
1 What's New in AMD APP SDK v2.7

Key features supported in SDK 2.7 and the Catalyst 12.4 drivers include:

OpenCL™ 1.2 adds the following key capabilities.

- Host access flags for memory objects enable more efficient buffer handling and provide added protection. For example, a buffer that is created as “write only” cannot be read from the host.
- Pattern-based GPU buffer and image initialization can help eliminate need for certain buffer/image transfers.
- Memory objects migration supports transfer of buffers before they are needed.
- New generalized image creation API.
- Enhanced image/buffer map operations.
- OpenCL 1.2 CPU device partition, including partition of a CPU after addition to a context.
- Generalized 1D and 2D images, image arrays, and image<->buffer interop.

The C++ Wrapper API provide the following new capabilities.

- Defaults for platform, queue, device, etc. significantly reduce the amount of boilerplate code required.
- Improved simplified constructors for `cl::Buffer` and addition of `cl::copy` functions.
- Additional support of events when using functors.

Notable C++ features supported by the OpenCL™ Static C++ Kernel language include:

- Kernel and function overloading.
- Inheritance:
 - Strict inheritance.
 - Friend classes.
 - Multiple inheritance.
- Templates:
 - Kernel templates.
 - Member templates.
 - Template default argument.
 - Limited class templates (the “virtual” keyword is not exposed).
 - Partial template specialization.

- Namespaces.
- References.
- `this` operator.
- with external symbols.
- Kernel reflection: the ability to query a kernel's arguments.
- Support for `printf` as a built-in function.

Additional features supported in SDK 2.7 and the Catalyst 12.4 drivers include:

- Support for Asynchronous PCI transfers.
- Video encode using VCE Encode (Win7).
- Open Encode update (12.4).
- The OpenCL extension `cl_khr_fp64` is now supported on Radeon HD™ 6900-series devices (Cayman).
- Added OpenGL™ interoperability under Linux for Radeon HD™ 7000 series devices.
- Stability Improvements.
- Performance improvements.
- Support for Radeon HD™ 7000 series devices.
- Support for AMD Second Generation APUs.
- Kernel Analyzer v1.12.
- APP Profiler v2.5.

gDEDebugger version 6.2. It can be downloaded for use with this SDK from <http://developer.amd.com/gDEDebugger>.

- Supports Linux®.
- New stand-alone user-interface for both Linux® and Windows®, with enhancements for better navigation and ease-of-use
- Supports OpenCL™ kernel and API-level debugging on AMD Radeon HD 7000 series devices.
- Supports OpenCL™ 1.2 beta drivers.
- Automatic updater to notify, and download, new product updates.
- Feature enhancements, including support for static arrays, union variables, and a Find feature.
- Stability improvements.

APP KernelAnalyzer v 1.12 enhancements include:

- Support for Catalyst revisions through 12.1 – 12.4.

APP Profiler v2.5 enhancements include:

- Support for OpenCL 1.2.
- Support for collecting performance counters on APU devices.
- Full support for profiling with AMD Radeon™ HD7000 series GPUs:
 - Added support for kernel occupancy analysis.
 - Added support for collecting performance counters for DirectCompute (DirectX 11) applications.
 - Addition of SALUBusy counter.
 - Fixed value reported for VALUBusy counter.
 - The values reported for LDSFetchInsts and LDSWriteInsts counters were inaccurate on AMD Radeon™ HD7000 series GPUs; thus, for those GPUs, those two counters have been replaced by a single LDSInsts counter.
 - Fixed display of kernel ISA.
- Improved OpenCL analysis module: added detection of deprecated OpenCL APIs.
- Added support for showing source and destination location, as well as zero-copy status for memory transfers initiated using `clEnqueueMapBuffer` or `clEnqueueMapImage`, which is shown in the API Trace view.
- Added support for Microsoft® Visual Studio® projects that use user-defined macros in the project settings.
- Fixed the `--workingdirectory (-w)` command line switch (set current directory) on Linux.
- Fixed some problems with importing previously generated profile results into Microsoft® Visual Studio®.
- Changed the default installation directory on Windows to `%PROGRAMFILES(X86)%\AMD\AMD APP Profiler` for consistency with other AMD tools.
- Stability improvements.

APPML 1.8 (APP Math Libraries) new features include:

- Support for real-to-complex FFT.
- Support for all functions in BLAS level 2 and BLAS level 3.

New and updated Samples.

