## General GUI Controls

Several CodeXL views and controls are used by the debugger, the Profiler, and the static analyzer.

* **[CodeXL Welcome Page](#_CodeXL_Welcome_Page)**
* [**Getting Started Dialog**](#_Getting_Start_Dialog)
* **[Project Settings](#_topic_ProjectSettings)**
* [**Execution Toolbar**](#_topic_ExecutionToolbar)
* [**Properties View**](#_topic_PropertiesView)
* [**CodeXL Explorer**](#_topic_CodeXLExplorer)
* [**Global Settings dialog**](#_topic_GlobalSettings)
* [**System Information Dialog**](#_topic_SystemInformationDialog)
* [**CodeXL Search Toolbar**](#_CodeXL_Search_Toolbar)

### CodeXL Welcome Page

CodeXL Welcome Page lets you access or create projects. It also contains links to common tasks you can be interested in.

The Welcome Page is opened in CodeXL startup. To access the Welcome Page, on the View menu, click Start Page.

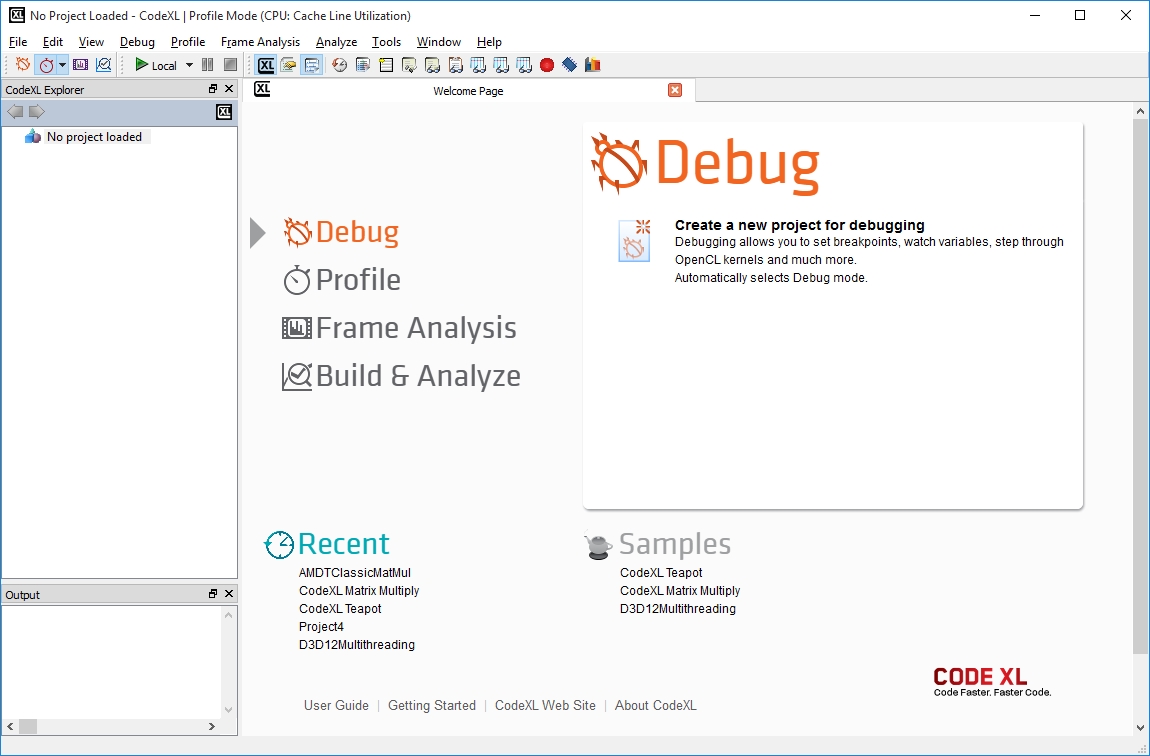
The Welcome Page is divided into three main sections:

**Modes menu:** Browse CodeXL modes, and select a task for quick project creation:

|  |  |
| --- | --- |
| Create New Project for Debugging | Use this option to create a new CodeXL project. After choosing this option, CodeXL will select “Debug Mode” in the execution mode toolbar, and will prepare the project for a debugging session. |
| Create New Project for Profiling | Use this option to create a new CodeXL project. After choosing this option, CodeXL will select “Profile Mode” in the execution mode toolbar, and will prepare the project for a profiling session. |
| Attach to Process | Select this option when you want to perform a CPU profiling session on an existing process. |
| System-wide Profiling | Use this option when you want to perform a CPU or power system-wide profiling session on the currently running processes on your machine. |
| Create a new OpenCL file for Analysis | Selecting this option will create an empty new OpenCL kernel file, with a default name. After editing the new create kernel name, CodeXL will allow you to paste code or edit the kernel code and then build and analyze this code on selected devices. |
| Add an existing OpenCL file for Analysis | Select this to add an existing OpenCL kernel file and use CodeXL to build and analyze the kernel code on selected devices. |

**Recent projects:** The list displays projects you have worked on recently. Clicking on one of the links will open the project for work.

