



Flutter and AGL Deep Dive

AGL Workshop

Joel Winarske

Toyota Connected North America

Introduction

- Qualcomm Windows BSP organization (~8 years)
 - MSM7200 to 8CX
 - Window Phone, Windows on ARM
 - Brought up UEFI on QC silicon MSM8660
 - Primary QC engineer involved in Windows on ARM bring-up. HAL, SD, USB, WiFi, etc.
 - Demonstrated first running case of the Windows Kernel power collapsing to Steven Sinofsky
 - Implemented Parking algorithm
 - 1st FTE engineer in Redmond
 - Many other highlights
- INRIX (OpenCar) – Automotive Javascript UI/UX
 - Chromium Browser customizations
 - Re-wrote the OpenCar server. Went from 20 second load to 750ms. 1+GB RAM runtime footprint down to 1.5MB.
 - Target Demos + device input drivers
 - Made the Android solution a viable product
 - Started work on embedded flutter - cross compile engine + store app demo
- MSFT Surface - PLE Team (Post Launch Engineering)
 - Surface ProX (8CX)
- Author/Creator of <https://github.com/meta-flutter/meta-flutter>
- Owner of ivi-homescreen (flutter-auto) at Toyota

Embedded Flutter

Components

- Development Environment
- Build Environment
- Target Environment

Flutter Engine

- Core component of the Flutter technology
- Written in C++ 17
- Common source tree for all platforms
 - Desktop - Mac/Windows/Linux
 - Mobile - Android / iOS
 - Web
 - Fuchsia
 - Custom Embedder

Flutter Engine

- Build environment based on Google GN
- Dart + SKIA
- Impeller
- Custom Embedder Backend support
 - OpenGL
 - Software
 - Metal
 - Vulkan

Benefits

- Premium User Experience
- Developer Experience
- Commodity Talent Pool
- Reduced NRE
- Time To Market

Development Environment

Goals

- Shortest path to run a Flutter App on AGL
- Easy to change between Flutter SDK versions
- Support unique configurations
- Archive friendly
- Ubuntu 20+ support

What does it do?

- Creates a Flutter Workspace
 - Clones Flutter SDK
 - Sets up local Flutter SDK config
 - Sets up local pub cache
 - Fetches defined artifacts and installs runtime dependencies
 - Clones defined repositories
 - Creates setup_env.sh

Demo

Workspace Components

- Flutter SDK
- Sandboxed Flutter SDK config
- Sandboxed pub cache
- Platform Setup
 - Runtime
 - Binary
 - Required Dependencies
 - Custom-Device Config
- Development Repositories
 - VS Code launch.json

Install Method – AGL Source Tree

- `cd $AGL_TOP`
- `external/meta-flutter/tools/setup_flutter_workspace.py meta-agl-devel/meta-agl-flutter/tools/flutter_workspace_config.json`

Install Method – Tip of Tree

- `mkdir -p $HOME/workspace && cd $HOME/workspace`
- `curl --proto '=https' --tlsv1.2 -sSf https://gerrit.automotivelinux.org/gerrit/gitweb?p=AGL/meta-agl-devel.git;a=blob_plain;f=meta-agl-flutter/tools/flutter_workspace_config.json;hb=HEAD -o flutter_workspace_config.json`
- `curl --proto '=https' --tlsv1.2 -sSf https://raw.githubusercontent.com/meta-flutter/meta-flutter/kirkstone/tools/setup_flutter_workspace.py | python3`

Script Options

- `./setup_flutter_workspace.py --help`
- usage: `setup_flutter_workspace.py [-h] [--clean] [--workspace-cfg WORKSPACE_CFG] [--flutter-version FLUTTER_VERSION]`
- `[--target-user TARGET_USER] [--target-address TARGET_ADDRESS]`
-
- options:
- `-h, --help` show this help message and exit
- `--clean` Wipes workspace clean
- `--workspace-cfg WORKSPACE_CFG`
- Selects custom workspace configuration file
- `--flutter-version FLUTTER_VERSION`
- Select flutter version. Overrides config file key: flutter-version
- `--target-user TARGET_USER`
- Sets custom-device target user name
- `--target-address TARGET_ADDRESS`
- Sets custom-device target address

Setup Caveats

- Flutter IDE tooling
 - Flutter tooling uses File Watching to trigger events. This conflicts with the setup script. To prevent this interaction run from a new system terminal instance, or temporarily disable the tooling.
- Multiple 'flutter' entries in system path
 - The setup script will attempt to remove the first occurring instance of 'flutter' from PATH when it runs, if the resolved path matches. To prevent this, remove all entries of 'flutter' from your path.
- Your Host Machine is expected to have hardware Hypervisor support enabled
 - If you explicitly want software Hypervisor support remove ``-enable-kvm -cpu kvm64`` from the QEMU arguments in `flutter_workspace_config.json` before running.

Debug on flutter-auto desktop

- Login via GDM Wayland Session
- Open Terminal and type
- `source ${FLUTTER_WORKSPACE}/setup_env.sh`
- Navigate to your favorite app
- `flutter run -d desktop-auto`

Debug on AGL QMEU

- Open Terminal and type
- `source ${FLUTTER_WORKSPACE}/setup_env.sh`
- Type `qemu_run`
- Wait until QEMU image reaches login prompt
- Run `ssh -p 2222 root@localhost` who to add remote host to `~/.ssh/known_hosts`
- Navigate to your favorite app
- `flutter run -d AGL-qemu`

Debug using Visual Studio Code

- Open Terminal and type
 - `source ${FLUTTER_WORKSPACE}/setup_env.sh`
 - `code .`
- Navigate to the debug pane
- Select application + runtime environment from drop down combo box
- Click the play icon to start debug session

VS Code launch.json creation

- `setup_flutter_workspace.py` creates a `.vscode/launch.json` file if one is not present
- It uses the repo configuration key ``pubspec_path``
- If this key is present in the repo entry, then it will add entry to `.vscode/launch.json`

Resources

- <https://github.com/meta-flutter/meta-flutter/tree/kirkstone/tools>
- <https://gerrit.automotivelinux.org/gerrit/gitweb?p=AGL/meta-agl-devel.git;a=blob;f=meta-agl-flutter/README.md>
- <https://github.com/flutter/flutter/wiki/Using-custom-embedders-with-the-Flutter-CLI>

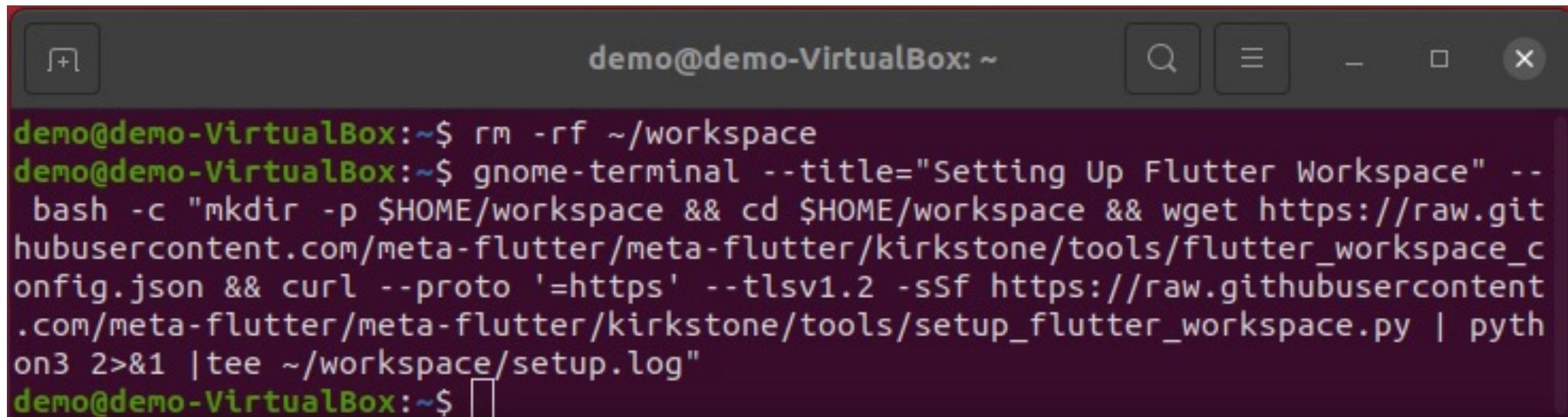
Labs

- Setup Flutter Workspace
- Create AGL Flutter Application
- Debug AGL Flutter Application – CLI
- Debug AGL Flutter Application – Visual Studio Code

Setup Flutter Workspace (1/3)

- Copy and paste the command below

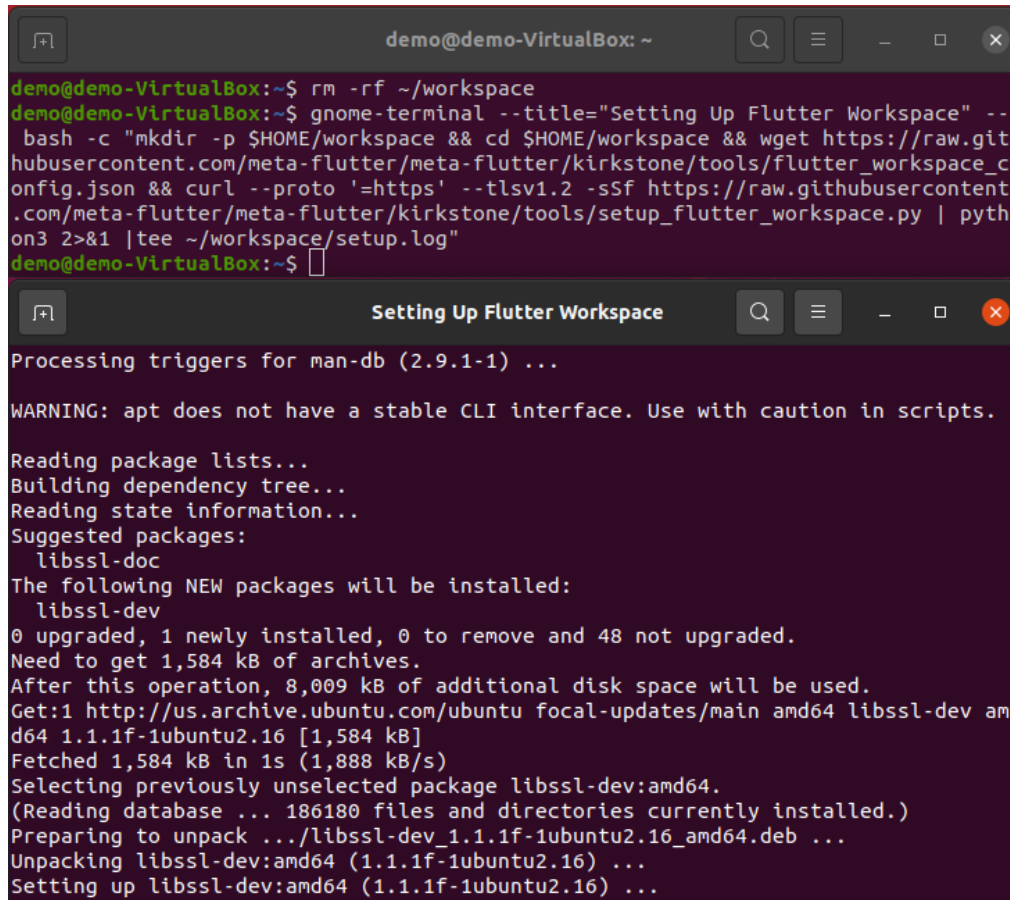
```
rm -rf ~/workspace
gnome-terminal --title="Setting Up Flutter Workspace" -- bash -c "mkdir -p $HOME/workspace && cd $HOME/workspace && wget https://raw.githubusercontent.com/meta-flutter/meta-flutter/kirkstone/tools/flutter_workspace_config.json && curl --proto '=https' --tlsv1.2 -sSf https://raw.githubusercontent.com/meta-flutter/meta-flutter/kirkstone/tools/setup_flutter_workspace.py | python3 2>&1 | tee ~/workspace/setup.log"
```

