Building Guide v. 0.1.1

http://timestampclient.sourceforge.net

This guide is intended to provide software developers with sufficient information to build TimeStampClient from sources under Microsoft Windows environment with Borland Turbo C++ Explorer IDE.

To build TimeStampClient you need following 3rd party software:

- Borland Turbo C++ Explorer
- OpenSSL 0.9.8c with OpenTSA patch
- Regex Boost C++ Library
- Tortoise SVN

To build OpenSSL 0.9.8c with OpenTSA patch you need following 3rd party software:

- Patch for Windows
- Tar for Windows
- Gzip for Windows
- ActiveState Perl
- The Netwide Assembler

Step 1. - Install Borland Turbo C++ Explorer

- Download and install "*Borland Turbo C++ Explorer*" and all of its prerequisites from http://www.turboexplorer.com/cpp
  (Please note that you will have to register at Borland Developer Network to receive free activation key file.)

Step 2. - Build OpenSSL 0.9.8c with OpenTSA patch

- Download zip archive with binaries of *"Patch for Windows"* from http://gnuwin32.sourceforge.net/packages/patch.htm

- Download zip archives with binaries of *"Tar for Windows"* and its dependencies from http://gnuwin32.sourceforge.net/packages/gtar.htm

- Download zip archive with binaries of *"Gzip for Windows"* from http://gnuwin32.sourceforge.net/packages/gzip.htm

- Download and install *"ActiveState Perl"* from http://www.activestate.com/

- Download *"OpenSSL 0.9.8c"* sources from http://openssl.org/

- Download *"Time Stamping Patch"* from http://opentsa.org/ts/ts-20060923-0_9_8c-patch.gz

- Download zip archive with binaries of *"The Netwide Assembler"* from http://sourceforge.net/projects/nasm

- Create building directory i.e. *“C:\openssl-ts-0.9.8c-build”*
- Extract *“Patch for Windows”*, *“Tar for Windows”*, *“Gzip for Windows”* and *“The Netwide Assembler”* into the building directory
- Copy *“openssl-0.9.8c.tar.gz”* and *“ts-20060923-0_9_8c-patch.gz”* files into the building directory
- Run following commands in the building directory to extract files from OpenSSL and OpenTSA archives:

```
C:\openssl-ts-0.9.8c-build> bin\gzip -d openssl-0.9.8c.tar.gz
C:\openssl-ts-0.9.8c-build> bin\gzip -d ts-20060923-0_9_8c-patch.gz
C:\openssl-ts-0.9.8c-build> bin\tar -xf openssl-0.9.8c.tar
```

- Copy extracted OpenTSA patch file *“ts-20060923-0_9_8c-patch”* into *“openssl-0.9.8c”* directory
- Apply OpenTSA patch on the OpenSSL sources with following command run from within *“openssl-0.9.8c”* directory:

```
C:\openssl-ts-0.9.8c-build\openssl-0.9.8c> type ts-20060923-0_9_8c-
patch | ..\bin\patch -p1
```

- Prepare OpenSSL sources for building with following commands run from within *“openssl-0.9.8c”* directory:

```
C:\openssl-ts-0.9.8c-build\openssl-0.9.8c> perl Configure BC-32
C:\openssl-ts-0.9.8c-build\openssl-0.9.8c> ms\do_nasm.bat
```

- Modify generated makefile *“C:\openssl-ts-0.9.8c-buil\openssl-0.9.8c\ms\bcb.mak”* by following instructions:
    - Change line
      `ASM=nasmw -f obj -d__omf__`
      to
      `ASM=C:\openssl-ts-0.9.8c-build\nasm-2.03.01\nasm.exe -f obj -d__omf__`
      (Please check the if path to nasm binary is correct.)
    - *Change line*
      `APP_EX_OBJ=c0x32.obj`
      to
      `APP_EX_OBJ=-L"C:\Program Files\Borland\BDS\4.0\lib" c0x32.obj`
      (Please check the if path to BDS libraries is correct.)
- Build OpenSSL with following command run from within *“openssl-0.9.8c”* directory:

```
C:\openssl-ts-0.9.8c-build\openssl-0.9.8c> make -f ms/bcb.bak
```

