



CodeXL 2.0 GA Release Notes

Contents

CodeXL 2.0 GA Release Notes	1
New in this version.....	2
System Requirements	3
Getting the latest Radeon software release	5
Note about installing CodeAnalyst after installing CodeXL for Windows.....	5
Fixed Issues	5
Known Issues.....	7
Support	9

Thank you for using CodeXL. We appreciate any feedback you have! Please use the [CodeXL Issues Page](#) to provide your feedback.

You can also check out the Getting Started guide and the latest CodeXL blog at GPUOpen.com

This version contains:

- For 64-bit Windows platforms
 - CodeXL Standalone application
 - CodeXL Microsoft® Visual Studio® 2010 extension
 - CodeXL Microsoft® Visual Studio® 2012 extension
 - CodeXL Microsoft® Visual Studio® 2013 extension
 - CodeXL Microsoft® Visual Studio® 2015 extension
 - CodeXL Remote Agent
- For 64-bit Linux platforms
 - CodeXL Standalone application
 - CodeXL Remote Agent

New in this version

The following items are new in this version:

- Open Source
 - CodeXL is now part of the exciting GPUOpen initiative – the CodeXL source code is now published on GitHub and accessible via [GPUOpen.com](https://gpuopen.com).
We believe that by adopting the open-source model and sharing the CodeXL source base with the world we can help developers make better use of CodeXL and make CodeXL a better tool.
 - Note: A few CodeXL components are released as binaries due to legal or IP confidentiality reasons.
 - To encourage 3rd party contributions and adopting of CodeXL, the AMD brand is removed from CodeXL product name and graphic assets.
- Frame Analysis
 - In this release we introduce Frame Analysis - a new CodeXL component focused on game development. CodeXL 2.0 is the first release to include game development features. This is a first step as we will add many more capabilities in the releases ahead.
 - Collect and display a frame timeline for applications that use Microsoft DirectX® 12
This feature is in Beta stage.
- CodeXL Visual Studio® 2015 Extension
- GPU Debugging
 - Combined Host and GPU Debugging on Linux for C/C++ apps
 - The CodeXL Visual Studio Extension provides combined Host and GPU Debugging on Windows for 32-bit C/C++ apps
- Static Analysis
 - Introducing 'Programs' support in the static analyzer – Vulkan® and OpenGL® shaders are now compiled and linked together in programs to produce more accurate ISA and performance statistics
 - Added support for Vulkan: generate ISA and performance statistics for Vulkan GLSL shaders
 - Export OpenGL and DirectX shader binaries
 - Register liveness report added to the analyzer command line tool. This feature is in Beta stage.
- GPU Profiling
 - HSA profiler support for Boltzmann (Fiji dGPU)
 - Remote GPU profiling from Linux to Windows and from Windows to Linux
- Initial HSA Debugging support for Boltzmann (Fiji dGPU)
- Power Profiling
 - Attributing power samples to specific processes
- Installer

- Move all CodeXL binaries to a common location on the hard-disk regardless of how many CodeXL Visual Studio extensions are installed.
- Move CodeXL samples to a path under the user home directory to allow write permission for non-admin users
- General
 - We modified CodeXL file names to adhere to a uniform naming pattern.
 - The following executable file names changed:
 - sprofile -> CodeXLGpuProfiler

System Requirements

CodeXL contains a host of development features with varying system requirements:

