



permission of the OpenSSL Project. 6. Redistributions of any form whatsoever must retain the acknowledgment: "This product includes software developed by the OpenS! for use in the OpenSSL Toolkit (http://www.openssl.org, \* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS' \* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITH IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PA \* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PR( \* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCI \* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, I \* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICE \* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) \* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN \* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERW: \* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN : \* OF THE POSSIBILITY OF SUCH DAMAGE. \_\_\_\_\_ \* This product includes cryptographic software written by E \* (eay@cryptsoft.com). This product includes software writ \* Hudson (tjh@cryptsoft.com). \*/ Original SSLeay License /\* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com) \* All rights reserved. \* This package is an SSL implementation written \* by Eric Young (eay@cryptsoft.com). \* The implementation was written so as to conform with Nets( \* This library is free for commercial and non-commercial us( \* the following conditions are aheared to. The following co \* apply to all code found in this distribution, be it the R( \* lhash, DES, etc., code; not just the SSL code. The SSL de \* included with this distribution is covered by the same co \* except that the holder is Tim Hudson (tjh@cryptsoft.com). \* Copyright remains Eric Young's, and as such any Copyright the code are not to be removed. If this package is used in a product, Eric Young should be

as the author of the parts of the library used. This can be in the form of a textual message at program st in documentation (online or textual) provided with the pac Redistribution and use in source and binary forms, with or modification, are permitted provided that the following co are met: 1. Redistributions of source code must retain the copyrigh notice, this list of conditions and the following disc. 2. Redistributions in binary form must reproduce the above notice, this list of conditions and the following disc. documentation and/or other materials provided with the 3. All advertising materials mentioning features or use of must display the following acknowledgement: "This product includes cryptographic software written k Eric Young (eay@cryptsoft.com)" The word 'cryptographic' can be left out if the rouines being used are not cryptographic related :-). 4. If you include any Windows specific code (or a derivat: the apps directory (application code) you must include "This product includes software written by Tim Hudson ( \* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND \* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMI IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PA \* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUT \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, \* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUL \* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS IN \* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN \* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARI \* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE P( SUCH DAMAGE. \* The licence and distribution terms for any publically ava: derivative of this code cannot be changed. i.e. this code \* copied and put under another distribution licence \* [including the GNU Public Licence.] \*/

### Win32 OpenSSL CVS Build

Win32 OpenSSL CVS Builds are designed for those who want to have the "latest-andgreatest." The Win32 OpenSSL CVS Builds are built such that they are successfully compiled, but there is no guarantee of stability or even if they work even though it \*\*LOOKS\*\* professionally packaged. The actual build is completely untested except for correct a compilation.

If the CVS snapshot is so bad that it requires heavy-duty modifications to make it work (minor fixes to the code to get rid of warnings in various places is okay), then the CVS snapshot will not be updated for the current week. So, there could be 2-3 week periods without any updates to the CVS build. The Win32 OpenSSL Installation Group makes suggestions to improve the distribution, but has no plans to modify the actual CVS source.

The major difference between the CVS package and the normal package is where libeay.dll and ssleay.dll are placed. In the normal package, the DLLs are put into the Windows system directory. In the CVS package, they are left in the installation directory.

# Installation

The installation tools used to build the Win32 OpenSSL Installation Project are Microsoft Visual Studio Professional v6.0 (SP5), Borland C++ Builder 5 Enterprise, MinGW, the OpenSSL source, and <u>InnoSetup</u>.

If you are just reading this and have installed OpenSSL for a third-party utility or application and don't care about developing software, feel free to run the uninstaller. The uninstaller will leave the OpenSSL DLLs on the system and remove the unneeded components, thus freeing up hard drive space while still allowing your OpenSSL enabled application(s) to work. Please note that some applications require OpenSSL to be installed (for instance, the Perl CPAN Net::SSLeay module).

Developers, however, need to perform additional steps before considering uninstalling. Find your compiler in the list above. If it is not listed, then it is not supported at this point.

# Microsoft Visual C++

OpenSSL is designed to build easily under Microsoft Visual C++. However, the requirement that you go and obtain the 25MB Win98 DDK and have the latest service pack (SP5 is 120MB) installed can prove to be a hinderance to some people (particularly those with modems).

So, the installation of the Win32 OpenSSL binaries is a fairly simple process (similar to the Borland C++ Builder process).

