The Power of Open-Source: How To Contribute To and Manage Communities

Caitlin Cabrera

About Me

- First of all thank you!
- Came here all the way from Omaha, NE.
- Software Engineer
- Writes a lot of Ruby on Rails in the information security space
- Writes a lot ABOUT Ruby on Rails and the information security space
- Probably lifting weights when I'm not writing code

Open source


KOLIDE

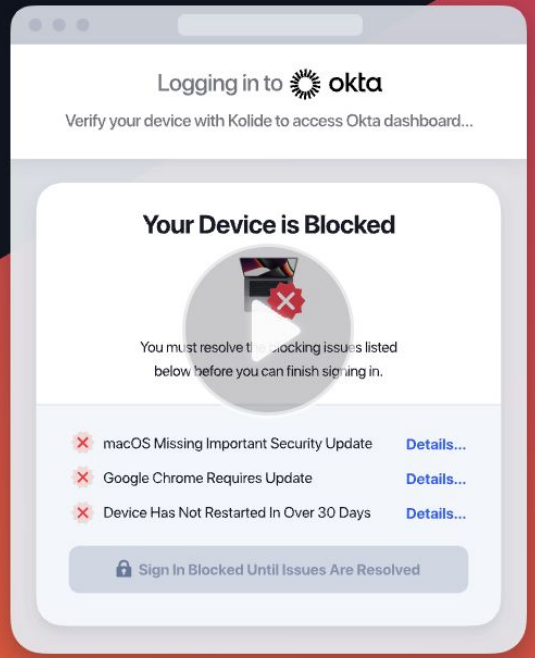
KOLIDE

Use Cases | Features | Pricing | Resources | Watch Demo | Talk To Sales | Sign In >

Kolide ensures that
if a device isn't secure, it can't access your apps.

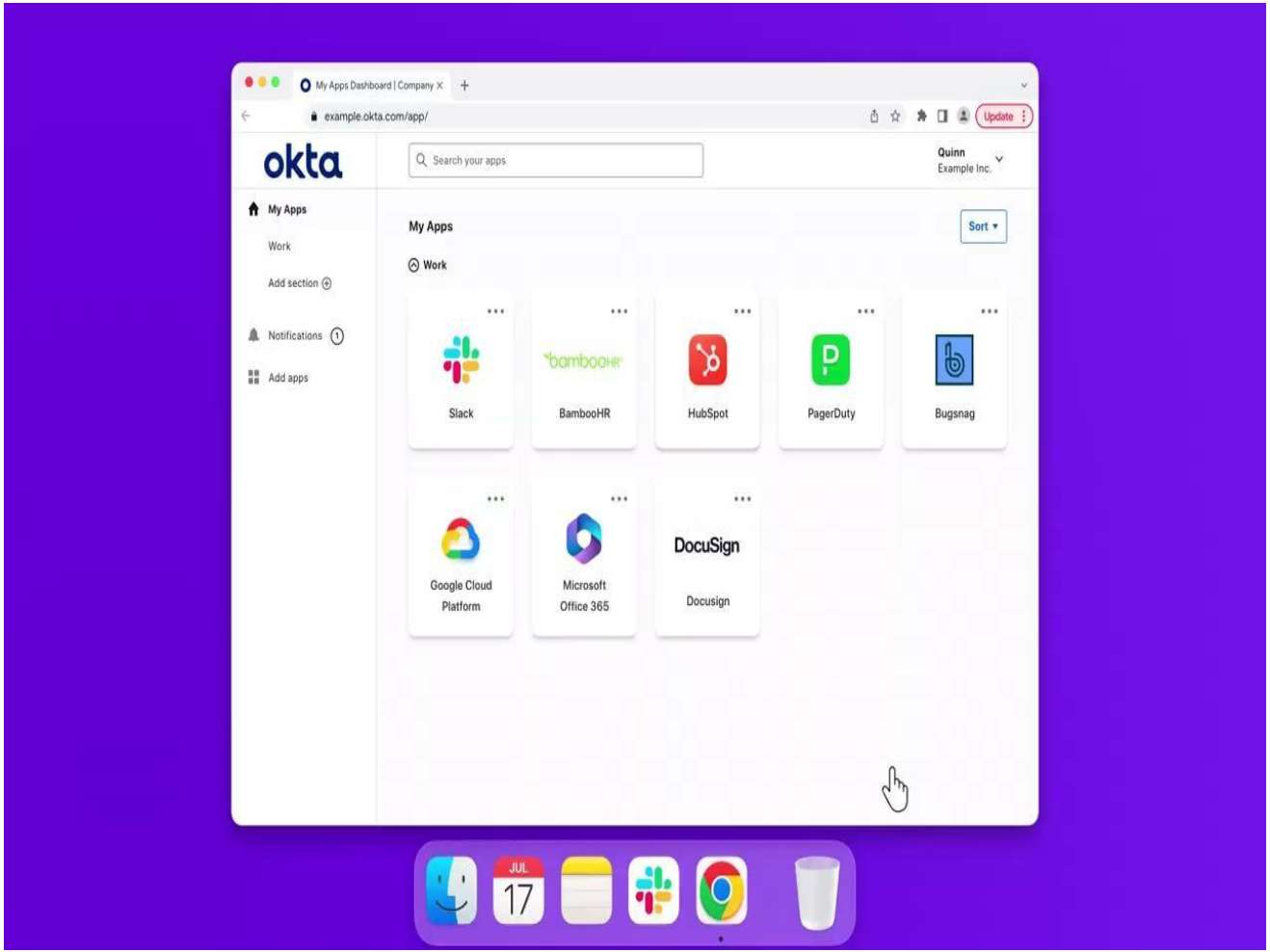
- Achieve Zero Trust Access
- Designed for Okta
- Works on Mac, Windows, Linux, iOS, and Android