A screenshot of a terminal window titled "demo@demo-VirtualBox: ~". The terminal shows the execution of the command to set up the Flutter workspace. The command is: `rm -rf ~/workspace`, `gnome-terminal --title="Setting Up Flutter Workspace" -- bash -c "mkdir -p $HOME/workspace && cd $HOME/workspace && wget https://raw.githubusercontent.com/meta-flutter/meta-flutter/kirkstone/tools/flutter_workspace_config.json && curl --proto '=https' --tlsv1.2 -sSf https://raw.githubusercontent.com/meta-flutter/meta-flutter/kirkstone/tools/setup_flutter_workspace.py | python3 2>&1 | tee ~/workspace/setup.log"`. The terminal output shows the command being executed and the prompt returning to `demo@demo-VirtualBox:~$`.

```
demo@demo-VirtualBox:~$ rm -rf ~/workspace
demo@demo-VirtualBox:~$ gnome-terminal --title="Setting Up Flutter Workspace" --
bash -c "mkdir -p $HOME/workspace && cd $HOME/workspace && wget https://raw.git
hubusercontent.com/meta-flutter/meta-flutter/kirkstone/tools/flutter_workspace_c
onfig.json && curl --proto '=https' --tlsv1.2 -sSf https://raw.githubusercontent
.com/meta-flutter/meta-flutter/kirkstone/tools/setup_flutter_workspace.py | pyth
on3 2>&1 | tee ~/workspace/setup.log"
demo@demo-VirtualBox:~$
```

Setup Flutter Workspace (2/3)

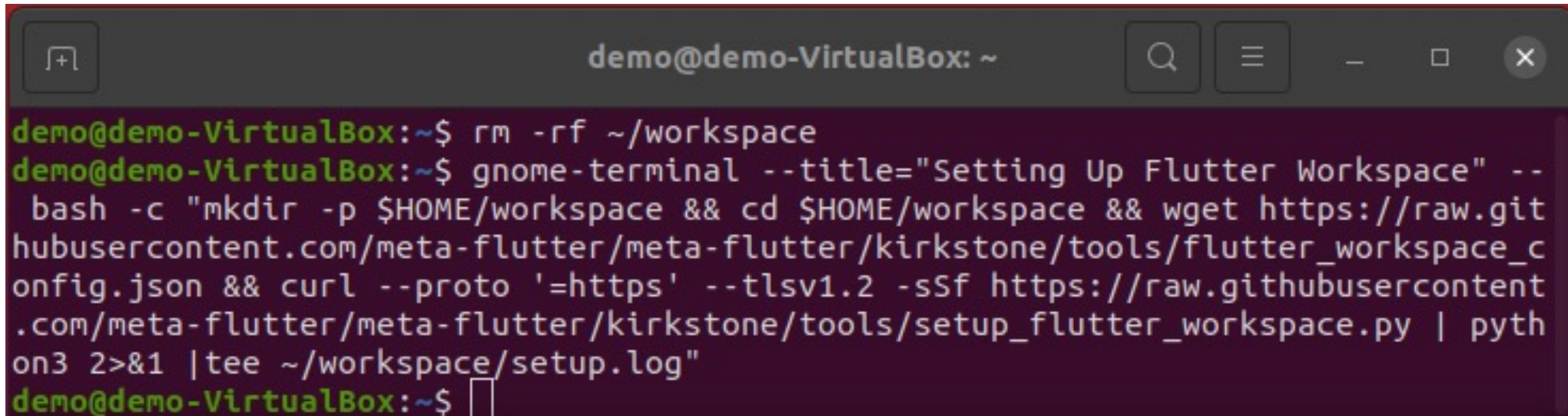
- The command brings up a new terminal to set up the flutter workspace.



```
demo@demo-VirtualBox: ~  
demo@demo-VirtualBox:~$ rm -rf ~/workspace  
demo@demo-VirtualBox:~$ gnome-terminal --title="Setting Up Flutter Workspace" --  
  bash -c "mkdir -p $HOME/workspace && cd $HOME/workspace && wget https://raw.git  
hubusercontent.com/meta-flutter/meta-flutter/kirkstone/tools/flutter_workspace_c  
onfig.json && curl --proto 'https' --tlsv1.2 -sSf https://raw.githubusercontent  
.com/meta-flutter/meta-flutter/kirkstone/tools/setup_flutter_workspace.py | pyth  
on3 2>&1 |tee ~/workspace/setup.log"  
demo@demo-VirtualBox:~$  
  
Setting Up Flutter Workspace  
Processing triggers for man-db (2.9.1-1) ...  
  
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.  
  
Reading package lists...  
Building dependency tree...  
Reading state information...  
Suggested packages:  
  libssl-doc  
The following NEW packages will be installed:  
  libssl-dev  
0 upgraded, 1 newly installed, 0 to remove and 48 not upgraded.  
Need to get 1,584 kB of archives.  
After this operation, 8,009 kB of additional disk space will be used.  
Get:1 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libssl-dev am  
d64 1.1.1f-1ubuntu2.16 [1,584 kB]  
Fetched 1,584 kB in 1s (1,888 kB/s)  
Selecting previously unselected package libssl-dev:amd64.  
(Reading database ... 186180 files and directories currently installed.)  
Preparing to unpack .../libssl-dev_1.1.1f-1ubuntu2.16_amd64.deb ...  
Unpacking libssl-dev:amd64 (1.1.1f-1ubuntu2.16) ...  
Setting up libssl-dev:amd64 (1.1.1f-1ubuntu2.16) ...
```


Setup Flutter Workspace (3/3)

- Once the setup is completed, the "Setting Up Flutter Workspace" terminal will close automatically.

A terminal window titled "demo@demo-VirtualBox: ~" with standard window controls. The terminal shows the execution of a command to remove the workspace directory, followed by a command to open a new terminal window titled "Setting Up Flutter Workspace" and run a complex command to create the workspace, download the Flutter workspace configuration, and execute the setup script. The terminal prompt returns to the user's shell.

```
demo@demo-VirtualBox:~$ rm -rf ~/workspace
demo@demo-VirtualBox:~$ gnome-terminal --title="Setting Up Flutter Workspace" --
  bash -c "mkdir -p $HOME/workspace && cd $HOME/workspace && wget https://raw.git
  hubusercontent.com/meta-flutter/meta-flutter/kirkstone/tools/flutter_workspace_c
  onfig.json && curl --proto '=https' --tlsv1.2 -sSf https://raw.githubusercontent
  .com/meta-flutter/meta-flutter/kirkstone/tools/setup_flutter_workspace.py | pyth
  on3 2>&1 | tee ~/workspace/setup.log"
demo@demo-VirtualBox:~$
```

Create AGL Flutter Application (1/2)

- Setup environment with the command below:

```
source ${FLUTTER_WORKSPACE}/setup_env.sh
```

```
demo@demo-VirtualBox:~/workspace$ source setup_env.sh
SCRIPT_PATH=/home/demo/workspace
*****
* Setting FLUTTER_WORKSPACE to:
* /home/demo/workspace
*****
[!] Flutter (Channel unknown, 3.3.2, on Ubuntu 20.04.5 LTS 5.15.0-48-generic, locale en_US.UTF-8)
! Flutter version 3.3.2 on channel unknown at /home/demo/workspace/flutter
! Upstream repository unknown
• Framework revision e3c29ec00c (3 weeks ago), 2022-09-14 08:46:55 -0500
• Engine revision a4ff2c53d8
• Dart version 2.18.1
• DevTools version 2.15.0

[✓] Linux toolchain - develop for Linux desktop
• clang version 10.0.0-4ubuntu1
• cmake version 3.24.2
• ninja version 1.10.0
• pkg-config version 0.29.1

[!] Flutter IDE Support (No supported IDEs installed)
• IntelliJ - https://www.jetbrains.com/idea/
• Android Studio - https://developer.android.com/studio/
• VS Code - https://code.visualstudio.com/

[✓] Connected device (1 available)
• Linux (desktop) • linux • linux-x64 • Ubuntu 20.04.5 LTS 5.15.0-48-generic

[✓] HTTP Host Availability
• All required HTTP hosts are available

! Doctor found issues in 2 categories.

No custom devices found in "/home/demo/workspace/.config/flutter/custom_devices.json"

*****
* Type 'qemu_run' to start the emulator *
*****
```

Create AGL Flutter Application (2/2)

```
cd ${FLUTTER_WORKSPACE}/app
flutter create hello_world -t app
cd hello_world
flutter run -d desktop-auto
```

