Technical Report

Integration of NetApp Manageability SDK 9.8P8 Binaries on OpenSSL 3.x in Linux 8.x and Windows

Hatim Ali Haidry, NetApp

Abstract

This document outlines how to test the NMSDK 9.8P8 binaries on OpenSSL 3.x in a Linux or Windows environment.

TABLE OF CONTENTS

Introduction

NetApp Manageability Software Development Kit (NMSDK) binaries with a version before 9.8P8 were integrated with the OpenSSL version 1.0.x, shipped with NMSDK. Beginning with NMSDK 9.8P8, the OpenSSL 3.x version is integrated, and OpenSSL 3.x binaries are no longer packaged and shipped with NMSDK.

This document outlines how to test the NMSDK 9.8P8 binaries on OpenSSL 3.x in a Linux or Windows environment.

NMSDK on Linux RHEL 8.x

First you must install OpenSSL 3.x using RPM. There are certain dependency challenges you might face during installation on Linux Red Hat Enterprise Linux (RHEL) 8.x.

As a result, to install the OpenSSL 3.x Federal Information Processing Standard (FIPS) compliance on Linux 8.x, build the OpenSSL libraries from its source package.

OpenSSL 3.x installation steps

Build and install OpenSSL 3.x on Linux 8.x machine.

NOTE: If OpenSSL 3.x is already installed on Linux RHEL machine, identify the location of the OpenSSL 3.x installation path, skip these steps, and directly move on to the 'Test NMSDK Binaries' section.

Steps

1. Download the latest version of OpenSSL 3.x: https://www.openssl.org/source/openssl-3.3.0.tar.gz
2. Copy the "openssl-3.3.0.tar.gz" file into your user folder location, for example, <UserLocation>.
3. Change the directory to a user location folder using the **cd <UserLocation>** command.
4. Extract the "openssl-3.3.0.tar.gz" file with the **tar -xvf openssl-3.3.0.tar.gz** command. This creates a folder named openssl-3.3.0 in <UserLocation>, such as <UserLocation>/openssl-3.3.0. You can now extract all the required source files within this folder.
5. Change the directory to openssl-3.3.0 using the **cd <UserLocation>/openssl-3.3.0** command.
6. Run the following command:
   ./Configure enable-fips shared no-rc5 no-idea no-asm -fno-strict-aliasing --prefix=<UserLocation>/openssl-3.3.0/LinuxBuild/Lib --openssldir=<UserLocation>/openssl-3.3.0/LinuxBuild/Lib/SSL
7. Install the perl-IPC-Cmd package using the **yum install perl-IPC-Cmd** command and re-run the command in step 6 if the following error occurs:

   Can't locate IPC/Cmd.pm in @INC (you may need to install the IPC::Cmd module)
8. Run the following command:

   make
9. Run the following command:

   make install
10. Once step 8 is complete, the command creates all the required OpenSSL binaries and other supporting files which are required for NMSDK binaries testing at the location <UserLocation>/openssl-3.3.0/LinuxBuild/Lib.

Test NMSDK Binaries

Before you begin, there are some pre-requisites for testing NMSDK binaries using a sample project.

Before you begin

- To use NMSDK binaries on a Linux 8.x RHEL platform, you must install the package libtirpc-devel, using the sudo yum install libtirpc-devel command.
- If a sample uses NMSDK binaries, you must add the following options during compilation:
  - To include RPC related headers:
    -I/usr/include/tirpc
  - To include RPC related libraries:
    -ltirpc

Steps

The following steps outline how to build and test the NMSDK binaries using OpenSSL 3.x on a Linux 8.x machine. In this scenario, it tests one of the C samples named “apitest”.

