

* gSOAP framework (gsoap\_2.8.23)
* Microsoft Visual Studio 2010 SP1
* Dos CMD command prompt

1. Coss compile gsoap-2.8.23 theo tài liệu firewall-build.sh

wsdl2h (schema importer)

wsdl2h <path and name of .wsdl file> -o <name of generated .h file e.g chatpp.h>

file.wsdl là file mô tả về các function/kết nối mà một web service nào đó đang chạy. Chương trình wsdl2h sẽ phân tích file.wsdl này để tạo ra file.h từ đó sẽ dùng để viết một chương trình client cho web service đã có đó.

File.h là file C/C++ define các function để kết nối giữa client và server.

soapcpp2 (stub/skeleton generator)

soapcpp2 -c <name of generated .h file e.g chatpp.h> cho chương trình C

soapcpp2 -i <name of generated .h file e.g chatpp.h> cho chương trình C++

Chương trình tạo ra các file.c và file.cpp implement các function đã define trong file.h

Trong gsoap-2.8-host/gsoap/bin/win32 đã có sẵn hai chương trình này cho Windows OS.

1. Xem ví dụ gsoap-2.8/gsoap/samples/calc

Calc.h

File này định nghĩa các function và các cài đặt về gsoap ns.

//gsoap ns service name: calc Simple calculator service

//gsoap ns service protocol: SOAP

//gsoap ns service style: rpc

//gsoap ns service encoding: encoded

//gsoap ns service namespace: http://10.49.46.201:2012/calc.wsdl

//gsoap ns service location: http://10.49.46.201:2012

//gsoap ns schema namespace: urn:calc

Với <http://10.49.46.201:2012> 🡺 máy Server có địa chỉ IP: 10.49.46.201, serrvice chạy qua port 2012.

DEPEND\_LIB\_DIR=/path/to/install

export CFLAGS="-I${DEPEND\_LIB\_DIR}/include -DWITH\_OPENSSL"

export CPPFLAGS=-I${DEPEND\_LIB\_DIR}/include

export LDFLAGS=-L${DEPEND\_LIB\_DIR}/lib

export PKG\_CONFIG\_PATH=${DEPEND\_LIB\_DIR}/lib/pkgconfig

export LD\_LIBRARY\_PATH=${DEPEND\_LIB\_DIR}/lib

export PATH=$PATH:${DEPEND\_LIB\_DIR}/bin:${DEPEND\_LIB\_DIR}/sbin

export LIBS="-lm -lssl -lcrypto -lpthread"

Compilation in C (see samples/calc):

$ soapcpp2 -c calc.h

$ gcc -o calcclient calcclient.c stdsoap2.c soapC.c soapClient.c ${CFLAGS} ${LDFLAGS} ${LIBS}

$ gcc -o calcserver calcserver.c stdsoap2.c soapC.c soapServer.c ${CFLAGS} ${LDFLAGS} ${LIBS}

Compilation in C++ (see samples/calc++):

$ soapcpp2 -i calc.h

$ g++ -o calcclient++ calcclient.cpp stdsoap2.cpp soapC.cpp soapcalcProxy.cpp ${CFLAGS} ${LDFLAGS} ${LIBS}

$ g++ -o calcserver++ calcserver.cpp stdsoap2.cpp soapC.cpp soapcalcService.cpp ${CFLAGS} ${LDFLAGS} ${LIBS}

