Welcome to the Win32 OpenSSL Installation Project. This project is designed to make it easy for Windows developers to get OpenSSL binaries (DLLs) installed and enable programmers to quickly get back to what they were doing. Note that this is an open-source project and Shining Light Productions is only hosting the project as a favor to the open-source community. This is an unofficial extension of the OpenSSL Project.

The Win32 OpenSSL Installation Group hopes that you enjoy this software and get many years of use out of it!

Legal Notice: This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)
LICENSE ISSUES
==============

The OpenSSL toolkit stays under a dual license, i.e. both the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License
---------------

/*  ===================================================================
*  Copyright (c) 1998-2004 The OpenSSL Project.  All rights reserved.
*
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*
*  1. Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*  2. Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in
*     the documentation and/or other materials provided with the
*     distribution.
*  3. All advertising materials mentioning features or use of this
*     software must display the following acknowledgment:
*     "This product includes software developed by the OpenSSL Project
*     for use in the OpenSSL Toolkit. (http://www.openssl.org)"
*  4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
*     endorse or promote products derived from this software without prior
*     written permission. For written permission, please contact
*     openssl-core@openssl.org.
*  5. Products derived from this software may not be called "OpenSSL"
*     nor may "OpenSSL" appear in their names without prior written
*     permission from the OpenSSL Project.
*/
permission of the OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following
   acknowledgment:
   "This product includes software developed by the OpenSSL Project
   for use in the OpenSSL Toolkit (http://www.openssl.org/)

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PURPOSE
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT
ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
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 OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
OF THE POSSIBILITY OF SUCH DAMAGE.

====================================================================

This product includes cryptographic software written by Eric Young
(eay@cryptsoft.com). This product includes software written by
Tim 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 Netscape
SSL specifications.

This library is free for commercial and non-commercial use
the following conditions are aheared to. The following code
apply to all code found in this distribution, be it the RL
lhash, DES, etc., code; not just the SSL code. The SSL
documented included with this distribution is covered by the same
copyrights 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 startup*
* in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:

1. Redistributions of source code must retain the copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above
   copyright notice, this list of conditions and the following disclaimer
   in the documentation and/or other materials provided with the
distribution.
3. All advertising materials mentioning features or use of this software
   must display the following acknowledgement:
   "This product includes cryptographic software written by
   Eric Young (eay@cryptsoft.com)"
   The word 'cryptographic' can be left out if the routines
   being used are not cryptographic related :-).
4. If you include any Windows specific code (or a derivative thereof)
   the apps directory (application code) you must include
   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG `AS IS' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMIT
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PART
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUB-
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) AR
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE PO
SUCH DAMAGE.

The licence and distribution terms for any publically availab
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 Builds are designed for those who want to have the "latest-and-greatest." 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.
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 InnoSetup.

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.
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 OpenSSL-capable code!
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 OpenSSL-capable code!
Using the tools provided by both Borland’s and MinGW’s compiler set, MinGW is now supported.

The Win32 OpenSSL Installation Project builds MinGW .a and .def 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\MinGW' and copy all of the files to your MinGW 'lib' directory.

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

That’s it! You are ready to go write OpenSSL-capable code!
Several users of the Win32 OpenSSL Installation Project have need of Net::SSLaye and/or Crypt::SSLaye 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::SSLaye 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_SSLaye'. If you have Perl 5.8 installed, go into the '5.8' sub-directory and then the sub-directory 'Net_SSLaye'.

5) Type 'install' and press enter.

At this point, several Perl scripts are run that
install the pre-compiled binaries.
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.
The Win32 OpenSSL Installation Group is proud to present the Win32 OpenSSL Installation to those who need OpenSSL on their machines as well as letting Windows programmers get back to what they do best...programming.

Sincerely,
Thomas J. Hruska, III
Shining Light Productions
"Meeting the Needs of Fellow Programmers"