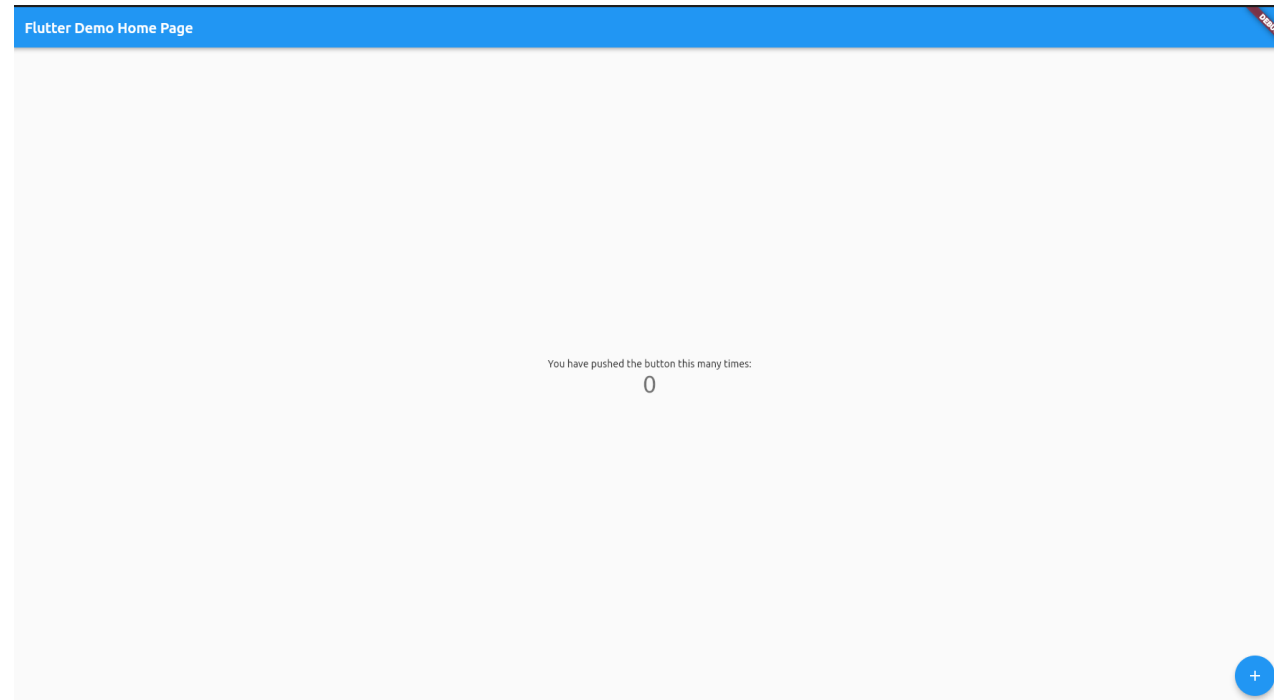
```
demo@demo-VirtualBox:~$ cd workspace/app
demo@demo-VirtualBox:~/workspace/app$ flutter create hello_world -t app
Creating project hello_world...
Running "flutter pub get" in hello_world... 3.9s
Wrote 18 files.

All done!
In order to run your application, type:

$ cd hello_world
$ flutter run

Your application code is in hello_world/lib/main.dart.

demo@demo-VirtualBox:~/workspace/app$ cd hello_world
demo@demo-VirtualBox:~/workspace/app/hello_world$ flutter run -d desktop-auto
```



Debug AGL Flutter Application – CLI

```
source workspace/setup_env.sh
cd workspace/app/gallery
flutter run -d desktop-auto
```

```
demo@demo-VirtualBox:~$ source workspace/setup_env.sh
SCRIPT_PATH=/home/demo/workspace
*****
* Setting FLUTTER_WORKSPACE to:
* /home/demo/workspace
*****
[!] Flutter (Channel unknown, 3.3.2, on Ubuntu 20.04.5 LTS 5.15.0-48-generic,
    locale en_US.UTF-8)
    ! Flutter version 3.3.2 on channel unknown at /home/demo/workspace/flutter
    ! Upstream repository unknown
    • Framework revision e3c29ec00c (3 weeks ago), 2022-09-14 08:46:55 -0500
    • Engine revision a4ff2c53d8
    • Dart version 2.18.1
    • DevTools version 2.15.0

[✓] Linux toolchain - develop for Linux desktop
    • clang version 10.0.0-4ubuntu1
    • cmake version 3.24.2
    • ninja version 1.10.0
    • pkg-config version 0.29.1

[!] Flutter IDE Support (No supported IDEs installed)
    • IntelliJ - https://www.jetbrains.com/idea/
    • Android Studio - https://developer.android.com/studio/
    • VS Code - https://code.visualstudio.com/

[✓] Connected device (2 available)
    • Linux (desktop) • linux • linux-x64 • Ubuntu 20.04.5
      LTS 5.15.0-48-generic
    • Toyota flutter-auto (mobile) • desktop-auto • linux-x64 • flutter-auto
      x86_64

[✓] HTTP Host Availability
    • All required HTTP hosts are available

! Doctor found issues in 2 categories.

List of custom devices in
"/home/demo/workspace/.config/flutter/custom_devices.json":
id: desktop-auto, label: Toyota flutter-auto, enabled: true
id: AGL-qemu, label: AGL x86_64 QEMU Image, enabled: true
id: stm32mp15-weston, label: STM32MP157F-DK2, enabled: true
id: pi-zero2w-weston, label: Raspberry Pi Zero 2W, enabled: true

*****
* Type 'qemu_run' to start the emulator
*****
demo@demo-VirtualBox:~$ cd workspace/app/gallery
demo@demo-VirtualBox:~/workspace/app/gallery$ flutter run -d desktop-auto
```

Debug AGL Flutter Application – Visual Studio Code (1/3)

```
cd workspace
source setup_env.sh
code .
```

```
demo@demo-VirtualBox:~$ cd workspace
demo@demo-VirtualBox:~/workspace$ source setup_env.sh
SCRIPT_PATH=/home/demo/workspace
*****
* Setting FLUTTER_WORKSPACE to:
* /home/demo/workspace
*****
[!] Flutter (Channel unknown, 3.3.2, on Ubuntu 20.04.5 LTS 5.15.0-48-generic,
    locale en_US.UTF-8)
! Flutter version 3.3.2 on channel unknown at /home/demo/workspace/flutter
! Upstream repository unknown
• Framework revision e3c29ec00c (3 weeks ago), 2022-09-14 08:46:55 -0500
• Engine revision a4ff2c53d8
• Dart version 2.18.1
• DevTools version 2.15.0

[✓] Linux toolchain - develop for Linux desktop
• clang version 10.0.0-4ubuntu1
• cmake version 3.24.2
• ninja version 1.10.0
• pkg-config version 0.29.1

[✓] VS Code
• VS Code at /snap/code/current
• Flutter extension version 3.48.0

[✓] Connected device (2 available)
• Linux (desktop) • linux • linux-x64 • Ubuntu 20.04.5 LTS 5.15.0-48-generic
• Toyota flutter-auto (mobile) • desktop-auto • linux-x64 • flutter-auto x86_64

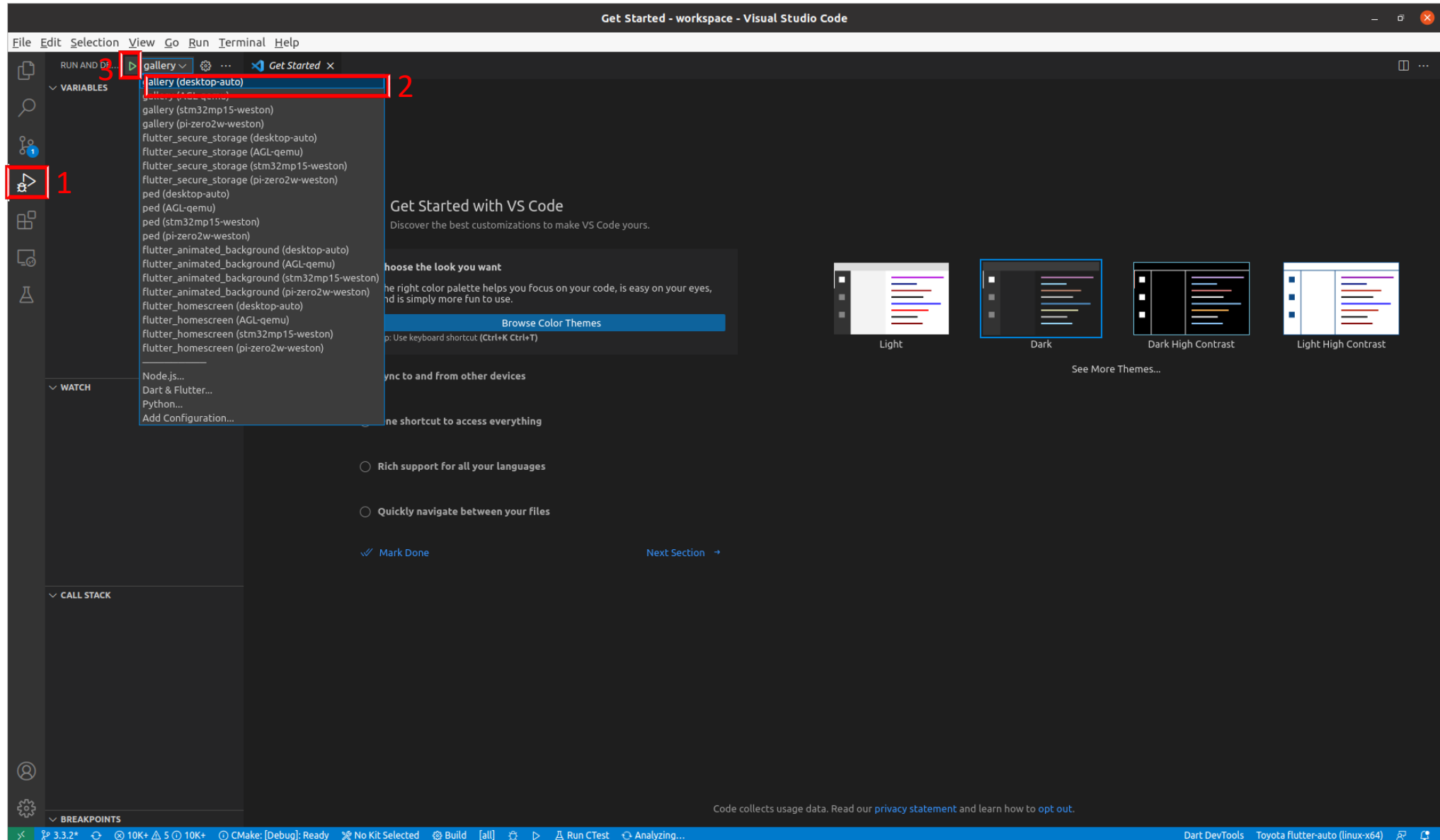
[✓] HTTP Host Availability
• All required HTTP hosts are available

! Doctor found issues in 1 category.

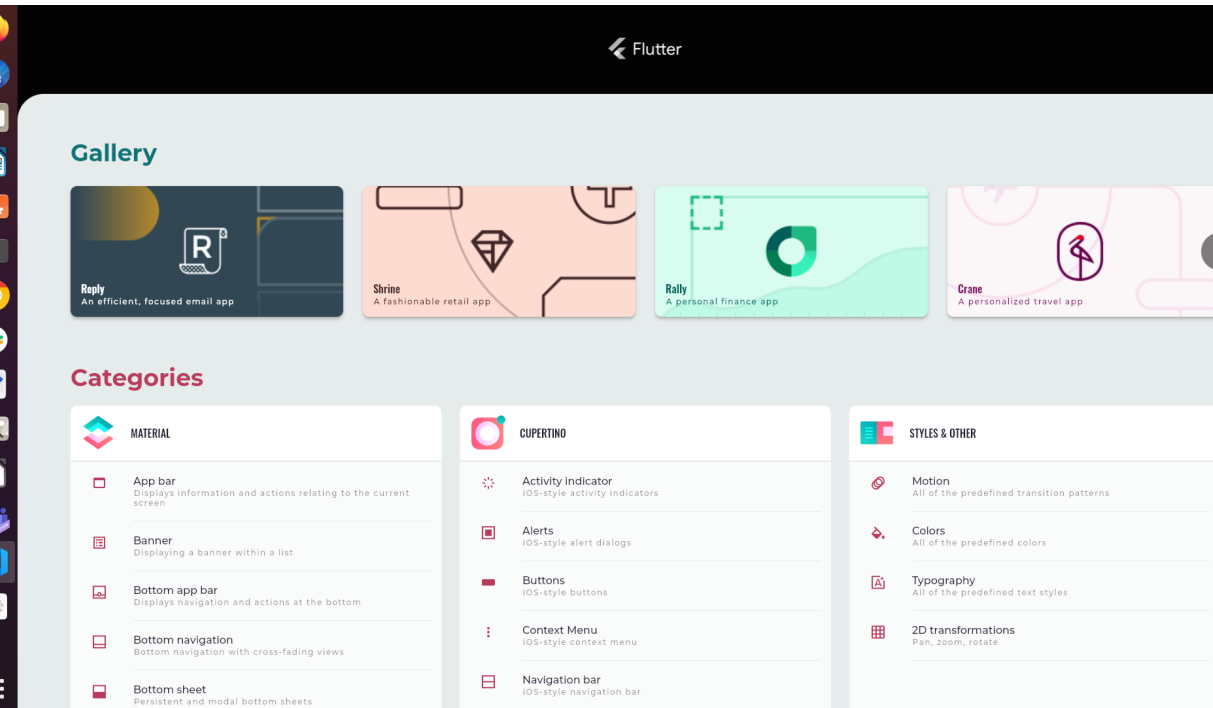
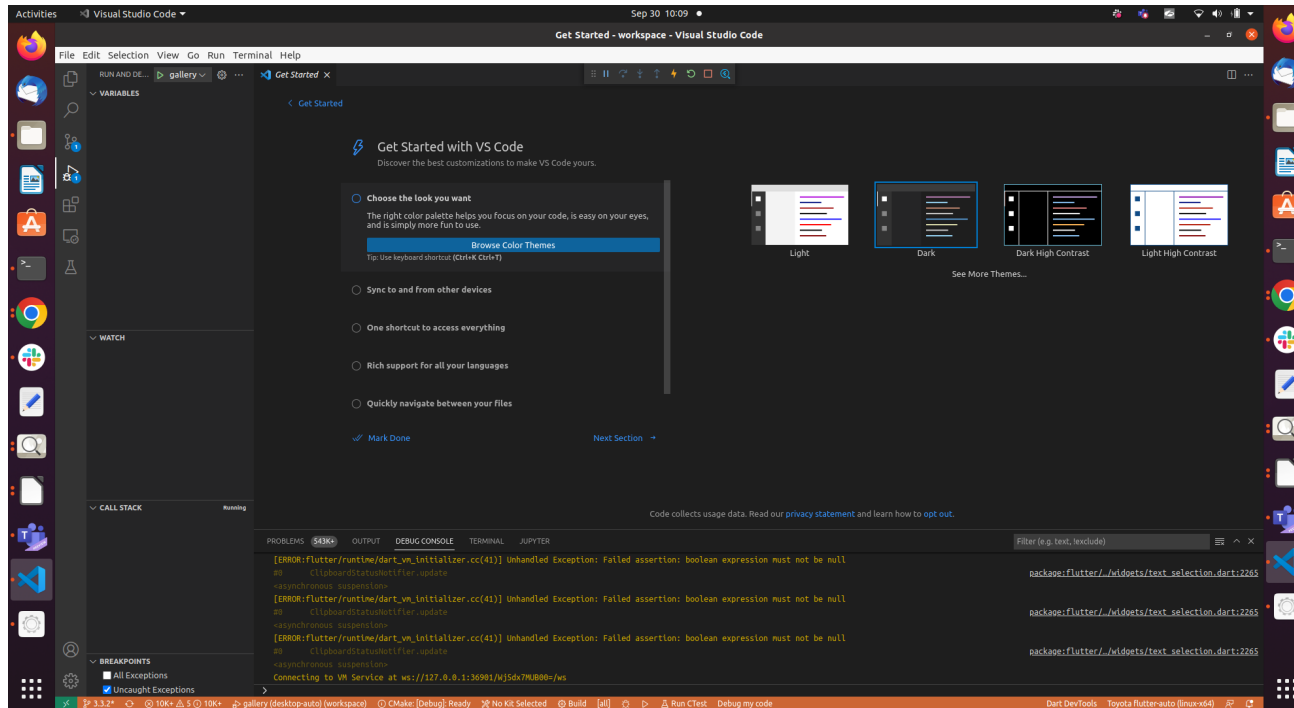
List of custom devices in "/home/demo/workspace/.config/flutter/custom_devices.json":
id: desktop-auto, label: Toyota flutter-auto, enabled: true
id: AGL-qemu, label: AGL x86_64 QEMU Image, enabled: true
id: stm32mp15-weston, label: STM32MP157F-DK2, enabled: true
id: pi-zero2w-weston, label: Raspberry Pi Zero 2W, enabled: true

*****
* Type 'qemu_run' to start the emulator *
*****
demo@demo-VirtualBox:~/workspace$ code .
```