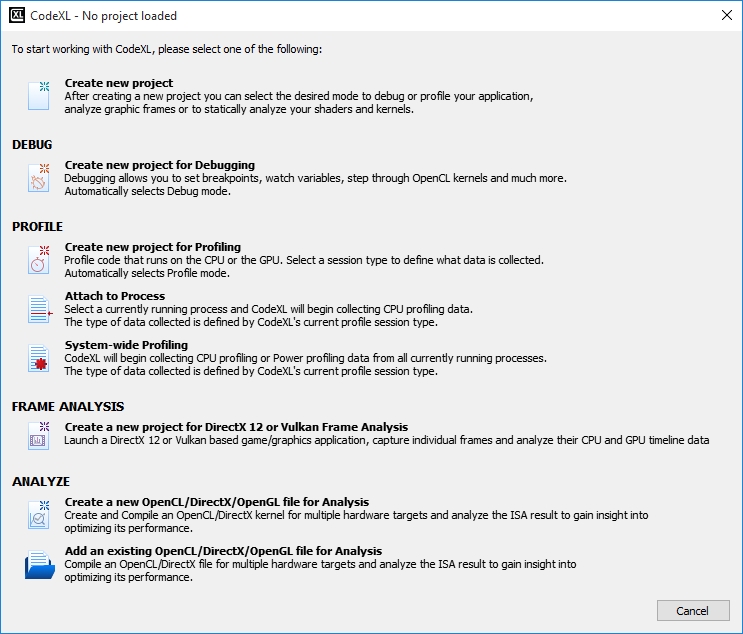
**Samples:** Click on CodeXL Teapot sample link, to open the sample project. The sample can be used for getting to know CodeXL capabilities.



### Getting Started Dialog

Another way to get start with CodeXL is using the startup dialog. Click the “Start” button (see the following screenshot). A dialog specifying the different options for getting started with CodeXL opens.