- Nbody: optimized for improved performance.
- DeviceFission: a new version of this sample using OpenCL 1.2 Device Fission capabilities. The old version is still included, but renamed as DeviceFission11Ext.
- ImageOverlap and GaussianNoiseGL are two new OpenCL™ 1.2 samples.
- DwtHaar1DCPPKernel: an additional version of DwtHaar1D, but modified to use the C++ kernel language.
- MatrixMultiplicationCPPKernel: an additional version of MatrixMultiplication, but modified to use the C++ kernel language. This sample supports multiplication of both int and float matrices through use of a template.
- TransferOverlapCPP: an additional version of TransferOverlap, but modified to use the C++ wrapper API.

- The URNGNoiseGL and HistogramAtomics samples have been modified to use the C++ wrapper API.
- The FFT, MersenneTwister, and EigenValue samples have been modified to use the C++ kernel language.
- Incremental improvements to a number of additional samples.

2 Important Notes

- The following values are returned when querying strings from OpenCL:
 - `CL_PLATFORM_VERSION`: OpenCL 1.2 AMD-APP (build #).
 - `CL_PLATFORM_NAME`: AMD Accelerated Parallel Processing.
 - `CL_PLATFORM_VENDOR`: Advanced Micro Devices, Inc.
- Check the Platform Vendor string, not the Platform Name, to determine AMD hardware. For example code that shows how to check and use the `CL_PLATFORM_VENDOR` string, see the AMD APP v 2.7 Samples.
- Support for OpenCL on 7xx generation GPUs is deprecated, this includes ATI Radeon™ HD and ATI Mobility Radeon™ HD 4000 series devices, ATI FirePro™ V8750, V8700, V7750, V5700, V2750, ATI Mobility FirePro™ M7740, and AMD FireStream™ 9270, 9250.
- To develop applications using deprecated OpenCL™ API calls, `#define CL_USE_DEPRECATED_OPENCL_1_1_APIS`.
- When parsing `#include` directives, the OpenCL compiler resolves relative paths using the current working directory of the application through the `-I` compiler option.
- You must install Catalyst 12.4 graphics drivers before installing SDK 2.7. Vital components of the AMD OpenCL solution are now contained within the drivers.
- Under Windows, making OpenCL runtime calls from `dllMain` can result in undefined behavior.
- The binary `clinfo.exe` is now located in the `\Windows\System32` directory.

3 Naming Convention

For Windows:

- The `__stdcall` calling convention is used for all Windows platforms.
- Function names are undecorated.
- It is not possible to use this OpenCL DLL on Windows with an application that was linked against a library using the `__cdecl` calling convention.

For Linux:

- The calling convention is `__cdecl`.

4 Resolved Issues

4.1 Runtime

- Fixed issue for some OpenCL applications on Linux platforms using 100% of the CPU time when using Catalyst 11.11 drivers.

4.2 APP Profiler

- APP Profiler now provides reliable results for applications running on AMD fusion APU devices.

4.3 printf Implementation

- Fixed the printf implementation to allow for infinite printf statements to occur, and all printf's are executed.

5 Known Issues

5.1 Compiler

- Use of dynamic pointer assignment in kernels that are executed on the GPU cause inefficient code generation.
- Many OpenCL specification compiler options are accepted by the AMD OpenCL compiler, but are not implemented. The implemented options are `-D`, `-I`, `w`, `Werror`, `-clsingle-precision-constant`, `-cl-opt-disable`, and `-cl-fp32-correctly-rounded-divide-sqrt`.
- The compiler may accept illegal cast-to-union (GNU GCC Extension) cases. In such cases, a warning is issued. This may be fixed in a future release.
- On 64-bit Win7/Vista systems, use of AVX code generation may produce incorrect results. If a problem is observed, in-line as many functions as possible within the kernel; you can use `__attribute__((always_inline))`. On rare occasions, the issue may persist, in such instances disable AVX code generation using the compiler option `-fdisable-avx`.
- If an argument to an OpenCL kernel function is optimized away late in the compilation process, the compiler may fail to build or produce undefined results. This occurs when either an argument gets entirely optimized away or when part of an argument gets optimized away. Part of an argument can get optimized away if two (for 64-bit data types) or four (for all other types) consecutive components of an argument are not used where the first unused component is a multiple of 2 or 4.
- When using the C++ wrapper API in the current release, it is recommended that the vector class not be used as a replacement for `std::vector` in production code except in very simple cases. Check the Khronos website for an updated version of `cl.hpp`.
- When using the C++ Wrapper API, some string copy operations do not function correctly. Check the Khronos website for an updated version of `cl.hpp`.
- When using the C++ Wrapper API, in some cases Functor interfaces do not correctly pass events. Check the Khronos website for an updated version of `cl.hpp`.

