## Remote GPU Profiling and GPU Debugging

CodeXL provides remote profiling and debugging capabilities. Using these features, you can execute GPU profiling and debugging sessions of applications that run on a remote machine. This is useful for working with tablets and headless server units.

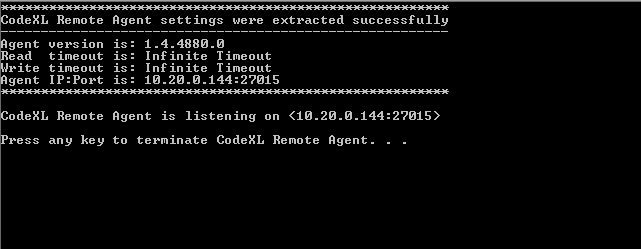
* [Running CodeXL Remote Agent](#_Running_CodeXL_Remote)
* [The Agent’s Configuration File](#_The_Agent’s_Configuration)
* [Performing Remote GPU Profiling](#_Performing_Remote_GPU)
* [Performing Remote GPU Debugging](#_Performing_Remote_GPU_1)

### Running CodeXL Remote Agent

As a first step, be sure that the remote machine has CodeXL Remote Agent installed. CodeXL Remote Agent ships with the CodeXL installer, and it is installed by default when installing CodeXL. You can also choose to install only CodeXL Remote Agent when using the installer.

CodeXL Remote Agent runs on the remote machine, and allows CodeXL clients located on other machines to connect the remote machine and execute GPU profiling and debugging sessions of applications on that machine.

When the remote is launched, it will output to the console a message in the following format:



In case the remote agent failed to launch successfully, it will output to the console a message describing the problem.

Please notice that by default the remote agent binds itself to the first valid IP address that it finds. In most cases, this would be your desired behavior. However, if the remote machine has multiple IP addresses, and you would like to force CodeXL remote agent to bind itself to a particular address, you can use the --ip command line switch (--ip <ip\_address>). For example, to force the remote agent to bind itself to 10.20.0.155, use the following command:

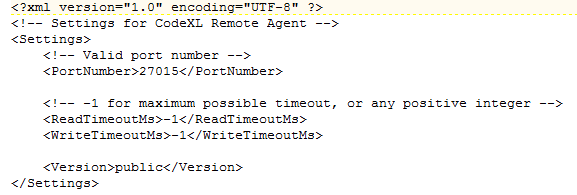
On Windows: CodeXLRemoteAgent --ip 10.20.0.155

On Linux: CodeXLRemoteAgent-bin --ip 10.20.0.155

### The Agent’s Configuration File

CodeXL Remote Agent ships with a simple configuration file named CodeXLRemoteAgent.xml, which is located at the same folder as the CodeXLRemoteAgent executable. This configuration file defines several parameters which are being used by the agent. These parameters are being read by the agent just before it starts running. Therefore, if you change one of the values in the configuration file, you must rerun the agent for it to read the new values.

Here is a screenshot of the agent’s configuration file:



|  |  |
| --- | --- |
| Field | **Description** |
| PortNumber | The port number on which the remote agent listens to incoming connections. |
| ReadTimeoutMs | Read timeout to be used by the remote agent for incoming connections. |
| WriteTimeoutMs | Write timeout to be used by the remote agent for incoming connections. |
| Version | Version type, do not change this value. |

Please pay special attention to the <PortNumber> parameter. This is the port on which the remote agent will listen to incoming connections. Prior to starting CodeXL remote Agent, you should verify that:

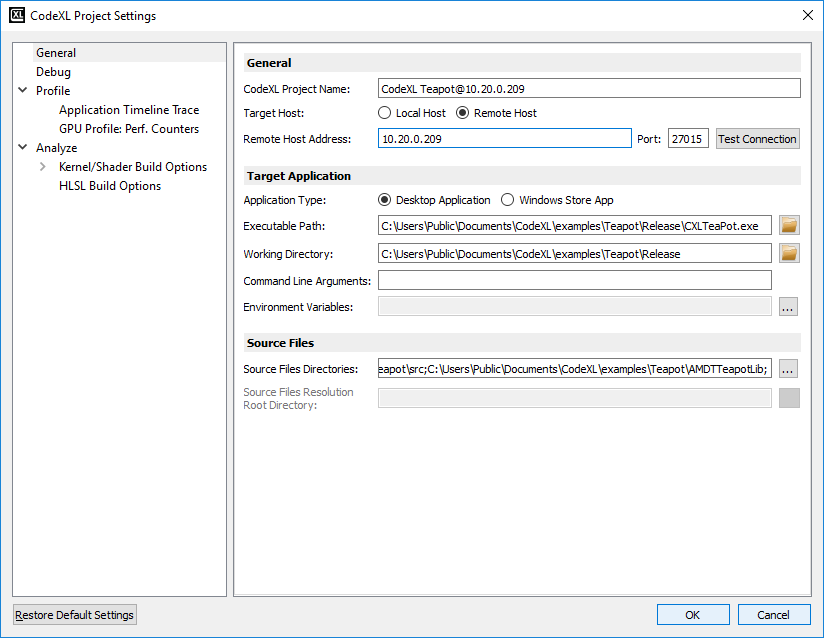
1. CodeXL Remote Agent is not blocked by the firewall on the remote machine.
2. The port number (27015 in the example above) is available (not being used by any other process), and that it is not blocked by a firewall. If there is a problem with the port, you can change this value and rerun the agent.

That’s it for the remote machine. Now, go back to the local machine where the CodeXL application is installed. The following sections describe how to perform remote GPU Profiling and Debugging.

### Performing Remote GPU Profiling

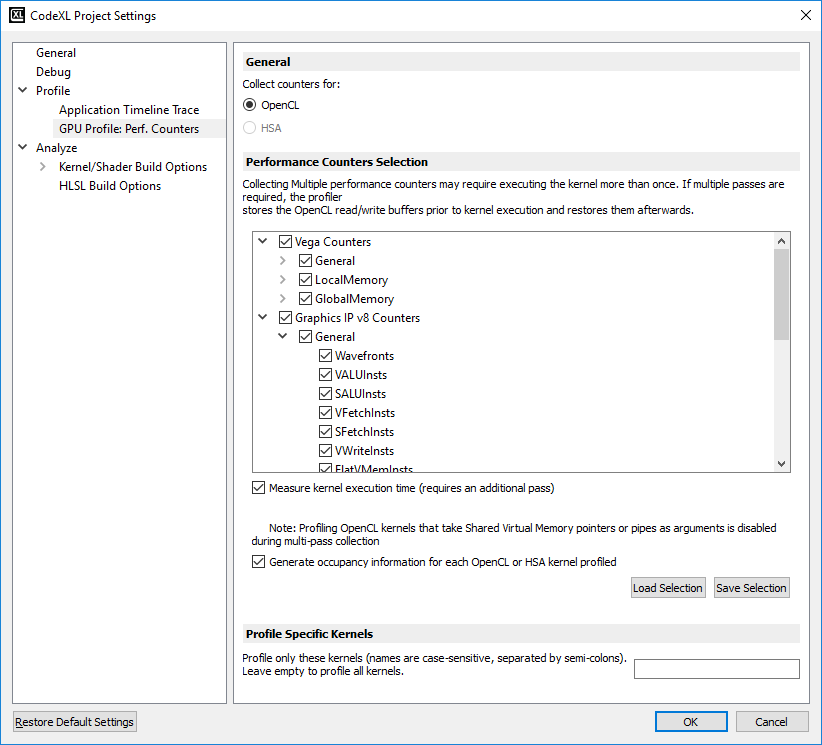
1. Setting the remote target:

When you open the CodeXL project settings, look at the general tab:



|  |  |
| --- | --- |
| UI Control | **Description** |
| Local Host and Remote Host radio buttons | These two radio buttons determine whether we are in a remote or local session. If Remote Host is selected, all locations under the Target Application box will refer to paths on the remote machine. Otherwise, all locations will refer to paths on the local machine, as usual. |
| Remote Host text box | The IP address of the remote machine (the machine on which the remote agent is running). In the above screenshot, the remote host is 10.20.0.144.  Alternatively, the host name of the remote machine can be used instead of its IP address, provided the name is recognized by the network DNS. |
| Port text box | The port (in the remote machine) on which the remote agent is listening. Note that the value of Port should match the PortNumber element value in CodeXLRemoteAgentConfig.xml which is located on the remote machine. |
| Test Connection Button | When this button is pressed, CodeXL will try to connect to the remote agent running on IP:Port (according to the text in the Remote Host Address and Port text boxes). This button can be used to verify that the remote agent is running on the remote machine and that it is reachable by the CodeXL client. Note that if the remote agent is not running on the remote machine or if a firewall blocks either CodeXL client or the remote agent – the connection test would fail. |

