86 Chrome 32-bit / Chrome OS 32-bit (Win1 32-bit: Use something like Chrome Beta 38.0.2125.77 beta-m).
4. So, I click on the ‘BitBucket’ dowload link.
5. Immediately the downloads bar at the bottom of Chrome pops up as the package starts downloading to my ‘Downloads’ folder.

Install: Vladikoff ARChon Runtime

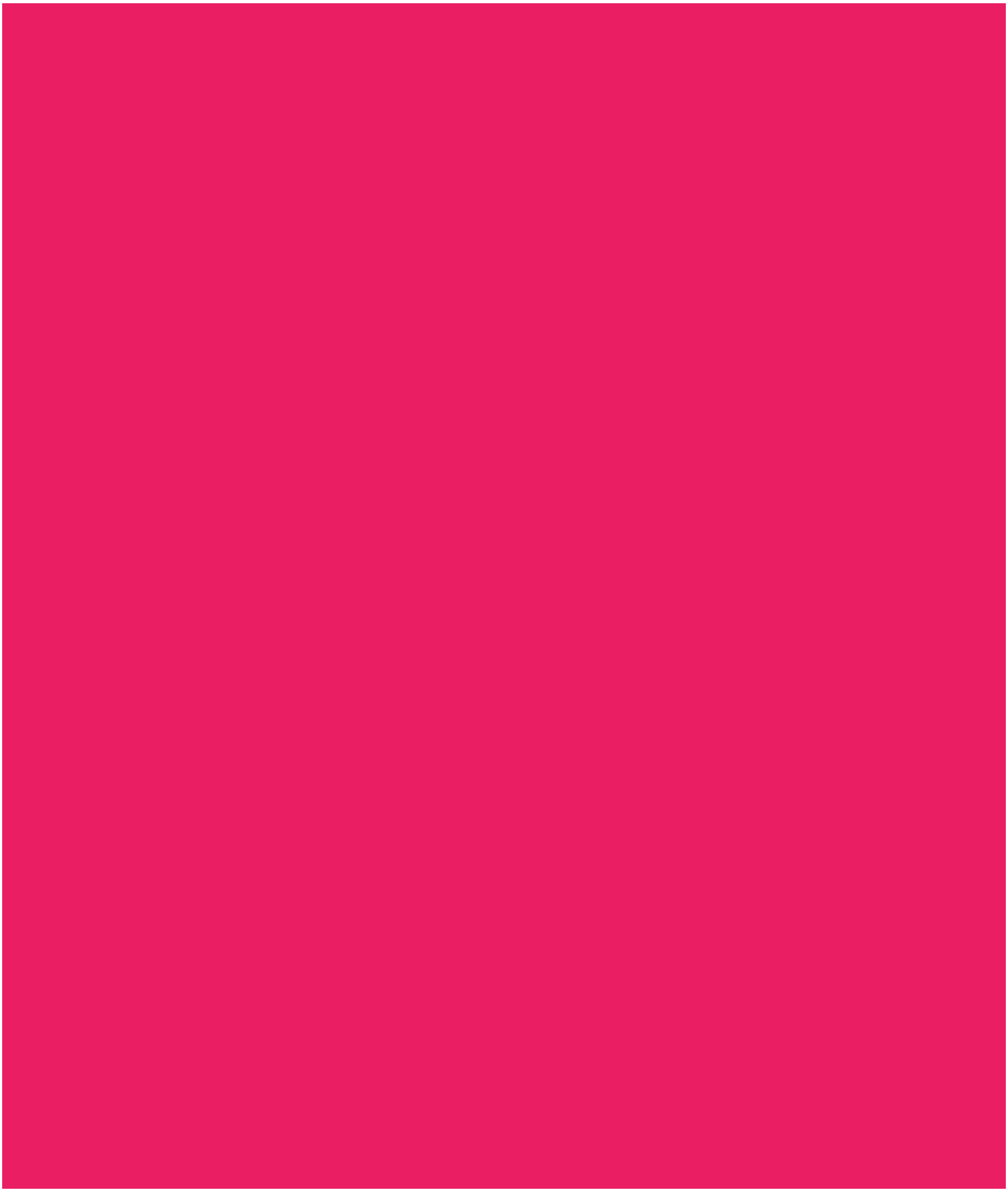
1. After it downloads, I click on the ‘vladikoff-archon-2d….zip’ file in the downloads bar at the bottom of Chrome.
2. A ‘vladikoff-archon-2d4c947b3f04.zip’ window pops open.
3. On the panel, I click on the ‘Files’ folder icon.
4. On the left sidebar, under ‘My Computer’, I click on ‘Documents’.
5. I double click on ‘Run Android Apps in Chrome’ folder.
6. From the ‘vladikoff-archon-2d4c947b3f04.zip’ window, I click and drag the folder ‘vladikoff-archon-2d4c947b3f04’ over to my ‘0 - Setup’ folder.
7. I navigate back to Chrome and type the following into the url bar: chrome://extensions.
8. Press enter.
9. In the top right corner, I ticker the box preceding ‘Developer mode’ by clicking on it.
10. Immediately, three buttons are made visible: Load unpacked extension…, Pack

extension…, and Update extensions now.

