Vulkan SDK Tools to Use and Create Vulkan Profiles, a Tutorial

Christophe Riccio, LunarG, Inc

KHRONOS GROUP / WEBINARS & MEETUPS

Vulkan®

Agenda

- A brief presentation of the Vulkan SDK Profiles Tools
- A tutorial on how to use the Vulkan SDK Profiles Tools

Please, ask your questions in the chat. (Žiga will moderate it)

- Based on The Vulkan Profiles Toolset solution whitepaper
  - https://www.lunarg.com/wp-content/uploads/2022/03/The-Vulkan-Profiles-Toolset-Solution-FEB2022.pdf

Vulkan Profiles

What's a Vulkan Profile?

- Released with Vulkan 1.3
  - But it's not really a part of the Vulkan specification, it's essentially developer tools.
- A collection of Vulkan Capabilities
  - Extensions
  - Features
  - Properties
  - Queue properties
  - Formats
  - Etc…
- A way to increase the minimum requirements of Vulkan specification
  - Based on our Vulkan developer needs

Vulkan Profiles usages:

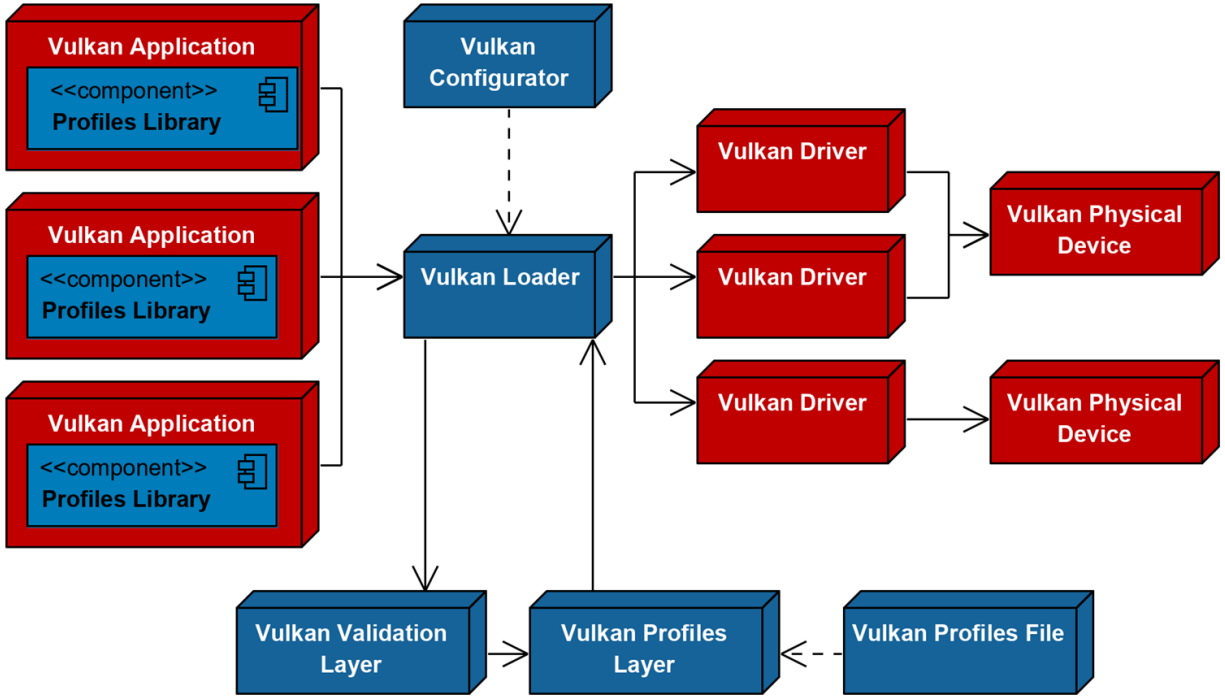
- *Roadmap profiles*: to express guidance on the future direction of Vulkan devices or projects. Eg: Vulkan Roadmap 2022.
- *Platform profiles*: to express the Vulkan support actually available on a platform. Eg: Android Baseline 2021.
- *Device profiles*: to express the Vulkan support of a single Vulkan driver for a Vulkan device. Eg: GPUinfo.org reports
- *Architecture profiles*: to express the Vulkan support of a class of GPUs. Eg: D3D12 Feature Level 12.1
- *Engine profiles*: to express some rendering code paths requirements of an engine. (Eg: VKD3D and `VK_EXT_mutable_descriptor_type`)
- Etc.

