Expat can be built on Windows in two ways:

using MS Visual Studio .NET or Cygwin.

\* Cygwin:

This follows the Unix build procedures.

\* MS Visual Studio 2013, 2015 and 2017:

A solution file for Visual Studio 2013 is provided: expat.sln.

The associated project files (\*.vcxproj) reside in the appropriate

project directories. This solution file can be opened in VS 2015 or VS 2017

and should be upgraded automatically if VS 2013 is not also installed.

Note: Tests have their own solution files.

\* All MS C/C++ compilers:

The output for all projects will be generated in the win32\bin

directory, intermediate files will be located in project-specific

subdirectories of win32\tmp.

\* Creating MinGW dynamic libraries from MS VC++ DLLs:

On the command line, execute these steps:

pexports libexpat.dll > expat.def

pexports libexpatw.dll > expatw.def

dlltool -d expat.def -l libexpat.a

dlltool -d expatw.def -l libexpatw.a

The \*.a files are mingw libraries.

\* Special note about MS VC++ and runtime libraries:

There are three possible configurations: using the

single threaded or multithreaded run-time library,

or using the multi-threaded run-time Dll. That is,

one can build three different Expat libraries depending

on the needs of the application.

Dynamic Linking:

By default the Expat Dlls are built to link statically

with the multi-threaded run-time library.

The libraries are named

- libexpat(w).dll

- libexpat(w).lib (import library)

The "w" indicates the UTF-16 version of the library.

One rarely uses other versions of the Dll, but they can

be built easily by specifying a different RTL linkage in

the IDE on the C/C++ tab under the category Code Generation.

Static Linking:

The libraries should be named like this:

Single-theaded: libexpat(w)ML.lib

Multi-threaded: libexpat(w)MT.lib

Multi-threaded Dll: libexpat(w)MD.lib

The suffixes conform to the compiler switch settings

/ML, /MT and /MD for MS VC++.

Note: In Visual Studio 2005 (Visual C++ 8.0) and later, the

single-threaded runtime library is not supported anymore.

By default, the expat-static and expatw-static projects are set up

to link statically against the multithreaded run-time library,

so they will build libexpatMT.lib or libexpatwMT.lib files.

To build the other versions of the static library,

go to Project - Settings:

- specify a different RTL linkage on the C/C++ tab

under the category Code Generation.

- then, on the Library tab, change the output file name

accordingly, as described above

An application linking to the static libraries must

have the global macro XML\_STATIC defined.