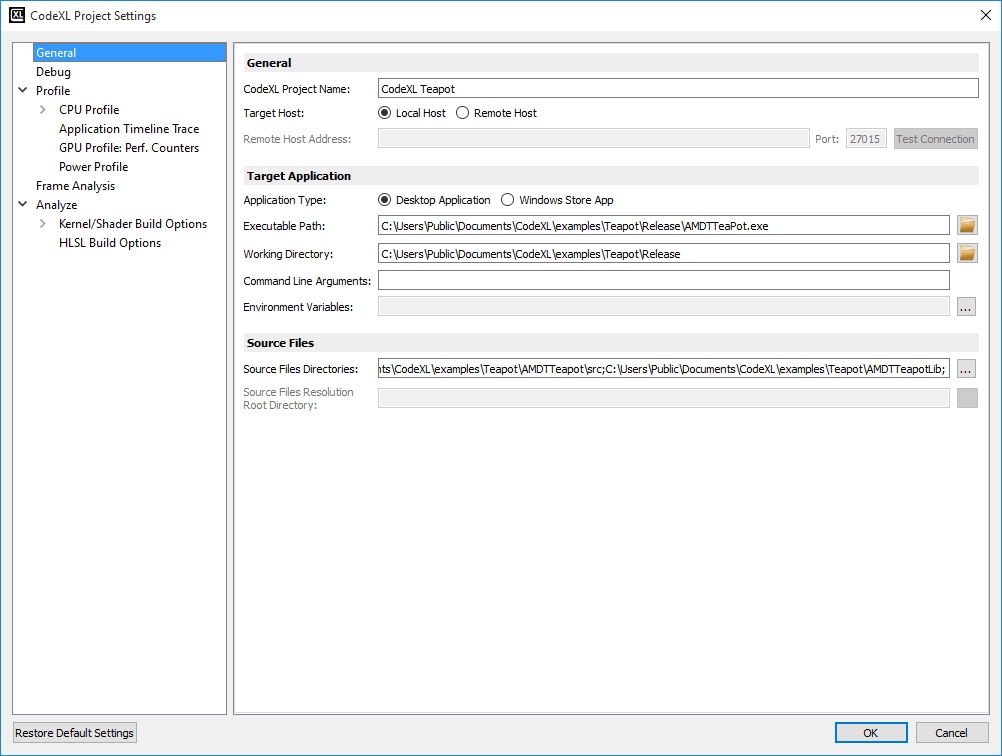


|  |  |
| --- | --- |
| Create New Project | Use this option to create a new CodeXL project. The project settings dialog will open and guide you through the steps to create a project with your desired settings. |
| Create New Project for Debugging | Use this option to create a new CodeXL project. After choosing this option, CodeXL will select “Debug Mode” in the execution mode toolbar, and will prepare the project for a debugging session. |
| Create New Project for Profiling | Use this option to create a new CodeXL project. After choosing this option, CodeXL will select “Profile Mode” in the execution mode toolbar, and will prepare the project for a profiling session. |
| Attach to Process | Select this option when you want to perform a CPU profiling session on an existing process. |
| System-wide Profiling | Use this option when you want to perform a CPU or power system-wide profiling session on the currently running processes on your machine. |
| Create a new OpenCL file for Analysis | Selecting this option will create an empty new OpenCL kernel file, with a default name. After editing the new create kernel name, CodeXL will allow you to paste code or edit the kernel code and then build and analyze this code on selected devices. |
| Add an existing OpenCL file for Analysis | Select this to add an existing OpenCL kernel file and use CodeXL to build and analyze the kernel code on selected devices. |

### Project Settings

To start debugging or profiling an application:

1. Create a CodeXL project.  
   A CodeXL project consists of general information for the debugged / profiled application, such as command-line arguments, environment variables. The project also configures debugging and profiling specific configurations.
2. Use the File > New Project menu to open the new project dialog.  
   The same dialog can be used later to configure the project settings during the debugging / profiling. (Debug > Debug Settings and Profile > Profile Settings menus)



General Project Settings

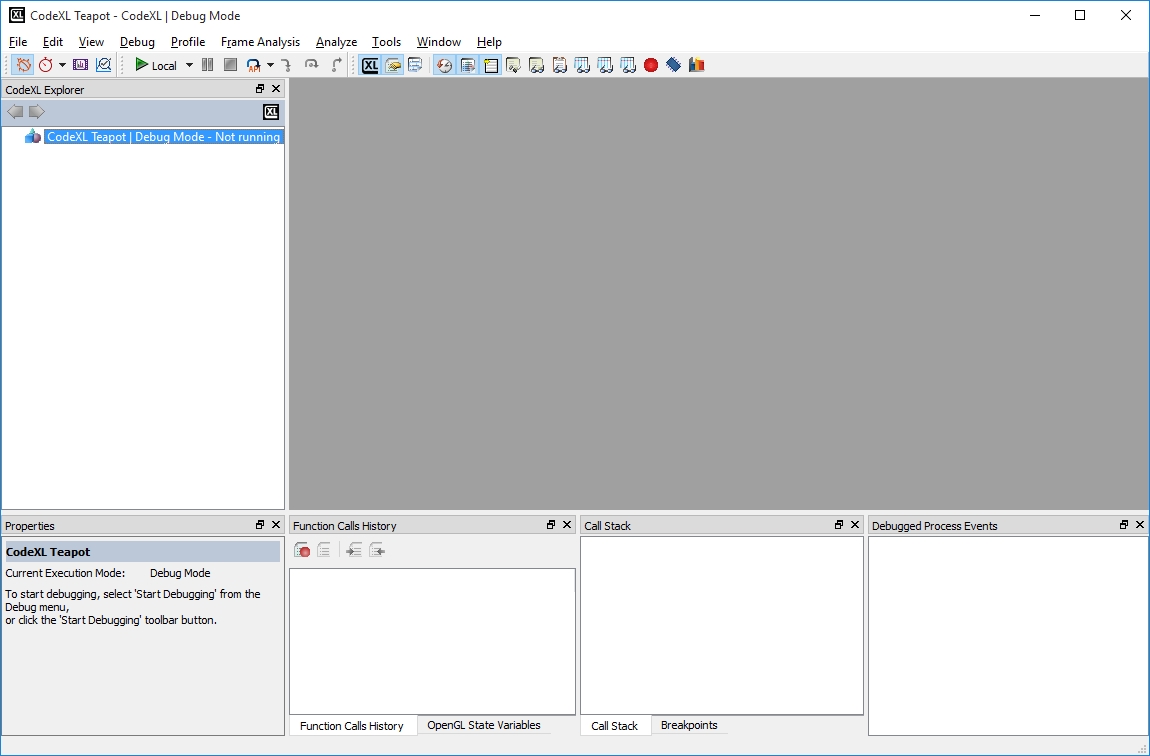
|  |  |
| --- | --- |
| CodeXL Project Name | The name of the current project used to identify a project in the Explorer views. |
| Target Host | The host on which the executable file will be debugged or profiled (Local / Remote). |
| Remote Host Address | The remote host address when the “Remote Host” option is selected. |
| Port | The port number on the remote host when the “Remote Host” option is selected. |
| Test Connection | Click to test the current connection settings. |
| Application Type | The type of application that will be debugged or profiled. The application can be either a desktop application or a Windows Store application. |
| Executable Path | The path to the executable / Windows Store application to be debugged or profiled. Use the Browse button for quick selection. |
| Working Directory | The directory in which the executable is to be debugged or profiled. Use the Browse button for quick selection. |
| Command Line Arguments | Arguments to be passed to the executable. |
| Environment Variables | A list of environment variables and their values to be set in the environment for the executable. |
| Source Files Directory | List of directories with the source code for the CPU profiled applications or the OpenCL™ kernels. Use the Browse button for quick selection. |

