

## Khronos Releases OpenCL 2.2 With SPIR-V 1.2



Today Khronos has released OpenCL 2.2 with SPIR-V 1.2.

The most important changes are:

**A static subset of the C++14 standard as a kernel language.** The OpenCL C++ kernel language includes classes, templates, lambda expressions, function overloads and many other constructs to increase parallel programming productivity through generic and meta-programming.

**Access to the C++ language from OpenCL library functions** to provide increased safety and reduced undefined behavior while accessing features such as atomics, iterators, images, samplers, pipes, and device queue built-in types and address spaces.

**Pipe storage, which are compile time pipes.** It's a device-side type in OpenCL 2.2 that is useful for FPGA implementations to enable efficient device-scope communication between kernels.

**Enhanced optimization of generated SPIR-V code.** Applications can provide the value of specialization constants at SPIR-V compilation time, a new query can detect non-trivial constructors and destructors of program scope global objects, and user callbacks can be set at program release time.

[KhronosGroup/OpenCL-Headers](#) repository has been flattened]. From now on, all version of OpenCL headers will be available not at separate branches, but all in master branch in separate directories named opcnl10, opcnl11 etc. Old branches are not removed, but they may not be updated in the future.

**OpenCL specifications are now open source.** OpenCL Working Group decided to publish sources of recent OpenCL specifications on GitHub, including just released OpenCL 2.2 and OpenCL C++ specifications. If you find any mistake, you can create an appropriate merge request fixing that problem.

This is what we said about the release:

"We are very excited and happy to see OpenCL C++ kernel language being a part of the OpenCL standard," said **Vincent Hindriksen, founder and managing director of StreamHPC**. "It's a great achievement, and it shows that OpenCL keeps progressing. After developing conformance tests for OpenCL 2.2 and helping finalizing OpenCL C++ specification, we are looking forward to work on first projects with OpenCL 2.2 and the new kernel language. My team believes that using OpenCL C++ instead of OpenCL C will result in improved software quality, reduced maintenance effort and faster time to market. We expect SPIR-V to heavily impact the compiler ecosystem and bring several new OpenCL kernel languages."

[Full news release is here.](#)

## OpenCL C++ porting guide

We've written [a guide to port C-kernels to C++-kernels](#). The guidelines are a set of guidelines for developers who know OpenCL C and plan to port their kernels to OpenCL C++, and therefore they need to know the main differences. As some differences may cause hard-to-detect bugs when porting to OpenCL C++, it's important to read the document carefully. Developers who are familiar with OpenCL C and C++ should find OpenCL C++ easy to learn.

## Help improve the guidelines

We put it on [Github](#), so you can send a [MR](#) if you see mistakes or omissions.

A lot of work already has been done, but there is enough room for improvement. This is what you can do:

- Extend sections about compilation by adding description how you can install and use Khronos OpenCL C++ compiler, the requirements (ICD, headers), or how to test it using Intel's OpenCL 2.1 implementation.
- Add missing details.
- Add your experience of porting OpenCL C to OpenCL C++. Add remarks to help others save time.
- Improving the examples.
- Add chapters or sections about new features introduced in OpenCL 2.2 and OpenCL C++.

## Lots is Open Sourced

Due to success with open sourcing almost anything with Vulkan, OpenCL follows its lead. At the Khronos Github, you'll find the following repos:

[OpenCL Compiler Test Suite \(CTS\)](#) (branches for 1.2, 2.0, 2.1, 2.2 (default)) - see also below]

[OpenCL CTS Framework](#) - for easy testing, see also below.]

[OpenCL ICD Loader](#) - this was already open source, updated to 2.2]

[OpenCL Reference Documentation](#) - OpenCL API, Extensions, and Environment Spec sources

[OpenCL Registry](#) - containing all the official registry files]

[OpenCL C++ frontend](#) - to compile C++ kernels to SPIR-V]

[SPIR-V Tools](#) - SPIR-V 1.2 Assembler, binary parser, disassembler and validator.]

Outside Khronos's Github you'll also find [OpenCL 2.0 support in LLVM Clang](#). This means that with the existing open source backends (like AMDGPU and NVPTX) the open source OpenCL compiler [POCL](#) could be made.

## Open sourced conformance tests

You might have heard [we upgraded the OpenCL Compiler Conformance Tests to 2.2](#). Besides the upgrade, we also put efforts in cleaning up the code together with the OpenCL working-group - all to prepare for the open sourcing of the tests.

All is under an Apache license. This goes two ways: easier testing and easier fixing. Khronos invites everybody to send merge requests for test they have developed. I think [Mikael Lepistö](#) and [the CLSmith team](#) are very pleased.

You can find the tests [here](#). There is also a testing framework by the University of Windsor that eases the building of the tests. It downloads all tests, builds itself, and then builds conformance tests. You can get it [here](#).

Important: only Khronos can label your product as conformant.

## Open sourced documentation

The conformance tests are tightly connected to the specifications. A problem in one is a problem in the other.

What if the specification is unclear and needs extra clarification? What if a test has been added, but the specification is less or more

restrictive? You can now create a [pull request](#) on Github to get it solved.

The documentation files are [here](#).

## Implementations

Imagination Technologies was ready in time to get the full 2.2 implemented as of today. On their new PowerVR Series8XT GPUs it will be possible to run OpenCL 2.2 C++ kernels or any other SPIR-V 1.2 frontend.

We're waiting for the rest to come up with their (beta) OpenCL 2.2 drivers.