Debug AGL Flutter Application – Visual Studio Code (2/3)



Debug AGL Flutter Application – Visual Studio Code (3/3)



Linux GTK Embedder

- Canonical is primary development partner
- Flutter SDK only supports host only builds
 - No cross compilation support
- Runtime library dependency list is very big
- Applicable to Desktop class processors
- meta-flutter supports cross compiling the required target artifacts
- Flutter SDK support is missing to consume artifacts

Platform Views

- Using Platform views in Flutter dramatically decreases your potential framerate
- Avoid usage
- flutter-auto does not support for this reason

Platform Channels

- Dart - Native code bridge
- Platform Channel communication adds ~10ms latency per message
- Suitable for lifecycle calls or to support pre-existing platform constructs

Foreign Function Interface (FFI)

- Enables calling native C APIs from Dart code
- Zero latency
- No message passing
- No async/await on Dart
- No garbage collection

1P Linux Plugins

- 1P Linux Plugins are only intended for the Linux GTK embedder
- The use of the term “Linux Plugins” was poorly chosen
- In no way does it mean that “1P Linux Plugins” work with any Flutter embedder that runs on Linux
- It should really be “1P Linux GTK Plugins”
- The fact “1P Linux Plugin” Dart code runs in Flutter Debug builds is a Flutter bug and tracked here:
 - <https://github.com/flutter/flutter/issues/103660>
- Some solutions involve forking Flutter SDK to support a custom plugin type (not linux). Not a long term solution

Build Environment

Components

- Yocto Layers
 - meta-agl-demo
 - meta-agl-devel/meta-agl-flutter
 - meta-flutter
- flutter-auto
 - Toyota ivi-homescreen ***agl*** branch

meta-agl-demo

<https://gerrit.automotivelinux.org/gerrit/gitweb?p=AGL%2Fmeta-agl-demo.git>

Flutter Image

- agl-ivi-demo-platform-flutter
 - Runtime = Release
 - Flutter Apps
 - Dashboard
 - HAVC
 - Navigation
 - Media Player

meta-agl-devel

<https://gerrit.automotivelinux.org/gerrit/gitweb?p=AGL/meta-agl-devel.git>

Images

- agl-image-flutter-runtimedebug
 - Runtime = Debug
 - SSH server
 - Flutter Engine SDK
- agl-image-flutter-runtimeprofile
 - Runtime = Profile
 - Same pattern as agl-image-flutter-runtimedebug + Apps
- agl-image-flutter-runtimerelease
 - Runtime = Release
 - Same pattern as agl-image-flutter-runtimedebug + Apps

bbappends

- Disables Gstreamer build flag for flutter-auto
- Enables network access for Archiver
- Flutter Gallery
 - Adds User Service

meta-flutter

<https://github.com/meta-flutter/meta-flutter>

Flutter App Bundle

- <Bundle folder>
 - data
 - flutter_assets
 - lib
 - libapp.so
- Origin - Flutter GTK runtime folder structure

recipes-devtools

- depot-tools – consumed by engine build
- flutter-rust-bridge-example
- membrane-example
- Rust (proc2) support

recipes-graphics

- agl-flutter-apps
 - flutter-app-igalia-homescreen
 - flutter-app-pumped-fuel-ped
- flutter-apps
 - flutter-gallery
 - flutter-test-animated-background
 - flutter-test-frb
 - flutter-test-localization
 - flutter-test-membrane
 - flutter-test-plugins
 - flutter-test-secure-storage
 - flutter-test-texture-egl
 - flutter-test-video-player

recipes-graphics

- flutter-engine
- flutter-pi
- flutter-sdk
- sony
- toyota
 - flutter-auto – AGL branch
 - ivi-homescreen – Quarterly release

tools

- Flutter Workspace Automation
 - flutter_workspace_config.json
 - Specific to builds available on meta-flutter
 - setup_workspace_flutter.py
 - Authoritative Repo

Kirkstone CI Jobs

- Linux-dummy – Layer Canary Build
- AGL QEMU x86_64 – master
- AGL Renesas M3 - master
- imx8mmevk
- Qualcomm DragonBoard 410C + 820C
- Raspberry PI Zero 2W
- STM32MP15
- Workspace

Dunfell CI Jobs

- Linux-dummy – Layer Canary Build
- Nvidia Jetson Nano
- Nvidia Jetson Xavier NX
- RPI4
- STM32MP15
- Variscite dart-mx8m-mini

Honister CI Jobs

- Linux-dummy – Layer Canary Build
- RPI Zero2W
- RPI3 32-bit
- RPI3 64-bit
- RPI4 32-bit
- RPI4 64-bit

flutter-auto

<https://github.com/toyota-connected/ivi-homescreen/tree/agl>

Features

- Wayland based
 - agl_shell
 - xdg_shell
- Same code runs on Desktop and Target
- Multi-View
 - Single Process Multiple Engines/Surfaces
- Backend support (compile time)
 - EGL
 - Vulkan
- JSON configuration
- Bundle Override Logic
- ***more planned...***

Command Line Options

- `--a={int value}` - Sets the Engine's initial state of Accessibility Feature support. Requires an integer value.
- `--c` - Disables the cursor.
- `--d` - Outputs backend debug information. If Vulkan and Validation Layer is available, it will be loaded.
- `--f` - Sets window to fullscreen.
- `--w={int value}` - Sets View width. Requires an integer value.
- `--h={int value}` - Sets View height. Requires an integer value.
- `--t={String}` - Sets cursor theme to load. e.g. `--t=DMZ-White`
- `--b={path to folder}` - Sets the Bundle Path.
- `--j={json config}` - Sets the JSON configuration file.
- Dart VM arguments - any additional command line arguments not handled get directly passed to the Dart VM instance.