Specific Project Settings

* [**GPU Debugging Settings**](#_topic_GPUDebuggingProjectsettings)
* [**CPU Profiling Settings**](#_topic_CPUProfileProjectOptions)
* [**GPU Profiling Settings**](#_topic_GPUProfilingProjectSettings)

### Execution Toolbar

Once a CodeXL project is started, the initial interface is displayed, as shown in the following screenshot.



CodeXL Initial Layout when a project is loaded

The CodeXL toolbar allows quick access to CodeXL views and main controls.



|  |  |  |
| --- | --- | --- |
| Debug Mode |  | Switch CodeXL to Debug mode. This is the default mode. |
| Profile Mode |  | Switch CodeXL to Profile mode. Use the Profile menu to switch between the various profile modes. |
| Frame Analysis Mode |  | Switch CodeXL to Frame Analysis mode. |
| Analyze Mode |  | Switch CodeXL to Analyze mode. |
| Start Debugging / Profiling |  | Start the startup project with CodeXL in the selected mode. |
| Pause / Stop Debugging / Profiling |  | Pause / Stop the startup project with CodeXL in the selected mode. |
| Debug Steps |  | API step / Draw Step / Frame Step / Step In / Step Over / Step Out the debugged application. |
| CodeXL Explorer |  | Show the CodeXL Explorer tree view. |
| Properties View |  | Show the Properties view. |
| Output View |  | Show the Output view |
| Function Calls History View |  | Show the Function Calls History view. |
| Debugged Process Events View |  | Show the Debugged Process Events view. |
| Call Stack View |  | Show the Debugger Call Stack view. |
| Locals View |  | Show the Debugger Locals view. |
| Debugger Watch View |  | Show the Debugger Watch view. |
| Debugger OpenGL State Variables View |  | Show the Debugger OpenGL State Variables view. |
| OpenCL Debugger Multiwatch View 1, 2, 3 |  | Show the OpenCL Debugger Multiwatch views. |
| Debugger Breakpoints View |  | Show the Debugger Breakpoints view. |
| Debugger Memory View |  | Show the Debugger Memory view. |
| Debugger Statistics View |  | Show the Debugger Statistics view. |

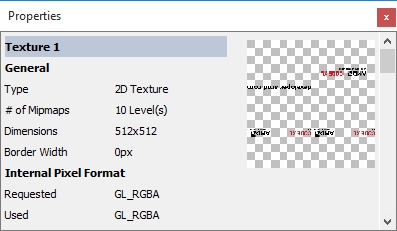
### Properties View

The CodeXL properties view displays detailed information about items appearing in other CodeXL views.

OpenCL Image Properties

1. Debug an OpenCL sample.
2. Break after clCreateContext.
3. Click the OpenCL Context object in the [**CodeXL Explorer**](#_topic_CodeXLExplorer) to see the properties view content.

As shown in the image, the properties view specifies the image type, the image handle, dimensions and format.