- **GPU Profiling and OpenCL™ Kernel Debugging**
 - An AMD GPU (Radeon HD 7700 series or newer, desktop or mobile version) or APU is required.
 - Radeon Software Crimson Edition 16.3.2 (driver 16.15) is the recommended driver. See "[Getting the latest Radeon software release](#)" section below.
 - Earlier HW configurations (Radeon HD 5000/6000 series) are no longer supported by Radeon Software Crimson Edition and CodeXL 2.0. For these configurations please install CodeXL 1.9 and the AMD Catalyst driver release 13.11 or later. Catalyst 15.9.1 (driver 15.201) is the recommended version.
- **HSA Profiling and Debugging**
 - Supported on the Linux® HSA stack, on Carrizo APUs and Boltzmann configurations:
 - CPU: Intel(c) Haswell or newer, Core i5, Core i7, Xeon E3 v4 & v5; Xeon E5 v3
 - GPU: Fiji ASIC (AMD R9 Nano, R9 Fury and R9 Fury X)
 - The Radeon Open Compute Runtime (ROCR) must be installed
 - Available here: <https://github.com/RadeonOpenCompute/ROCR-Runtime>
 - Should a new version of the ROCr-Runtime become available, the version of the ROCm GPU Debug backend binaries included in CodeXL will need to be updated in order to be compatible with that version. If/when a new runtime is published to GitHub, we will also publish new ROCm GPU Debug backend binaries on the ROCm-Debugger repository on GitHub (<https://github.com/RadeonOpenCompute/ROCr-Debugger>). In order to continue using the HSA Debugging with CodeXL, you will need to copy the following files from the GitHub repository to the CodeXL installation directory to overwrite the versions of the files released with CodeXL:
 - libAMDGPUDebugHSA-x64.so
 - libAMDHwDbgFacilities-x64.so
 - Should a new version of the ROCr become available, the version of the profiler included in CodeXL will need to be updated in order to be compatible with that version. If/when a new runtime is published to GitHub, we will also publish new HSA Profiler binaries on GitHub (<https://github.com/RadeonOpenCompute/ROCr-Profiler>). In order to continue using the HSA Profiler with CodeXL, you will need to copy the following files from the

GitHub repository to the CodeXL installation directory to overwrite the versions of the files released with CodeXL:

- bin/libGPUPerAPIHSA.so
 - bin/libHSAProfileAgent.so
 - bin/libHSATraceAgent.so
- For **GPU API-Level Debugging**, a working OpenCL/OpenGL configuration is required (AMD or other).
- **CPU Profiling**
 - Time-Based Profiling can be performed on any x86 or AMD64 (x86-64) CPU/APU.
 - The Event-Based Profiling (EBP) and Instruction-Based Sampling (IBS) session types require an AMD CPU or APU processor.
 - CPU Profiling on Linux platforms - Limitations of PERF
 - CPU profiling uses PERF which requires kernel 2.6.32 or later. CPU Profiling with Call Stack Sampling requires Linux kernel 3.0 or later. However, we recommend using kernel 3.2 and above which has shown to be more stable.
 - Call chain analysis on Linux currently depends on the call chain information provided by Linux PERF. This requires the profiled binaries to have stack frame pointer. (i.e. compiled with -fno-omit-frame-pointer).
 - For non-root users to run CodeXL CPU profiling, "/proc/sys/kernel/perf_event_paranoid" needs to be set to "-1".
 - Instruction-Based Profiling on Linux requires Linux kernel 3.5 and above.
 - Call chain information (stack trace) for inline functions is not available.
- **CPU Profiling on VMWare**
 - Time-Based Profiling (TBP) and Event-Based Profiling (EBP) are supported in guest OS running on VMware Workstation 11.0 or later.
 - If VMWare Workstations's Virtual Performance Monitoring Counters (VPMC) is not supported on a given CPU, then only time-based profiling will be available. Event-Based Profiling will not capture any data other than CPU cycles.
 - Event-Based Profiling works on Windows and Linux guest OS in these scenarios:
 - Host OS: Windows, Guest OS: Window 7, Windows 8.1, Win10, Ubuntu 14.04, RHEL 7
 - Host OS: Linux, Guest OS: Window 7, Windows 8.1, Win10, Ubuntu 14.04, RHEL 7
 - Basic CPU configuration requirements:
 - CPU should support SVM or AMD-V feature. Without this VMware will not be able to do hardware virtualization.
 - This CPU feature can be enabled/disabled from BIOS settings.
- **Power Profiling**
 - Supported on:
 - Carrizo, Kaveri, Mullins and Temash APUs.
 - The majority of the Graphics IP 7 GPUs (code name "Sea Islands") or more recent, including AMD Radeon™ and AMD FirePro™ models.

- **Static Analysis**
 - OpenCL/DirectX 11 kernel/shader analysis requires a working AMD OpenCL/DirectX 11 configuration
 - OpenGL shader analysis on Windows requires Catalyst 15.9. (driver 15.20) or later
 - OpenGL shader analysis on Linux requires Radeon Software Crimson Edition (driver 15.30) or later
 - Vulkan shader analysis is not driver dependent.