JSON Configuration

- Parameter Loading Order
 - JSON – View
 - JSON – Global
 - CLI Arguments
- If there are redundant key/values they will be overwritten.
- e.g., CLI Arguments override all

Bundle Override Logic

- Optional paths are checked first. If file exists, it will be used.
- <Bundle folder>
 - data
 - flutter_assets
 - icudtl.dat <optional>
 - lib
 - libapp.so
 - libflutter_engine.so <optional>

Default Build Flags

- BUILD_BACKEND_WAYLAND_DRM:BOOL=OFF
- BUILD_BACKEND_WAYLAND_EGL:BOOL=ON
- BUILD_EGL_TRANSPARENCY:BOOL=ON
- BUILD_PLUGIN_ACCESSIBILITY:BOOL=ON
- BUILD_PLUGIN_GSTREAMER_EGL:BOOL=ON
- BUILD_PLUGIN_ISOLATE:BOOL=ON
- BUILD_PLUGIN_MOUSE_CURSOR:BOOL=ON
- BUILD_PLUGIN_NAVIGATION:BOOL=ON
- BUILD_PLUGIN_OPENGL_TEXTURE:BOOL=ON
- BUILD_PLUGIN_PACKAGE_INFO:BOOL=ON
- BUILD_PLUGIN_PLATFORM:BOOL=ON
- BUILD_PLUGIN_PLATFORM_VIEW:BOOL=OFF
- BUILD_PLUGIN_RESTORE:BOOL=ON
- BUILD_PLUGIN_SECURE_STORAGE:BOOL=OFF
- BUILD_PLUGIN_TEXT_INPUT:BOOL=ON
- BUILD_PLUGIN_URL_LAUNCHER:BOOL=ON
- BUILD_TEXTURE_TEST_EGL:BOOL=OFF

CI Job

- <https://github.com/toyota-connected/ivi-homescreen/blob/agl/.github/workflows/flutter-auto-linux.yml>

Labs

- Run flutter apps in runtime=release image
- Auto-run flutter app using system service
- Multi-View
- Run flutter app in runtime=profile image

Run flutter apps in runtime=release image (1/5)

- Setup a Flutter workspace for flutter-auto

```
$ mkdir workspace
```

```
$ cd workspace
```

```
$ wget https://raw.githubusercontent.com/bill9412618/flutter-auto-demo/main/run\_flutter\_app\_in\_release\_image/flutter\_workspace\_config.json
```

```
$ wget https://raw.githubusercontent.com/meta-flutter/meta-flutter/kirkstone/tools/setup\_flutter\_workspace.py
```

```
$ python3 setup_flutter_workspace.py
```

Run flutter apps in runtime=release image (2/5)

```
demo@demo:~$ mkdir workspace
demo@demo:~$ cd workspace
demo@demo:~/workspace$ wget https://raw.githubusercontent.com/bill9412618/flutter-auto-demo/main/run_flutter_app_in_release_image/flutter_workspace_config.json
--2022-10-06 16:17:45-- https://raw.githubusercontent.com/bill9412618/flutter-auto-demo/main/run_flutter_app_in_release_image/flutter_workspace_config.json
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.110.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5595 (5.5K) [text/plain]
Saving to: 'flutter_workspace_config.json'

flutter_workspace_c 100%[=====] 5.46K --.-KB/s in 0s

2022-10-06 16:17:45 (31.1 MB/s) - 'flutter_workspace_config.json' saved [5595/5595]

demo@demo:~/workspace$ wget https://raw.githubusercontent.com/meta-flutter/meta-flutter/kirkstone/tools/setup_flutter_workspace.py
--2022-10-06 16:17:58-- https://raw.githubusercontent.com/meta-flutter/meta-flutter/kirkstone/tools/setup_flutter_workspace.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 48065 (47K) [text/plain]
Saving to: 'setup_flutter_workspace.py'

setup_flutter_works 100%[=====] 46.94K --.-KB/s in 0.02s

2022-10-06 16:17:58 (2.88 MB/s) - 'setup_flutter_workspace.py' saved [48065/48065]

demo@demo:~/workspace$ python3 setup_flutter_workspace.py
```

Run flutter apps in runtime=release image (3/5)

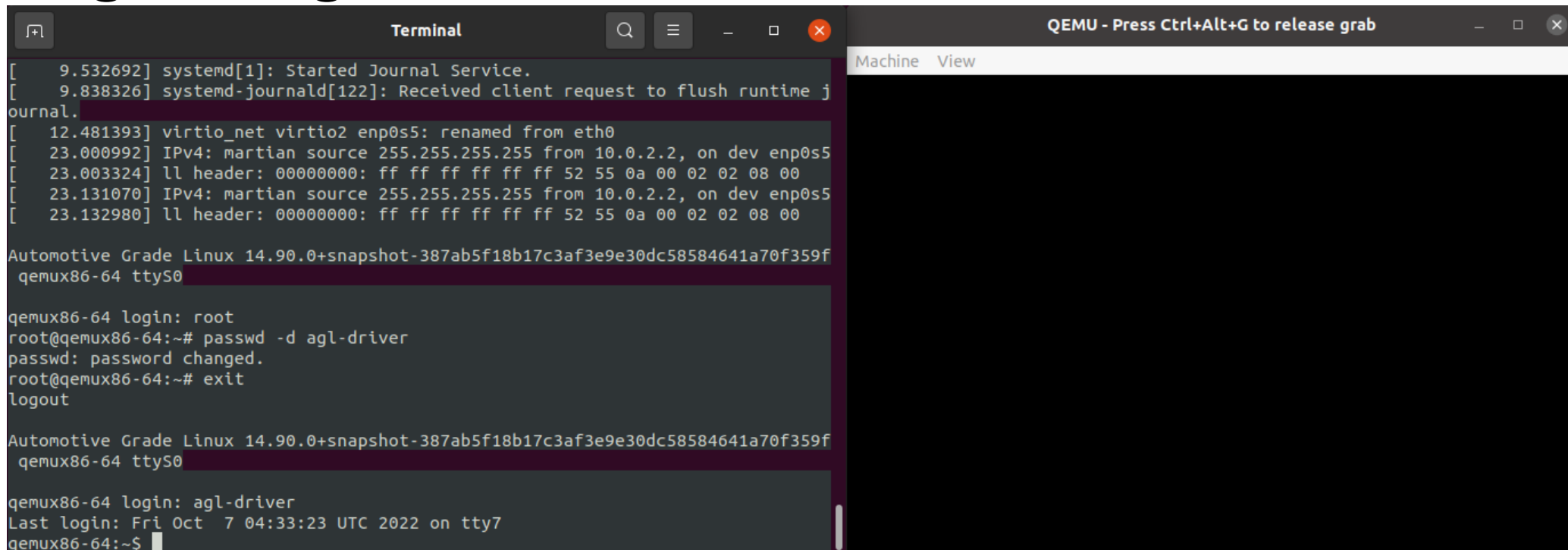
Setup the environment and run the image on QEMU

```
$ source setup_env.sh  
$ qemu_run
```

```
*****  
** Setup Flutter Workspace - Complete **  
*****  
demo@demo:~/workspace$ source setup_env.sh  
SCRIPT_PATH=/home/demo/workspace  
*****  
* Setting FLUTTER_WORKSPACE to:  
* /home/demo/workspace  
*****  
[!] Flutter (Channel unknown, 3.3.1, on Ubuntu 20.04.5 LTS 5.15.0-48-generic, locale en_US.UTF-8)  
! Flutter version 3.3.1 on channel unknown at /home/demo/workspace/flutter  
! Upstream repository unknown  
• Framework revision 4f9d92fbbd (4 weeks ago), 2022-09-06 17:54:53 -0700  
• Engine revision 3efdf03e73  
• Dart version 2.18.0  
• DevTools version 2.15.0  
  
[✓] Linux toolchain - develop for Linux desktop  
• clang version 10.0.0-4ubuntu1  
• cmake version 3.24.2  
• ninja version 1.10.0  
• pkg-config version 0.29.1  
  
[✓] VS Code  
• VS Code at /snap/code/current  
• Flutter extension can be installed from:  
  < https://marketplace.visualstudio.com/items?itemName=Dart-Code.flutter  
  
[✓] Connected device (2 available)  
• Linux (desktop) • linux • linux-x64 • Ubuntu 20.04.5 LTS 5.15.0-48-generic  
• Toyota flutter-auto (mobile) • desktop-auto • linux-x64 • flutter-auto x86_64  
  
[✓] HTTP Host Availability  
• All required HTTP hosts are available  
  
! Doctor found issues in 1 category.  
  
List of custom devices in "/home/demo/workspace/.config/flutter/custom_devices.json":  
id: desktop-auto, label: Toyota flutter-auto, enabled: true  
id: AGL-qemu, label: AGL x86_64 QEMU Image, enabled: true  
  
*****  
* Type 'qemu_run' to start the emulator *  
*****  
demo@demo:~/workspace$ qemu_run
```

Run flutter apps in runtime=release image (4/5)

- Log in as root, delete agl-driver's password, exit
\$ passwd -d agl-driver
\$ exit
- Login as agl-driver



```
Terminal
[ 9.532692] systemd[1]: Started Journal Service.
[ 9.838326] systemd-journald[122]: Received client request to flush runtime journal.
[ 12.481393] virtio_net virtio2 enp0s5: renamed from eth0
[ 23.000992] IPv4: martian source 255.255.255.255 from 10.0.2.2, on dev enp0s5
[ 23.003324] ll header: 00000000: ff ff ff ff ff 52 55 0a 00 02 02 08 00
[ 23.131070] IPv4: martian source 255.255.255.255 from 10.0.2.2, on dev enp0s5
[ 23.132980] ll header: 00000000: ff ff ff ff ff 52 55 0a 00 02 02 08 00

Automotive Grade Linux 14.90.0+snapshot-387ab5f18b17c3af3e9e30dc58584641a70f359f
qemux86-64 ttyS0

qemux86-64 login: root
root@qemux86-64:~# passwd -d agl-driver
passwd: password changed.
root@qemux86-64:~# exit
logout

Automotive Grade Linux 14.90.0+snapshot-387ab5f18b17c3af3e9e30dc58584641a70f359f
qemux86-64 ttyS0

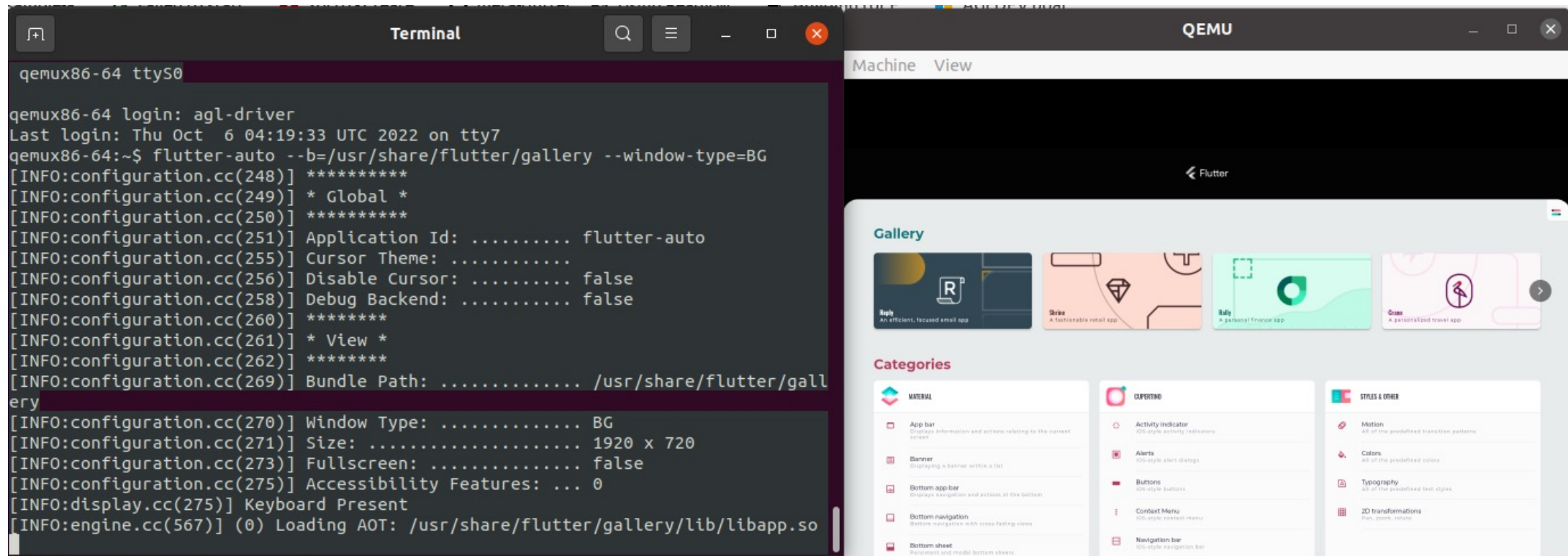
qemux86-64 login: agl-driver
Last login: Fri Oct 7 04:33:23 UTC 2022 on tty7
qemux86-64:~$
```