- Create directory for binary distribution of OpenSSL i.e. *“C:\openssl-ts-0.9.8c”* with subdirectories called *“bin”*, *“lib”* and *“include\openssl”*.
- Copy OpenSSL demonstration application to the distribution directory with the following commands:

```
C:\> copy C:\openssl-ts-0.9.8c-
build\openssl-0.9.8c\out32\openssl.exe  C:\openssl-ts-0.9.8c\bin\
C:\> copy C:\openssl-ts-0.9.8c-
build\openssl-0.9.8c\apps\openssl.cnf C:\openssl-ts-0.9.8c\bin\
```

- Copy OpenSSL libraries to the distribution directory with the following commands:

```
C:\> copy C:\openssl-ts-0.9.8c-
build\openssl-0.9.8c\out32\libeay32.lib C:\openssl-ts-0.9.8c\lib\
C:\> copy C:\openssl-ts-0.9.8c-
build\openssl-0.9.8c\out32\ssleay32.lib C:\openssl-ts-0.9.8c\lib\
```

- Copy C header files to the distribution directory with the following commands:

```
C:\> copy C:\openssl-ts-0.9.8c-
build\openssl-0.9.8c\inc32\openssl\*.*  C:\openssl-
ts-0.9.8c\include\openssl\
C:\> copy C:\openssl-ts-0.9.8c-build\openssl-0.9.8c\ms\applink.c
C:\openssl-ts-0.9.8c\include\openssl\
```

- You can delete building directory *“C:\openssl-ts-0.9.8c-build”* and all of its content

Step 3. - Build Regex Boost C++ Library

- Download *“Boost C++ Libraries”* from
  http://www.boost.org/

- Extract downloaded archive somewhere on the harddrive i.e. *“C:\boost_1_35_0”*

- Build regex library from within *“C:\boost_1_35_0\libs\regex\build”* directory with following commands:

```
C:\boost_1_35_0\libs\regex\build> make -f bcb6.mak
C:\boost_1_35_0\libs\regex\build> make -f bcb6.mak install
C:\boost_1_35_0\libs\regex\build> make -f bcb6.mak clean
```

- Do not delete directory *“C:\boost_1_35_0”* because it contains header files needed to build TimeStampClient

Step 4. - Download TimeStampClient sources from SVN repository

- Download and install *“Tortoise SVN”* client software from
  http://tortoisesvn.tigris.org/

- Create directory where you want to keep TimeStampClient sources i.e.
  “C:\svn\timestampclient”

- Righ-click the directory in windows explorer and from the context menu choose
  “SVN Checkout...” option

- Enter *“https://timestampclient.svn.sourceforge.net/svnroot/timestampclient/current”* as *“URL of repository”* and press the *“OK”* button

Step 5. - Building TimeStampClient binary

- Run *“Turbo C++”* IDE and open *“TimeStampClient.bdsproj”* file from directory with TimeStampClient sources

- Choose *“Project > Options..”* from the main menu of Turbo C++

- Under *“C++ Compiler (bcc32) > Paths and Defines”* set additional
  “Include search path (-I)” to *“c:\openssl-ts-0.9.8c\include”* and *“C:\boost_1_35_0”*

- Under *“Linker (ilink32) > Paths and Defines”* set additional *“Library search path (-L)”* to *“c:\openssl-ts-0.9.8c\lib”*
- Modify file *“c:\openssl-ts-0.9.8c\include\openssl\applink.c”* by following instructions:
    - Change line
      `{ return _setmode (_fileno(fp),mod=='b'?_O_BINARY:_O_TEXT); }`
      to
      `{ return setmode (_fileno(fp),mod=='b'?_O_BINARY:_O_TEXT); }`
    - *Change line*
      `OPENSSL_ApplinkTable[APPLINK_LSEEK] = _lseek;`
      to
      `OPENSSL_ApplinkTable[APPLINK_LSEEK] = lseek;`
- Build TimeStampClient by pressing the *“F9”* button