Supported platforms:

- Windows platforms
 - Windows 7 64-bit, 8.1 64-bit and 10 64-bit.
 - Windows 7 & 8.1 require to install Microsoft update KB2999226
<https://support.microsoft.com/en-us/kb/2999226>
 - Note: For the CodeXL Visual Studio 2010/2012/2013/2015 Package, the station must be installed with Visual Studio 2010/2012/2013/2015, respectively. However, the CodeXL Standalone Application does not require Visual Studio to be installed.
- Linux platforms
 - Red Hat EL 7 64-bit
 - Ubuntu 15.04 64-bit
 - SUSE 11 SP3 64-bit

Getting the latest Radeon software release

The way to get the latest beta driver is to use the links "Latest Windows Beta Driver" and "Latest Linux Beta Driver" on the Graphics Drivers support page:

<http://support.amd.com/us/gpudownload/Pages/index.aspx>

Note about installing CodeAnalyst after installing CodeXL for Windows

AMD CodeAnalyst has reached End-of-Life status and has been replaced by CodeXL. CodeXL installer will refuse to install on a Windows station where AMD CodeAnalyst is already installed. Nevertheless, if you would like to install CodeAnalyst, do not install it on a Windows station already installed with CodeXL. Uninstall CodeXL first, and then install CodeAnalyst.

Fixed Issues

The following are the major fixes that were not part of the 1.9 release and are new to this version:

- Static Analyzer resource usage reference tables – the SGPR resource usage was corrected to match runtime allocation scheme. The reference table ranges shifted 2 registers up.
- AMDTClassicMatMul sample asks for Build every time before profiling. (109)
- Repeated CPU Profiler Assess Performance sessions of a CLR application provides inconsistent samples count. (130)
- In the CPU Profiler the same function is shown as both parent and child when profiling CodeXL. (288)
- HSA Performance Counters profiling is not supported on Carrizo. (826)

- Carrizo APU is identified as Kaveri in the GPU Profiler's timeline chart. (1159)
- After performing application timeline trace, the GPU Profiler's Kernel Occupancy view shows empty brackets. (1338)
- "Failed to load driver" error on HyperV enabled system on upgrading from 1.8 to 1.9. (1362)
- Using the static analyzer to build OpenGL fragment shaders, the ISA shows V_NOP instructions. (1437)
- CodeXL displays inappropriate error message when Power Profiling an FX core platform which does not support power profiling: "Communication to CodeXL Power Profiling driver... returned error 2013". (1468)
- Teapot smoke appears corrupted on Windows, Radeon Software Crimson Edition with Bonaire and Fiji devices. (1475)
- The Power Profiler's Summary tab doesn't get updated unless it is in focus. (1522)
- CPU Profiler's Call Graph view may display empty parent lists and call graphs when attaching the profiler to a running process on Windows. (1529)
- In Static Analyzer on Linux, OpenGL geometry shaders fail to build for some GFX IP v7 and v8 devices. (1537)
- A message pops-up when creating a new HSA profiling project, warning that a 2-pass counter combination was selected. (1563)
- Installing the CodeXL Debian package directly from the web browser on Ubuntu 15.04 displays "Package is of bad quality". This is a browser issue. Workaround: Save the Debian package to the home directory and launch it from there outside of the web browser. (1574)
- 32-bit profiled applications crash if the GPU Profiler used from the Visual Studio extension and the standalone CodeXL client concurrently. (1605)
- Very Low FPS while running teapot in Debug Mode. The debugger forces a build without optimizations using the -o0 flag. The Radeon Software Crimson Edition dropped all optimizations for -o0, therefore the OpenCL kernels are executed with unoptimized code. This is the cause for the performance drop. (1611)
- Linux may hang when GPU Profiling of GL samples is performed after enabling the navigation to source code option. (1613)
- Performing HSA Profiling application trace on a HCC application causes the application to hang. (1623)
- CodeXL shows Power Profiling counters of the local machine instead of the remote machine's Power Profiling counters. (1639)
- An empty statistics tree node appears after cancelling the static analyzer build operation midway. (1640)
- While performing HSAIL debugging the 'Add watch' is disabled on right click on any variable in Locals view. (1647)
- GPU Debugger hangs when clicking 'Break' while debugging the APP SDK GaussianNoiseGL sample on Windows 10. (1681)
- GPU Debugger 'Buffer Display Options' for OpenCL buffers show graphic buffer options. (1683)
- Power profiling data collection stops when the APP SDK MonteCarloMultiGPU sample is launched. (1693)
- If a breakpoint is set inside an OpenCL kernel file before debugging begins then the GPU debug session execution will not break at that breakpoint. (1702)
- On Ubuntu 15.04, the static analyzer displays empty statistics view after building an OpenCL kernel. (1705)

