XCode

**Virtuos Games**

Jurgen Kluft

A maven like environment for C++ and C#

# About

XCode is a tool that provides the following features:

* Dependency management
* Release management
* Configuration management

To do the above it introduces a couple of new concepts:

* Package (or in C++ we might call this a Library or Module)
  + A package is everything needed by a library to build, test and publish itself.
* Root Package
  + A package with possible dependencies
  + Has the following files to enable itself to function within XCode
    - <file://pom.targets>
    - <file://pom.props>
    - <file://pom.xml>
* Dependency Package
  + A package used as a dependency by a root package
* Version
  + Every package has a version in the following format: Major.Minor.Fix, e.g.: 1.0.0
* Version Range
  + A dependency package is defined by specifying a single Version or a Version Range. An example of a Version Range is [1.0,), this means any version equal or higher than 1.0.0. And XCode will always obtain the highest version. If you want a defined range you can specify [1.0, 1.5] which means any version between and or equal to 1.0 to 1.5 is ok with a preference to the highest version that is available.
* Install and Deploy
  + Install: This will install a published package to the local package repository or cache.
  + Deploy: This will deploy a published package to the remote package repository and will make it publically available.

# Purpose 🡺 Convention

XCode is meant to enforce a ‘convention’ or standard where it is possible to deviate but not encouraged. The conventions that are enforced are:

* Directory Structure
* Configuration and Platform names
* Test (Unit-Test, Integration-Test) before Install/Deploy

Note: xproject is the example project used in this manual.

## Directory Layout

|  |  |  |
| --- | --- | --- |
| Package Structure of xproject (example) | | |
| Name | Size | Description |
| [\xproject\](file:///C:\Documents%20and%20Settings\jurgen\Local%20Settings\Temp\xproject\) |  |  |
| <docs> | <DIR> | Package related documentation |
| <source> | <DIR> | Package source code (C++, C#, ...) |
| <target> | <DIR> | Package build and dependencies (all temporary) |
| <tools> | <DIR> | Package utilities and tools |
| pom.props | 387 | MsBuild property file of this package |
| xproject.sln | 2 071 | Visual Studio solution file (generated) |
| pom.targets | 241 | MsBuild targets file of this package |
| pom.xml | 2 248 | The package description file |
| [\xproject\docs\](file:///C:\Documents%20and%20Settings\jurgen\Local%20Settings\Temp\xproject\docs\) |  |  |
| <manuals> | <DIR> |  |
| <references> | <DIR> |  |
| [\xproject\docs\manuals\](file:///C:\Documents%20and%20Settings\jurgen\Local%20Settings\Temp\xproject\docs\manuals\) |  |  |
| [\xproject\docs\references\](file:///C:\Documents%20and%20Settings\jurgen\Local%20Settings\Temp\xproject\docs\references\) |  |  |
| [\xproject\source\](file:///C:\Documents%20and%20Settings\jurgen\Local%20Settings\Temp\xproject\source\) |  |  |
| <main> | <DIR> | Main source of this package |
| <test> | <DIR> | UnitTest and/or IntegrationTest source code |
| [\xproject\source\main\](file:///C:\Documents%20and%20Settings\jurgen\Local%20Settings\Temp\xproject\source\main\) |  |  |
| <cpp> | <DIR> |  |
| <include> | <DIR> |  |
| <resources> | <DIR> |  |
| [\xproject\source\main\cpp\](file:///C:\Documents%20and%20Settings\jurgen\Local%20Settings\Temp\xproject\source\main\cpp\) |  |  |
| xproject.vcxproj | 7 575 | Visual Studio project file (generated) |
| [\xproject\source\main\include\xproject\](file:///C:\Documents%20and%20Settings\jurgen\Local%20Settings\Temp\xproject\source\main\include\) |  | Public header files |
| [\xproject\source\main\include\xproject\](file:///C:\Documents%20and%20Settings\jurgen\Local%20Settings\Temp\xproject\source\main\include\)private |  | Private header files |
| [\xproject\source\main\resources\](file:///C:\Documents%20and%20Settings\jurgen\Local%20Settings\Temp\xproject\source\main\resources\) |  | Images, Icons, Database files |
| [\xproject\source\test\](file:///C:\Documents%20and%20Settings\jurgen\Local%20Settings\Temp\xproject\source\test\) |  |  |
| <cpp> | <DIR> |  |
| <include> | <DIR> |  |
| <resources> | <DIR> |  |
| [\xproject\source\test\cpp\](file:///C:\Documents%20and%20Settings\jurgen\Local%20Settings\Temp\xproject\source\test\cpp\) |  |  |
| xproject\_test.vcxproj | 3 924 | Visual Studio project file (generated) |
| [\xproject\source\test\include\xproject\](file:///C:\Documents%20and%20Settings\jurgen\Local%20Settings\Temp\xproject\source\test\include\) |  |  |
| [\xproject\source\test\resources\](file:///C:\Documents%20and%20Settings\jurgen\Local%20Settings\Temp\xproject\source\test\resources\) |  |  |
| [\xproject\target\](file:///C:\Documents%20and%20Settings\jurgen\Local%20Settings\Temp\xproject\target\) |  |  |
| [\xproject\tools\](file:///C:\Documents%20and%20Settings\jurgen\Local%20Settings\Temp\xproject\tools\) |  |  |
|  |  |  |

