



## CodeXL 2.2 GA Release Notes

### Contents

CodeXL 2.2 GA Release Notes .....	1
New in this version.....	2
System Requirements .....	2
Getting the latest Radeon software release .....	5
Note about installing CodeAnalyst after installing CodeXL for Windows.....	5
Fixed Issues .....	5
Known Issues.....	6
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](http://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:

- Support Linux systems installed with amdgpu-pro driver.
- Support for AMD Radeon RX 480 and RX 460 GPUs
- Support Ubuntu 16.04
- Support Windows 10 Anniversary Update
- Frame Analysis
  - Support for timeline trace of Vulkan® frames on Linux®.
  - Capturing the trace of multiple frames in a single capture operation, controlled by the project settings.
  - UI improvements
  - 'CPU only' and 'GPU only' trace capture
- CPU Profiling
  - New DB-based session data store allowing for quicker session post-processing and display of longer sessions
  - Support for the new A9, A6 and E2 series of the AMD 7<sup>th</sup> generation APUs.
- Power Profiling
  - Module level attribution (available in the Command Line tool on Windows only)
- Static Analysis
  - Support static analysis of OpenCL kernels, OpenGL and Vulkan GLSL shaders on Linux systems installed with amdgpu-pro driver.
  - Extract AMDIL code for DX11 shaders (available in the command line tool).
- GPU Debugging
  - Display referenced structures/values when viewing pointer variables
  - Support OpenGL-OpenCL interop
- Source Control and Build Environment
  - Remove git sub-module
  - Let the build succeed even if JDK and ROCm are not installed
  - Build compatible with both gcc 4.x on Red Hat 6 and gcc 5.x on Ubuntu 16.04
- UI Improvements
  - The CodeXL toolbar's Play button text reflects CodeXL's mode and action
  - The total number of frames captured in the session are displayed in the session node of the Explorer tree
  - Frame Analysis navigation bar displays a graphic indication of the precise location in the frame timeline when moving the mouse over the focus area in the timeline chart
  - Many more improvements

## System Requirements

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

- **Frame Analysis**

- Radeon Software Crimson Edition 16.7.3 (driver 16.30.2311) is the recommended driver. See "[Getting the latest Radeon software release](#)" section below.
- **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 16.8.1 (driver 16.30) is the recommended driver on Windows, and the latest amdgpu-pro (driver 16.30) on Linux.
  - 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 (available [here](#)) 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)
  - CodeXL supports ROCm version 1.1.2.
  - The following ROCm packages must be installed:
    - Radeon Open Compute Runtime (ROCR).  
Available here: <https://github.com/RadeonOpenCompute/ROCR-Runtime>
    - Radeon Open Compute GPU Debug SDK.  
Available here: <https://github.com/RadeonOpenCompute/ROCm-GPUDebugSDK>
  - 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/ROCm-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
    - Radeon Software Crimson Edition 16.7.3 (driver 16.30.2311) is the recommended driver.
  - OpenGL shader analysis on Windows requires Catalyst 15.9. (driver 15.20) or later
  - OpenGL shader analysis on Linux requires either
    - Radeon Software Crimson Edition (driver 15.30)
    - or
    - amdgpu-pro driver version 16.40 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 (including Windows 10 Anniversary Update).
  - 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 16.04 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 2.1 release and are new to this version:

- Linux OpenGL applications exit immediately when run with CodeXL GPU Debugger on Ubuntu 15.10 + Mesa driver. (1722)
- Breaking a GPU Debugging session with an API breakpoint displays a 'source file was not found' message. (2656)
- Static Analyzer offline OpenCL build ignores the -l option with driver 16.2.1 and later. Fixed in driver version 16.30 (2794, 90709)
- Wrong Radeon Software version shown in CodeXL's About dialog. Fixed and supported starting Radeon Software release 16.5.2 or later. (2879)
- Clicking the Power profiler Summary page in Visual Studio fails to open the page. (2897)
- 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. Fixed and CodeXL is now on par with gdb behavior. (2955)
- The GPU Debugger's "skip breakpoints and enter kernel debugging" dialog requires clicking 'No' multiple times when running in the Visual Studio extension. (2975)
- CodeXL Frame Analysis fails to detect the exit of game, when game is launched via Steam. (2977)
- Selecting to install only the Remote Agent component on Windows does not install the Power Profiler driver, which may cause crashes when a counters selection is later initiated. (2990)
- Static Analyzer fails to build Vulkan programs on Linux when run using a non-root login. (2993)
- Frame Analysis Capture/Stop buttons active even after profiled game exits. (2999)

