

# The Vulkan SDK License Registry

**Karen Ghavam**

February 2022

## Introduction

LunarG delivers the Vulkan SDK for Windows, Linux, and macOS. We've received inquiries from users who want to know the license agreements for the SDK components. The Vulkan SDK is comprised of tools and layers from 13 different open source repositories that are built, integrated, tested, and packaged for installation on the user's system. Every tool or layer is built from source with multiple open-source licenses. It isn't appropriate to assign a license to the SDK itself, since that would be re-attributing a license to all of the SDK components; so the SDK itself is "unlicensed." Providing proper attribution for every component requires that we be able to report the exact licensing for each component. Customers may also need to understand low-level licensing details to assess the SDK for installation in their corporate environments.

The Vulkan SDK Registry is LunarG's method of communicating the licenses for all of the SDK components.

## Key Challenges

The licenses for a binary should be a list of all the licenses from the sources and binaries included in that binary. What are the licenses for each of the binaries included in the Vulkan SDK and how do we determine the list of sources and binaries that contribute to a binary?

Many licenses require reporting copyrights; hence all the copyrights need to be identified for binaries and sources in the SDK as well.

What if licenses of source files change over time? How will this automatically get recorded in the licenses assigned to a binary?

The Linux tarball SDK has about 30,000 files in the SDK (since it includes all of the repositories and scripts to rebuild all of the binaries). Depending upon the product included in the SDK (number of source files for that product), it is possible that a large number of files will have no license within the source. How do we detect a component with no explicit licensing, update the recorded license for the component as needed, and not have to do this for each SDK release?

## Solution Approach

Wherever possible, license determination of source files and binary files was automated with software.

### Scanning files for a license

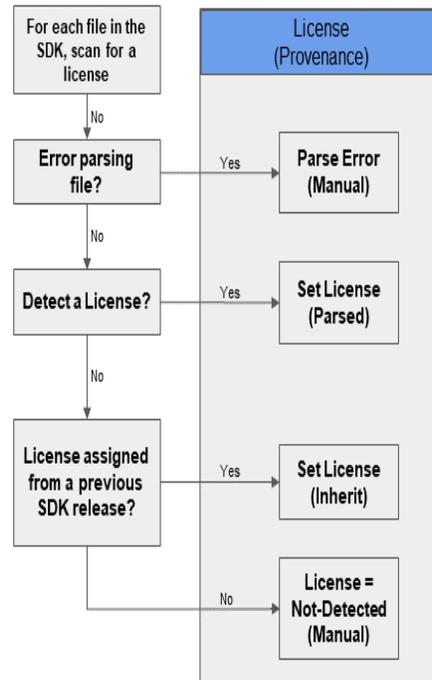
Source files are scanned for both license text and copyrights. The licenses are defined by a license definition file. The license definition file is a mapping between license names/codes and the license text that could be found in the source code. The text of the licenses can be found at: <https://spdx.org/licenses>

### Building license and copyright lists for binaries

Using output from CMake, it is possible to build a dependency tree that identifies all the sources and binaries that contribute to a built binary target. Each source file can then be scanned for a license and copyright to build the list of licenses and copyrights to assign to a binary.

### License inheritance from a previous SDK release

A database is built with the results from file scanning and the source/dependency analysis that identifies a list of licenses and copyrights for binaries. Each entry will list the license(s) and copyright(s) for a file. Each entry also has a provenance that indicates how the license was determined. See the diagram below.



If the provenance is set to “Manual,” then for that file the license is identified as “Not-Detected.” For these files a manual audit needs to occur to assign a license. During the manual audit the following can occur:

1. Examine the file for license text. It may be a new license not yet recognized by the license scanning system. In this case, a new entry is added to the License Definition File.
2. If there is no license text in the file, inspect the home repository for the file for an overarching license file. If one exists, assign this license.
3. If the file is a binary file (and hence can't be parsed), it is possible that this is an external binary brought in during the build process. So an evaluation needs to be done to identify the license for the external binary and assign it.

The auditing activity happens once for the file, and will be inherited by the next SDK release since the license scanning process again will not detect a license. This approach removes the need to redo manual licensing edits for a file. This is important because there can be hundreds of files that did not detect a license during the parsing process.

Note that if a file previously did not have a license but now does contain a license, or the license within the file changed from what was detected the last time it was parsed, the parsing activity will give precedence to a parsed license over an inherited license. Thus changes in source file licenses over time will be automatically detected.

