

Unimatrix Seven

Tuesday, 27 March 2012

Compiling Eclipse Paho MQTT C client in Visual Studio C++

The Paho code comes with a Makefile which can be used on most OSes, but for Windows, it needs a lot of dependencies: Cygwin with make, .Net Framework, Visual Studio compiler, etc ...

After some frustrating time trying to get all this working, copying headers and libraries all over the place, and trying to find all the headers and libraries it needed, I decided to change tactic ...

Since by now I had installed Visual Studio and the .Net framework, why not just use those ? Visual Studio will give me an environment where all the libraries and headers are in their respective paths, and *should* make it easier to build.

The following was done using Visual Studio C++ 2010, .Net Framework 3.5 and 4 installed.

Get the Paho code

First we start by installing Git for Windows from the msysgit project <http://code.google.com/p/msysgit/>.

I followed the installation instructions from this blog post: <http://nathanj.github.com/gitguide/tour.html>.

Once installed, you can either open the **Git Bash** from the Programs list, or navigate to a folder where you want to put the Paho code, and then right click and select '**Git clone ..**'.

From Andy Piper's blog <http://andypiper.co.uk/2012/03/10/paho-gets-started/>, use the git command mentioned if using the Git Bash, or just the git URL if using the Git UI.

We have the code now, in the org.eclipse.paho.mqtt.c directory.
Let's start Visual Studio C++.

Create a project in Visual Studio

In Visual Studio, you want to create a '**New Project**' of type '**Win32 Console Application**'.

In the **New Project Wizard**, click Next to get to the **Application Settings** panel, choose '**DLL**' and tick '**Empty Project**' then click Finish.

Adding the source files

Open the project in the Solution Explorer, then right click **Header Files**, and select from the menu: '**Add**' -> '**Existing Item..**', navigate to the directory where Paho MQTT C was copied and go into the 'src' folder.

Type *.h in the Filename and press Enter, then select all the header files and click '**Add**'.

Do the same for the **Source Files** folder, selecting *.c files only.

Configuring the project properties

There are a few tweaks to do in the project's properties, before it can build cleanly.

First, you need to go enable the expert settings, in '**Tools**' -> '**Settings**' -> '**Expert Settings**'. Then right-click the project and select '**Properties**'.

Now you need to check and change a few things:

* Check the library name:

'**Configuration Properties**' -> '**General**' -> '**Target Name**', the default is `mqttv3c`

* Built as C code:

'**Configuration Properties**' -> '**C/C++**' -> '**Advanced**' -> '**Compile As**' should be set to '**Compile as C Code (/TC)**'.

* Add ws2_32.lib (because the code uses winsock2.h/socket library) to the library path:

'**Configuration Properties**' -> '**Linker**' -> '**Additional Dependencies**', add '**ws2_32.lib**;' in the list,

Blog Archive

► 2013 (2)

▼ 2012 (4)

► April (2)

▼ March (2)

[Compiling Eclipse Paho MQTT C client in Visual Stu...](#)

[Resistance is futile ...](#)

About Me

Flo

[View my complete profile](#)

between the last library odbccp32.lib and \${AdditionalDependencies}.

Save everything so far, click Apply.

Finally, if you want a Release build by default, click the **Configuration Manager's** button at the top right of the Properties window and select 'Release' instead of 'Debug' in the Configuration column.

Click OK out of the Configuration Manager's and the Properties' windows.

We're ready to Build.

Right click the project and select 'Build'.

You should see the build output passing by, and finally you can see the .dll has been created:

```
Finished generating code  
paho-mqtt.vcxproj -> C:\<path-to-your-project>\Release\mqttv3c.dll
```

All done! you now have mqttv3c.lib and mqttv3c.dll files which you can use in your MQTT project.

Posted by [Flo](#) at 02:47

[Recommend this on Google](#)

No comments:

Post a Comment

Comment as:

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)