- GPU Debugger crashes when stepping into Teapot's OpenCL kernel during GUI automation testing. (3015)
- Static Analyzer fails to build OpenCL kernel which uses recursion and raises unhandled exception. Fixed in driver version 16.40. (3063, 93647)
- In Frame Analysis, irrespective of "Number of Frames to capture" option, the UI displays a single frame number per capture. (3070)
- Some combinations of Vertex and Fragment shaders may cause the Static Analyzer to not generate statistics when building a rendering pipeline Vulkan program. (3086)
- Cannot use Frame Analysis on Windows if the 'Remote Agent' component was not installed (3101)
- CPU Profiler Assess Performance sessions display wrong IPC at the process level. (3102)
- Teapot crashes when exiting, running with amdgpu-pro driver on Ubuntu 16.04. Fixed in driver 16.40. (3227, 97562)
- Crash when Static Analyzer builds OpenCL kernels with binary output format set to BIF3.0. (3141)
- Static Analyzer's build of DirectX shaders fails when targeting A6-85xx, A8-86xx, A10-87xx, A12-88xx APUs or Radeon R5 M255, R7 M260, R9 285/380/380X/M295X/M390X/M395/M395X/M485X GPUs. Fixed in driver 16.40. (3271, 97975)

## Known Issues

- Debugging OpenCL kernels that use read-modify-write atomic operations is not supported.
- GPU Debugging of AMD APP SDK samples on Linux fails because of RPATH value in the SDK samples binaries. Workaround:
  1. Remove RPATH from the SDK sample using `chrpath -d sample_path`  
Example: `chrpath -d /opt/AMDAPP/samples/ocl/bin/x86_64/BoxFilterGL`
  2. Verify that RPATH has indeed been removed using  
`readelf -d sample_path | grep PATH`  
This should not show RPATH (it might show RUNPATH, which is okay).  
Example `readelf -d /opt/AMDAPP/samples/ocl/bin/x86_64/BoxFilterGL | grep PATH`
  3. If RPATH is showing in step #2 above then perhaps you did not have permission to make the change, so try step #1 again with su permissions: "`sudo chrpath -d ...`"
- 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)
  1. 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)
- 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 "AMDTBaseTools-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)
- 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)
- Cannot open a GPU Profiler session once we rename it, after re-starting Visual Studio. (2912)
- CPU Profiler does not display symbol information on importing a .prd file. (2942)
- FLAT\_\* ops get classified incorrectly as 'Vector ALU' in Analyze Mode. (3080, [GitHub issue #25](#))
- The GPU debugger does not display OpenGL static buffers when running inside the Visual Studio extension and 'Break' is clicked. (3167)
- Modules are identified as 'Unknown' in Power Profiler sessions when the profiled process is run after the profiler's command line tool session began. (3168)
- Some Steam games may crash when Steam is launched from CodeXL Frame Analysis mode. (3172)
- Visual Studio displays error "The following session files could not be deleted" when deleting a CodeXL Power Profiling session. (3179)
- Filtering the CPU Profiling display based on CPU Core/Numa is disabled in the display filter. (3233)
- When CPU Profiling on Linux, C++/Java inline functions are shown with generic names or missing. (3240)
- The Vulkan versions of Doom and The Talos Principle fail to start when launched from Steam as part of a Frame Analysis session on Windows. (3364)
- When launching the Vulkan version of DOTA2 from Steam as part of a Frame Analysis session on Linux, The game must be manually shutdown at the end of the session else viewing frame traces and session export will fail. (3381)
- CPU Profiling command line tool throws "Output directory does not have write permission" on using relative paths in command line arguments. (3409)
- Function information is sometimes missing from the csv file generated by the CPU Profiler command line tool's 'report' command. (3480)
- Changing functions in the CPU Profiler's source view for a session of a profiled Java application causes a crash. (3484)
- The CPU Profiler's Overview '5 Hottest functions' table does not filter out JVM functions when profiling Java applications on Linux. (3497)
- Double clicking a function name in the CPU Profiling session's Functions view displays an empty source view for profiled Java applications on Linux. (3498)
- Double clicking a module name in the CPU Profiling session's Overview displays an empty source view for profiled Java applications on Linux. (3499)



- Stopping a CPU Profiling session where the CodeXLPowerProfiler is the profiled application and the session was initiated from CodeXL GUI causes kernel panic on Linux (Ubuntu). (3500)
- For CPU Profiling system-wide profile, right clicking a module and selecting "Display in function view" opens the Functions tab displaying all of the session functions. (3516)
- For a session containing multiple profiled processes, the CPU Profiler's source code view's TID drop-list contains only the 'all threads' value. (3519)
- On Linux platform, switching between [PLT] and regular functions in the CPU Profiler's Source code view drop down list, leads to crash. (3524)
- CPU Profiler displays an empty source code view on double clicking any function with "Unknown Module(PID:)" entry. (3529)
- Considerable number of "Unknown Function" entries in "Functions" table of call graph tab with CPU Profiler's system-wide profile. (3530)
- Callstack data is not displayed when importing a CPU Profiling session from an ebp file. (3537)
- 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 or Radeon Software installed, 'iGPU Frequency' is constantly shown as 100MHz. If Catalyst or Radeon Software 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.