1. I click on the first button: Load unpacked extension…
2. A ‘Select the extension directory’ window pops open.
3. I click on ‘Documents’ folder.
4. I double click on ‘Run Android Apps in Chrome’ folder.
5. I also double click on ‘0 - Setup’ folder.
6. Then I click (only one time to select) on ‘vladikoff-archon-2d4c947b3f04’.
7. And then I click the ‘Open’ button in the bottom right corner of that same window.
8. I see a green circle icon (a cross between Android and Chrome logos) with ‘ARChon Custom Runtime 1.2 - x86_32 to the right of it.
9. The box preceding ‘Enabled’ is tickered by default.
10. I also can’t help but notice a big pink box with warnings, which I also ignore:

    - ‘minimum_chrome_version’ is only allowed for extensions, hosted apps,

legacy packaged apps, and packaged apps, but this is a shared module.
- Unrecognized manifest key ‘arc_build_tag’.



Step-By-Step: Twerk App for Chrome

Install: Twerk App for Chrome

1. In Chrome, I click the ‘Apps’ shortcut on the bookmarks bar.
2. Click on the Chrome Web Store app and wait for the store to load.
3. In the search field, I type: twerk.
4. Press enter.
5. Immediately I see Twerk by nerd.of.steel and click the blue ‘+ Free’ button.
6. A ‘Confirm New App’ pop-up box opens about adding Twerk.
7. In the bottom right corner, I click on the ‘Add’ button.
8. The download bar at the bottom of the browser shows the app downloading. The Chrome Web Store app box also opens and I see the pink and white Twerk logo with a greyed-out installation load bar underneath.
9. ‘An error has occurred’ pop-up box appears about the download being interrupted. Just great (not really). And, I am presented with two options to pick from: reload or close.
10. I select ‘Reload’ and all that happens is a page

refresh. Right.

1. Now I have to troubleshoot: 1) I click and drag the url to my bookmarks bar, 2) Ctrl+H, 3) click on ‘Clear browsing data…’ button at the top, 4) make sure ‘the beginning of time’ is showing in the drop down window, 5) make sure all boxes are tickered, 6) in the bottom right corner, click on ‘Clear browsing data’, and 7) close the browser.
2. Reopen Chrome.
3. From my bookmarks bar, I click on my shortcut to the Twerk search results in the Chrome Web Store.
4. Click on the blue ‘+ Free’ button again.
5. The ‘Confirm New App’ pop-up box re-opens.
6. I click on the ‘Add’ button in the bottom right corner.
7. The download bar at the bottom of the browser shows the app downloading. The Chrome Web Store app box also re-opens and I see the pink and white Twerk logo with a greyed-out installation load bar underneath, which turns blue after a few seconds.

8. The download bar disappears, the ‘Chrome Web Store - twerk’ page refreshes, and an ‘Added’ green tag is now affixed to the Twerk thumbnail image.
9. I right click on the ‘Chrome Web Store’ shortcut from my bookmarks bar.
10. Select ‘Delete’ from the menu.

Run: Twerk App for Chrome

1. From the bookmarks bar in Chrome, I click on the ‘Apps’ shortcut.
2. Click on Twerk, the pink and white Android icon.
3. A ‘Twerk’ window pops open.

Step-By-Step: Google Play Store

Find & Copy: Angry Birds™ Stella App

1. In Chrome, I go to the Google Play Store.
2. In the search field, I type in ‘Angry Birds Stella’.
3. Press enter.
4. I see ‘Angry Birds Stella’ by Rovio Entertainment in the results.
5. Right click on the title.
6. Select ‘Copy link address’.

Step-By-Step: Evozi APK Downloader

Download: Angry Birds Stella App APK

1. In Chrome, I go to the Evozi APK Downloader.
2. In the ‘Package name or Google Play URL’ field, Ctrl+V.
3. Highlight and delete everything from ‘h’ to ‘=’.
4. I am left with ‘com.rovio.angrybirdsstella’.
5. Click on the blue ‘Generate Download Link’ button and wait.
6. In a few seconds, I see the APK details (e.g. package name, file size, QR code, MD5 file hash, last fetched, and version) along with its thumbnail.
7. Right click on the thumbnail.
8. Select the ‘Save image as…’ option.
9. The ‘Save File’ window appears.
10. In the left sidebar titled ‘Places’, I click on ‘Documents’.
11. Double click on ‘Run Android Apps in Chrome’ folder.
12. Double click on ‘1 - APK Downloader Downloads’ folder.

3. In the bottom right corner, click on the ‘Save’ button.
4. The downloads bar at the bottom of Chrome appears as the thumbnail downloads. (I will come back to this later.)
5. Underneath the blue ‘Generate Download Link’ button, click on the green ‘Click here to download com.rovio.angrybirdsstella now’ button.
6. The ‘Save File’ window appears.
7. In the left sidebar titled ‘Places’, I click on ‘Documents’.
8. Double click on ‘Run Android Apps in Chrome’ folder.
9. Double click on ‘1 - APK Downloader Downloads’ folder.
0. In the bottom right corner, click on the ‘Save’ button.
1. The APK downloads and shows up in the downloads bar at the bottom of Chrome.

