Using Premake to create a Build Project to compile Orx

> This article is no longer recommended as Orx now provides a setup script in the root of the Orx repository (setup.sh) which will automatically create build projects for all IDEs for your operating system. For more information: Cloning Orx from Github
>
> If you wish to do this manually, continue with the article below.

This article is a short guide on building Orx from scratch using an IDE of your choice (Visual Studio, Codelite, XCode etc) on an operating system of your choice (Windows, Linux, Mac, Android etc).

What is a build project?

When you clone a copy of the very latest orx, look in the \orx\code\build folder. In here you will see folders to Windows, Linux, Android etc. And in these folders, you'll find subfolders to VisualStudio2010, Codelite, CodeBlocks, XCode, jni, etc.

These are build projects. With previous releases of orx, there were several build projects for some IDEs but not all. Premake now builds these Projects for us, for your chosen operating system and editor.

Your build may not exist in these folders, but you can create it. Then you can build orx for your own project.

Creating your build project

This article assumes that you are using the latest cloned Orx.

Step 1: cd your way into the build output folder: `cd orx\code\build`

Step 2: run premake and specify the editor your want to create a build project for (in this case, windows codelite): `..\..\extern\premake\bin\windows\premake4 codelite`

Your project for building Orx from scratch will now be created.

All operating systems, architectures, build projects

To get a list of all possible combinations, do a:

`extern\premake\bin\windows\premake4 –help` (dash dash help)

At the current time, the available oses are:

- OpenBSD, NetBSD, or FreeBSD
- Haiku
- Linux
- Apple Mac OS X
- Solaris
- Microsoft Windows

And the available build projects are:

- codeblocks
- codelite
- gmake
- vs2002
- vs2003
- vs2005
- vs2008
- vs2010
- vs2012
- vs2013
- xcode3
- xcode4

There's some interesting architectures there as well.

Compiling Orx using your Generated Project

Opening your IDE of choice and compiling should be fairly straight forward, but if you require any help, you can pick from one of the following guides:

- Building the Orx library on Windows with Visual Studio
- Building the Orx library on Windows with Codelite
- Compiling Orx with mingw32/gmake on Windows
- Building the Orx library on Linux with Codelite
- Building the Orx library on Linux with Code::Blocks
- Cloning and building the Orx library on the Mac
- Building the Orx library on a Mac with Codelite
- Building the Orx library on a Mac with Code::Blocks
- Building the Orx Library with Xcode on Mac OS X
- Compiling the Orx Library and Demo Project for Android

For troubleshooting help building Orx please see: Troubleshoot Building Orx



And you're done!

Troubleshooting

If you encounter any issues, visit the troubleshooting page for building Orx.

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

Permanent link:
https://www.orx-project.org/wiki/en/tutorials/orx/using-premake-to-create-build-projects

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