Watch On-Demand Demo



databricks | octopus energy | Robinhood | Clio

ANDURIL | MERCURY | LaunchDarkly | JOBBER


My Apps Dashboard | Company
example.okta.com/app/
Update
okta
Search your apps
Quinn
Example Inc.
My Apps
Work
Add section
Notifications
Add apps
My Apps
Sort
Work
Slack
BambooHR
HubSpot
PagerDuty
Bugsnag
Google Cloud Platform
Microsoft Office 365
Docusign
JUL
17

ARP Cache Entries
network

Atom Packages
software, extensions, developers, ide

BIOS Platforms
hardware, operating-system

Certificates
identity, security, trust

Chrome Extensions
google, software, web-browsers, extensions

Crontab Entries
unix, scripts, autoruns

/etc/hosts Entries
network, dns

Firefox Add-ons
mozilla, software, web-browsers, extensions

Linux Debian Packages
debian, software, packages

Linux RPM Packages
centos, rpm, software, packages

Mac Active Directory Options
active-directory, enterprise

Mac App Schemes
apps, network, default-software

Mac Apps
apps, software

Mac Battery Status
batteries, hardware

Mac Crashes
operating-system, hardware, stability

Mac Firewall Settings
firewall, operating-system, security

Mac Homebrew Packages
software, packages, developers

Mac Kernel Extensions
kernel, security, stability, extensions

Mac Launchd Entries
autoruns, daemons, software

Mac Location Services Authori...
location, privacy, tcc

Mac Location Services Status
location, privacy, tcc

Mac Package Install History
software, apps, install-history, updates

Mac Profiles
mdm, device-management, polices

Mac Safari Extensions
apple, software, web-browsers, extensions

Mac Screenlock Status
screenlock, security, privacy

Mac SIP Settings
security, integrity, boot-security, operating-...

Mac Startup Security Settings
security, startup, boot-security, operating-...

Mac System Extensions
operating-system, kernel, extensions, stabi...

Mac XProtect Reports
anti-virus, threats, security

Microsoft Office Add-ins
microsoft, extensions, software, productivity

NPM Packages
developers, software, packages

Operating Systems
operating-system, updates

Python Packages
developers, software, packages

SSH Keys
developers, identity, security, trust

TPM Chips
tpm, hardware-security, integrity, hardware

Users
operating-system, identity, login, access

VSCode Extensions
software, extensions, ide, developers

Windows Chocolatey Packages
software, packages, developers

Windows Defender Settings
defender, anti-virus, security

Windows Defender Threat Det...
defender, anti-virus, security, threats

Windows Drivers
hardware, drivers

Windows Environment Variables
environment

Windows Microsoft Licenses
software, licenses

Windows Pending Updates
updates, operating-system, security

Windows Programs
software

Windows Screenlock Status
screenlock, security, privacy

Windows Update Settings
updates, operating-system, security


osquery

osquery

- Created by engineers at Facebook
- Open-source framework
- Provides a way to get structured data back from and endpoint
  - Enables a user to write queries to gather information about a machine or a fleet of machines
- This data can then be queried using SQL