Step-By-Step: APK Twerk Conversion

Convert: Angry Birds Stella APK in Twerk

1. From the downloads bar at the bottom of Chrome, I click and drag the file over to the Twerk app.
2. In the ‘Application Name’ field, type: Angry Birds Stella.
3. Near the top right corner, I click on the green and white Android thumbnail.
4. The ‘Open File’ window appears.
5. In the left sidebar titled ‘Places’, I click on ‘Documents’.
6. Double click on ‘Run Android Apps in Chrome’ folder.
7. Double click on ‘1 - APK Downloader Downloads’ folder.
8. Select ‘com.rovio.angrybirdsstella.png’ file.
9. In the bottom right corner, click on the ‘Open’ button.
10. I no longer see the default green and white Android icon, but the Angry Birds Stella icon.
11. Leave ‘ARChon Runtime’ option alone.
12. Change ‘Phone’ to ‘Tablet’ (the app will display bigger) by clicking on its greyed-out

circle in the right column, which will turn blue.

3. Change ‘Portrait’ to ‘Landscape’ (on a phone or tablet, the app displays this way by default) by clicking on its greyed-out circle in the right column, which will turn blue.
4. Leave ‘Fixed Size’ option alone.
5. Leave ‘Disabled External Directory’ option alone.
6. Leave ‘Offline Enabled’ option alone, too.
7. Click on the pink and white Twerk icon at the bottom.
8. The ‘Select Folder’ window appears.
9. My ‘Run Android Apps in Chrome’ folder is already open, so I just click on ‘2 - Twerk APK Conversions’.
10. In the bottom right corner, click on the ‘Open’ button.
11. More times than not, the Twerk app will display a message about the success of the app being build. However, for the first time, it did not show up. And, that’s not a problem.
12. I simply went to my ‘2 - Twerk APK Conversions’ folder to ensure the folder

‘com.rovio.angrybirdsstella_twerk’ was there. And, it was successful.

Install: Angry Birds Stella APK by Twerk in Chrome

1. In Chrome, I type the following into the url bar: chrome://extensions.
2. Press enter.
3. I click on the first button: Load unpacked extension…
4. A ‘Select the extension directory’ window pops open.
5. I click on ‘Documents’ folder.
6. I double click on ‘Run Android Apps in Chrome’ folder.
7. I also double click on ‘2 - Twerk APK Conversions’ folder.
8. Then I click (only one time to select) on ‘com.rovio.angrybirdsstella_twerk’.
9. And then I click the ‘Open’ button in the bottom right corner of that same window.
10. The extensions are listed alphabetically. And, I see Angry Birds Stella at the top with another pink box of warnings, but this time just one, which I will also ignore: Unrecognized manifest key ‘arc_metadata’.

1. The box preceding ‘Enabled’ is tickered by default.

Play Options: New Chrome App

Time to Let Android Play in His New Playing Field

Run: Angry Birds Stella App in Chrome

Now, I have three options to run Angry Birds Stella:

1. On Google Chrome Extensions™ page (chrome://extensions), I can click on the 'Launch' link under the Angry Birds Stella app name.
2. On the Chrome bookmarks bar, I can click on the 'Apps' shortcut and click on the Angry Birds Stella app.
3. On the panel, click on 'Menu', click on 'Chrome Apps', and then click on Angry Birds Stella app.

Since I have the Google Chrome Extensions page still open, I will just choose option #1 and click the 'Launch' link under the Angry Birds Stella app name to run the app.

Results: It's Not Working!

Okay, I attempted to launch the app three times through the Chrome Extensions page to no avail. The only positive sign of it potentially working was the app window opening with the Angry Bird Stella icon inside. That is promising and encouraging. So, I will restart my computer. Usually, that works in making the apps run and smoothly in Chrome. So, I will be back.

Results: Android is Now Playing in His New Playing Field!

I am back. And, I am happy to say that my PC restarter trick worked like a charm, again! I was successfully able to play Angry Birds Stella in Chrome and without it ever crashing.

More Troubleshooting

The ‘WebGL Error’ in Google Chrome Extensions

There was a time when I received a yellow box with a ‘WebGL Error’ after uploading a Twerk converted APK of the original Angry Birds game into Google Chrome Extensions. Here’s what I did to rectify the problem:

1. In Chrome, I typed the following into the url bar: chrome://flags.
2. Pressed enter.
3. I read the red ‘WARNING’ at the top. Afterward, I saw a rather long list of ‘experiements’ which is also known as Google Chrome Experiments™. I will just copy and paste what mine looks like below, which you can mirror in your chrome://flags if you feel comfortable with the warning:

Override software rendering list Mac, Windows, Linux, Chrome OS, Android

Overrides the built-in software rendering list and enables GPU-acceleration on unsupported system configurations. #ignore-gpu-blacklist

Enable

Disable layer squashing Mac, Windows, Linux, Chrome OS, Android

Prevents the automatic combining of composited layers. #disable_layer_squashing

Enable

Enable experimental canvas features Mac, Windows, Linux, Chrome OS, Android

Enables the use of experimental canvas features which are still in development. #enable-experimental-canvas-features

Enable

Disable accelerated 2D canvas Mac, Windows, Linux, Chrome OS, Android

Disables the use of the GPU to perform 2d canvas rendering and instead uses software rendering. #disable-accelerated-2d-canvas

Enable

Enable display list 2D canvas Mac, Windows, Linux, Chrome OS, Android

Enables the use of display lists to record 2D canvas commands. This allows 2D canvas rasterization to be performed on separate thread. #enable-display-list-2d-canvas