1. Copy the NMSDK binaries “netapp-manageability-sdk- XXXXX-linux.zip” zip file to the <UserLocation> folder.
2. Unzip the “netapp-manageability-sdk-XXXXX-linux.zip” file so it can extract all the NMSDK binaries into the <UserLocation>/netapp-manageability-sdk- XXXXX-linux/netapp-manageability-sdk- XXXXX location. Refer to this path as <NMSDKBinariesRoot>.
3. Copy files named “libssl.so.3” and “libcrypto.so.3” from the <UserLocation>/openssl-3.3.0/LinuxBuild/Lib/lib64/ location to <NMSDKBinariesRoot>/lib/linux-64.
4. Change the directory to a sample application with the command
   cd <NMSDKBinariesRoot>/src/util/apitest
5. If OpenSSL 3.x is installed on the Linux RHEL machine, or you skipped the ‘OpenSSL 3.x installation steps’ section and the location of OpenSSL 3.x installation path is something other than “/usr/”, then you must define an environment variable named “OPENSSL_PATH” with values from either of following commands.

   If OpenSSL is already installed and the installation path is something other than /usr/:

   setenv OPENSSL_PATH <Already installed OpenSSL 3.x installation path>

   OR

   If you installed OpenSSL with the ‘OpenSSL 3.x installation steps’:

   setenv OPENSSL_PATH <UserLocation>/openssl-3.3.0/LinuxBuild/Lib

   The OPENSSL_PATH environment variable adds the following options when used with the **make** command.

   a. Includes the path of OpenSSL 3.3.0 related headers:
      -I<UserLocation>/openssl-3.3.0/LinuxBuild/Lib/include
   b. Includes the path of OpenSSL 3.3.0 related libraries:
      -L<UserLocation>/openssl-3.3.0/LinuxBuild/Lib/lib64
   c. Includes OpenSSL 3.3.0 libraries:
      -lssl -lcrypto

NOTE: The above commands should be added to “Makefile” if you create your own sample using NMSDK libraries.

6. Run the following command:
   make

   Once the above command is complete, it generates the “apitest” binary sample.
7. Set the library path:
   setenv LD_LIBRARY_PATH <NMSDKBinariesRoot>/lib/linux-64
8. Run the **./apitest <FilerIP> <user> <password> system-get-version** command. It should return a similar output to the following:

```
<results status="passed">
        <build-timestamp>1688367881</build-timestamp>
        <is-clustered>true</is-clustered>
        <version>NetApp Release Clawhammer__9.14.1: Mon Jul 03 07:04:41 UTC
2023</version>
        <version-tuple>
                <system-version-tuple>
                        <generation>9</generation>
                        <major>14</major>
                        <minor>1</minor>
                </system-version-tuple>
        </version-tuple>
</results>
```