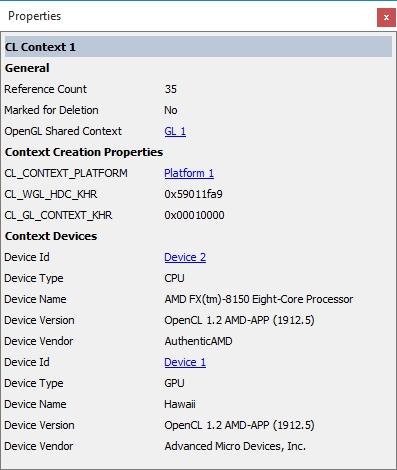


Clicking on the image thumbnail in the properties view will open an [**image view**](#_topic_ObjectImageview).

OpenCL Context Properties

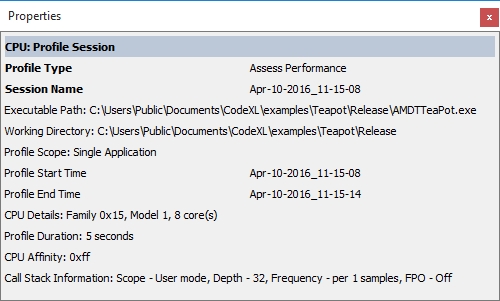
1. Debug an OpenCL sample with images
2. Break the application.
3. Click on an OpenCL context in the [**CodeXL Explorer**](#_topic_CodeXLExplorer) to see the properties view content.  
   The properties of an OpenCL context contains the context run time properties.

By clicking the context platform or devices link, the [**System Information Dialog**](#_topic_SystemInformationDialog) is displayed. This shows platform / devices details.



Profile Session

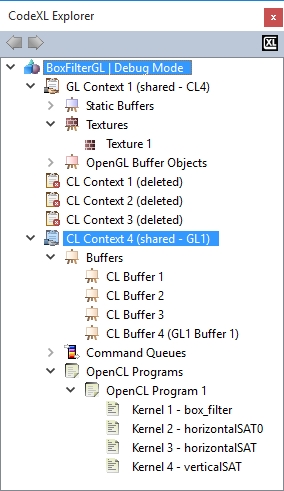
1. Set CodeXL execution mode to profile.
2. Click on one of the sessions in the [**CodeXL Explorer**](#_topic_CodeXLExplorer) to see the properties view content.  
   Session properties view displays the session type, the session file path and working folder, and few of the session properties.



### CodeXL Explorer

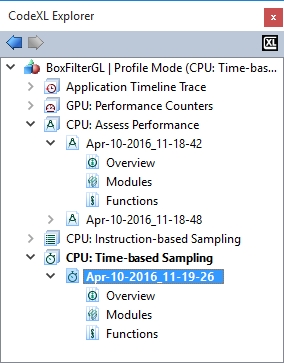
The CodeXL Explorer tree lets you navigate while debugging, profiling, or analyzing.

The tree structure is different for each of the execution modes. In Debug mode, the tree contains the debugged application details **only when the debugged application is paused.** Only one debug tree can be viewed while debugging. In Profile mode, on the other hand, the tree contains all the profiled sessions for the current project. You can view multiple sessions. When switching to Analyze mode, the tree will display all the OpenCL kernel files added to the current project, with build and analyze results for each of them.



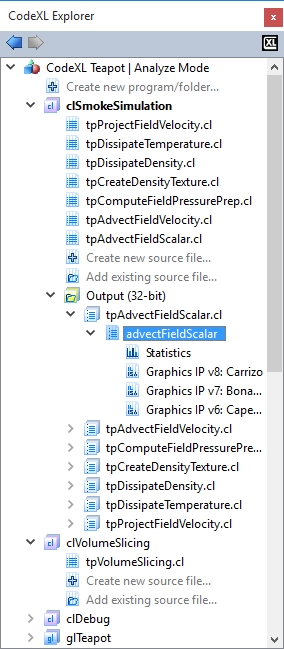
OpenCL / OpenGL Debugged Objects Tree

While debugging, the CodeXL Explorer tree view lets you navigate and access all OpenCL and OpenGL objects during an application run. The Objects tree displays all the debugged application's allocated objects in the hierarchy: Application –> Context –> (Object Type) –> (Specific Object). Contexts marked with the sharing icon have their memory allocated objects shared by at least one other context. See cl\_khr\_gl\_sharing in the *OpenCL Specification*, and wglShareLists in the *OpenGL Specification*, for more details about context resource sharing. Note that sharing contexts can have memory sizes even after being deleted, as long as there is at least one other context sharing their objects.