Default ▾

Composited render layer borders Mac, Windows, Linux, Chrome OS, Android

Renders a border around composited Render Layers to help debug and study layer compositing. #composited-layer-borders

Enable

FPS counter Mac, Windows, Linux, Chrome OS, Android

Shows a page's actual frame rate, in frames per second, when hardware acceleration is active. #show-fps-counter

Enable

Disable WebGL Mac, Windows, Linux, Chrome OS, Android

Enabling this option prevents web applications from accessing the WebGL API. #disable-webgl

Enable

Compositing for RenderLayers with transitions. Mac, Windows, Linux, Chrome OS, Android

Enabling this option will make RenderLayers with a transition on opacity, transform, or filter have their own composited layer.#enable-compositing-for-transition

Default ▾

Native Client Mac, Windows, Linux, Chrome OS, Android

Enable Native Client for all web applications, even those that were not installed from the Chrome Web Store. #enable-nacl

Enable

Native Client GDB-based debugging Mac, Windows, Linux, Chrome OS

Enable GDB debug stub. This will stop a Native Client application on startup and wait for nacl-gdb (from the NaCl SDK) to attach to it. #enable-nacl-debug

Enable

Restrict Native Client GDB-based debugging by pattern Mac, Windows, Linux, Chrome OS

Restricts Native Client application GDB-based debugging by URL of manifest file. Native Client GDB-based debugging must be enabled for this option to work. #nacl-debug-mask

Debug everything except secure shell and the PNaCl translator. ▾

Experimental Extension APIs Mac, Windows, Linux, Chrome OS

Enables experimental extension APIs. Note that the extension gallery doesn’t allow you to upload extensions that use experimental APIs. #extension-apis

Enable

Extensions on chrome:// URLs Mac, Windows, Linux, Chrome OS, Android

Enables running extensions on chrome:// URLs, where extensions explicitly request this permission. #extensions-on-chrome-urls

Enable

Enable fast tab/window close Mac, Windows, Linux, Chrome OS, Android

Enables fast tab/window closing - runs a tab’s onunload js handler independently of the GUI. #enable-fast-unload

Enable

Enable ‘window-controls’ element Mac, Windows, Linux, Chrome OS

Enables using ‘window-controls’ HTML elements in packaged apps. #enable-app-window-controls

Enable

Disable hyperlink auditing Mac, Windows, Linux, Chrome OS, Android

Disable sending hyperlink auditing pings. #disable-hyperlink-auditing

Enable

Show Autofill predictions Mac, Windows, Linux, Chrome OS, Android

Annotates web forms with Autofill field type predictions as placeholder text. #show-autofill-type-predictions

Enable

Smooth Scrolling Linux

Enable the experimental smooth scrolling implementation. #enable-smooth-scrolling

Enable

Overlay Scrollbars Windows, Linux, Chrome OS

Enable the experimental overlay scrollbars implementation. You must also enable threaded compositing to have the scrollbars animate. #overlay-scrollbars

Default ▾

Enable Panels Mac, Windows, Linux, Chrome OS

Enable Panel windows that open outside of the browser frame. Attempts to open a Panel will open a popup instead if not enabled. Panels are always enabled on the dev and canary channels. #enable-panels

Enable

Save Page as MHTML Mac, Windows, Linux

Enables saving pages as MHTML: a single text file containing HTML and all sub-resources. #save-page-as-mhtml

Enable

Experimental QUIC protocol. Mac, Windows, Linux, Chrome OS, Android

Enable experimental QUIC protocol support. #enable-quic

Default ▾

Enable SPDY/4 Mac, Windows, Linux, Chrome OS, Android

Enable SPDY/4, which is the HTTP/2 standard. Currently experimental. #enable-spdy4

Enable

Built-in Asynchronous DNS Mac, Windows, Linux, Chrome OS

Enable experimental asynchronous DNS client. #enable-async-dns

Default ▾

Disable Media Source API. Mac, Windows, Linux, Chrome OS, Android

Disable the MediaSource object. This object allows JavaScript to send media data directly to a video element. #disable-media-source

Enable

Enable Encrypted Media Extensions. Mac, Windows, Linux, Chrome OS, Android

Enable Encrypted Media Extensions on video and audio elements. This enables the latest version of the Encrypted Media Extensions. #enable-encrypted-media

Enable

Disable prefixed Encrypted Media Extensions. Mac, Windows, Linux, Chrome OS, Android

Disable the experimental version of Encrypted Media Extensions on video and audio elements. #disable-prefixed-encrypted-media

Enable

Enable Experimental JavaScript Mac, Windows, Linux, Chrome OS, Android

Enable web pages to use experimental JavaScript features. #enable-javascript-harmony

Enable

Enable experimental Web Platform features. Mac, Windows, Linux, Chrome OS, Android

Enable experimental Web Platform features that are in development. #enable-experimental-web-platform-features

Enable

Disable NTP ‘Other devices’ menu. Mac, Windows, Linux, Chrome OS

Disable the new tab page menu for accessing tabs on other devices. #disable-ntp-other-sessions-menu

Enable

Enable Material Design NTP. Mac, Windows, Linux, Chrome OS

Enable the Material Design New Tab Page. #enable-material-design-ntp

Default ▾

Enable Developer Tools experiments. Mac, Windows, Linux, Chrome OS

Enable Developer Tools experiments. Use Settings panel in Developer Tools to toggle individual experiments. #enable-devtools-experiments