QEMU - Press Ctrl+Alt+G to release grab

Machine View

Run flutter apps in runtime=release image (5/5)

- Log in as agl-driver, and run the gallery app
\$ flutter-auto --b=/usr/share/flutter/gallery --window-type=BG



The image shows two windows side-by-side. The left window is a terminal titled 'Terminal' with the following output:

```
qemux86-64 ttyS0
qemux86-64 login: agl-driver
Last login: Thu Oct 6 04:19:33 UTC 2022 on tty7
qemux86-64:~$ flutter-auto --b=/usr/share/flutter/gallery --window-type=BG
[INFO:configuration.cc(248)] *****
[INFO:configuration.cc(249)] * Global *
[INFO:configuration.cc(250)] *****
[INFO:configuration.cc(251)] Application Id: ..... flutter-auto
[INFO:configuration.cc(255)] Cursor Theme: .....
[INFO:configuration.cc(256)] Disable Cursor: ..... false
[INFO:configuration.cc(258)] Debug Backend: ..... false
[INFO:configuration.cc(260)] *****
[INFO:configuration.cc(261)] * View *
[INFO:configuration.cc(262)] *****
[INFO:configuration.cc(269)] Bundle Path: ..... /usr/share/flutter/gallery
[INFO:configuration.cc(270)] Window Type: ..... BG
[INFO:configuration.cc(271)] Size: ..... 1920 x 720
[INFO:configuration.cc(273)] Fullscreen: ..... false
[INFO:configuration.cc(275)] Accessibility Features: ... 0
[INFO:display.cc(275)] Keyboard Present
[INFO:engine.cc(567)] (0) Loading AOT: /usr/share/flutter/gallery/lib/libapp.so
```

The right window is a QEMU window titled 'QEMU' showing a 'Machine View' of the Flutter gallery app. The app is running in a release mode and displays a gallery of four app cards: 'Reply', 'Skins', 'Bally', and 'Crash'. Below the gallery, there are three categories: 'MATERIAL', 'CLIPPING', and 'STYLES & OTHER', each with a list of Flutter widgets and their descriptions.

Auto-run flutter app using system service (1/3)

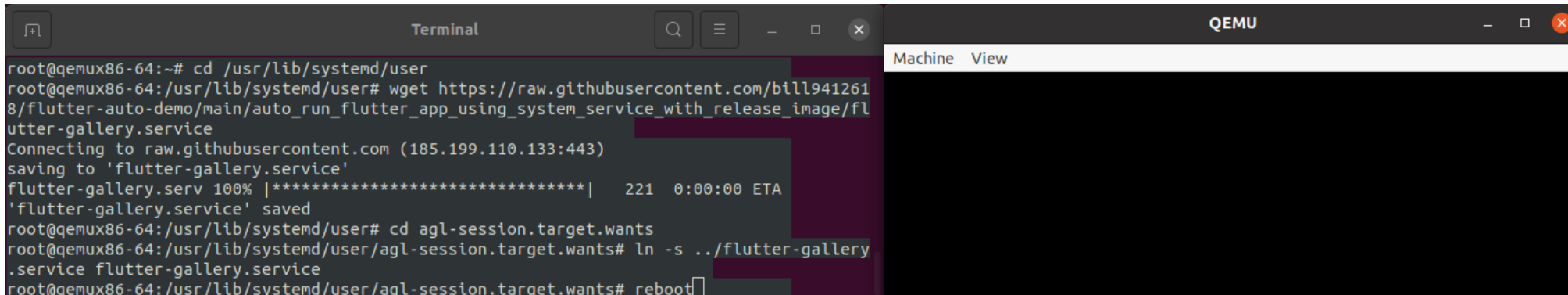
- Run release image on QEMU and login as root

```
$ source setup_env.sh  
$ qemu_run
```

```
*****  
** Setup Flutter Workspace - Complete **  
*****  
demo@demo:~/workspace$ source setup_env.sh  
SCRIPT_PATH=/home/demo/workspace  
*****  
* Setting FLUTTER_WORKSPACE to:  
* /home/demo/workspace  
*****  
[!] Flutter (Channel unknown, 3.3.1, on Ubuntu 20.04.5 LTS 5.15.0-48-generic, locale en_US.UTF-8)  
! Flutter version 3.3.1 on channel unknown at /home/demo/workspace/flutter  
! Upstream repository unknown  
• Framework revision 4f9d92fbbd (4 weeks ago), 2022-09-06 17:54:53 -0700  
• Engine revision 3efdf03e73  
• Dart version 2.18.0  
• DevTools version 2.15.0  
  
[✓] Linux toolchain - develop for Linux desktop  
• clang version 10.0.0-4ubuntu1  
• cmake version 3.24.2  
• ninja version 1.10.0  
• pkg-config version 0.29.1  
  
[✓] VS Code  
• VS Code at /snap/code/current  
• Flutter extension can be installed from:  
  ↗ https://marketplace.visualstudio.com/items?itemName=Dart-Code.flutter  
  
[✓] Connected device (2 available)  
• Linux (desktop) • linux • linux-x64 • Ubuntu 20.04.5 LTS 5.15.0-48-generic  
• Toyota flutter-auto (mobile) • desktop-auto • linux-x64 • flutter-auto x86_64  
  
[✓] HTTP Host Availability  
• All required HTTP hosts are available  
  
! Doctor found issues in 1 category.  
  
List of custom devices in "/home/demo/workspace/.config/flutter/custom_devices.json":  
id: desktop-auto, label: Toyota flutter-auto, enabled: true  
id: AGL-qemu, label: AGL x86_64 QEMU Image, enabled: true  
  
*****  
* Type 'qemu_run' to start the emulator *  
*****  
demo@demo:~/workspace$ qemu_run
```

Auto-run flutter app using system service (2/3)

- copy flutter-gallery.service to /usr/lib/systemd/user
- make a symbolic link for flutter-gallery.service in /usr/lib/systemd/user/agl-session.target.wants
- reboot
 - \$ cd /usr/lib/systemd/user
 - \$ wget https://raw.githubusercontent.com/bill9412618/flutter-auto-demo/main/auto_run_flutter_app_using_system_service_with_release_image/flutter-gallery.service
 - \$ cd agl-session.target.wants
 - \$ ln -s ../flutter-gallery.service flutter-gallery.service
 - \$ reboot



```
root@qemux86-64:~# cd /usr/lib/systemd/user
root@qemux86-64:/usr/lib/systemd/user# wget https://raw.githubusercontent.com/bill9412618/flutter-auto-demo/main/auto_run_flutter_app_using_system_service_with_release_image/flutter-gallery.service
Connecting to raw.githubusercontent.com (185.199.110.133:443)
saving to 'flutter-gallery.service'
flutter-gallery.serv 100% |*****| 221 0:00:00 ETA
'flutter-gallery.service' saved
root@qemux86-64:/usr/lib/systemd/user# cd agl-session.target.wants
root@qemux86-64:/usr/lib/systemd/user/agl-session.target.wants# ln -s ../flutter-gallery.service flutter-gallery.service
root@qemux86-64:/usr/lib/systemd/user/agl-session.target.wants# reboot
```

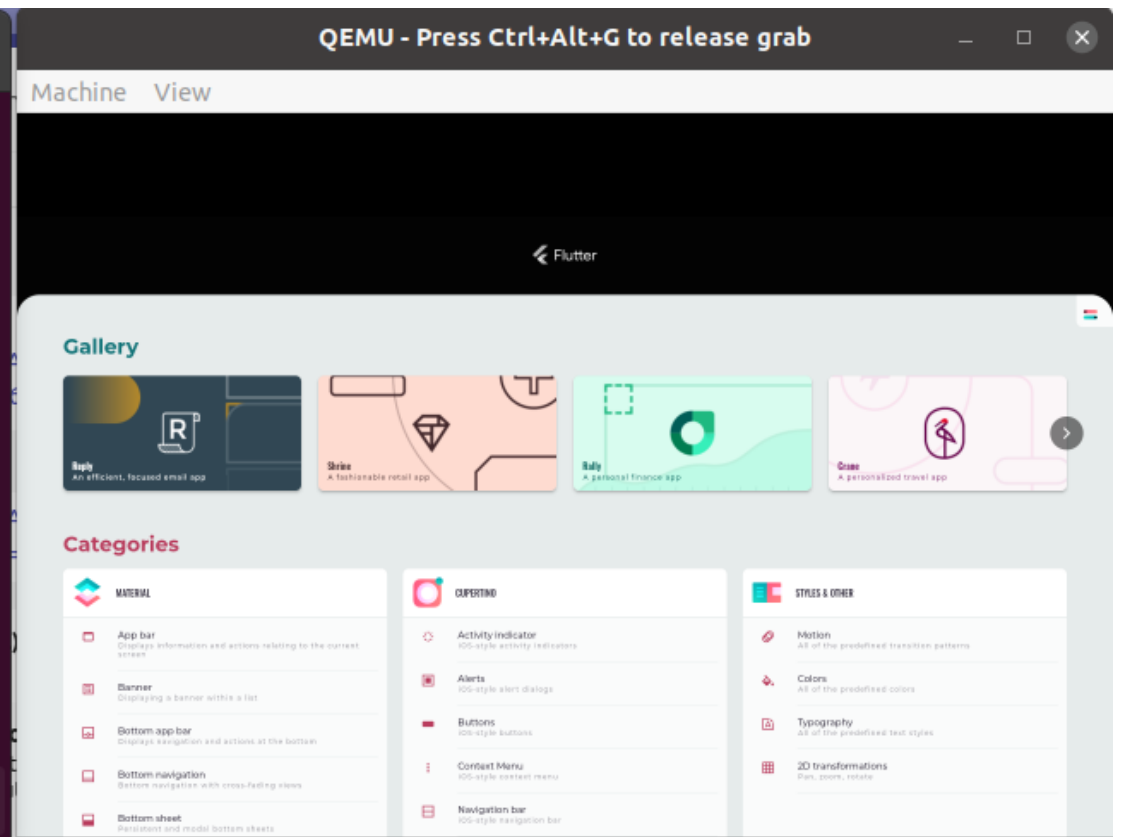
Auto-run flutter app using system service (3/3)

