Compiling Orx dependencies for new Visual Studio editions

This document is a cheat sheet for compiling up orx dependencies for new editions of Visual Studio in order to contribute back the compiled libraries into the release versions of orx.

This page is not for general use, it is low traffic, and prone to frequent changes and errors. If you are looking for compiled orx dependancies, you can download a precompiled orx, or clone from the the regular repo.

Clone https://github.com/orx/orx-extern

Build libwebp

1. Switch to 32 bit compiler with: C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\bin\amd64_x86\vcvarsamd64_x86.bat
2. In a Visual Studio console, go to the libwebp folder and: nmake /f Makefile.vc CFG=release-static RTLIBCFG=static OBJDIR=output/32
3. Ignore errors, locate the lib at: libwebp\output\release-static\x86\lib\webpdecoder.lib
4. Switch to 64 bit compiler mode with: C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\bin\x86_amd64\vcvarsx86_amd64.bat
5. nmake /f Makefile.vc CFG=release-static RTLIBCFG=static OBJDIR=output/64
6. Ignore errors, locate the lib at: libwebp\output\release-static\x64\lib\webpdecoder.lib

Build freetype

1. Install the latest version of CMake
2. Create a Visual Studio project with: cmake -DLIBTYPE=STATIC -G “Visual Studio 16 2022”
3. Open the freetype.sln solution with Visual Studio.
4. Switch to release, right click on the freetype project and select build to make the 32-bit static lib.
5. Find the 32 bit version in \Release\freetype.lib
6. For 64bit: Select the dropdown that says win32
7. Click and select Configuration Manager
8. In the Active Solution Platform dropdown, click <New>
9. Select x64 and copy from win32 (leave all options default)
10. In the freetype project properties
    1. In C/C++→Code Generation
        1. select Multithreaded (/MT)
    2. In Librarian / Command Line
        1. remove additional options relating to /machine:X86
11. In the common project properties
    1. In Librarian / Command Line
        1. remove additional options relating to /machine:X86
12. Right click the freetype project and select build to make the 64-bit static lib.

13. Find the 64bit version in x64\Release\freetype.lib

Build Liquidfun

1. Command window.
2. Go to folder to C:\Work\liquidfun\liquidfun\Box2D
3. cmake -DLIBTYPE=STATIC -G “Visual Studio 16 2022”
4. Open in Visual Studio
5. Go to Box2D project properties
6. There is only Win32 project. Set Code Gen / Runtime Library to Multi-threaded (/MT)
7. Add to disable warnings: 4456;4457;%(DisableSpecificWarnings)
8. Do for both Debug and Release
9. Copy Win32 to X64 with configuration editor
10. Go to Box2D project properties
11. In Librarian, remove Additional Options, remove 32bit Machine

Build GLFW

1. Copy build VC13 to VC14
2. Create a lib/msvs2014/32 and 64 folders
3. Open solution is Visual Studio
4. Allow it to convert the project
5. Convert the solution by selecting the solution in solution explorer, and save as over the top of the .sln
6. In both Win32 and x64 configurations, get properties on the Box2D project
7. In General / Output Directory - change to your new lib path and add a trailing slash.
8. Compile both Win32 and x64 configurations.

From:
https://www.orx-project.org/wiki/ - **Orx Learning**

Permanent link:
https://www.orx-project.org/wiki/compiling_visual_studio_deps?rev=1648414381

Last update: **2025/09/30 17:26 (14 months ago)**