## Standard: MsDev 2010 C++ Project Configurations

* DevDebug, DevRelease, DevFinal
  + Used for builds during development
* ClientDebug, ClientRelease, ClientFinal
  + Used for sending builds to the client
* RetailDebug, RetailRelease, RetailFinal
  + Used for sending builds to publisher
* ToolDebug, ToolRelease, ToolFinal
  + Used for Tools/Editor builds
* ProfileDebug, ProfileRelease, ProfileFinal
  + Used for profile builds
* Doxygen?
  + Used for documentation builds

## Standard: MsDev 2010 C++ Project Platforms

* WII
* NDS
* N3DS
* PS3
* PS3\_SPU
* Xbox 360
* Win32

## Pom.xml: Special markers; Include and Exclude

‘#’ is used as a marker before items to indicate that they are valid when the package is used as a dependency package and when generating project files for this package as the root package then these items will not be included.

‘@’ is used as a marker before items to indicate that they are valid when the package is used as the root package and when generating project files for this package as the root package then these items will be included.

Unmarked items will always be included.

The above markers are used for:

* PreprocessorDefinitions (C++)
* AdditionalIncludeDirectories (C++)
* AdditionalLibraryDirectories (C++)
* AdditionalDependencies (C++)
* DefineConstants (C#)

## Pom.xml: MsDev Project configuration (under construction)

Rule: You **need** to specify group items containing conditions for the platforms and configurations that you support in the package.

## Rule: Include Directory (under construction)

Include directory should follow the rule of being located at ‘source\main\include\xproject’, there should be a minimal amount of exceptions to this rule. To use a header file in your .cpp file is to write the following ‘#include “xproject\header.h”. Make sure that this include directory is specified for all platforms and configurations in the pom.xml file.

**Reason:** This rule is here to solve conflicts between header files with the same name.

### Working on the ‘default’ branch (under construction)

Normally you would be working on this branch and would build packages for the different platforms from this branch. Versioning is done automatically for the Year.Month.Day.Minute part of the version, the Fix part of the version needs to be manually increased before doing a Deploy. The Major and Minor have to increased manually as well when necessary.

### Working on a feature branch (under construction)

Sometimes you may want to implement a new feature for only one or so platforms; in this case you can create a new branch. When you create a new feature branch you need to create a tag with an increased Minor version, keep in mind that the other branches when building a new Fix will now have the same Minor version as this feature branch. The reason for this is that the versioning and features should never be able to divert. In the revision history you have to mention that the now new Major.Minor version has a new feature for the platforms that you have implemented them, for the other platforms have you to specify that they are ‘To be implemented’.

### Working on an experimental branch (under construction)

If you want to do some experiments the proposed procedure is to create a branch but skip the creation of the tag and not increase the Minor version. When doing this you may run XCode until the Install stage but should never do a Deploy.

# Usage

* Init (done in the root of the development directory):  
  msbuild dev.targets /t:Init /p:PackageName=NameOfYourProject /p:Language=C++
* Init Directories (in the package directory, after editing the pom.xml <DirectoryStructure> element):  
  msbuild pom.targets /t:InitDirs
* Construct Microsoft Visual Studio Projects and Solution (after editing the pom.xml <Project> elements):  
  msbuild pom.targets /t:Construct
* Clean:  
  msbuild pom.targets /t:Clean /p:PackagePlatform=YourPlatform(Win32)
* Compile:  
  msbuild pom.targets /t:Compile /p:PackagePlatform=YourPlatform(Win32)
* Package:  
  msbuild pom.targets /t:Package /p:PackagePlatform=YourPlatform(Win32)
* Install:  
  msbuild pom.targets /t:Install /p:PackagePlatform=YourPlatform(Win32)
* Deploy:  
  msbuild pom.targets /t:Deploy /p:PackagePlatform=YourPlatform(Win32)