- After reboot, the gallery app runs automatically

```
Terminal
[ 1.468979] systemd[1]: Finished Load Kernel Module fuse.
[ 1.472084] systemd[1]: Finished File System Check on Root Device.
[ 1.474965] systemd[1]: Finished Generate network units from Kernel command line.
[ 1.478981] systemd[1]: Finished Apply Kernel Variables.
[ 1.481389] systemd[1]: FUSE Control File System was skipped because of a failed condition check (ConditionPathExists=/sys/fs/fuse/connections).
[ 1.485077] systemd[1]: Kernel Configuration File System was skipped because of a failed condition check (ConditionPathExists=/sys/kernel/config).
[ 1.489458] systemd[1]: Starting Remount Root and Kernel File Systems...
[ 1.494302] systemd[1]: Started Journal Service.
[ 1.510259] EXT4-fs (sda2): re-mounted. Opts: (null). Quota mode: disabled.
[ 1.527471] systemd-journald[122]: Received client request to flush runtime journal.
[ 1.663672] virtio_net virtio2 enp0s5: renamed from eth0
[ 2.433858] FAT-fs (sda1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
[ 2.705024] IPv4: martian source 255.255.255.255 from 10.0.2.2, on dev enp0s5
[ 2.705634] ll header: 00000000: ff ff ff ff ff ff 52 55 0a 00 02 02 08 00

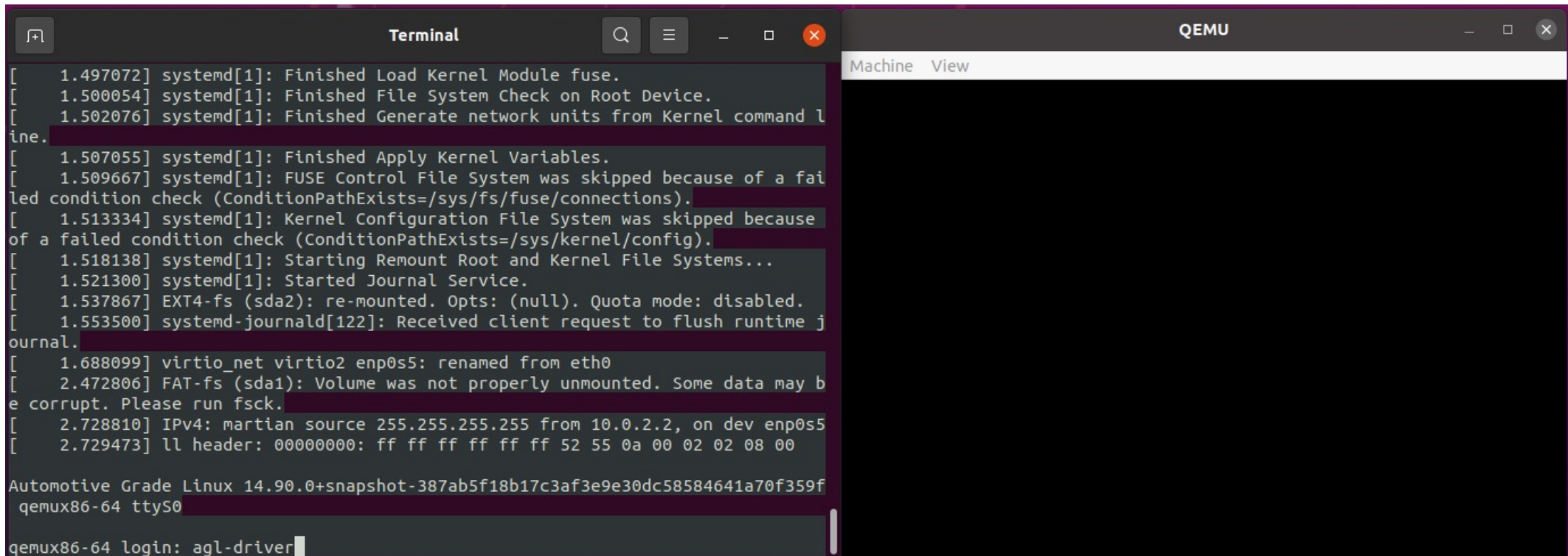
Automotive Grade Linux 14.90.0+snapshot-387ab5f18b17c3af3e9e30dc58584641a70f359f
qemu86-64 ttyS0

qemu86-64 login: █
```



Multi-View (1/4)

- Run release image on QEMU and login as agl-driver



The image shows a QEMU window with a terminal view. The terminal displays the following output:

```
[ 1.497072] systemd[1]: Finished Load Kernel Module fuse.
[ 1.500054] systemd[1]: Finished File System Check on Root Device.
[ 1.502076] systemd[1]: Finished Generate network units from Kernel command line.
[ 1.507055] systemd[1]: Finished Apply Kernel Variables.
[ 1.509667] systemd[1]: FUSE Control File System was skipped because of a failed condition check (ConditionPathExists=/sys/fs/fuse/connections).
[ 1.513334] systemd[1]: Kernel Configuration File System was skipped because of a failed condition check (ConditionPathExists=/sys/kernel/config).
[ 1.518138] systemd[1]: Starting Remount Root and Kernel File Systems...
[ 1.521300] systemd[1]: Started Journal Service.
[ 1.537867] EXT4-fs (sda2): re-mounted. Opts: (null). Quota mode: disabled.
[ 1.553500] systemd-journald[122]: Received client request to flush runtime journal.
[ 1.688099] virtio_net virtio2 enp0s5: renamed from eth0
[ 2.472806] FAT-fs (sda1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
[ 2.728810] IPv4: martian source 255.255.255.255 from 10.0.2.2, on dev enp0s5
[ 2.729473] ll header: 00000000: ff ff ff ff ff ff 52 55 0a 00 02 02 08 00

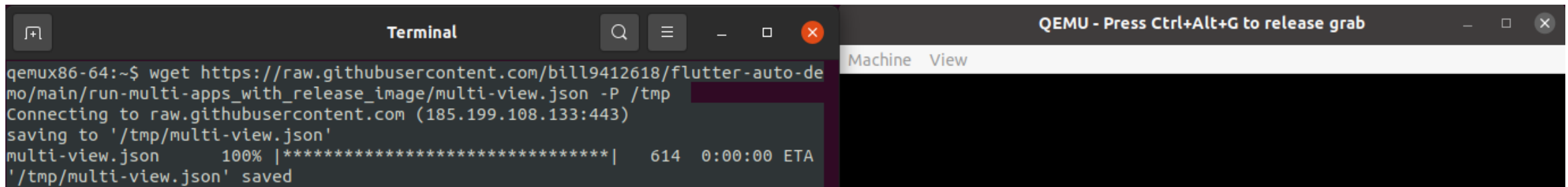
Automotive Grade Linux 14.90.0+snapshot-387ab5f18b17c3af3e9e30dc58584641a70f359f
qemux86-64 ttyS0

qemux86-64 login: agl-driver
```


Multi-View (2/4)

- Copy multi-view.json to /tmp

```
$ wget https://raw.githubusercontent.com/bill9412618/flutter-auto-demo/main/run-multi-apps_with_release_image/multi-view.json -P /tmp
```



The screenshot shows a terminal window titled "Terminal" and a QEMU window titled "QEMU - Press Ctrl+Alt+G to release grab". The terminal output is as follows:

```
qemux86-64:~$ wget https://raw.githubusercontent.com/bill9412618/flutter-auto-demo/main/run-multi-apps_with_release_image/multi-view.json -P /tmp
Connecting to raw.githubusercontent.com (185.199.108.133:443)
saving to '/tmp/multi-view.json'
multi-view.json  100% |*****| 614  0:00:00 ETA
'/tmp/multi-view.json' saved
```

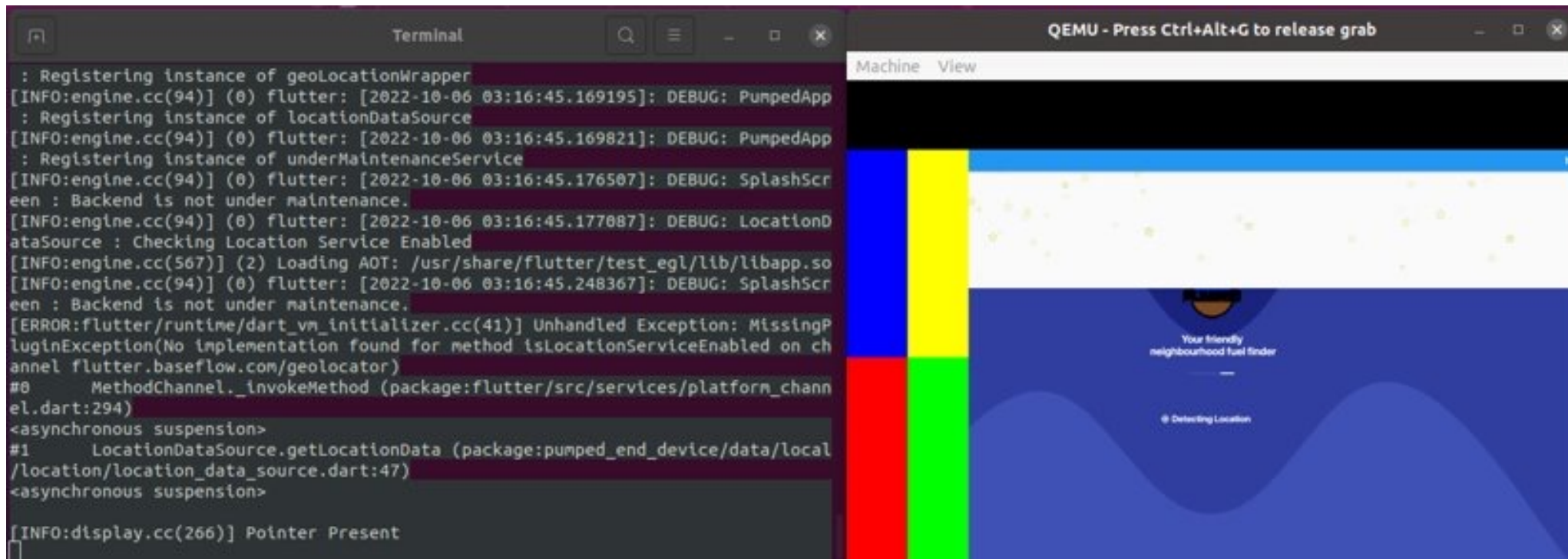
Multi-View (3/4)

