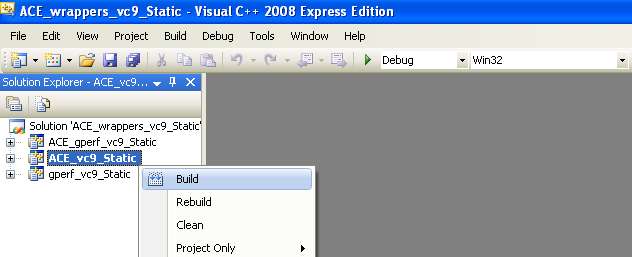
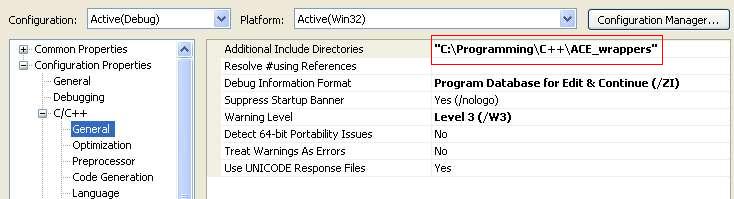
**Building ACE on Windows using Visual Studio**

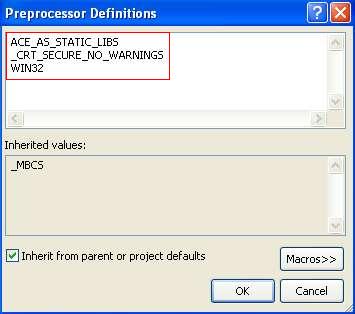
Posted on [November 16, 2009](http://yuval.bar-or.org/blog/2009/11/building-ace-on-windows-using-visual-studio/) by [yuval](http://yuval.bar-or.org/blog/author/yuval/)

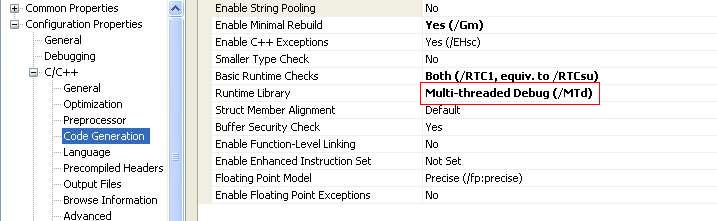
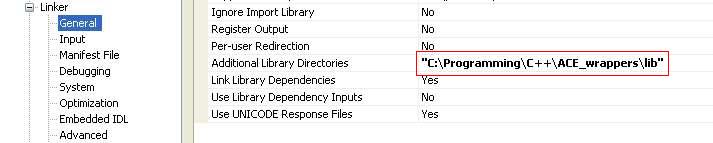
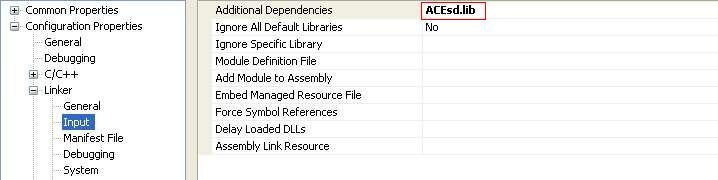
The [Adaptive Communication Environment](http://www.cs.wustl.edu/~schmidt/ACE-overview.html) (ACE)  framework is an extensive set of cross-platform tools in C++. In this post I will describe in detail the steps required to set up the ACE framework on Windows using Visual Studio, and to set up a project using the ACE framework. The process is not complicated, but there are several pitfalls that one should be aware of.

Here we go!

Note: Steps 1-6 are partially covered in the [ACE Build Guide](http://www.dre.vanderbilt.edu/~schmidt/DOC_ROOT/ACE/ACE-INSTALL.html#msvc). I give them here with some extra hints and tips.

1. Download the ACE framework from [here](http://download.dre.vanderbilt.edu/). We will be using the full version of ACE.zip (last line under “Latest Micro Release Kit”).
2. Unzip this file into the directory where you want your ACE files to be located. I put my ACE files under C:\Programming\C++. Unzipping the file creates an ACE\_Wrappers directory (C:\Programming\C++\ACE\_Wrappers in my case). We’ll call this directory the ACE\_ROOT directory.
3. Go to the ACE\_ROOT\ace directory (C:\Programming\C++\ACE\_Wrappers\ace in my case) and create a file called “config.h”. Edit this file with notepad or any other text editor and put the following text in the file:  
   #include “ace/config-win32.h”
4. In the ACE\_ROOT directory there are several solutions for different versions of Visual Studio and for different types of libraries. In this tutorial I will be using Visual C++ 2008 Express Edition (which is also called vc9) and building ACE as a static library. So the solution I  need is ‘ACE\_wrappers\_vc\_9\_Static.sln’. Select the solution that fits your needs.  
   \* Note: Visual Studio 2010 users can use the vc9 solutions. The Conversion Wizard will upgrade the solution to VS 2010 format.
5. Select your configuration (Debug or Release) and build the ACE project in your solution. In my case this is ‘ACE\_vc9\_Static’.  
   [](http://yuval.bar-or.org/blog/wp-content/uploads/2009/11/BuildACE.jpg)
6. After a few minutes of building, you should have the ACE library file in your ACE\_ROOT\lib directory. I built using the Debug configuration, so the file that was created is c:\Programming\C++\ACE\_Wrappers\lib\ACEsd.lib
7. We are now ready to create our first ACE project! Create an empty C++ solution and then edit the project properties.
8. In “Properties”–”C/C++”–”General”–”Additional Include Directories” add the ACE\_ROOT directory.  
   [](http://yuval.bar-or.org/blog/wp-content/uploads/2009/11/AdditionalIncludeDirs.jpg)
9. In “Properties”–”C/C++”–”Preprocessor”–”Preprocessor Definitions” set the following:  
   ACE\_AS\_STATIC\_LIBS  
   \_CRT\_SECURE\_NO\_WARNINGS  
   WIN32

[[](http://yuval.bar-or.org/blog/wp-content/uploads/2009/11/PreprocessorDefs.jpg)](http://yuval.bar-or.org/blog/wp-content/uploads/2009/11/PreprocessorDefs.jpg)

1. In “Properties”–”C/C++”–”Code Generation”–”Runtime Library” set to Multi-Threaded debug (/MTd) for the Debug configuration and Multi-Threaded (/MT) for the Release configuration.  
   [](http://yuval.bar-or.org/blog/wp-content/uploads/2009/11/CodeGeneration.jpg)
2. In “Properties”–”Linker”–”General”–”Additional Library Directories” add the ACE\_ROOT\lib directory.  
   [](http://yuval.bar-or.org/blog/wp-content/uploads/2009/11/AdditionalLibDirs.jpg)
3. In “Properties”–”Linker”–”Input”–”Additional Dependencies” add ACEsd.lib for the Debug configuration and ACEs.lib for the Release configuration.  
   [](http://yuval.bar-or.org/blog/wp-content/uploads/2009/11/LinkerInput.jpg)
4. Your project is now configured to work with ACE. You can create your project code and start using the ACE framework. Following is a simple example program that runs a thread to perform some work.