Installing the C++ Library

1. Prerequisites

The library requires OpenSSL for cryptographic support. Xalan-C is also required if XPath and/or XSLT transformations are required.

Version 1.00 of the library has been tested with version 2.2 and 2.3 of Xerces-C, version 1.6 of Xalan-C and Version 0.9.7 (and above) of OpenSSL.

2. Getting the source

You can download the sources via WWW in the distribution directory from one of the Apache mirrors .

This project's CVS repository can be checked out through anonymous (pserver) CVS with the following instruction set. The module you wish to check out must be specified as the modulename. When prompted for a password for anonymous, simply enter "anoncvs" without quotes:

```
cvs -d :pserver:anoncvs@cvs.apache.org:/home/cvspublic login
       password: anoncvs
       cvs -d :pserver:anoncvs@cvs.apache.org:/home/cvspublic checkout
       xml-security
```

A HTTP interface to browse the sources online is available via http://cvs.apache.org/viewcvs.cgi/xml-security/

3. Building for UNIX

XML-Security-C is currently fully supported on Linux, FreeBSD and Solaris. It is partially supported (in cases where Xalan is not required) on NetBSD and Cygwin. It has been built and tested using GNU gcc 3.2, gcc 2.95.4, Forte C++ 5.4 (Solaris) and GNU make.

Note:

The UNIX XML-Security-C build process has changed radically since beta 0.0.2 as the Makefiles and configure scripts have been completely rewritten.

3.1. Set up the Environment

The build process has been automated as much as possible. To start the process, three environment variables *can* be set :

- *XERCESCROOT* - points to the base of your Xerces distribution
- *XALANCROOT* - points to the base of your Xalan distribution
- *OPENSSL* - points to the base of your OpenSSL distribution

For example (on my Debian Linux box under Bash) :

```
export XERCESCROOT=~/prog/extlibs/xerces-c-src2_3_0
export XALANCROOT=~/prog/extlibs/xalan-c-src1_6/c
export OPENSSL=~/prog/extlibs/openssl-0.9.7a
```

If these environment variables are not set, configure will try to find the necessary include and library files in the system directories. The configure script is created through *autoconf* so you can also tell your compiler where to find these things via the *CXXFLAGS* and *LDFLAGS* environment variables.

If configure cannot find anything for Xalan, it will assume that you are not interested in XPath or XSLT support and will compile XSEC without linking to Xalan. Any attempt to use these features will raise an exception in the library.

3.2. Configure

Now go to the $XSECCROOT/src directory and run the command *./configure*. This will create the necessary makefiles and header files necessary to build the package.

In addition to the standard options, *configure* can be passed a number of XSEC specific options :

- *--without-xalan* disable linkage to Xalan.
- *--enable-debug* cause the library to be built with symbols

> **Note:**
> Using the *--without-xalan* option will automatically mean that the library does not support XPath or XSLT transformations (although envelope transforms will work as the library can now perform these without going through an XPath transform).

3.3. Compile

Assuming the output of the above command looks reasonable simply type *make* (or *gmake* - you must use the GNU make utility) in the *src* directory. This will make the shared library. In

addition, *make tools* will make the tools (or examples) in the src/tools directory.

The make process will create three directories in the distribution directory:

1. *include* - All public header files are copied here
2. *bin* - Where the tools are placed once compiles
3. *lib* - Where the shared library is place

You will need to set up your *LD_LIBRARY_PATH* environment variable to ensure ld.so will find the new shared libraries.

Finally - you can use *make clean* and *make distclean* to remove all binaries and libraries (former) and build scripts (latter)

3.4. Install

make install can be used to install the library and the include files into the relevant directories (which can be set via the *configure* script using the various *--prefix=* options.

4. Building for Windows

XML-Security-C has been built and tested on Microsoft's Visual C++ 6.0 compiler only. (VC++ .NET support is currently being worked on.) The following subsections briefly describe how to rebuild the library, tools and samples using the supplied workspaces.