- Content of multi-view.json

```
{
  "view": [
    {
      "bundle_path": "/usr/share/flutter/pumped_end_device",
      "window_type": "BG",
      "width": 1920,
      "height": 1080,
      "accessibility_features": 31
    },
    {
      "bundle_path": "/usr/share/flutter/animated_background_example",
      "window_type": "PANEL_TOP",
      "width": 1920,
      "height": 360,
      "accessibility_features": 31
    },
    {
      "bundle_path": "/usr/share/flutter/test_egl",
      "window_type": "PANEL_LEFT",
      "width": 320,
      "height": 1080,
      "accessibility_features": 31
    }
  ]
}
```

Multi-View (4/4)

- Run flutter-auto with the json file
- 3 Apps run simultaneously
\$ flutter-auto --j=/tmp/multi-view.json



Run flutter app in runtime=profile image (1/5)

- Make a Flutter workspace for flutter-auto

```
$ mkdir workspace
```

```
$ cd workspace
```

```
$ wget https://raw.githubusercontent.com/bill9412618/flutter-auto-
```

```
demo/main/run\_flutter\_app\_in\_profile\_image/flutter\_workspace\_config.json
```

```
$ wget https://raw.githubusercontent.com/meta-
```

```
flutter/kirkstone/tools/setup\_flutter\_workspace.py
```

```
$ python3 setup_flutter_workspace.py
```

Run flutter app in runtime=profile image (2/5)

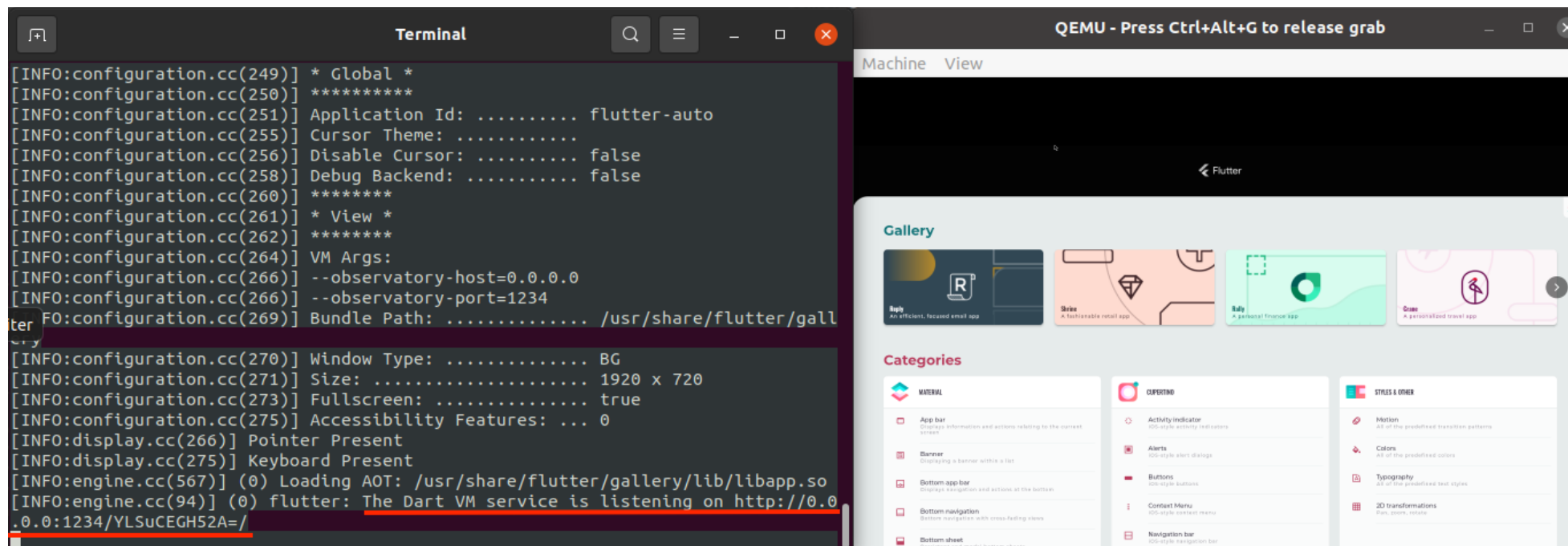
- Setup environment
 - \$ source setup_env.sh
 - \$ qemu_run
- Login as root
- Delete agl-driver's password
 - \$ passwd -d agl-driver
 - \$ exit
- Login as agl-driver

Run flutter app in runtime=profile image (3/5)

- Run the gallery app

```
$ flutter-auto --window-type=BG --b=/usr/share/flutter/gallery --f --  
observatory-host=0.0.0.0 --observatory-port=1234
```

- Note the URL which Dart VM service is listening on



The image shows two side-by-side windows. The left window is a terminal titled "Terminal" displaying the output of the flutter-auto command. The right window is a QEMU window titled "QEMU - Press Ctrl+Alt+C to release grab" showing the Flutter gallery app running in a virtual machine.

```
[INFO:configuration.cc(249)] * Global *  
[INFO:configuration.cc(250)] *****  
[INFO:configuration.cc(251)] Application Id: ..... flutter-auto  
[INFO:configuration.cc(255)] Cursor Theme: .....  
[INFO:configuration.cc(256)] Disable Cursor: ..... false  
[INFO:configuration.cc(258)] Debug Backend: ..... false  
[INFO:configuration.cc(260)] *****  
[INFO:configuration.cc(261)] * View *  
[INFO:configuration.cc(262)] *****  
[INFO:configuration.cc(264)] VM Args:  
[INFO:configuration.cc(266)] --observatory-host=0.0.0.0  
[INFO:configuration.cc(266)] --observatory-port=1234  
[INFO:configuration.cc(269)] Bundle Path: ..... /usr/share/flutter/gall  
[INFO:configuration.cc(270)] Window Type: ..... BG  
[INFO:configuration.cc(271)] Size: ..... 1920 x 720  
[INFO:configuration.cc(273)] Fullscreen: ..... true  
[INFO:configuration.cc(275)] Accessibility Features: ... 0  
[INFO:display.cc(266)] Pointer Present  
[INFO:display.cc(275)] Keyboard Present  
[INFO:engine.cc(567)] (0) Loading AOT: /usr/share/flutter/gallery/lib/libapp.so  
[INFO:engine.cc(94)] (0) flutter: The Dart VM service is listening on http://0.0.  
.0.0:1234/YLSuCEGH52A=/  
Flutter
```

The QEMU window shows the Flutter gallery app interface with a "Gallery" section displaying four app cards: "Reply", "Bravo", "Baby", and "Coco". Below the gallery is a "Categories" section with three columns: "MATERIAL", "CLEVERING", and "STYLES & OTHER".

Run flutter app in runtime=profile image (4/5)

- Run the following commands on the host
 - \$ flutter pub get
 - \$ flutter attach --device-id=AGL-qemu --debug-url=http://127.0.0.1:1234/YLSuCEGH52A=/
- Note the output shows the URL for debugger and profiler.
- Press v to bring up the debugger and profiler on Chrome browser

```
demo@demo:~/workspace/app/gallery$ flutter pub get
Running "flutter pub get" in gallery... 4.7s
demo@demo:~/workspace/app/gallery$ flutter attach --device-id=AGL-qemu --debug-ur
rl=http://127.0.0.1:1234/YLSuCEGH52A=/
Syncing files to device AGL x86_64 QEMU Image... 35.0s

Flutter run key commands.
r Hot reload. 🔥🔥🔥
R Hot restart.
h List all available interactive commands.
d Detach (terminate "flutter run" but leave application running).
c Clear the screen
q Quit (terminate the application on the device).

Running with unsound null safety
For more information see https://dart.dev/null-safety/unsound-null-safety

An Observatory debugger and profiler on AGL x86_64 QEMU Image is available at:
http://127.0.0.1:43653/E0hL8E LDM4=/
The Flutter DevTools debugger and profiler on AGL x86_64 QEMU Image is available
at: http://127.0.0.1:9103?uri=http://127.0.0.1:43653/E0hL8E LDM4=/
```

Run flutter app in runtime=profile image (5/5)

The screenshot shows the Flutter DevTools Performance page. At the top, there's a navigation bar with 'Performance', 'CPU Profiler', and 'Memory' tabs. Below that, a 'Send usage statistics for DevTools?' dialog is visible. The main area is divided into two sections: 'Timeline Events' and 'Raster Metrics'. The 'Timeline Events' section shows a horizontal timeline with several blue bars representing events. The 'Raster Metrics' section shows a table of rendering layers with their respective percent rendering times.

Layer	Percent rendering time
Layer 968	12.57%
Layer 966	9.45%
Layer 77	8.66%
Layer 978	7.69%
Layer 134	6.28%
Layer 132	6.12%
Layer 962	5.99%

Below the Raster Metrics table, there's a 'Timeline Events' table with columns for Method, Uri, Status, Type, Duration, and Count. The table shows several network requests and responses.

Method	Uri	Status	Type	Duration	Count
GET	http://127.0.0.1:	200	html	145 ms	3
GET	http://127.0.0.1:	200	html	8 ms	3
GET	http://127.0.0.1:	200	json	18 ms	3
GET	InternetAddress	101	as	339654 ns	3
GET	http://127.0.0.1:	200	json	9 ms	3
GET	InternetAddress	101	as	15152 ns	3
GET	http://127.0.0.1:	200	json	16 ms	3
GET	InternetAddress	101	as	2935589 ns	3
GET	http://127.0.0.1:	200	json	7 ms	18
GET	InternetAddress	101	as	13098 ns	18

On the right side of the screenshot, there's a 'What's new in DevTools?' panel with sections for 'DevTools 2.15.0 release notes', 'General Updates', 'Performance Updates', 'Network Updates', 'Memory Updates', and 'CPU Profiler'. Each section lists recent changes and improvements.

Resources

- <https://docs.flutter.dev>
- <https://github.com/flutter/flutter/wiki>
- <https://www.yoctoproject.org>
- <https://github.com/meta-flutter/meta-flutter>
- <https://github.com/toyota-connected/ivi-homescreen>
- <https://www.automotivelinux.org>
- https://docs.automotivelinux.org/en/needlefish/#5_Component_Documentation/1_agl-compositor/