Enable

Enable Silent Debugging. Mac, Windows, Linux, Chrome OS

Do not show the infobar when an extension attaches to a page via chrome.debugger API. This flag is required to debug extension background pages. #silent-debugger-extension-api

Enable

Enable Automatic Spelling Correction Windows, Linux, Chrome OS

Turn on autocorrection of text while typing. Synchronous spellchecking is not compatible with this feature. #spellcheck-autocorrect

Enable

Enable scroll prediction Mac, Windows, Linux, Chrome OS

Predicts the finger’s future position during scrolls allowing time to render the frame before the finger is there. #enable-scroll-prediction

Enable

Enable touch events Mac, Windows, Linux, Chrome OS

Force touchscreen support to always be enabled or disabled, or to be enabled when a touchscreen is detected on startup (Automatic, the default). #touch-events

Automatic ▾

Disable touch adjustment. Windows, Linux, Chrome OS, Android

Disables touch adjustment support. Touch adjustment is the process of refining the position of a touch gesture in order to compensate for touches having poor resolution compared to a mouse. #disable-touch-adjustment

Enable

Enable Download Resumption Mac, Windows, Linux, Chrome OS

Allow downloads that have been interrupted to be continued or restarted, using the Resume context menu item. #enable-download-resumption

Enable

NaCl Socket API. Mac, Windows, Linux, Chrome OS

Allows applications to use NaCl Socket API. Use only to test NaCl plugins. #allow-nacl-socket-api

Enable

Disallow shelf to minimize-on-click. Mac, Windows, Linux, Chrome OS, Android

Disallow the shelf to minimize a window if a shelf item gets clicked which has only a single, already active, window associated with it. #disable-minimize-on-second-launcher-item-click

Enable

Show HUD for touch points Mac, Windows, Linux, Chrome OS, Android

Enables a heads-up display at the top-left corner of the screen that lists information about the touch-points on the screen.#show-touch-hud

Enable

Enable pinch scale. Windows, Linux, Chrome OS

Enables experimental support for scale using pinch. #enable-pinch

Default ▾

Enable pinch virtual viewport. Windows, Linux, Chrome OS, Android

When zoomed in, fixed-position elements and scaled scrollbars attach to this viewport. #enable-pinch-virtual-viewport

Default ▾

Enable viewport meta tag. Mac, Windows, Linux, Chrome OS

Enables handling of the viewport meta tag to allow pages to set the layout width and user zoom properties. #enable-viewport-meta

Enable

Debugging keyboard shortcuts Mac, Windows, Linux, Chrome OS, Android

Enables additional keyboard shortcuts that are useful for debugging Chromium. #ash-debug-shortcuts

Enable

Enable debugging for packed apps. Mac, Windows, Linux, Chrome OS

Enables debugging context menu options such as Inspect Element for packed applications. #debug-packed-apps

Enable

Enable password generation. Mac, Windows, Linux, Chrome OS, Android

Allow the user to have Chrome generate passwords when it detects account creation pages. #enable-password-generation

Default ▾

Save passwords automatically. Mac, Windows, Linux, Chrome OS

Skip the passwords prompt and save passwords automatically. #enable-automatic-password-saving

Enable

Enable remote password management link Mac, Windows, Linux, Chrome OS, Android

Show a link in the password manager settings page to manage your synced passwords online. #enable-password-link

Default ▾

Enable deferred image decoding. Mac, Linux, Chrome OS

Defer image decoding operations in WebKit until painting. #enable-deferred-image-decoding

Enable

Use Wallet sandbox servers Mac, Windows, Linux, Chrome OS, Android

For developers: use the sandbox service for Wallet API calls for requestAutocomplete(). #wallet-service-use-sandbox

Default ▾

Overscroll history navigation Mac, Windows, Linux, Chrome OS, Android

Experimental history navigation in response to horizontal overscroll. #overscroll-history-navigation

Enabled ▾

Enable Mojo based IPC channel for renderers Mac, Windows, Linux, Chrome OS, Android

Enable Mojo based IPC channel for communication between browser process and renderer processes. #enable-renderer-mojo-channel

Enable

Enable touch based text editing Windows, Linux, Chrome OS

Touch editing can be initiated by tapping on a textfield or a selected text. #enable-touch-editing

Default ▾

Enable the stale-while-revalidate cache directive Mac, Windows, Linux, Chrome OS, Android

Enable the experimental implementation of the Cache-Control: stale-while-revalidate directive. This permits servers to specify that some resources may be revalidated in the background to improve latency. #enable-stale-while-revalidate

Enable

Enable the supervised user host blacklist Mac, Windows, Linux, Chrome OS, Android

Enable the host blacklist for use by supervised users. #enable-supervised-user-blacklist

Default ▾

Enable the supervised user SafeSites filter Mac, Windows, Linux, Chrome OS, Android

Enable SafeSites filtering for supervised users. #enable-supervised-user-safesites

Default ▾

Enable Synced Notifications Mac, Windows, Linux, Chrome OS

Enable experimental Synchronized Notifications. #enable-sync-synced-notifications

Default ▾

Enable App Launcher sync Mac, Windows, Linux, Chrome OS

Enable App Launcher sync. This also enables Folders where available (non OSX). #enable-sync-app-list