CodeXL Profile Sessions Tree

In Profile mode, the CodeXL Explorer lists the profiling sessions done for the current project; it indicates the type of profile for each session. When a CodeXL project is opened, the Profiler Session Explorer automatically displays all previous sessions associated with that project. Double-click on a session to open the session data view. Right-click on a session to rename it, delete it or to open the folder that contains the session output file. You can also import a previously-generated Profiler session, or delete all sessions using the context menu in the Profiling Session Explorer. A previously-generated Profiler session can be imported by dragging and dropping the session file from the system file browser to the Session Explorer.



CodeXL Analyze Tree

In Analyze mode, the tree displays 2 sub trees:

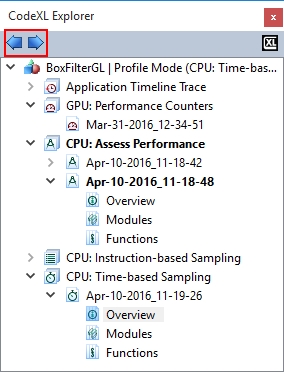
1. Tree of programs created by the user, with the output build results.
2. Tree of OpenCL / DirectX / OpenGL / Vulkan shader and kernel files added by the user and used while building the programs.

* Double clicking on a source file will open the source file in CodeXL
* Double clicking on an output build results will open open a view that will show the source and IL and/or ISA if applicable for the specific device. The build results are grouped by kernel and then by device families.
* Double-clicking the “Statistics” and “Analysis” nodes will open the appropriate information view.
* You can drag a source file from the tree, in order to place it on a program.

The Tree also allows quick activation of the “Create new source file” and “Add existing source File” commands via double-clicking the two nodes at the bottom of the tree.

Back / Forward Buttons

The two tree navigation buttons are located in CodeXL Explorer top panel. Use these buttons to navigate to the previously viewed objects in the tree. The navigation history resets when opening another project or a debug session is terminated.

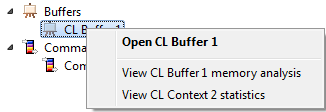


CodeXL Explorer context menu

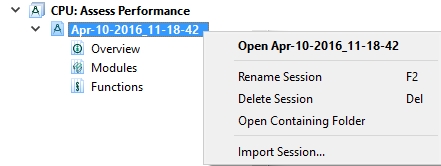
Use the tree context menu to navigate and manage the current displayed project data.

For example, see the following context menus:

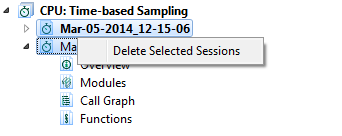
* Context menu on a CL buffer item when the debugged process is paused. The menu will contain the option to display the buffer in an MDI window and to view the buffer memory analysis and statistics.



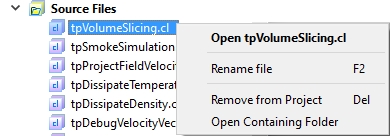
* Context menu on a Time-Based CPU profile session. The menu will contain the options to open the session in an MDI window, rename or delete the session, and open the folder that contains the session.



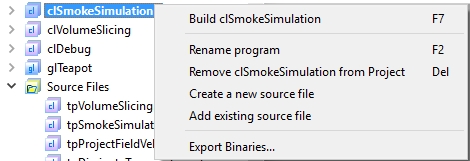
* Context menu on multiple sessions in profile mode. The menu will enable the deletion of multiple sessions.



* Context menu on source file. This menu enables one to open the source file, rename it, remove it from the CodeXL project, and open the containing folder.

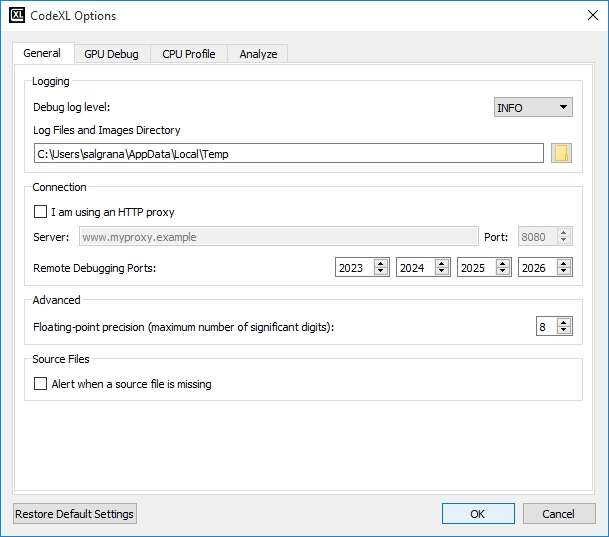


* Context menu on a program. This menu enables one to run the build program command, to rename, remove or change the source of a program.