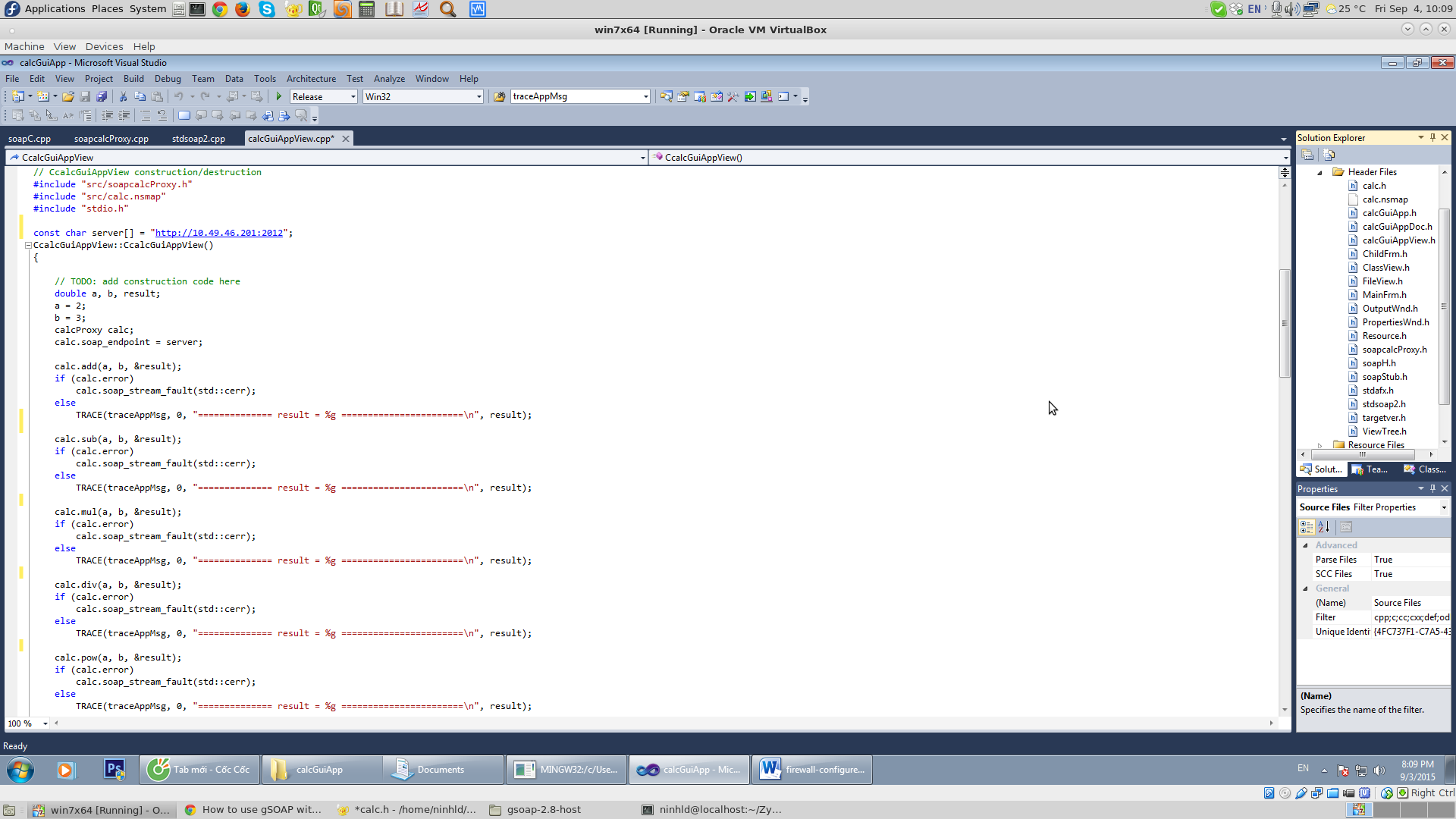
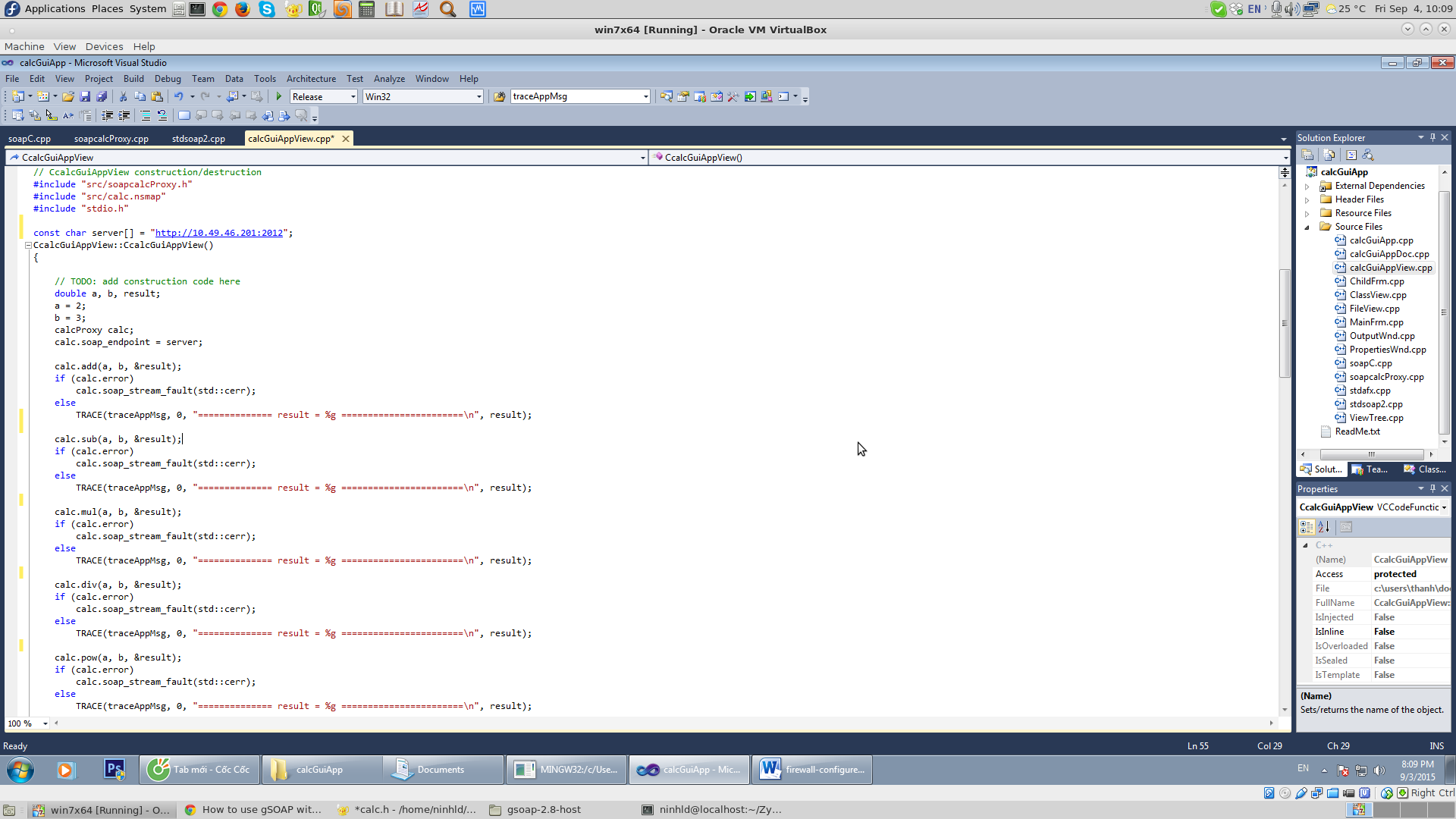
Chú ý: stdsoap2.c và stdsoap2.cpp được lấy từ gsoap-2.8/gsoap

