



GATS Companion to Building a Visual C++ Library

Author: Garth Santor

Editors: -

Copyright Dates: 2016, 2018, 2019

Version: 1.0.0 (2019-01-14)

Overview

Precompiled libraries are essential to efficiently distributing and building code. This document provides a sequence of steps to extract code from an existing project and place it in a library anchored to the same solution. It can easily be adapted to a stand-alone library solution.

Conversion steps

1. Create the folders '**applib-dist/include**, **applib-dist/lib-x64**, **applib-dist/lib-x86**' within the solution folder.
2. Move the library's .cpp files to the library project folder.
3. Move the library's .hpp files to '**applib-dist/include**' folder.
4. Add the library's .cpp/.hpp files to the *applib* project. Remove them from their original project.
5. Fix the include path in all projects to **\$(SolutionDir)applib-dist/include**
6. Fix the output directory in the library project to **\$(SolutionDir)applib-dist/lib-x64**, and **\$(SolutionDir)applib-dist/lib-x86** (note that the setting **does** have a trailing slash. An absent trailing slash will break the configuration).
7. Set the target names to: **applib-mt-gd** for the *debug* configuration, and **applib-mt** for the *release* configuration.
8. Set library search paths in application project to **\$(SolutionDir)applib-dist/lib-x86** and **\$(SolutionDir)applib-dist/lib-x64**.
9. Set project dependencies (*MyApp* depends on *applib*).
10. Make static versions of the libraries:
-> Configuration Manager -> Active solution configuration:
-> new -> Debug Static (from Debug), Release Static (from Release)
11. Set Static compile, etc.
 - a. Multithread Debug + **applib-mt-sgd**
 - b. Multithread Release + **applib-mt-s**
12. Test:
 - a. Batch build should build 16 projects. Four 32-bit executables, four 64-bit executables, and 8 libraries.