- The static analyzer's statistics view 'Dynamic LDS Usage' field is not editable when building OpenCL kernels. (1706)
- CPU Profiler does not collect callstack data when 'Kernel mode' is selected in the project settings. (1719)
- Static analyzer statistics cannot be open when locale is not set to EN. (1748)
- The GPU Debugger's texture properties slider control does not change the layer/index for texture arrays, cubemaps and volumetric textures. (2552)
- Power profiler sometimes fails to detect that "Intel CPU + AMD dGPU" configuration is a supported platform for power profiling. (2641)
- GPU Debugger may crash when debugged application uses clSetKernelArgSVMPointer. (2714)
- OpenCL runtime crashes when launching from Visual Studio extension and collecting performance counters using the Teapot sample on Carrizo. Fixed in Radeon Software 16.3.2 (80512)

Known Issues

- Debugging OpenCL kernels that use read-modify-write atomic operations is not supported.
- GPU Debugging on OpenCL Static C++ Kernels is not supported. (334415)
- OpenCL 1.2 keyword printf and barriers are not supported during kernel debugging.
- Building kernels with OpenCL 1.2 clCreateProgramWithBinaries and clLinkProgram API prevents the display of source code when debugging these kernels. (369171)
- Performing CPU Profiling with Call-Stack Sampling (CSS) enabled, on systems with discrete graphics card (Radeon HD 5000, 6000 or 7000 series) and Linux kernel version 3.0 or lower, may result in Linux kernel panic. This kernel panic does not occur with Linux kernel version 3.2 onwards. (352399)
- CPU Profiling is disabled on Windows 8 and 8.1 if Hyper-V is enabled. (438549)
 - Note that installing Microsoft Windows Phone 8.0 SDK activates Hyper-V.
- PERF call chains which contain call stacks across modules have shown to be truncated. This results in inaccurate "Deep Samples", "Downstream Samples", and "Call Path" analysis.
- If gDEDebugger 6.x is installed on the machine, mouse click doesn't start text fields editing in CodeXL Visual Studio Extension. [Workaround: Navigate to the text fields using TAB or uninstall gDEDebugger before installing CodeXL.](#) (344811)
- Menu items are present but not visible after minimization and restore of CodeXL in Ubuntu system using Unity theme. [Workaround: Use Unity 2D theme instead of Unity theme.](#) (353082)
- AMDTTeapot sample may crash while debugging OpenCL kernels after multiple step operations (45 or more). (357741)
- CPU Profiling on Windows 8 shows two target applications in Profile Overview. The conhost.exe process is an actual executable. This process fixes a fundamental problem in the way previous versions of Windows handled console windows, which broke drag & drop in Vista.
- If CodeXL is installed in path that includes non-ASCII Unicode characters, profiling does not work (365118).
- GPU Debugger does not display locals when debugging a kernel with extremely large buffers or worksize. (23, 1156)
- Power Profiling of Tonga and Iceland dGPUs is disabled. (36, 1497)
- The Call-graph view for CPU Profiling with callstack collection of 32-bit applications may show two separate paths for a function that has a single path. (223)