1. Viết chương trình điều khiển trên Windows

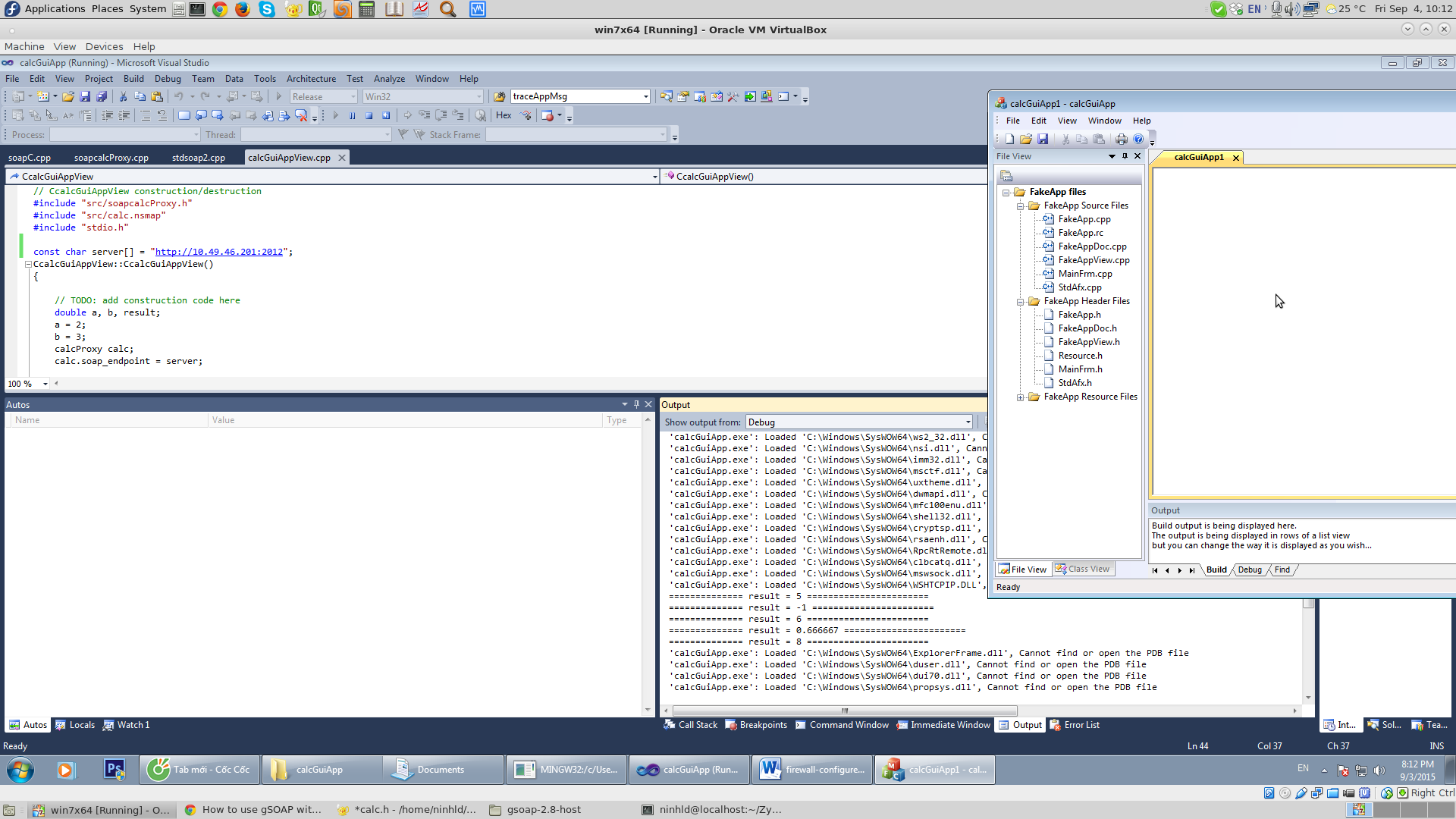
* Chương trình client có giao diện đồ họa
* Sử dụng Microsoft Visual Studio 2010 Utimate SP1
* Sử dụng DOS CMD command prompt để chạy **soapcpp2.exe**

Xem chương trình ví dụ calcGuiApp:

* Tạo MFC Application default trong VS2010 có tên calcGuiApp
* Import các file **soapC.cpp, soapcalcProxy.cpp, stdsoap2.cpp** vào Source Files. Thêm khai báo header **#include “stdafx.h”** vào đầu các file cpp trên
* Import các file **calc.h, calc.nsmap, soapH.h, soapcalcProxy.h, soapStub.h, stdsoap2.h** vào Header Files
* Mở file calcGuiAppView.cpp và paste vào đoạn code như hình dưới



Chạy chương trình server dưới board linux sau đó chạy chương trình client trên windows.



* gSOAP with OpenSSL
* <http://www.activestate.com/activeperl/downloads> download and install ActivePerl
* <http://www.openssl.org/source/> download OpenSSL source code, e.g **openssl-1.0.0s**
* Open the Visual Studio Command Prompt (2010)

Start > Microsoft Visual Studio 2010 > Visual Studio Tools

* cd to your OpenSSL source folder for **32-bit**

(e.g. cd C:\Users\ninhld\Documents\**openssl-1.0.0s**).

* Run the following:

*perl Configure* ***VC-WIN32*** *--prefix=C:\Build-OpenSSL-VC-32*

*ms\do\_ms*

*nmake -f ms\nt.mak*

*nmake -f ms\nt.mak install*

* Your outputs will be in C:\Build-OpenSSL-VC-32
* Setup on VC++

Right click on the project and select Properties from the right click menu. Under Configuration Properties, C/C++ and General, add include paths to gSOAP and Open SSL. Click Apply.

Under Configuration Properties, C/C++ and Preprocessor, add “WITH\_OPENSSL” (to enable SSL in gSOAP) and “DEBUG” (to enable logging) to the list of Preprocessor Definitions. Click Apply.

Under Configuration Properties, Linker and General, add the path to the Open SSL lib directory, where the compiled lib files reside, to Additional Library Directories. Click Apply.

Under Configuration Properties, Linker and Input, add “libeay32.lib” and “ssleay32.lib” to Additional Dependencies. These

are the Open SSL libraries. Click Apply.