Osquery goals

- Operating system support
  - Windows
  - macOS
  - Linux
- Provide reliability while using few system resources (CPU, RAM, etc.)
- Consistent access to data across multiple platforms, making it easy to expose and correlate structured data

```
~ osqueryi
osquery> SELECT * FROM time;
       weekday = Friday
          year = 2023
         month = 7
           day = 14
          hour = 19
       minutes = 4
       seconds = 54
      timezone = UTC
local_timezone = CDT
     unix_time = 1689361494
     timestamp = Fri Jul 14 19:04:54 2023 UTC
      datetime = 2023-07-14T19:04:54Z
      iso_8601 = 2023-07-14T19:04:54Z
```

```
~ osqueryi
osquery> SELECT * FROM battery;
                  model = bq40z651
          serial_number = AF8Y1419703LNM0J9TAY
            cycle_count = 112
                 health = Good
                  state = Battery Power
               charging = 0
                charged = 0
      designed_capacity = 8694
           max_capacity = 100
        current_capacity = 71
      percent_remaining = 71
               amperage = -695
                voltage = 11826
    minutes_until_empty = 483
minutes_to_full_charge = 35
```

osquery

275 Tables

- account_policy_data
- acpi_tables
- ad_config
- alf
- alf_exceptions
- alf_explicit_auths
- app_schemes
- apparmor_events
- apparmor_profiles
- appcompat_shims
- apps
- apt_sources
- arp_cache
- asl
- atom_packages
- augeas
- authenticode
- authorization_mechanisms
- authorizations
- authorized_keys
- autoexec
- azure_instance_metadata
- azure_instance_tags
- background_activities_modera
- battery
- bitlocker_info
- block_devices
- bpf_process_events
- bpf_socket_events
- browser_plugins
- carbon_black_info
- carves
- certificates
- chassis_info
- chocolatey_packages
- chrome_extension_content_sc
- chrome_extensions
- connectivity
- cpu_info
- cpu_time
- cpuid
- crashes
- crontab
- cups_destinations
- cups_jobs
- curl
- curl_certificate
- deb_packages
- default_environment
- device_file
- device_firmware
- device_hash
- device_partitions
- disk_encryption
- disk_events
- disk_info
- dns_cache
- dns_resolvers
- docker_container_envs
- docker_container_fs_changes
- docker_container_labels
- docker_container_mounts
- docker_container_networks
- docker_container_ports
- users
- video_info
- virtual_memory_info
- wifi_networks
- wifi_status
- wifi_survey
- winbaseobj
- windows_crashes
- windows_eventlog
- windows_events
- windows_firewall_rules
- windows_optional_features
- windows_security_center
- windows_security_products
- windows_update_history
- wmi_bios_info
- wmi_cli_event_consumers
- wmi_event_filters
- wmi_filter_consumer_binding
- wmi_script_event_consumers
- xprotect_entries
- xprotect_meta
- xprotect_reports
- yara
- yara_events
- ycloud_instance_metadata
- yum_sources

https://honest.security/

Honest Security v 1.0

A guide to endpoint security and device management that doesn't erode your values.