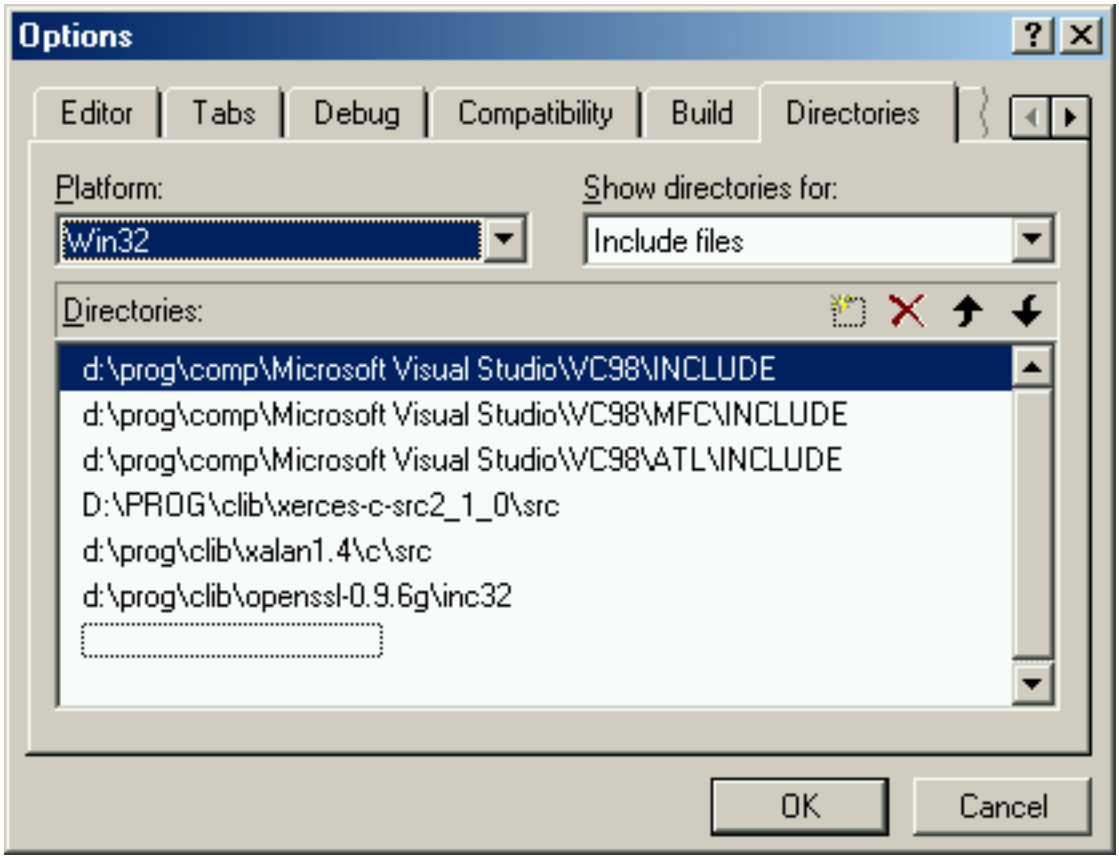
Note:

As of version 0.2, the library can be built without OpenSSL on a Windows platform. (The WinCAPI provider will be used instead). See below for details on how to do this. This is still experimental, but should work.

