CROSS-PLATFORM UTILITIES (CXUTILS) LIBRARY 2.0

USER DOCUMENTATION

1 CONTENTS

2 INTRODUCTION

2.1 CROSS-PLATFORM UTILITIES LIBRARY (CXUTILS)

CxUtils is a multi-platform C++ library containing many useful functions and classes for rapid development of applications. It contains tools for threads, network communication, joysticks, serial communication, shared memory, timers, and basic math operations (matrices, quaternion rotations, coordinate transformations). Using this library it should be a simple task to create a C++ application that can easily be ported between Windows, Linux, and other platforms.

2.2 NAMING CONVESTIONS AND LICENSE

The CxUtils Library uses naming conventions defined by the ACTIVE Laboratory. The CxUtils library is all within the CxUtils namespace. CxUtils is released under the BSD License.

2.3 CXUTILS LIBRARY DIRECTORY STRUCTURE

- include – Contains all header files for library
    - cxutils – Main folder for cxutils library header files
        - math – Subdirectory for math related header files
        - networking – Subdirectory for networking related header files
        - ipc – Shared/Mapped memory header files
        - images – Subdirectory for image related header files
            - jpg
            - png
- src
    - Identical structure layout as include folder
    - examples – Example programs.
- lib – Contains all compiled CxUtils library files produced
- bin – Contains all compiled CxUtils executable files produced
- docs – Documentation
- build - Contains folders for compilation on different platforms
    - cmake - Contains CMake project files
    - codelite – CodeLite project files and workspaces for Linux
    - linux – Standard Linux Makefile for compilation and installation
- ext – Contains any external libraries needed to build CxUtils

3 INSTALLATION

This section explains how to install the CxUtils library on different platforms.

3.1 LIBRARY DEPENDENCIES

In Windows using Visual Studio, you will not need to install any additional dependencies if you are using a full version (not express edition). For details on using Express Editions of VS, see the Visual Studio section below.

The following is the list of all external dependencies used by CxUtils in *NIX:

- libX11
- libXTst
- libpthread
- libpng
- libz
- libcxjpeg-5b – Modified build of libjpeg provided with CxUtils in the cxutils\version\ext\libjpeg-6b\linux folder and is compiled automatically by the Makefile or by the CodeLite workspace.

You can install these dependencies in Ubuntu with the following line in the terminal:

sudo apt-get install libpng-dev libX11-dev libxtst-dev

3.2 CMAKE

The preferred method for building and installing CxUtils is to use CMake, as it will allow you to support multiple platforms and IDE's not already provided with the download. In *NIX using CMake, from a command line navigate to the cxutils/version/build/cmake folder and type the following to build and install:

```
cmake .
make
sudo make install
```

Once installed you will need to update your ld.so.conf file to find the .so file at runtime. Add /usr/local/lib/active (or whatever install path you may have set) to your ld.so.conf file (in Ubuntu this is located at /etc/ld.so.conf). Make sure you use sudo to edit. When you finish this step, run:

```
sudo ldconfig
```

To update your system settings.

3.3 VISUAL STUDIO

3.3.1 BUILDING THE LIBRARY

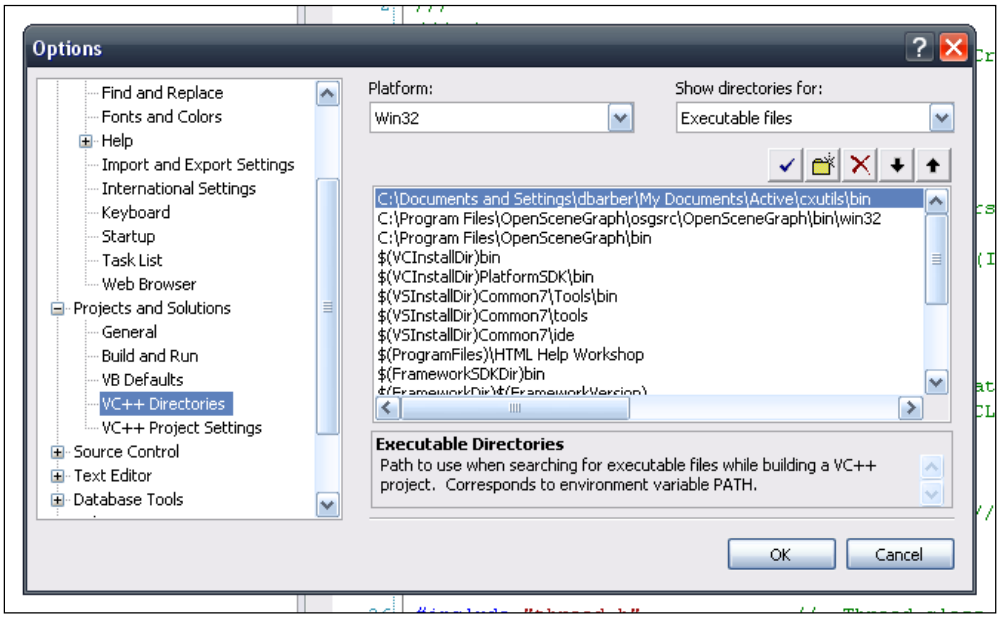
If the library is not already compiled, or you wish to rebuild from the source, you can use the cxutils.sln file located in cxutils/build/msvc9/. Open the solution and build the version of the library you want.

3.3.2 CONFIGURING VISUAL STUDIO 2005/2008 GLOBAL SETTINGS

From the main menu bar, select **Tools->Options…** which will bring up the **Visual Studio Options** Dialog. In the left section of the Dialog, select **Projects and Solutions**, **VC++ Directories**. In the following sections, substitute version with whatever version of CxUtils you are using (e.g. 1.0, 2.0).