Start Reading →

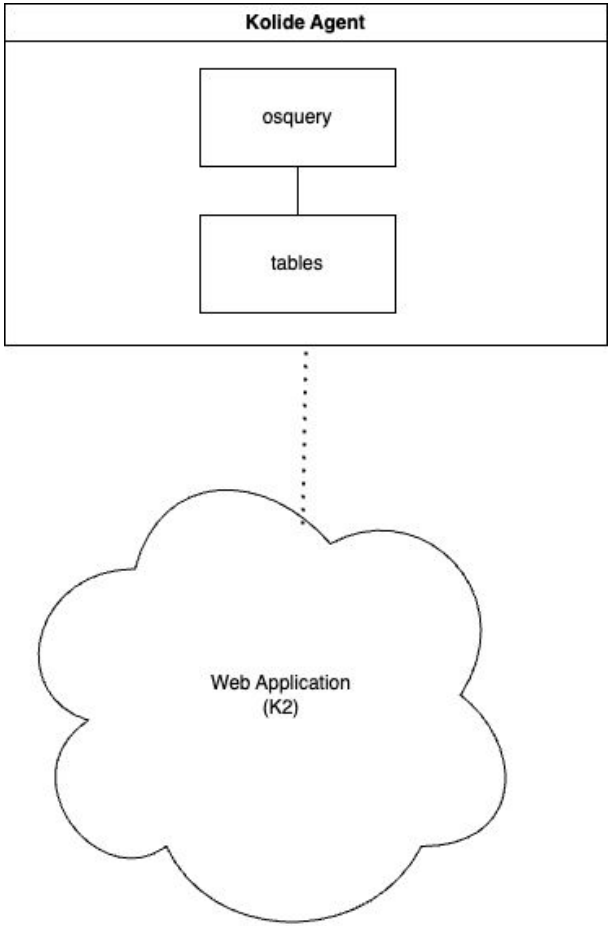
Get the PDF

THE TENETS OF HONEST SECURITY

1. The values your organization stands behind **should be well-represented in your security program.**
2. **A positive working relationship** between the end-user and the security team is incredibly valuable and worth fostering.
3. This relationship is built on a foundation of trust that is demonstrated through **informed consent and transparency.**
4. The security team should **anticipate and expect that end-users use their company owned devices for personal activities** and design their detection capabilities with this in mind.
5. End-users are capable of making **rational and informed decisions about security risks** when educated and honestly motivated.

osquery + Kolide

- Kolide's launcher agent runs on osquery to gather ground truth from devices so that we can determine device compliance
  - https://github.com/kolide/launcher
- Our launcher itself is also open source, and it offers a few extra elements on top of osquery such as automatic updating
- Osquery is essential for our product


Kolide Agent
osquery
tables
Web Application
(K2)

Checks Search Checks

6 New Checks | Add New Checks

88 Active Checks | 18 Paused

40%	macOS Software Update - Ensure OS Is Up-to-Date	22 Devices Failing (5 Blocked)	Blocks after **7 days**
75%	Windows Update - Ensure Important Updates Are Installed	3 Devices Failing (2 Blocked)	Blocks after **14 days**
75%	Windows Screenlock - Require Secure Screenlock Config COMPLIANCE CRITICAL	3 Devices Failing (3 Blocked)	Blocks **immediately**
77%	EFI SecureBoot Should Be Enabled	4 Devices Failing	Reporting Only Configure Blocking
83%	Windows Explorer - Require File Extensions Be Visible	2 Devices Failing	Reporting Only Configure Blocking
83%	Bitlocker - Require Bitlocker Encryption on Primary Disk	2 Devices Failing (2 Blocked)	Blocks after **14 days**
83%	Linux Firewall - Require Uncomplicated Firewall (UFW) To Be Enabled INFO	1 Device Failing	Reporting Only Configure Blocking

- We just learned what osquery is and how it works with Kolide
- Let’s talk about making a contribution
- The fun part: coming up with an idea and shipping it!

n8felton commented on Feb 25, 2022 Contributor

Feature request

What new feature do you want?

Expand the `video_info` table to macOS. Example of where the information available is in `system_profiler -json SPDisplaysDataType`

How is this new feature useful?

This would allow collecting information available in the `video_info` table for macOS hosts.

How can this be implemented?

I haven't dived deep into how this is possible yet, however, Core Graphics may be involved.

video_info

Retrieve video card information of the machine.

Improve this Description on Github

COLUMN	TYPE	DESCRIPTION
color_depth	INTEGER	The amount of bits per pixel to represent color.
driver	TEXT	The driver of the device.
driver_date	BIGINT	The date listed on the installed driver.
driver_version	TEXT	The version of the installed driver.
manufacturer	TEXT	The manufacturer of the gpu.
model	TEXT	The model of the gpu.
series	TEXT	The series of the gpu.
video_mode	TEXT	The current resolution of the display.

My desk

```
~ osqueryi
osquery> SELECT * FROM connected_displays;
                     name = ASUS PA328CGV
               product_id = 326c
            serial_number = ccb2
                vendor_id = 6b3
        manufactured_week = 44
        manufactured_year = 2021
               display_id = 2
                   pixels = 2560 x 1440
               resolution = 2560 x 1440 @ 120.00Hz
ambient_brightness_enabled = -1
           connection_type = -1
              display_type = -1
                     main = 1
                   mirror = 0
                   online = 1
                 rotation = 1
                  ......
```

```
      ......
                        name = Color LCD
                  product_id = a050
               serial_number = fd626d62
                   vendor_id = 610
           manufactured_week = 0
           manufactured_year = 0
                  display_id = 1
                      pixels = 3456 x 2234
                  resolution = 1728 x 1117 @ 120.00Hz
ambient_brightness_enabled = -1
             connection_type = spdisplays_internal
                display_type = spdisplays_built-in-liquid-retina-xdr
                        main = 0
                      mirror = 0
                      online = 1
                    rotation = 0


        ......
```

```
      ......
                        name = Hp Vh240A
                  product_id = 3499
               serial_number = 1010101
                   vendor_id = 220e
           manufactured_week = 0
           manufactured_year = 0
                  display_id = 5
                      pixels = 1080 x 1920
                  resolution = 1080 x 1920 @ 60.00Hz
ambient_brightness_enabled = -1
             connection_type = -1
                display_type = -1
                        main = 0
                      mirror = 0
                      online = 1
                    rotation = 1


          ......
```