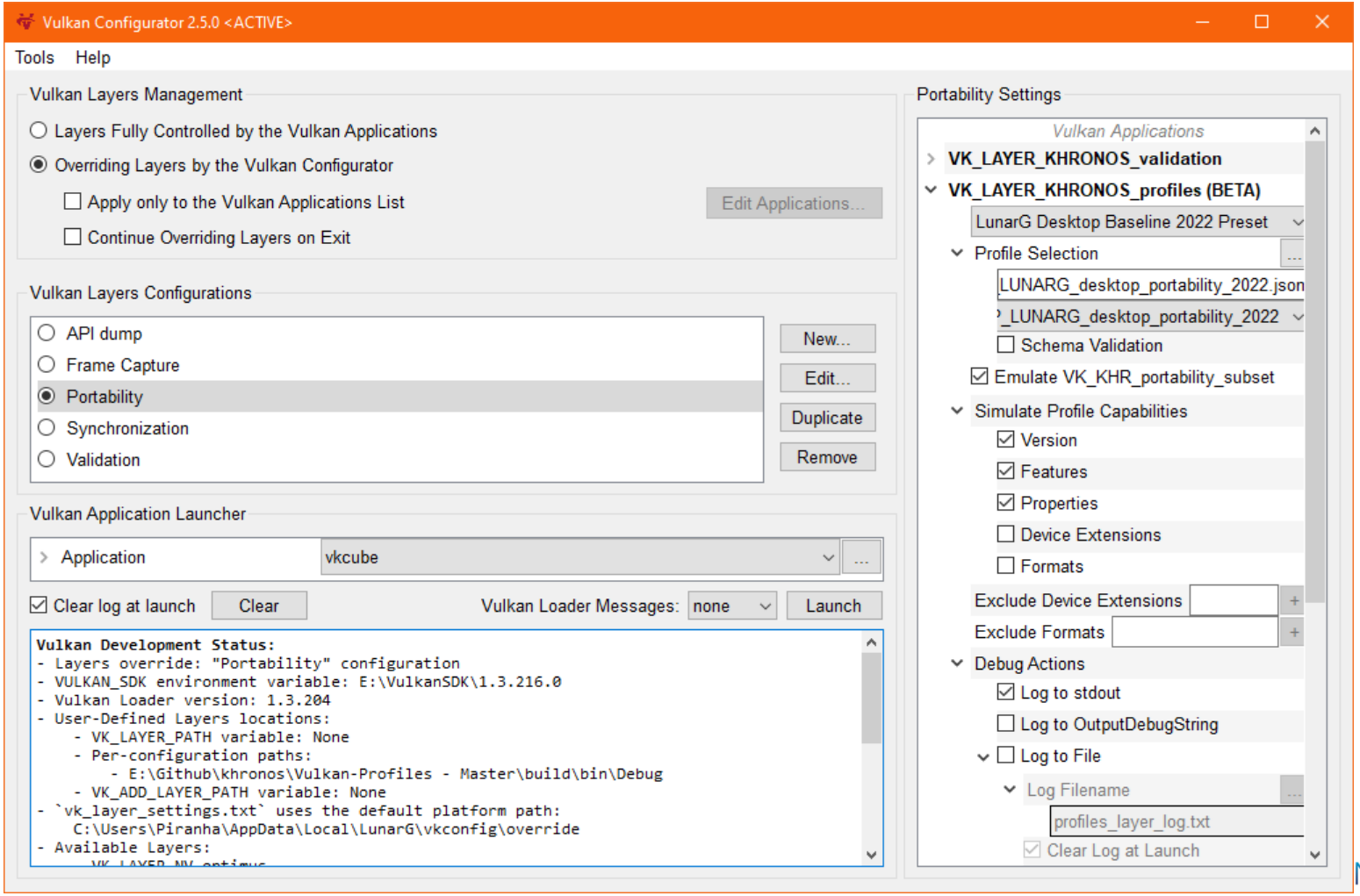
The Vulkan SDK Profiles Tools