- If the profiled station goes into Sleep/Hibernate state during a Power Profiling session, only data collected before hibernation is displayed, and the navigation slider does not respond. (459572, 224)
- GPU Debugger does not stop at breakpoints inside kernels that take a very long time to execute and cause a driver TDR. (240)
- Performing 2 GPU Profiling sessions concurrently - Timeline Application Trace and Performance Counters - on a Red Hat Linux System may cause a system hang after several minutes. (259, 68176)
- CPU Profiler runs out of memory and closes down while performing post-processing of a system-wide profile session that combines IBS, CLU and Time-based sampling for over 5 minutes. (265)
- CPU Profiling multiple processes with call stack collection may result in call graph view displaying addresses instead of function names for functions used by more than one process. (289)
- The GPU Debugger can't step into a kernel if blocks that contain a return statement. (349)
- Windows system crash (Blue Screen Of Death) is observed, if CPU Profiling using Event-Based-Profiling is run both in guest and host OS or if CPU Profiling using Event-Based-Profiling is run on host OS while the guest OS is launched. This is an issue in the VMWare VMM driver. (907)
- GPU Profiler does not display any hsa_*_get_info calls in host thread calls list if they are callbacks encompassed by hsa_iterate_agents calls. (980)
- When using the static AMDTPwrProfile library for the Power Profiler API on Linux, the user must build his application with "-Wl,--whole-archive -lpthread -Wl,--no-whole-archive". Otherwise not all the symbols from pthread library will be linked, since most of them are WEAK symbols. Failing to use these flags will lead to a crash. (1040)
- CPU Profiler time-based profiling on a VM may produce more samples than the session duration and sampling interval suggest. (1125)
- Power Profiler displays zero values for 'Others' counters in Summary view if only dGPU counters are selected. (1200)
- GPU Debugger skips the internal loop in APP SDK nBody sample. (1250)
- In CPU Profiler's Time-Based Profiling, an unexpected low number of samples is collected when running on guest Win10-64 OS. (1277)
- Step-in operations require over a minute when debugging clFFT sample. (1324)
- Unable to launch GPU profiler - cannot allocate memory error on starting profiling after running 2 or 3 GPU Profiler timeline trace sessions for 2-3 min. (1347)
- CodeXL throws segmentation fault while launching on Linux through SSH. (1533)
- The HSAIL Debugger's Debugged Process Events viewer shows SIGPIPE or SIGBUS error while debugging HSAIL Applications. (1590)
- Multiwatch view is disabled while debugging HSAIL. (1628)
- API/Draw/Frame steps should be disabled while doing HSAIL debugging. (1648)
- Newly registered Windows Store Apps do not appear in the Project Settings list of apps. (1688)
- CPU Profiler doesn't launch Windows Store App that is specified in project settings. (1689)
- System Information dialog's OpenCL Devices tab appears empty on Linux. (1954)
- GPU debugger backend crashes when we close the Teapot window on I+A system. (2201)
- On some Windows 7 stations the GPU profiler command-line tool fails with error message "AMDTPBaseTools-x64.dll is missing". (2361)
- In Visual Studio Host+GPU debug session, execution of the debugged application resumes and doesn't break when performing a 'step out' operation. (2412)
- For some debugged applications, the HSA Debugger breakpoints are not hit. (2516)

- Frame Analysis runs out of memory when loading and displaying ~40 captured frame traces at once. (2561)
- Static Analyzer offline OpenCL build ignores the -l option with driver 16.2.1 and later. (2794)
- On Linux, the GPU Debugger updates the display of OpenCL and OpenGL runtime objects only when app execution is suspended for API breakpoint. Breaking at other types of breakpoints does not trigger CL/GL objects update. (2870)
- In Visual Studio, after using Frame Analysis to capture frame traces, clicking a frame thumbnail without stopping the session may lead to "Session Aborted" error pop up followed by "Failed to load frame data" error. (2893)
- Clicking the Power profiler Summary page in Visual Studio fails to open the page. (2897)
- Cannot open a GPU Profiler session once we rename it, after re-starting Visual Studio. (2912)
- Combining host + GPU debugger displays incomplete call stack when stepping into system functions on Linux. (2916)
- CPU Profiler does not display symbol information on importing a .prf file. (2942)
- Using combined host + GPU debugging from Visual Studio and clicking API step-in button multiple times may lead to Visual Studio hang. (2950)
- Combined host+GPU debugger fails to debug programs containing nested inlined C/C++ function. (2955)
- On Linux, GPU Profiling Performance Counters of an OpenGL application may cause a system hang after a few seconds. (68152)
- In Power Profile sessions on machines without Catalyst installed, 'iGPU Frequency' is constantly shown as 100MHz. If Catalyst is installed, then CodeXL reports the proper integrated-GPU frequency. (459364)
- Collecting GPU Profiler performance counters on the integrated GPU on an APU while another 3D app is running outside CodeXL can lead to a display hang. (68176)
- Debugging OpenCL kernels with optimizations disabled may cause kernel hang / driver not responding (TDR) in Radeon Software Crimson Edition (driver 15.30). (80095)

Support

Please use our [CodeXL Issues Page](#) for bug reports, support and feature requests.