cacab commented on Feb 24 Contributor

What?

This table adds a connected_displays virtual table which produces multiple rows (depending on the number of displays) giving useful information about a Macbook's internal battery.

Here is what the table looks like for my 15" Macbook Pro 2021:

name	product_id	serial_number	vendor_id	display_week	display_year	display_id	pixels	resolution	ambient_brightness	connection_type	display_type	main	mirror	online	rotation
LC32G7xT	705a	43583645	4c2d	13	2021	2	2560 x 1440	2560 x 1440 @ 120.00Hz	0	2560 x 1440 @ 120.00Hz		1	0	1	1
Color LCD	a050	fd626d62	610	0	0	1	3456 x 2234	1728 x 1117 @ 120.00Hz	0	1728 x 1117 @ 120.00Hz	spdisplays_built-in-liquid-retina-xdr	0	0	1	0
Hp Vh240A	3499	1010101	220e	0	0	32	1080 x 1920	1080 x 1920 @ 60.00Hz	0	1080 x 1920 @ 60.00Hz		0	0	1	1
Hp Vh240A	3499	1010181	220e	0	0	33	1080 x 1920	1080 x 1920 @ 60.00Hz	0	1080 x 1920 @ 60.00Hz		0	0	1	1

```
osquery>
```

Why?

This is meant to resolve #7486. This table provides helpful insights into display age, serial numbers, and resolution.

🎉 1 ❤️ 1

Easy, right?

Add `connected_displays` table on macOS ✓ macOS virtual tables

#7946 by cacab was merged on May 3 • Approved

- We all know software engineering isn’t that easy
- Time at Kolide = a little over a year
- Ruby on Rails != C++ || Python
- Blockers

Contribution

- Research, gathering resources
  - Community written blogs, and the official documentation

» Development » Creating New Tables

Edit on GitHub

Creating tables

SQL tables are used to represent abstract operating system concepts, such as running processes.

A table can be used in conjunction with other tables via operations like sub-queries and joins. This allows for a rich data exploration experience. While osquery ships with a default set of tables, osquery provides an API that allows you to create new tables.

You can explore current schema here: https://osquery.io/schema. Tables that are up for grabs in terms of development can be found on GitHub issues using the "virtual tables" + "up for grabs tag".

New Table Walkthrough

Let's walk through an exercise where we build a 'time' table. The table will have one row, and that row will have three columns: hour, minute, and second.

Column values (a single row) will be dynamically computed at query time.

TUTORIALS

How to Write a New Osquery Table

Jason Meller

Introduction

One of my favorite features of osquery is the delightful user experience associated with developing new virtual tables. In this guide, we will work together to implement a new high-value table from scratch that currently doesn't exist in osquery. Specifically, we will implement a `bluetooth` table that works on macOS.

Caitlin Cabrera

I'm trying to compile my updated fork of osquery and I'm getting some errors in the configuration stage: anyone in this channel has have insight? `build git:(master) cmake -DCMAKE_OSX_DEPLOYMENT_TARGET=10.14 -DCMAKE_C_COMPILER=clang -DCMAKE_CXX_COMPILER=clang++...`

becca

if you run `git describe --tags --always --dirty` what do you see?

Caitlin Cabrera

```
a11449053
```

becca

you'll want the git describe command to output something that looks like semver (e.g. for launcher i've got `v0.13.5-10-g40629c0-dirty`), that's how you'll know it's fixed

- There are multiple data sources to pull from to write an osquery table
    - Reading a property list (.plist) file
    - Reading an SQLite database file
    - Using a macOS API
    - Shelling out a binary (or utility)

n8felton commented on Feb 25, 2022 Contributor

Feature request

What new feature do you want?