Default ▾

Impl-side painting Mac, Windows, Linux, Chrome OS, Android

If enabled, painting is done on a separate thread instead of the main thread. #impl-side-painting

Default ▾

LCD text antialiasing Mac, Windows, Linux, Chrome OS

If disabled, text is rendered with grayscale antialiasing instead of LCD (subpixel) when doing accelerated compositing. #lcd-text-aa

Default ▾

Maximum tiles for interest area Mac, Windows, Linux, Chrome OS, Android

Specify the maximum tiles for interest area. #max-tiles-for-interest-area

Default ▾

Enable Offline Auto-Reload Mode Mac, Windows, Linux, Chrome OS, Android

Pages that fail to load while the browser is offline will be auto-reloaded when the browser is online again. #enable-offline-auto-reload

Default ▾

Only Auto-Reload Visible Tabs Mac, Windows, Linux, Chrome OS, Android

Pages that fail to load while the browser is offline will only be auto-reloaded if their tab is visible. #enable-offline-auto-reload-visible-only

Default ▾

Enable Offline Load Stale Button Mac, Windows, Linux, Android

When a page fails to load, if a stale copy of the page exists in the browser, a button will be presented to allow the user to load that stale copy. #enable-offline-load-stale-cache

Default ▾

Default tile width Mac, Windows, Linux, Chrome OS, Android

Specify the default tile width. #default-tile-width

Default ▾

Default tile height Mac, Windows, Linux, Chrome OS, Android

Specify the default tile height. #default-tile-height

Default ▾

Disable the Extensions Info dialog. Windows, Linux, Chrome OS

Disables the Extensions Info dialog from being launched from the chrome://extensions page (reverts to the old-style dialog).#disable-extension-info-dialog

Enable

Simple Cache for HTTP. Mac, Windows, Linux, Chrome OS

The Simple Cache for HTTP is a new cache. It relies on the filesystem for disk space allocation. #enable-simple-cache-backend

Default ▾

Enable TCP Fast Open Linux, Chrome OS, Android

Enable the option to send extra authentication information in the initial SYN packet for a previously connected client, allowing faster data send start. #enable-tcp-fast-open

Enable

Device Discovery Notifications Mac, Windows, Linux, Chrome OS

Device discovery notifications on local network. #device-discovery-notifications

Default ▾

Enable Google Cloud Devices Mac, Windows, Linux, Chrome OS

Enabled Google Cloud Devices features. #enable-cloud-devices

Enable

Enable Print Preview Registration Promos Mac, Windows, Linux, Chrome OS

Enable registering unregistered cloud printers from print preview. #enable-print-preview-register-promos

Enable

Tab capture upscaling quality. Mac, Windows, Linux, Chrome OS, Android

Specifies quality setting for images captured if scaling up. #tab-capture-upscale-quality

Default ▾

Tab capture downscaling quality. Mac, Windows, Linux, Chrome OS, Android

Specifies quality setting for images captured if scaling down. #tab-capturedownscale-quality

Default ▾

Spelling Feedback Field Trial. Mac, Windows, Linux, Chrome OS, Android

Enable the field trial for sending user feedback to spelling service. #enable-spelling-feedback-field-trial

Enable

Enable WebGL Draft Extensions Mac, Windows, Linux, Chrome OS, Android

Enabling this option allows web applications to access the WebGL Extensions that are still in draft status. #enable-webgl-draft-extensions

Enable

Enable Web MIDI API Mac, Windows, Linux, Chrome OS, Android

Enable Web MIDI API experimental support. #enable-web-midi

Enable

Enable new profile management system Mac, Windows, Linux, Chrome OS, Android

Enables the new profile management system, including profile lock and the new avatar menu UI. #enable-new-profile-management

Default ▾

Enable identity consistency between browser and cookie jar Mac, Windows, Linux, Chrome OS, Android

When enabled, the browser manages signing in and out of Google accounts. #enable-account-consistency

Default ▾

Enable fast user switching in the avatar menu Mac, Windows, Linux

Enables fast switching between users in the avatar menu. #enable-fast-user-switching

Enable

Enable the new avatar menu Mac, Windows, Linux

Enables the new avatar menu. When combined with new-profile-management, it simply shows the new profile management avatar menu. Otherwise it shows a redesigned avatar menu with the same functionality as the old one, plus a tutorial card at the top prompting the user to try out the new profile management UI. #enable-new-avatar-menu

Default ▾

Enable pure web-based sign-in flows Mac, Windows, Linux

When enabled, will use a pure web-based sign-in flow on first run/NTP/wrench menu/settings page. Otherwise use a native flow with embedded webview. #enable-web-based-signin

Enable

Enables webview-based Chrome sign-in flows. This flag overrides —enable-web-based-signin. Mac, Windows, Linux

When enabled, will use a webview-based Chrome sign-in flow. #enable-webview-based-signin

Enable

Enable Google profile name and icon Mac, Windows, Linux

Enables using Google information to populate the profile name and icon in the avatar menu. #enable-google-profile-info

Enable

Reset the App Launcher install state on every restart. Mac, Windows, Linux

Reset the App Launcher install state on every restart. While this flag is set, Chrome will forget the launcher has been installed each time it starts. This is used for testing the App Launcher install flow. #reset-app-list-install-state