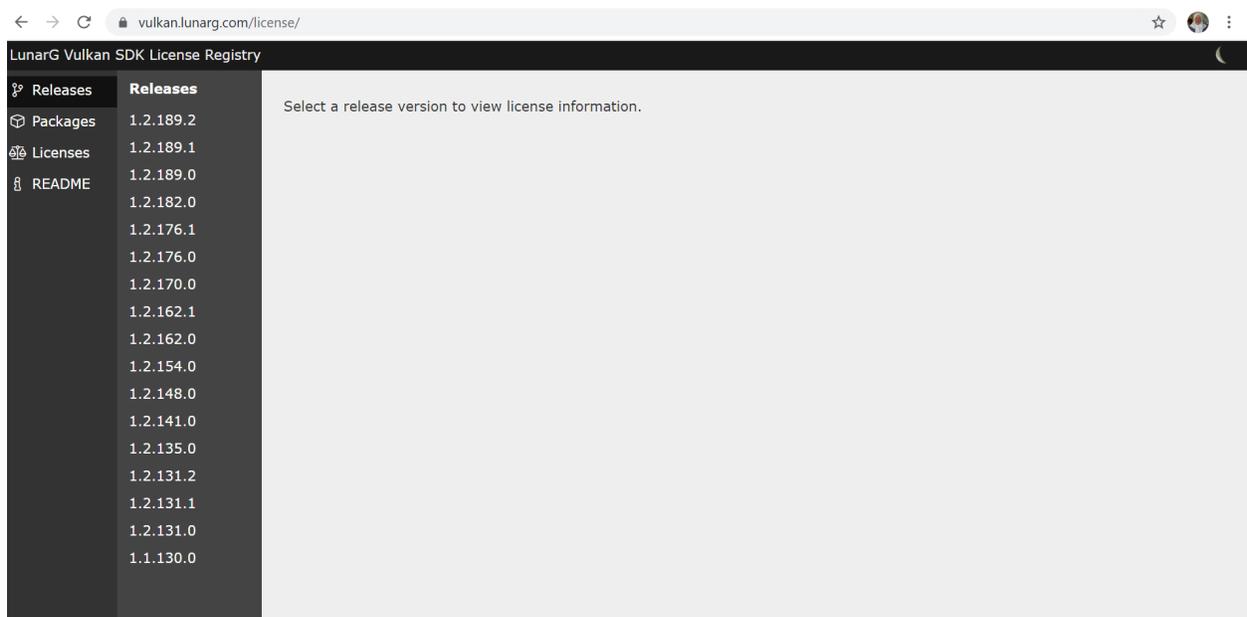
## Presentation of the License and Copyright Results

Once all the processing is completed for all the files in each of the Linux, Windows, and macOS SDKs, the results are presented for each SDK to the user as follows:

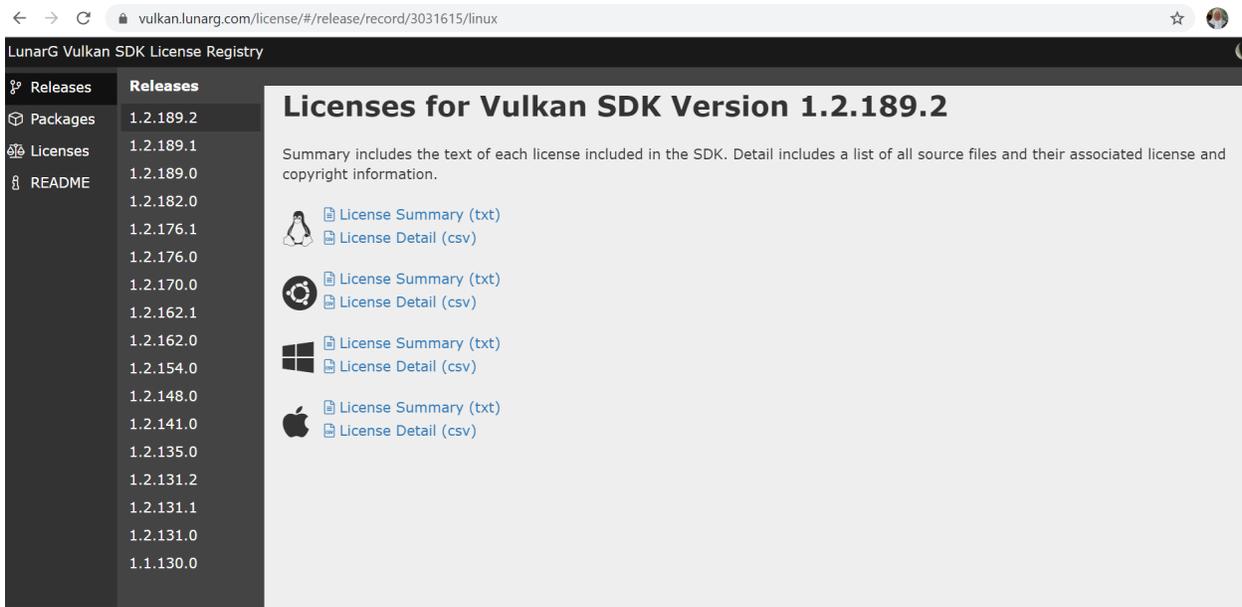
1. A CSV file is available for download that for each SDK entry will detail the license(s) and copyright(s) for each file in the SDK
2. A text file is available that will list all the licenses found in the SDK with the license text for each of these licenses

The information is presented to the user via the Vulkan SDK License registry found at [vulkan.lunarg.com/license](https://vulkan.lunarg.com/license). The full disclosure of all licenses for each SDK can be found for each SDK release.

First the specific release is selected:



Next the License Summary or the License Detail for the appropriate operating system is selected:



The screenshot shows a web browser window with the URL `vulkan.lunarg.com/license/#/release/record/3031615/linux`. The page title is "LunarG Vulkan SDK License Registry". On the left, there is a navigation menu with "Releases" selected. The main content area is titled "Licenses for Vulkan SDK Version 1.2.189.2". Below the title, there is a summary paragraph: "Summary includes the text of each license included in the SDK. Detail includes a list of all source files and their associated license and copyright information." Below this, there are four operating system icons (Linux, macOS, Windows, and Apple) each with a "License Summary (txt)" and "License Detail (csv)" link.

## Summary

The volume of files that need to be evaluated for a license is large. Without software automation, LunarG would not be able to do this evaluation for each SDK release. The tools that have been written automate much of this process (wherever possible) to provide a detailed license listing for items delivered in the Vulkan SDK.