Expand the `video_info` table to macOS. Example of where the information available is in `system_profiler -json SPDisplaysDataType`

How is this new feature useful?

This would allow collecting information available in the `video_info` table for macOS hosts.

How can this be implemented?

I haven't dived deep into how this is possible yet, however, Core Graphics may be involved.

cc@Caitlins-MacBook-Pro-3:~ ⌥⌘1

```
➜  ~ system_profiler SPDisplaysDataType
Graphics/Displays:

    Apple M1 Pro:

      Chipset Model: Apple M1 Pro
      Type: GPU
      Bus: Built-In
      Total Number of Cores: 16
      Vendor: Apple (0x106b)
      Metal Support: Metal 3
      Displays:
        Color LCD:
          Display Type: Built-in Liquid Retina XDR Display
          Resolution: 3456 x 2234 Retina
          Main Display: Yes
          Mirror: Off
          Online: Yes
          Automatically Adjust Brightness: No
          Connection Type: Internal

➜  ~
```

What About Shelling Out to a Binary?

Sometimes when a user of osquery is advocating for a new table, they point to a command-line tool that produces the exact output they are looking for (in our case, `system_profiler SPBluetoothDataType` does the job). These users might expect the table to be easily developed by quickly asking the osquery process to execute the command-line tool, read its output, and produce a table.

```
table_name("connected_displays")
description("Provides information about the connected displays of the machine.")
schema([
    Column("name", TEXT, "The name of the display."),
    Column("product_id", TEXT, "The product ID of the display."),
    Column("serial_number", TEXT, "The serial number of the display."),
    Column("vendor_id", TEXT, "The vendor ID of the display."),
    Column("manufactured_week", INTEGER, "The manufacture week of the display. This field is 0 if not supported"),
    Column("manufactured_year", INTEGER, "The manufacture year of the display. This field is 0 if not supported"),
    Column("display_id", TEXT, "The display ID."),
    Column("pixels", TEXT, "The number of pixels of the display."),
    Column("resolution", TEXT, "The resolution of the display."),
    Column("ambient_brightness_enabled", TEXT, "The ambient brightness setting associated with the display. This will be 1 if enabled and is 0 if disabled or not supported."),
    Column("connection_type", TEXT, "The connection type associated with the display."),
    Column("display_type", TEXT, "The type of display."),
    Column("main", INTEGER, "If the display is the main display."),
    Column("mirror", INTEGER, "If the display is mirrored or not. This field is 1 if mirrored and 0 if not mirrored."),
    Column("online", INTEGER, "The online status of the display. This field is 1 if the display is online and 0 if it is offline."),
    Column("rotation", TEXT, "The orientation of the display."),
])
implementation("connected_displays@genConnectedDisplays")
```

```
#pragma clang diagnostic pop

    cleanup = [&]() {
      CFRelease((__bridge CFTypeRef)document);
      CFBundleUnloadExecutable(bundle);
      CFRelease(bundle);
    };

    NSDictionary* report = [[[document reportForDataType:@"SPDisplaysDataType"]
        objectForKey:@"_items"] lastObject];
    NSArray* data = [report valueForKeyPath:@"spdisplays_ndrvs"];

    for (NSString* obj in data) {
      Row r;

      if (data == nullptr) {
        return results;
      }

      if ([obj valueForKey:@"_name"]) {
        r["name"] = TEXT([[obj valueForKey:@"_name"] UTF8String]);
      }

      if ([obj valueForKey:@"_spdisplays_display-product-id"]) {
        r["product_id"] = TEXT(
            [[obj valueForKey:@"_spdisplays_display-product-id"] UTF8String]);
      }

      if ([obj valueForKey:@"_spdisplays_display-serial-number"]) {
        r["serial_number"] = TEXT([[obj
            valueForKey:@"_spdisplays_display-serial-number"] UTF8String]);
      }
```

Caitlin Cabrera

I'm working on setting up an integration test and running `cmake --build . --target test` gives the following: `make: *** No rule to make target test'. Stop. File has been added to `CMakeLists.txt`. Running from the build dir. Any suggestions?

Stefano Bonicatti

hum so, It's a bit weird that it has tried to reconfigure the compiler; I would delete the CMakeCache.txt file in the build folder, then configure again, then build all (since tests do not have a common target to build them, the `test` target only runs them), and try again.

Caitlin Cabrera

That worked. Thank you!

cacab commented on Feb 24 Contributor

What?