4.1. Setup Directories

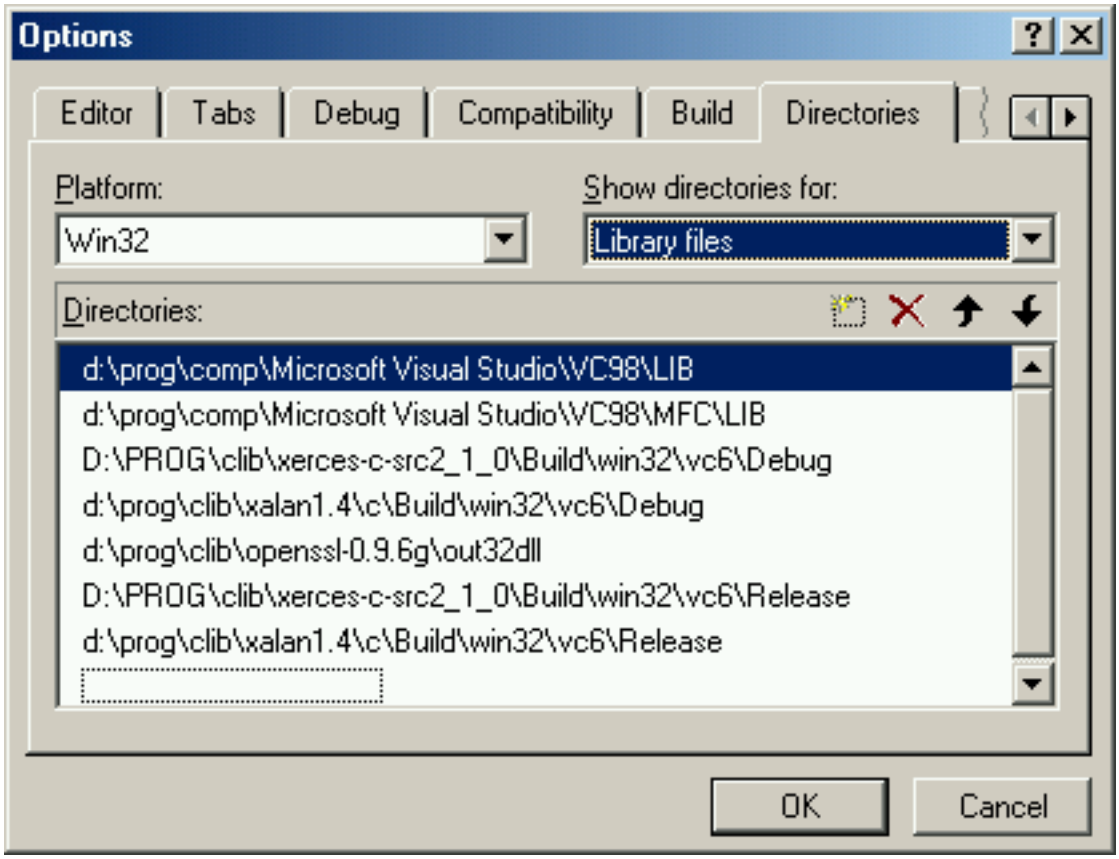
The workspace and project files provided do not make any assumptions about where Xerces, Xalan or OpenSSL might be on the system. The first step is therefore to configure VC directories under Tools->Options (Directories).

For the Include directories you will need something similar to my setup below (replacing D:\PROG\CLIB\.. with the appropriate path on your system).



Include Directories

Similarly the library directories will need to be added to. Note that in the example below, I use both Debug and Release libraries for Xalan and Xerces. As provided, the workspace projects link to the debug libraries for XSEC Debug and Release for XSEC Release.



Link Directories

4.2. Configure

If you are using Xalan and OpenSSL, no configuration is required when building from the Visual C++ v6.0 workspace.

If you wish to disable OpenSSL, you should edit the file *.../src/framework/XSECW32Config.hpp* and comment out the line `#define HAVE_OPENSSL 1`. This will effectively remove support for OpenSSL from the library as it is being compiled.

You will also need to remove the library module `libeay32.lib` from the link->General settings in each of the projects in the XSEC workspace.

To enable support for the Windows Crypto API, edit the XSECW32Config.hpp file and uncomment the line `#define HAVE_WINCAPI 1`

To disable support for Xalan, a similar process is followed. Edit the XSECW32Config.hpp file, and **uncomment** the XSEC_NO_XALAN line. This will remove all support for Xalan from the various source code files.

When compiling, using the "...No Xalan configurations for each project. These are the same as the normal debug or release builds, but the Xalan library is not linked in.

4.3. Build Library and Tools

The main workspace is found in :

.../Projects/VC6.0/xsec/xsec.dsw

You can load this to build the tools or the library using the relevant project. (The library is xsec_lib.)

Samples can be built using the workspace found in :

.../src/Projects/VC6.0/Samples/Samples.dsw

All output will be sent to

.../Build/Win32/VC6/Debug

for the debug builds and Release for the release.