Enable

Enable the App Launcher. Linux

Enable the App Launcher. Upon enabling, creates operating system shortcuts to the App Launcher. #enable-app-list

Enable

Disable the app info dialog. Windows, Linux, Chrome OS

Disable the app info dialog. If not disabled, the user will be able to select the ‘App Info’ context menu in the App Launcher to show the app info dialog for the selected app. #disable-app-list-app-info

Enable

Enable one-copy rasterizer Mac, Windows, Linux, Chrome OS, Android

If enabled, raster threads write to GPU memory. #enable-one-copy

Default

Enable zero-copy rasterizer Mac, Windows, Linux, Chrome OS, Android

If enabled, raster threads write directly to GPU memory associated with tiles. #enable-zero-copy

Enable

Enable experimental streamlined hosted apps. Windows, Linux, Chrome OS

Enables an experimental, more streamlined hosted app experience. #enable-streamlined-hosted-apps

Enable

Enable experimental ephemeral apps. Mac, Windows, Linux, Chrome OS, Android

Enables experimentation with ephemeral apps, which are launched without installing in Chrome. #enable-ephemeral-apps

Enable

Enable experimental linkable ephemeral apps. Mac, Windows, Linux, Chrome OS, Android

Enables experimentation with launching ephemeral apps from hyperlinks. For example, links to Chrome Web Store app detail pages in Google search results will launch the app rather than navigate to the detail page. #enable-linkable-ephemeral-apps

Enable

Enable support for ServiceWorker background sync event. Mac, Windows, Linux, Chrome OS, Android

ServiceWorker background synchronization lets ServiceWorkers send messages and update resources even when the page is in the background. #enable-service-worker-sync

Enable

Disable rect-based targeting in views Windows, Linux, Chrome OS

Disables rect-based targeting in views. Rect-based targeting uses a heuristic to determine the most probable target of a gesture, where the touch region is represented by a rectangle. #disable-views-rect-based-targeting

Enable

Enable show-on-first-paint for apps. Mac, Windows, Linux, Chrome OS

Show apps windows after the first paint. Windows will be shown significantly later for heavy apps loading resources synchronously but it will be insignificant for apps that load most of their resources asynchronously. #enable-apps-show-on-first-paint

Enable

Enable Enhanced Bookmarks Mac, Windows, Linux, Chrome OS, Android

Provides an off switch for enhanced bookmarks experiment #enhanced-bookmarks-experiment

Default ▾

Number of raster threads Mac, Windows, Linux, Chrome OS, Android

Specify the number of raster threads. #num-raster-threads

Default ▾

Origin chip Mac, Windows, Linux, Chrome OS

Hides the URL in the Omnibox and instead shows the hostname in a clickable chip. #origin-chip-in-omnibox

Default ▾

Enable search button in Omnibox Mac, Windows, Linux, Chrome OS

Places a search button in the Omnibox. #search-button-in-omnibox

Default ▾

Ignore autocomplete=‘off’ (Autofill) Mac, Windows, Linux, Chrome OS

Ignore autocomplete=‘off’ for forms that Chrome can Autofill with credit card or address data. #ignore-autocomplete-off-autofill

Enable

Use Bubbles for Permission Requests Mac, Windows, Linux, Chrome OS

Show content permission requests (e.g. notifications, quota, camera use, microphone use) in bubbles instead of info bars.#enable-permissions-bubbles

Default ▾

Enable session restore bubble UI. Windows, Linux

If enabled, the session restore UI will be shown in a bubble instead of an info bar. #enable-session-crashed-bubble

Default ▾

Enable out of process PDF. Mac, Windows, Linux, Chrome OS

Enable the out of process PDF plugin. #out-of-process-pdf

Enable

Disable support for Cast Streaming hardware video encoding. Mac, Windows, Linux, Chrome OS, Android

This option disables support in Cast Streaming for encoding video streams using platform hardware. #disable-cast-streaming-hw-encoding

Enable

Enable the experimental App Launcher. Windows, Linux, Chrome OS

Enable the experimental version of the App Launcher. #enable-experimental-app-list

Enable

Center the App Launcher. Windows, Linux, Chrome OS

Positions the App Launcher in the center of the screen with a landscape aspect. #enable-centered-app-list

Enable

Touch scrolling mode. Windows, Linux, Chrome OS, Android

Change the touch event behavior while scrolling. “touchcancel” is what Chrome has historically used, and “async-touchmove” is the new preferred mode. #touch-scrolling-mode

Default ▾

Disable threaded scrolling. Windows, Linux, Chrome OS, Android

Disabled threaded handling of scroll-related input events, forcing all such scroll events to be handled on the main thread. Note that this can dramatically hurt scrolling performance of most websites and is intended for testing purposes only. #disable-threaded-scrolling

Enable

Show settings in a window Mac, Windows, Linux, Chrome OS

If enabled, Settings will be shown in a dedicated window instead of as a browser tab. #enable-settings-window

Default ▾

Enable Save Password Bubble UI Mac, Windows, Linux, Chrome OS

Enable experimental bubble-based UI for saving passwords; replaces the existing infobar. #enable-save-password-bubble

Default

Allow insecure WebSocket from https origin Mac, Windows, Linux, Chrome OS, Android

This flag makes Chrome unsafe. Use this only if you understand what this does. Note that this flag may be removed without any notice. If enabled, frames with an https origin can use WebSockets with an insecure URL (ws://). #allow-insecure-websocket-from-https-origin