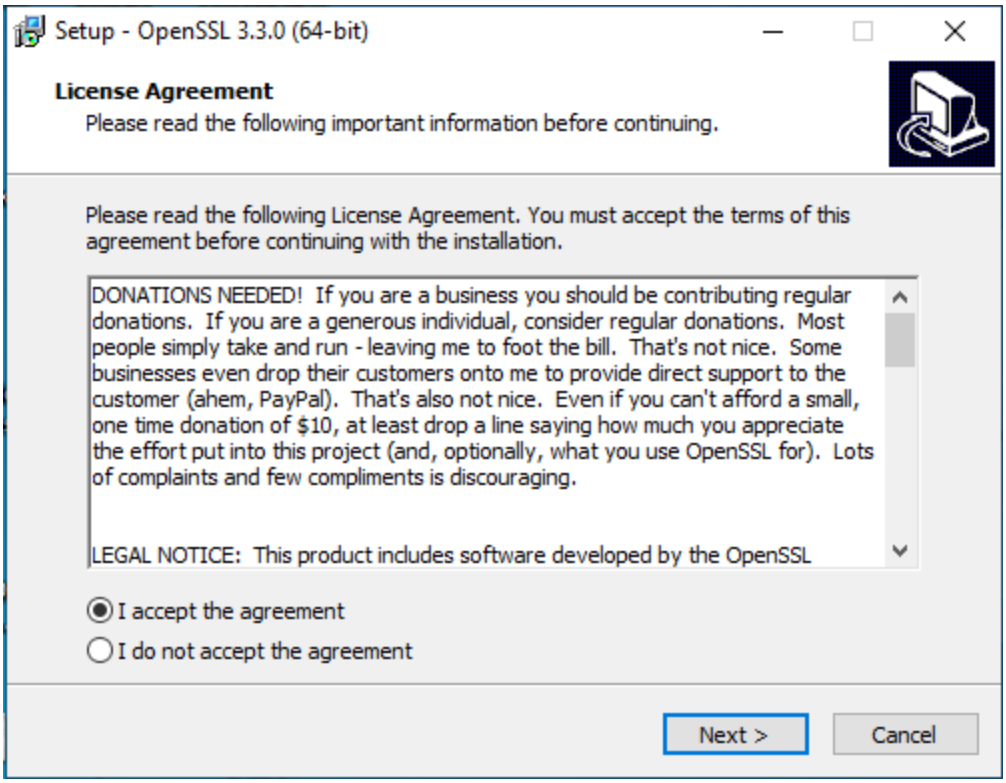
NMSDK on Windows

OpenSSL 3.x installation steps

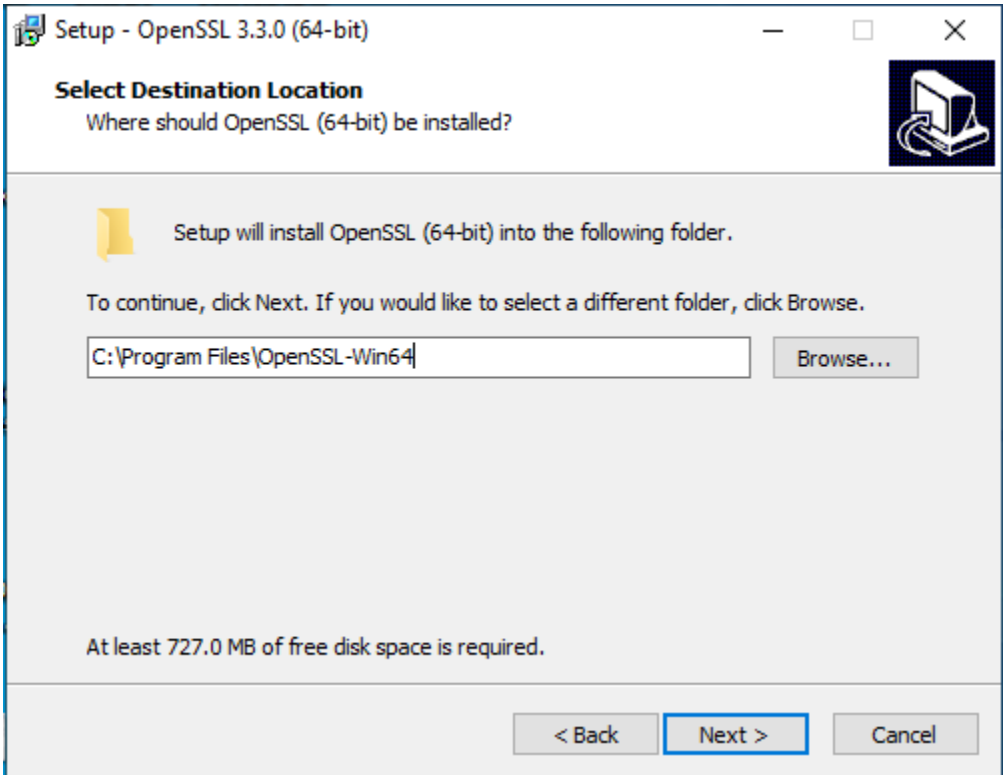
Build and install OpenSSL 3.x on a Windows machine.