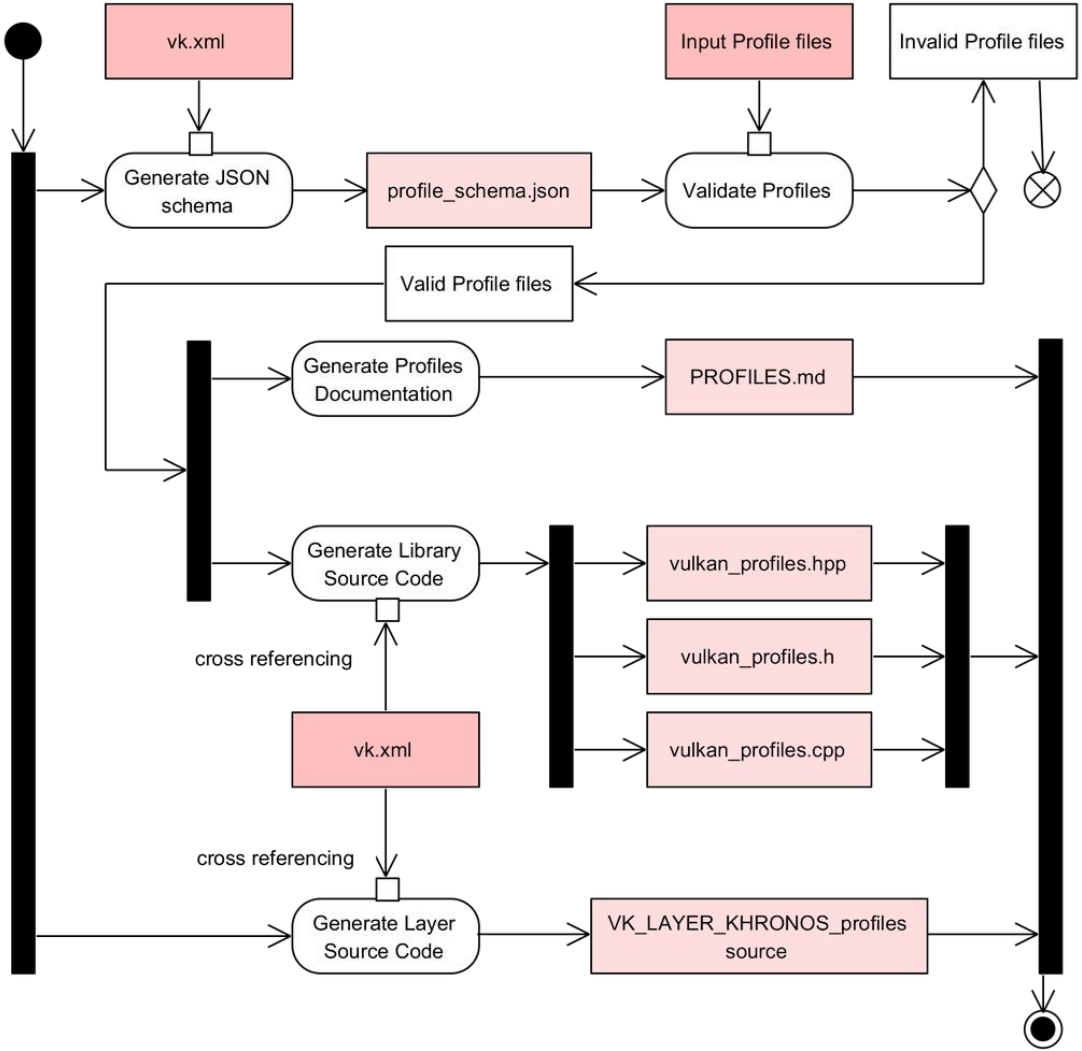
The Vulkan SDK Profiles Tools:

- Vulkan Profiles JSON schema
  - A JSON format to exchange Vulkan capabilities programmatically
  - One JSON schema per Vulkan Header revision
- Vulkan Profiles file generation
  - `Vulkaninfo` and GPUinfo.org export *Device Profile JSON files*
  - Command line tool for multiple profiles intersection or union of capabilities
    - `VP_LUNARG_desktop_baseline_2022` provided as a profile example
- Vulkan Profiles layer
  - A layer to emulate/clamp profile capabilities on Vulkan developer system
- Vulkan Profiles API library
  - C++ code ; Header-only or Header + Source
  - A library for Vulkan applications code to check profiles support, to create `VkDevice` with features enabled
  - A KhronosGroup/Vulkan-Samples sample is available on github for demonstrating the library usage
- Vulkan Profiles comparison table
  - Markdown documentation, to easily read, search, compare capabilities across profiles

Deployment of the tools




Vulkan Configurator 2.5.0 <ACTIVE>
Tools
Help
Vulkan Layers Management
Layers Fully Controlled by the Vulkan Applications
Overriding Layers by the Vulkan Configurator
Apply only to the Vulkan Applications List
Edit Applications...
Continue Overriding Layers on Exit
Vulkan Layers Configurations
API dump
Frame Capture
Portability
Synchronization
Validation
New...
Edit...
Duplicate
Remove
Vulkan Application Launcher
Application
vkcube
Clear log at launch
Clear
Vulkan Loader Messages:
none
Launch
Vulkan Development Status:
- Layers override: "Portability" configuration
- VULKAN_SDK environment variable: E:\VulkanSDK\1.3.216.0
- Vulkan Loader version: 1.3.204
- User-Defined Layers locations:
- VK_LAYER_PATH variable: None
- Per-configuration paths:
- E:\Github\khronos\Vulkan-Profiles - Master\build\bin\Debug
- VK_ADD_LAYER_PATH variable: None
- `vk_layer_settings.txt` uses the default platform path:
C:\Users\Piranha\AppData\Local\LunarG\vkconfig\override
- Available Layers:
Portability Settings
Vulkan Applications
VK_LAYER_KHRONOS_validation
VK_LAYER_KHRONOS_profiles (BETA)
LunarG Desktop Baseline 2022 Preset
Profile Selection
LUNARG_desktop_portability_2022.json
_LUNARG_desktop_portability_2022
Schema Validation
Emulate VK_KHR_portability_subset
Simulate Profile Capabilities
Version
Features
Properties
Device Extensions
Formats
Exclude Device Extensions
Exclude Formats
Debug Actions
Log to stdout
Log to OutputDebugString
Log to File
Log Filename
profiles_layer_log.txt
Clear Log at Launch


vk.xml
Input Profile files
Invalid Profile files
Generate JSON schema
profile_schema.json
Validate Profiles
Valid Profile files
Generate Profiles Documentation
PROFILES.md
Generate Library Source Code
vulkan_profiles.hpp
vulkan_profiles.h
vulkan_profiles.cpp
cross referencing
vk.xml
cross referencing
Generate Layer Source Code
VK_LAYER_KHRONOS_profiles source


Tutorial!


Questions?

More Information

Vulkan: https://www.vulkan.org/

- Vulkan 1.3 specification: https://khr.io/vulkan13spec
- Vulkan 1.3 Spec GitHub Repo: https://khr.io/vulkan13github
- Vulkan SDK: https://www.lunarg.com/vulkan-sdk/
- Vulkan Profiles Toolset: https://github.com/KhronosGroup/Vulkan-Profiles
- Vulkan Profiles Toolset Solution white paper: https://khr.io/xv
- Vulkan 1.3 and Roadmap blog post: https://khr.io/xp
- Khronos Validation Layer Repository: https://khr.io/xu
- Discord Link for community discussion: https://khr.io/vulkan13discord

A recording + slides from today can be found at:
https://www.khronos.org/events/japan-vulkan-meetup-september-9-2022

KHRONOS GROUP

Get Involved!

- **Discord:** https://discord.gg/vulkan
- **Twitter:** https://twitter.com/vulkanapi
- **Slack:** https://khr.io/slack [Vulkan Channel]
- **Reddit:** https://www.reddit.com/r/vulkan
- **Github:** https://github.com/khronosgroup [search for Vulkan]
- **Stackoverflow:** https://stackoverflow.com/questions/tagged/vulkan
- **Forums** https://community.khronos.org

KHRONOS GROUP