### Global Settings

This dialog box lets you specify global settings for the CodeXL environment.



General Global Settings

|  |  |
| --- | --- |
| Debug Log Level | The debug log level is an internal mechanism that helps fix CodeXL problems. There are three levels.   * Error: Log errors that occur while running CodeXL. * Info: Log errors and CodeXL internal information seen while running CodeXL (default level). * Debug: Log CodeXL debugging information, errors, and other internal information that is generated while running CodeXL. (This can be used by the CodeXL support team to locate a problem inside CodeXL). * Extensive: Log extensive CodeXL debugging information, errors, and other internal information that is generated while running CodeXL. (This can be used by the CodeXL support team to locate a problem inside CodeXL). |
| Log Files and Images Directory | The directory in which the API call log files and temporary image and source files are saved. |
| I am using an HTTP proxy server | Check this option if your computer connects to the Internet through an HTTP proxy. If you are experiencing problems with CodeXL's online features (check for updates and send error report), changing the proxy settings might be the solution. |
| Proxy server / Port number | If you are using a proxy server, please input its information here. The server name could be a DNS address or an IP address. If you are unsure of what your HTTP proxy server or port are, contact your network administrator or copy the settings from your web browser. CodeXL does not currently support the HTTP\_PROXY environment variable. |
| Remote Debugging Ports | Specifies the port numbers that are being used for remote debugging sessions. On the client machine, all four ports should be able to receive incoming connections, and on the remote machine all four ports should not be blocked for outgoing connections. |
| Floating-point Precision | The display precision for displaying floating-point numbers. |
| Alert when a source file is missing | Alert the user when a source file is not found during the profile session source code display. |
| Restore Default Settings | Click this button to restore all settings on all pages to their default values. |

Plugin Global Settings

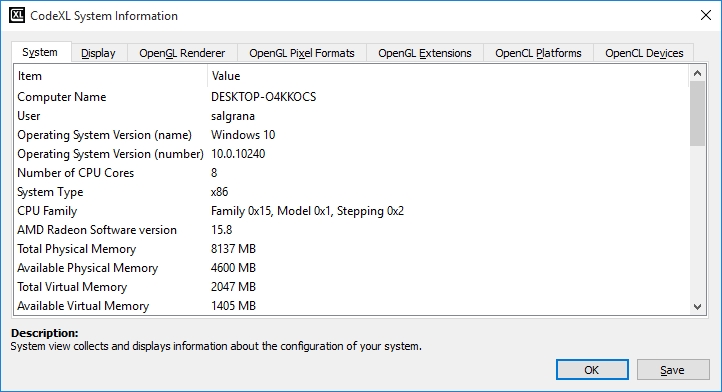
[**GPU Debugging Global Settings**](#_topic_GPUDebuggingGlobalsettings)

[**CPU Profiling Global Settings**](#_topic_CPUProfileGlobalSettings)

**[Static Analysis Global Settings](#_Analyze_Mode_Options)**

### System Information Dialog

The system information dialog displays computer configuration details and OpenCL / OpenGL implementation details. These include operating system, memory, graphic card and driver details; available OpenCL platforms and devices; monitor details, available pixel formats, and available OpenGL extensions.

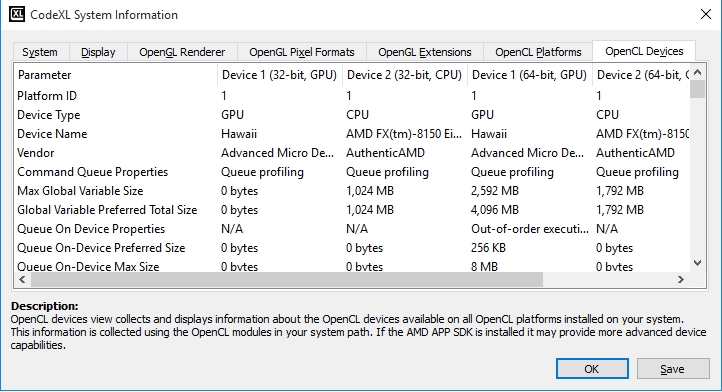


System Information Tabs

|  |  |
| --- | --- |
| System | Collects and displays information about the system configuration. |
| Display | Collects and displays information about the display system configuration. |
| OpenGL Renderer | Collects and displays information about the graphics accelerator configuration. |
| OpenGL Pixel Formats | Collects and displays information about the system’s supported pixel format. |
| OpenGL Extensions | Collects and displays information about the system’s supported OpenGL extensions. |
| OpenCL Platforms | Collects and displays information about the supported OpenCL platforms. |
| OpenCL Devices | Collects and displays information about the supported OpenCL devices and their capabilities and limits. |
| Save Button | Exports the system information data from all views to a .csv file. |