1. For remote GPU performance counters profiling, have a look at the GPU Profile: Performance Counters tab (application trace settings act the same as in local sessions):



In a remote session, CodeXL does not know which counters are supported by the remote target’s GPU. Therefore, all possible HW families are available. In case that you picked a counter that is not supported by the remote host, you will see a blank column in the result spreadsheet (can be easily eliminated by using the “Hide zero columns” feature).

1. To launch Remote GPU profiling, click OK. Then, start profiling as usual.

### Performing Remote GPU Debugging

1. Setting the remote debugging ports:

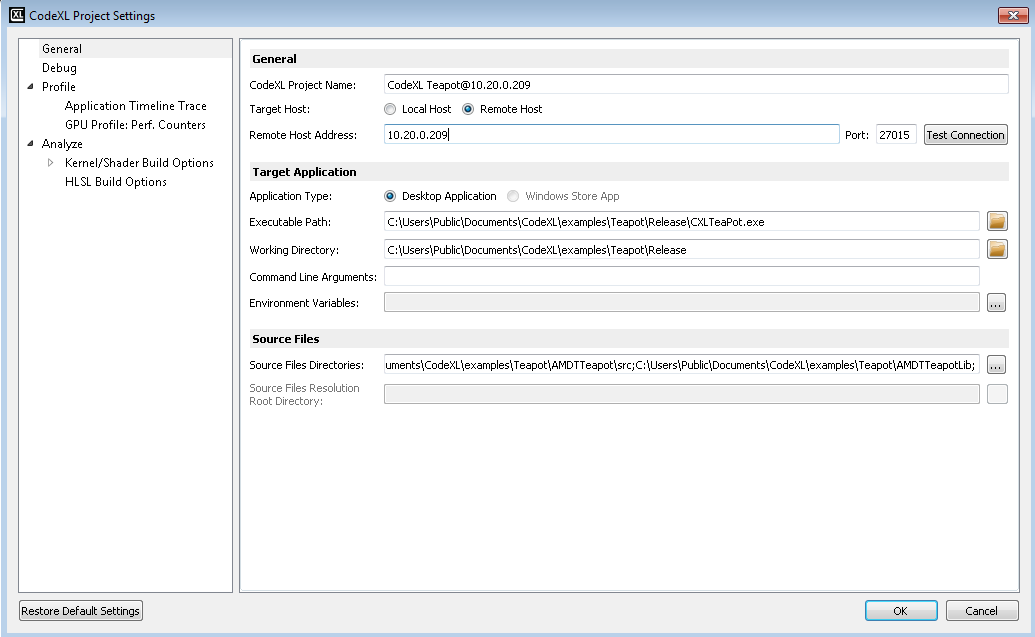
Go to Tools->Options, open the General tab and look at the Connection section. In the Remote Debugging Ports subsection, there are 4 ports which are being used by CodeXL on the local machine for remote debugging purposes. Please verify that the following requirements are fulfilled:

* 1. All 4 port values are distinct.
  2. None of the 4 ports is blocked by a firewall.
  3. None of the 4 ports is being used by another process.

If necessary, you can change the port values to fulfill the above requirements.

1. Setting the remote target:

When you open the CodeXL project settings, look at the general tab:



|  |  |
| --- | --- |
| UI Control | **Description** |
| Local Host and Remote Host radio buttons | These two radio buttons determine whether we are in a remote or local session. If Remote Host is selected, all locations under the Target Application box will refer to paths on the remote machine. Otherwise, all locations will refer to paths on the local machine, as usual. |
| Remote Host text box | The IP address of the remote machine (the machine on which the remote agent is running). In the above screenshot, the remote host is 10.20.0.144.  Alternatively, the host name of the remote machine can be used instead of its IP address, provided the name is recognized by the network DNS. |
| Port text box | The port (in the remote machine) on which the remote agent is listening. Note that the value of Port should match the PortNumber element value in CodeXLRemoteAgentConfig.xml which is located on the remote machine. |

1. Click OK, and start debugging as usual.