Steps

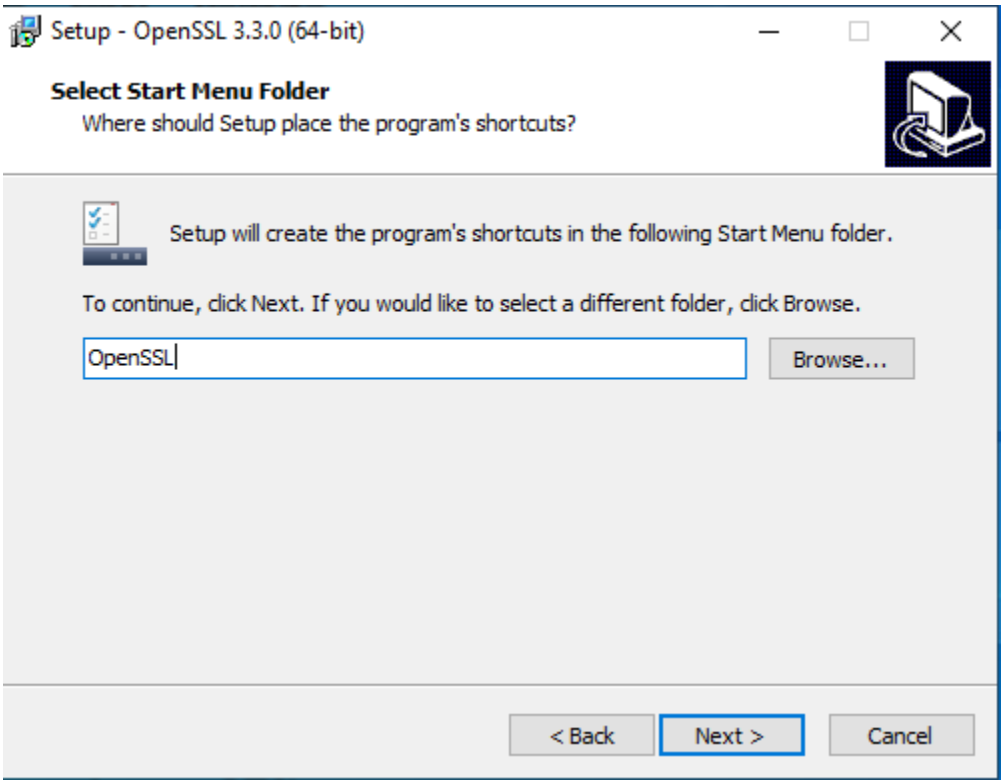
1. Download the latest version of the OpenSSL 3.x 64-bit Windows installer from https://slproweb.com/products/Win32OpenSSL.html.
   For testing the NMSDK binaries on Windows platform, the OpenSSL v3.3.0 64-bit was downloaded from https://slproweb.com/download/Win64OpenSSL-3_3_0.exe .
2. To install OpenSSL 3.x, double click on the downloaded Win64OpenSSL-3_3_0.exe file.
3. Read and accept the License Agreement.