Enable

Enable distance field text Mac, Windows, Linux, Chrome OS, Android

If enabled, text is rendered with signed distance fields rather than bitmap alpha masks. #distance-field-text

Default

Extension Content Verification Mac, Windows, Linux, Chrome OS

This flag can be used to turn on verification that the contents of the files on disk for extensions from the webstore match what they’re expected to be. This can be used to turn on this feature if it would not otherwise have been turned on, but cannot be used to turn it off (because this setting can be tampered with by malware). #extension-content-verification

Default

Experimental text input focus manager. Windows, Linux, Chrome OS

Enable an experimental focus manager to track text input clients. #text-input-focus-manager

Default

User consent for extension scripts Mac, Windows, Linux, Chrome OS, Android

Require user consent for an extension running a script on the page, if the extension requested permission to run on all urls.#extension-active-script-permission

Enable

HarfBuzz for UI text. Mac, Windows, Linux, Chrome OS

Cross-platform HarfBuzz engine for UI text. Doesn't affect web content. #harfbuzz-rendertext

Default ▾

Enable experimental 'Ok Google' hotword detection features. Mac, Windows, Linux, Chrome OS

Enables experimental 'Ok Google' hotword detection features, such as using the built-in extension. The hotword extension from the Chrome Web Store will no longer be used. #enable-experimental-hotwording

Enable

Enable embedded extension options. Mac, Windows, Linux, Chrome OS

Display extension options as an embedded element in chrome://extensions rather than opening a new tab. #enable-embedded-extension-options

Enable

Enables the website settings manager. Mac, Windows, Linux, Chrome OS

Enables the website settings manager on the settings page. #enable-website-settings-manager

Enable

Remember decisions to proceed through SSL errors for a specified length of time. Mac, Windows, Linux, Chrome OS, Android

Remember decisions to proceed through SSL errors for a specified length of time. #remember-cert-error-decisions

Default ▾

Drop sync credentials from password manager. Mac, Windows, Linux, Chrome OS, Android

If enabled, the password manager will not offer to save the credential used to sync.

#enable-drop-sync-credential

Default ▾

Enable extension toolbar redesign. Windows, Linux, Chrome OS

Enables the (in development) new extension toolbar toolbar design. #enable-extension-action-redesign

Enable

Autofill sync credential Mac, Windows, Linux, Chrome OS, Android

How the password manager handles autofill for the sync credential. #autofill-sync-credential

Default ▾

Enable experiments that message center always scroll up upon notification removal. Mac, Windows, Linux, Chrome OS

Enables experiment that message center always scroll up when a notification is removed. #enable-message-center-always-scroll-up-upon-notification-removal

Enable

Enable tab audio muting UI control. Mac, Windows, Linux, Chrome OS

When enabled, the audio indicators in the tab strip double as tab audio mute controls. This also adds commands in the tab context menu for quickly muting multiple selected tabs. #enable-tab-audio-muting

Enable

Enable the experimental Credential Manager API. Mac, Windows, Linux, Chrome OS, Android

Enables an experimental implementation of the Credential Manager API. No promises that this won't expose all your passwords to every site on the web; don't enable this unless you know what you're doing. #enable-credential-manager-api

Enable

Enable Plugin Power Saver. Mac, Windows, Linux, Chrome OS

Enables experimental power saver mode for peripheral plugin content. #enable-plugin-power-saver

Enable

Minimum SSL/TLS version supported. Mac, Windows, Linux, Chrome OS, Android

Configure the minimum SSL/TLS version that can be used when connecting to HTTPS servers. #ssl-version-min

Default ▾

Twerk Orientation Settings

I learned that sometimes the settings in the Twerk app—before building the Twerk APK version—need to be changed. The settings to focus on is the device setting and orientation setting. So, any of the combinations below:

- Phone + Portrait
- Phone + Landscape
- Tablet + Portrait
- Tablet + Landscape

I messed around with the other Twerk app settings, but they had no impact on getting the Android app to run in Chrome. So, I'm not going to waste time telling you what combinations of settings I did there.

Method No.1 Abandonment

Abandon: Method No. 1

There are times I have to remind myself that Method No.1 is not 100% full-proof in getting Android apps to run in Chrome browser. Sometimes, Twerk conversions just refuse to work in Chrome and on my system. And, when this happens I just call it quits on troubleshooting the method any further.

Coming Soon: Alternative Method No. 2 & Method No. 3

Now that I have learned two other ways to get an Android app to run in Chrome, I rarely troubleshoot Method No.1 when I hit a snag. One of these methods involves me manually going into the code. (Not to worry, it’s more like a plug-in-play kind of setup - really simple to do). The other method involves me using another converter tool. Again, neither is 100% full-proof, but they are still worth doing when Twerk is giving me grief.

Moreover, I will be working on writing out the details of how I get the alternative methods to work for me in another comprehensive step-by-step guide. And, if you do not want to do the research like I did and problem solve like I did, then be sure to keep checking back to know when I have released the new how-tos for getting Android to play in his new Chrome playground!

Alternatively, you can reach out to me using my email links on the ‘Thank You‘ page in the beginning of this book, if you want to be notified when I’m done writing the books and when I have released them.

Thanks again for being a customer and your continued support!