This table adds a connected_displays virtual table which produces multiple rows (depending on the number of displays) giving useful information about a Macbook's internal battery.

Here is what the table looks like for my 15" Macbook Pro 2021:

name	product_id	serial_number	vendor_id	display_week	display_year	display_id	pixels	resolution	ambient_brightness	connection_type	display_type	main	mirror	online	rotation
LC32G7xT	705a	43583645	4c2d	13	2021	2	2560 x 1440	2560 x 1440 @ 120.00Hz	0	2560 x 1440 @ 120.00Hz		1	0	1	1
Color LCD	a050	fd626d62	610	0	0	1	3456 x 2234	1728 x 1117 @ 120.00Hz	0	1728 x 1117 @ 120.00Hz	spdisplays_built-in-liquid-retina-xdr	0	0	1	0
Hp Vh240A	3499	1010101	220e	0	0	32	1080 x 1920	1080 x 1920 @ 60.00Hz	0	1080 x 1920 @ 60.00Hz		0	0	1	1
Hp Vh240A	3499	1010181	220e	0	0	33	1080 x 1920	1080 x 1920 @ 60.00Hz	0	1080 x 1920 @ 60.00Hz		0	0	1	1

```
osquery>
```

Why?

This is meant to resolve #7486. This table provides helpful insights into display age, serial numbers, and resolution.

🎉 1 ❤️ 1

Easy, right?

Add `connected_displays` table on macOS ✓ macOS virtual tables

#7946 by cacab was merged on May 3 • Approved

Open source projects

- Managing
- Approaching as a new or first time contributor
- Caveat: there are many types of different types of open source projects
  - Vendor driven
  - Sole proprietor passion project
  - Heavily community based
  - Etc.

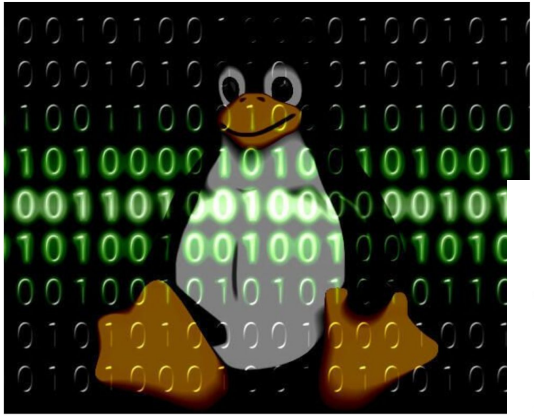
Managing and contributing

- What incentivises developers, managers and contributors?
- How can we best align our projects' mission to serve our use case?
- How can we best encourage the people who are current community members?

8 of the worst open source innovations of the decade

by **Jack Wallen** in **Software**
on December 11, 2019, 2:00 PM PST

Open source innovations aren't all successes. Jack Wallen shares his picks for the biggest open source failures of the 2010s.



The most forgotten open source projects

Unity

While the Unity 3D engine is triumphing in the video game world, its "relative" **Unity** (Canonical's graphical shell for GNOME) is now almost disappeared in action. It was a great success when Ubuntu implemented it, most users liked it, however, now they have changed it to GNOME and few seem to miss it.

Mir

Another of Canonical's jewels that ended up failing miserably, along with Unity, Ubuntu Touch, Ubuntu Edge, etc. Although it is still in development, the truth is that the adoption of this **graphic server** It's void. Most distros still use X or Wayland.

Ubuntu Touch and FirefoxOS

Canonical also wanted to reach mobile devices with its **Ubuntu Touch**, and although it is a system that opens possibilities compared to Android, and that is still under development by UBports, the truth is that few have launched to adopt it. Their list of compatible devices is very limited.

Firefox OS it was also one of those wonders that seemed like they would go a long way. In this case from Mozilla, and also with the intention of conquering the mobile sector. However, now it is also conspicuous by its absence ...

Diaspora

La **Diaspora social network** It was intended to be a success, with a distributed system, with groups of independently owned nodes operating together, and with the idea of improving privacy and avoiding the control of large corporations. However, that good idea turned into a chimera and few remember it anymore.

How I Wasted Two Years In Opensource Development

The story of the side-project failure that let me grow up

Daniele Fontani · Follow
Published in Towards Data Science · 6 min read · Apr 8, 2021

288 5

I'm an open-source love. I started developing an open-source side project in 2006 and that was the secret that boosts my career. Thanks to the experiments that I have made during this trip I'm now a better developer and I give back something to the community — almost I hope.