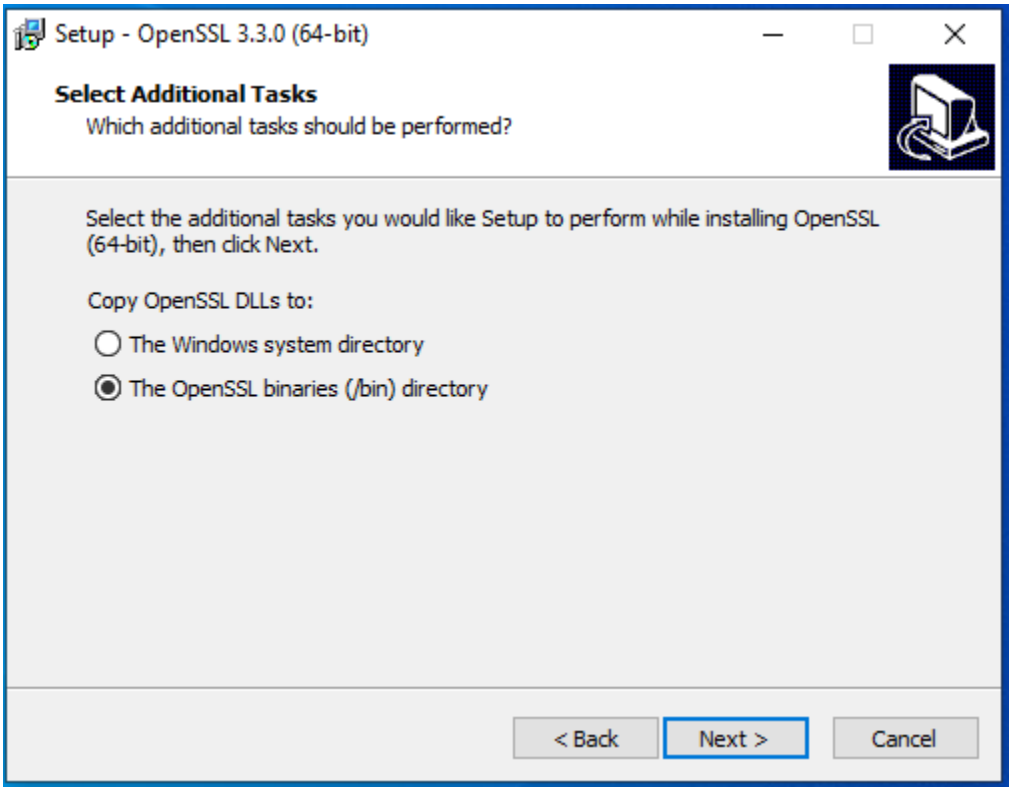
4. Select the **Next** button to continue. When asked for the installation location, keep the location as C:\Program Files\OpenSSL-Win64.



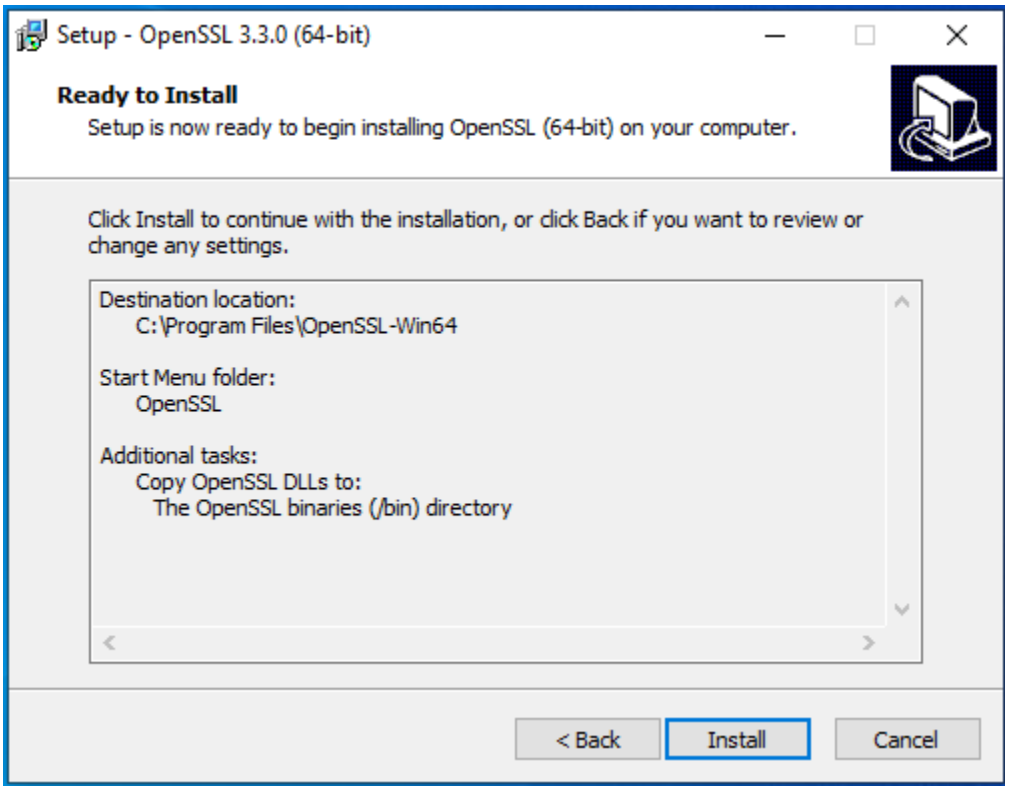
5. Select the **Next** button to continue and choose your Start Menu folder as OpenSSL.

