c_installation

Installing the Apache XML Security for C++ Library

Prerequisites

The library requires one of OpenSSL, NSS, or WinCAPI for cryptographic support, but only OpenSSL is well-tested. Xalan-C is also required if XPath and/or XSLT transformations are required.

Version 1.6.0 of the library has been tested with versions 3.x and 2.8.0 of Xerces-C, and Version 0.9.7 (and above) of OpenSSL. Xalan-C has no official releases that will build successfully with Xerces-C 3.x.

Getting the source

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

This project's SVN repository can be checked out directly from here

A HTTP interface to browse the sources online is available here

Building for UNIX

XML Security C++ is currently fully supported on Red Hat and SUSE Linux, Solaris, and Mac OS X. It has been built and tested using GNU gcc 3.x/4.x and Sun Workshop (Solaris).

Set up the Environment

The build process has been corrected as much as possible to rely on standard configure and make commands and not 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 without linking to Xalan. Any attempt to use these features will raise an exception in the library.

Configure

Now go to the root 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.
- --with-xerces=<dir> define Xerces directory
- --with-openssl=<dir> define OpenSSL directory
- --with-nss=<dir> define NSS directory (must be used to get NSS support)
- --enable-debug cause the library to be built with symbols

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 perform these without going through an XPath transform).

Compile

Assuming the output of the above command looks reasonable simply type make (or gmake) in the base directory. This will build the 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 may 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)

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.

Building for Windows

Apache XML Security for C++ has been built and tested on Microsoft's Visual C++ 6.0 and 2005/2008/2010 compilers. The following subsections briefly describe how to rebuild the library, tools and samples using the supplied workspaces. The specific information is for VC6, but the process for later versions is similar.

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 experimental, but should work for some scenarios.

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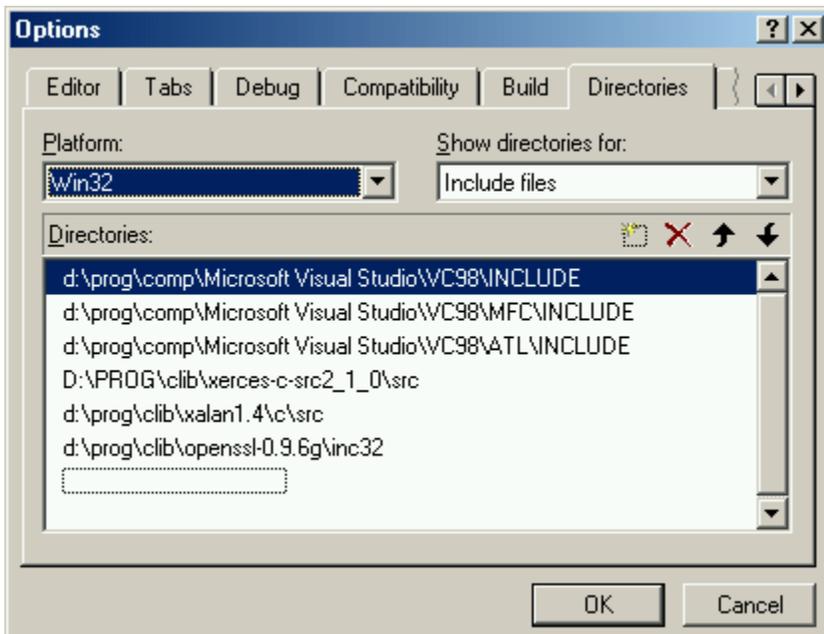
Similarly the library directories will need to be added to. Note that in the example below, I use both Debug and Release libraries for Xalan and Xerces. As provided, the workspace projects link to the debug libraries for XSEC Debug and Release for XSEC Release.
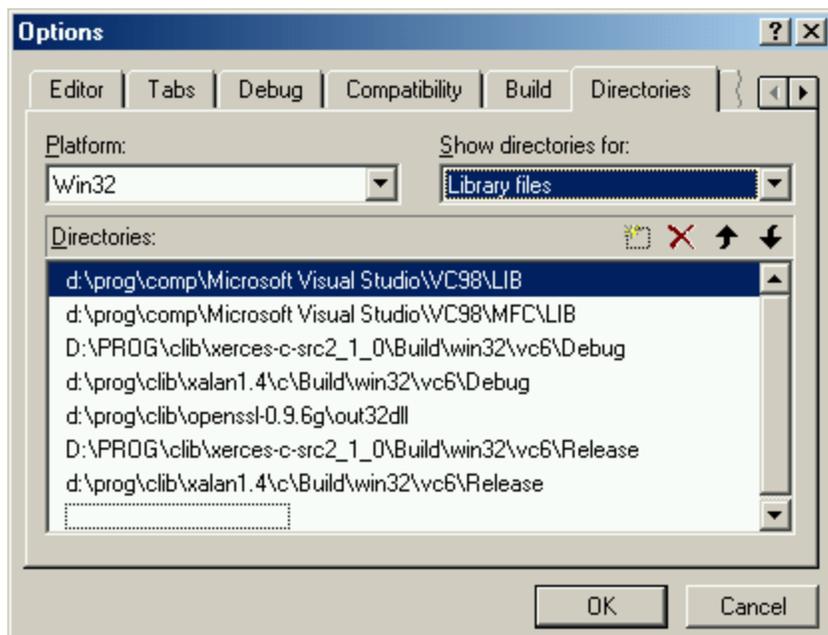
Configure

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.

Build Library and Tools

The main workspace (for VC 6.0) 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.) Project files for VC++ 7.0 and 8.0 also exist in the appropriate Project directories.

Note that as of version 1.3.1, Xerces 3.0 is supported. However this uses an updated library file name, so this must be modified in all the projects in the workspace. This can be set in the Linker option for all versions of Visual C++.

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.