- On CPUs that support AVX, querying the values of native and preferred vector widths returns incorrect values (4 for float, instead of 8; 0 for double, instead of 4).

5.2 Runtime

- The OpenCL runtime currently does not validate handles to OpenCL memory objects.
- Under Windows Vista, to prevent long programs from causing a dialog to be displayed indicating that the display driver has stopped responding, disable the Vista Timeout Detection and Recovery (TDR) feature, which is trying to detect hangs in graphics hardware. To do this, use `regedit.exe` to create the following `REG_DWORD` entry in the registry, and set its value to 0:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\GraphicsDrivers\TdrLevel
```

This avoids the constant polling by the driver and the kernel to prevent long work units from monopolizing the device. (To restore default functionality, set the `TdrLevel` to 3.)

Note that Microsoft strongly discourages disabling this feature, and only recommends doing so for debugging purposes. Do so at your own risk.

- On Linux platforms, if a kernel deadlocks the GPU, the system becomes unresponsive for a few minutes, and both the X-window server and the application become defunct processes. The system must be rebooted in order to use the GPU again.
- For error-free processing on 7xx-based GPUs, the maximum work-group size is 64. Specifying a larger size can result in undefined behavior.
- If the `clGetPlatformIDs()` failed error is issued with a properly installed ICD while running 32-bit code on a 64-bit system, ensure that all necessary 32-bit libraries are installed. The specifics of this vary between Linux distributions; consult your OS documentation for more information. The `libGLU.so` library is known to trigger this problem, but there may be others, depending on the specific installation.
- In Linux for non-Southern_Islands platforms, the OpenCL runtime currently exposes less than the total amount of memory physically available on the card. In Windows and on Linux or Southern-Islands-based platforms, the OpenCL runtime reports the total amount of physical memory for boards with up to 2 GB.
- To access multiple GPUs under Linux, ensure the `DISPLAY` environment variable is set to `:0` instead of `:0.0`. Alternatively, set the `COMPUTE` environment variable to `:0`. If `COMPUTE` is set, the OpenCL SDK uses the this variable, rather than `DISPLAY`, to determine the available GPU devices.

5.3 Interoperability

- For OpenGL interoperability with OpenCL, there currently is a requirement on when the OpenCL context is created and when texture/buffer shared allocations can be made. To use shared resources, the OpenGL application must create an OpenGL context and then an OpenCL context. All resources (GL buffers and textures) created after creation of the OpenCL context can be shared between OpenGL and OpenCL. If resources are allocated before the OpenCL context creation, they cannot be shared between OpenGL and OpenCL.

5.4 Tools

- When using APP Profiler to collect an application trace with the “Write trace data periodically during program execution” option enabled (`--timeout` at the command line), the profile results do not contain GPU timestamp data unless the application calls `clReleaseContext`. The

workaround is to either enable the “Write trace data at program termination” option, or to ensure the application calls `clReleaseContext` to clean up any OpenCL context that it creates.

5.5 MPEG-2 Decode

- There is a known issue in this release when trying to check the capabilities of the Cypress GPU: it incorrectly reports that it can decode MPEG-2 VLD. This GPU, which has UVD 2.2, does not support MPEG-2 decode capability. AMD hopes to address this issue in a subsequent version of the driver.

Contact

Advanced Micro Devices, Inc.
One AMD Place
P.O. Box 3453
Sunnyvale, CA, 94088-3453
Phone: +1.408.749.4000

For AMD Accelerated Parallel Processing:
URL: developer.amd.com/appsdk
Developing: developer.amd.com/
Support: developer.amd.com/appsdksupport
Forum: developer.amd.com/openclforum



The contents of this document are provided in connection with Advanced Micro Devices, Inc. (“AMD”) products. AMD makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication and reserves the right to make changes to specifications and product descriptions at any time without notice. The information contained herein may be of a preliminary or advance nature and is subject to change without notice. No license, whether express, implied, arising by estoppel or otherwise, to any intellectual property rights is granted by this publication. Except as set forth in AMD’s Standard Terms and Conditions of Sale, AMD assumes no liability whatsoever, and disclaims any express or implied warranty, relating to its products including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or infringement of any intellectual property right.

AMD’s products are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications intended to support or sustain life, or in any other application in which the failure of AMD’s product could create a situation where personal injury, death, or severe property or environmental damage may occur. AMD reserves the right to discontinue or make changes to its products at any time without notice.

Copyright and Trademarks

© 2011 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, ATI, the ATI logo, Radeon, FireStream, and combinations thereof are trademarks of Advanced Micro Devices, Inc. OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos. Other names are for informational purposes only and may be trademarks of their respective owners.