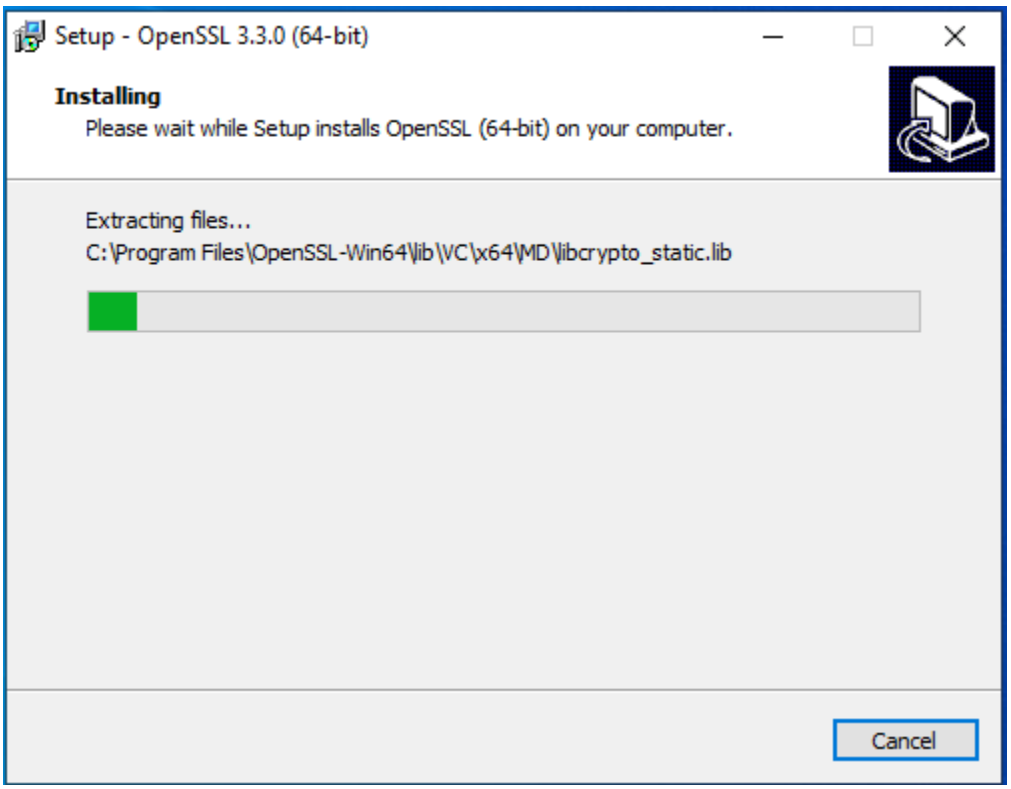


6. Select the **Next** button to continue to an additional task to copy OpenSSL DLLS to the Windows system's directory or OpenSSL directory. Select **The OpenSSL binaries (/bin) directory**.