The first thing to do (assuming a default installation of 'C:\OpenSSL') is to go to 'C:\OpenSSL\lib\VC' and copy all of the files to your Visual C++ 'lib' directory. This directory is sometimes located in a somewhat cryptic location such as 'C:\Program Files\Microsoft Visual Studio\VC98\lib' or 'C:\Program Files\Microsoft Visual C++\lib'.

Next, copy everything in the 'C:\OpenSSL\include' directory to your Visual C++ 'include' directory.

That's it! You are ready to go write OpenSSLcapable code!

# **Borland Builder**

Borland C++ Builder is one of the more difficult compilers to build OpenSSL for. However, Borland/Inprise have very nice tools to extract information from pre-made DLL files. The problem is, is that most people only own Microsoft's OR Borland's compilers. So, finding a pre-built DLL is difficult for a Borland user in the open-source arena.

The Win32 OpenSSL Installation Project builds Borland C++ Builder-compatible LIB files as part of the project. So, it is a matter of copying files to the proper locations.

Assuming a default installation (C:\OpenSSL), go to 'C:\OpenSSL\lib\Builder5' and copy all of the files to your C++ Builder 'lib' directory. Note that the source directory says 'Builder5', but there is a high probability that it will work under Builder 3, 4, and 5. However, only Builder 5 has been tested at this point (anyone verifying 3 and 4 should send a notice to the Win32 OpenSSL Installation Group as to whether or not the .lib files work).

Next, copy everything in the 'C:\OpenSSL\include' directory to your C++ Builder 'include' directory.

That's it! You are ready to go write OpenSSLcapable code!

| MinGW   |
|---|
| Using the tools provided by both Borland's and MinGW's compiler set, MinGW is now supported.  |
| The Win32 OpenSSL Installation Project builds<br>MinGW .a and .def files as part of the project.<br>So, it is a matter of copying files to the proper<br>locations. |
| Assuming a default installation (C:\OpenSSL),<br>go to 'C:\OpenSSL\lib\MinGW' and copy all of<br>the files to your MinGW 'lib' directory.                           |
| Next, copy everything in the<br>'C:\OpenSSL\include' directory to your MinGW<br>'include' directory.  |
| That's it! You are ready to go write OpenSSL-<br>capable code!  |
|   |
| © 2000-2003 Shining Light Productions   |

### E-Commerce (Net::SSLeay)

Several users of the Win32 OpenSSL Installation Project have need of Net::SSLeay and/or Crypt::SSLeay for Merchant Account E-Commerce providers. Since many Windows users do not have access to or can afford a compiler, the Win32 OpenSSL Installation Project team provides default builds of binaries for these Perl modules. The following instructions are for use with the Win32 OpenSSL Installation Project with the CPAN Net::SSLeay module:

1) Make sure Perl is installed in C:\Perl (this is the default location for most PCs). If it is located elsewhere, you will have to modify a lot of code referencing this hardcoded location.

2) Start a Command Prompt.

3) Go into the directory where OpenSSL is installed (e.g. c:\OpenSSL).

4) Inside the main OpenSSL directory is a directory called 'perl', go into that directory. If you have Perl 5.6 installed, go into the '5.6' sub-directory and then the sub-directory 'Net\_SSLeay'. If you have Perl 5.8 installed, go into the '5.8' sub-directory and then the sub-directory 'Net\_SSLeay'.

5) Type 'install' and press enter.

At this point, several Perl scripts are run that

install the pre-compiled binaries.

#### E-Commerce (Crypto::SSLeay)

Several users of the Win32 OpenSSL Installation Project have need of Net::SSLeay and/or Crypt::SSLeay for Merchant Account E-Commerce providers. Since many Windows users do not have access to or can afford a compiler, the Win32 OpenSSL Installation Project team provides default builds of binaries for these Perl modules. The following instructions are for use with the Win32 OpenSSL Installation Project with the CPAN Crypto::SSLeay module:

1) Make sure Perl is installed in C:\Perl (this is the default location for most PCs). If it is located elsewhere, you will have to modify a lot of code referencing this hardcoded location.

2) Start a Command Prompt.

3) Go into the directory where OpenSSL is installed (e.g. c:\OpenSSL).

4) Inside the main OpenSSL directory is a directory called 'perl', go into that directory. If you have Perl 5.6 installed, go into the '5.6' sub-directory and then the sub-directory 'Crypto\_SSLeay'. If you have Perl 5.8 installed, go into the '5.8' sub-directory and then the sub-directory 'Crypto\_SSLeay'.

5) Type 'install' and press enter.

At this point, several Perl scripts are run that

install the pre-compiled binaries.