I wrote about my vision of opensource as a driver for companies and developers growth here but that is another story 😄

In this article, I want to talk about my experience since 2018 on an open-source low-code platform called RawCms.

Why do open source projects fail?

- The most popular reasons for the failure of an open source project are lack of time, lack of interest, competition, and outdated/obsolete technologies (Coelho, 2017)
- There are a lot of elements that can make or break a project
- Each of these elements share a common thread

Community

Open source communities

- Collaboration and culture
- Technology and innovation
- Processes and vision

Open source communities: collaboration and culture

- The “people” piece
- Maintaining relationships with existing developers as well as a focus on creating new relationships with new developers
- Venues for discussion
  - Garnering feedback and discussion with developers/ core team
  - The culture associated with these venues
- Code of Conduct
- Office hours (held by core team)
- Active communities with an overall positive culture

Open source communities: technology and innovation

- Strong, well-defined use-case
- Strong product direction (from core team)
- Emphasis on quality
- Processes in place
  - Tests, linters, straight-forward pull request process
  - CI/CD pipeline

Open source communities: processes and vision

- Core team
- Documentation
  - Blogs (official and unofficial)
- License agreements such as Contributor License Agreements (CLAs)

How can we implement these practices as maintainers and/or contributors?

Checklist (for maintainers)

- Core team with clear roles and responsibilities
  - Approving pull requests, driving product direction and maintaining a standard of quality
- Code of conduct to ensure interactions within your community are governed by a set of standards and promote inclusivity
- Clear process for CLAs (CI/CD pipeline or PR)
  - Should be easy to find, understand, and complete

Checklist (for maintainers)

- Strong documentation, tutorials
  - Usually the first thing potential contributors see
- Formal communities- also listed in the documentation, usually moderated and managed by the core team
- Transparent process for reviewing contributions
- Office hours

Guidelines (for contributors)

- Lurk first
- Start small
- If you have a question, search the community first
- If you can’t find an answer to your question, ask!
- Lean on your coworkers and the community
- Advocate for yourself
- Is there a code of conduct? How is it written?
- Still unsure of where to start? Ask!

Summary

- Was able to get assistance and feedback from core team, community members (collaboration and culture)
- Added a new feature to the project (technology and innovation)
- Issue the pull request resolved drove the projects main goals (processes and vision)

Wrapping up

- If your team utilizes a piece of open source software, create a culture that encourages contribution
- Allow engineers time to devote to projects and communities they are passionate about
- Encourage mentorship and a culture of teaching and learning

Open source is for everyone

Thank you / where you can find me

- **caitlincabrera.com**
- **github.com/cacab**
- **linkedin.com/in/caitlincabrera**
- twitter.com/cc_codes
- ruby.social/@cc_codes
- @cacab.bsky.social

Sources

- Why you should contribute to open source
  - https://www.forbes.com/sites/forbesinsights/2020/01/15/diversity-confirmed-to-boost-innovation-and-financial-results/?sh=2662492cc4a6
  - https://greatergood.berkeley.edu/article/item/how_diversity_makes_us_smarter
  - https://cointelegraph.com/news/the-importance-of-open-source-in-computer-science-and-software-development
- Failed open source projects / why open source projects fail
  - https://www.techrepublic.com/article/8-of-the-worst-open-source-innovations-of-the-decade/
  - https://arxiv.org/pdf/1707.02327.pdf
  - https://www.linuxadictos.com/en/los-8-peores-proyectos-de-codigo-abierto.html
  - https://en.wikipedia.org/wiki/OpenOffice.org
- Guidelines for open source participation and contribution
  - https://www.linuxfoundation.org/resources/open-source-guides/participating-in-open-source-communities

Sources

- Writing an osquery table
  - https://www.kolide.com/blog/how-to-write-a-new-osquery-table
- How to write documentation
  - https://open.spotify.com/episode/28jv8wYRC61lMXP1uJ02TH?si=2bfa65b611804951
- How osquery works
  - https://www.kolide.com/blog/osquery-under-the-hood
- Writing and evaluating CoC's
  - https://www.palantir.net/blog/secret-sauce-podcast-ep-10-why-codes-conduct-matter
  - https://confcodeofconduct.com/
- MacOS system profiler
  - https://apple.fandom.com/wiki/Apple_System_Profiler#:~:text=Apple%20System%20Profiler%20is%20a,on%20which%20it%20is%20running
- Honest Security Blog
  - https://honest.security/