With the "**Show directories for:**" drop down menu set to "**Executable files**" add the following directories:

<Install Path>\cxutils\version\bin



With the "**Show directories for:**" drop down menu set to "**Include files**" add the following directories:

<Install Path>\cxutils\version\include\

With the "**Show directories for:**" drop down menu set to "**Library files**" add the following directories:

<Install Path>\cxutils\version\lib\

Finally, you'll need to add the bin directory your Windows system PATH, or copy the DLL file to your Windows\system32\ folder. It is recommended you update your system PATH so that you do not need to keep making copies of the files if updates are made to the library.

Once all of this is done you are ready to use CxUtils within your own projects using VS2005/2008.

3.3.3 NOTE FOR VISUAL STUDIO EXPRESS EDITIONS

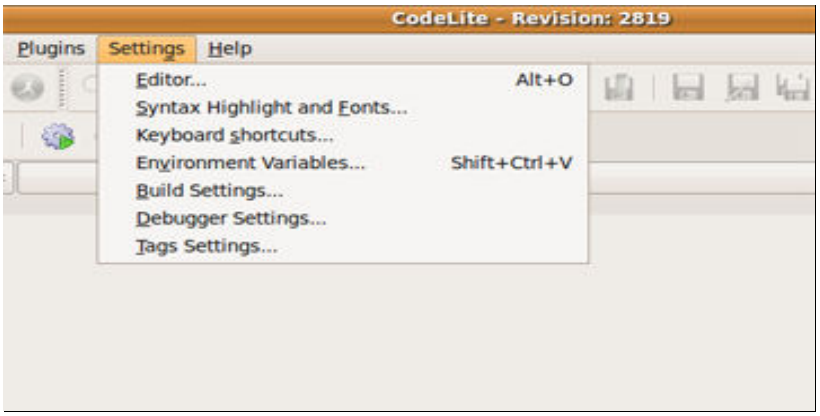
In order to build CxUtils with VS2005 Express you must have the Windows Platform SDK installed and setup. Make sure you following all additional instructions for installation of Visual Studio 2005 Express (http://www.microsoft.com/express/2005/download/default.aspx) including SP1, **2008 Express has not been evaluated with CxUtils at this time, but 2008 Standard works fine**. If you are still unable to build the CxUtils library with the Express Edition of Visual Studio 2005 because you get an error related to atlthunk.lib, then you'll need to do the following:

Open atlbase.h which is located in the Platform SDK install directory under Include\atl\ and comment out line 293 (#pragma comment(lib, "atlthunk.lib")). Once this is done you should have no issues with building the library and commenting out the line should not affect anything else on your system as it has yet to do anything for the author of this document. This is necessary because the Express edition of VS2005 does not include atlthunk.lib, so commenting out the line will change nothing on your system.

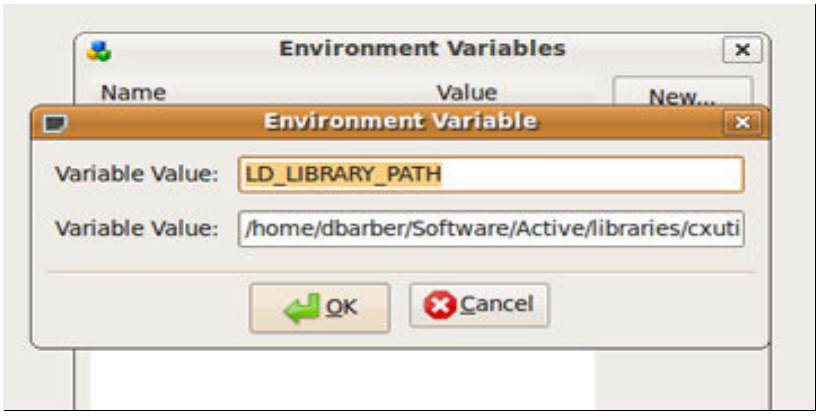
3.4 LINUX DISTRIBUTIONS

3.4.1 CODELITE

There is also support for the CodeLite IDE under Linux. The CodeLite project files are located in the build/codelite folder. These projects will build a dynamic library of CxUtils, and install the library files to your system at /usr/local/lib/active for the library files, and /usr/local/include/active for the header files. When this happens, you will be prompted to enter your password for installation. Once installed, you will be able to use CxUtils or run the example programs after you update either your /etc/ld.so.conf file to add the library install directory, or update your LD_LIBRARY_PATH in CodeLite as shown in the following figures by going to **Settings->Environment Variables…** and adding a new variable.



From there you'll have the option to add a variable. Press the **New…** button and add a variable called LD_LIBRARY_PATH and set the value to the <Your System Path>cxutils\<version>\lib folder as show in the following image.



Press the OK button and save your settings you shouldn't have any problems running any of the example programs from IDE.

However, the recommended method is to add the library path to your ld.so.conf file. First add the path to your /etc/ld.so.conf, and when complete type sudo ldconfig.

4 EXAMPLES

For additional information on what the Library offers or how the interfaces are defined, please refer to the HTML documentation provided in the docs directory of the CxUtils download, or from the online site (http://active-ist.sourceforge.net). There are plenty of example applications and source files provided with the library under <Install Path>/cxutils/version/src/examples and workspace files for Visual Studio 2005/2008 and CodeLite that will help any developer figure out what is available and how to do the most common tasks required of them.