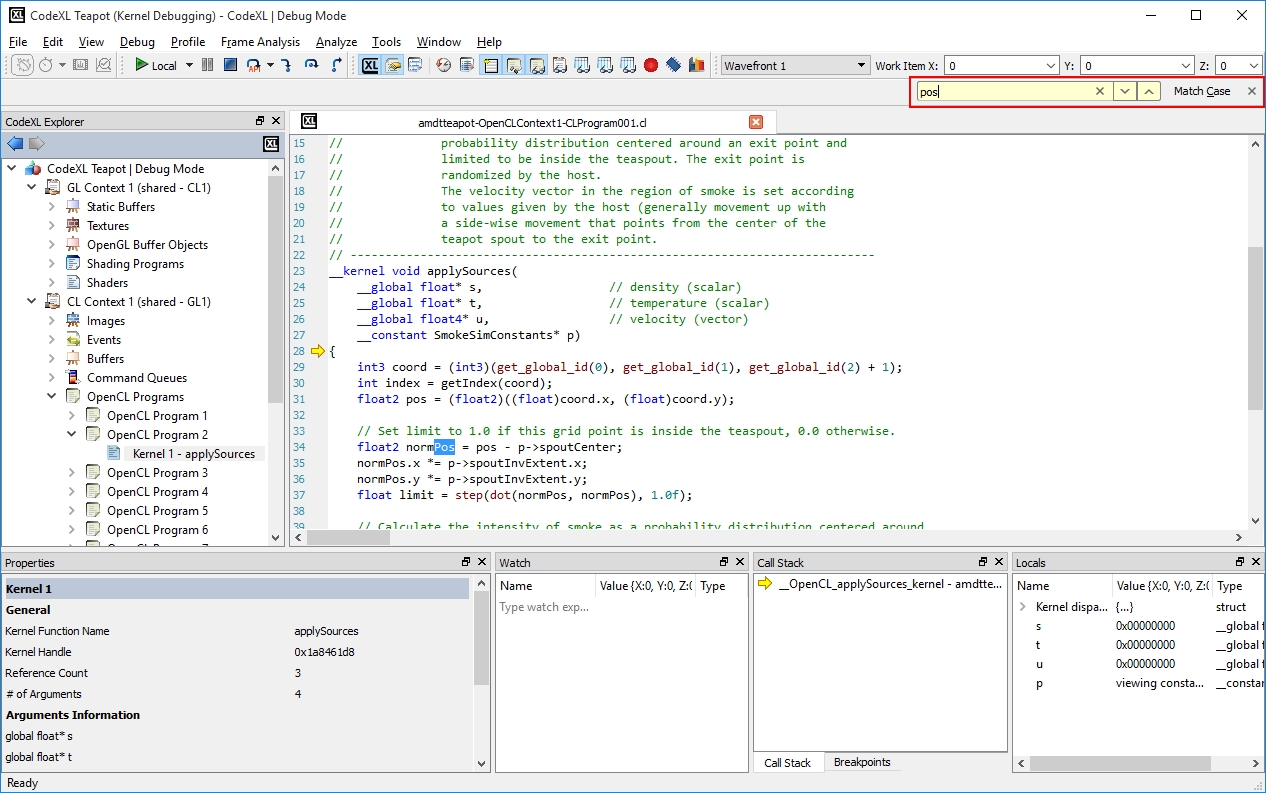
OpenCL Devices View

The OpenCL Devices View displays the OpenCL devices available in the 32-bit and 64-bit OpenCL runtimes that are installed on the local station. The information is collected using the modules from the system path and may differ from runtimes installed elsewhere, such as OpenCL modules installed in the AMD APP SDK folder.



### CodeXL Search Toolbar

Search toolbar appears when a view with searchable text is in focus and the Ctrl+F shortcut is used or the Find command is clicked in the Edit menu:



The user can navigate through a view’s text content using the Previous and Next buttons. Use the Match Case toggle button to control the case sensitivity of the search.