7. Select the **Next** button to continue and then select the **Install** button.





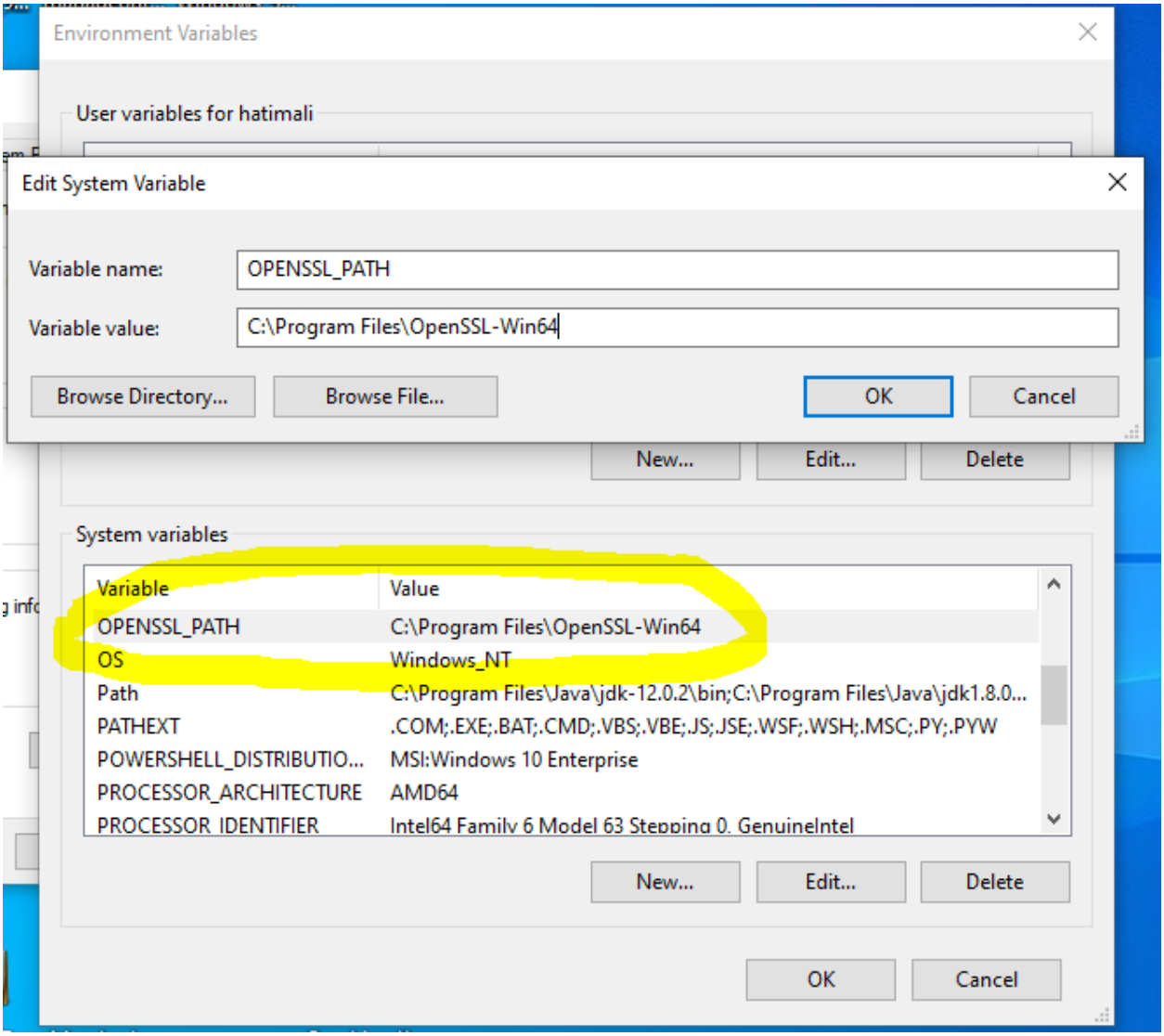
8. Once the installation is complete, all OpenSSL 3.x related headers, libraries, and binaries are copied to the C:\Program Files\OpenSSL-Win64 location.

Test NMSDK Binaries

Steps

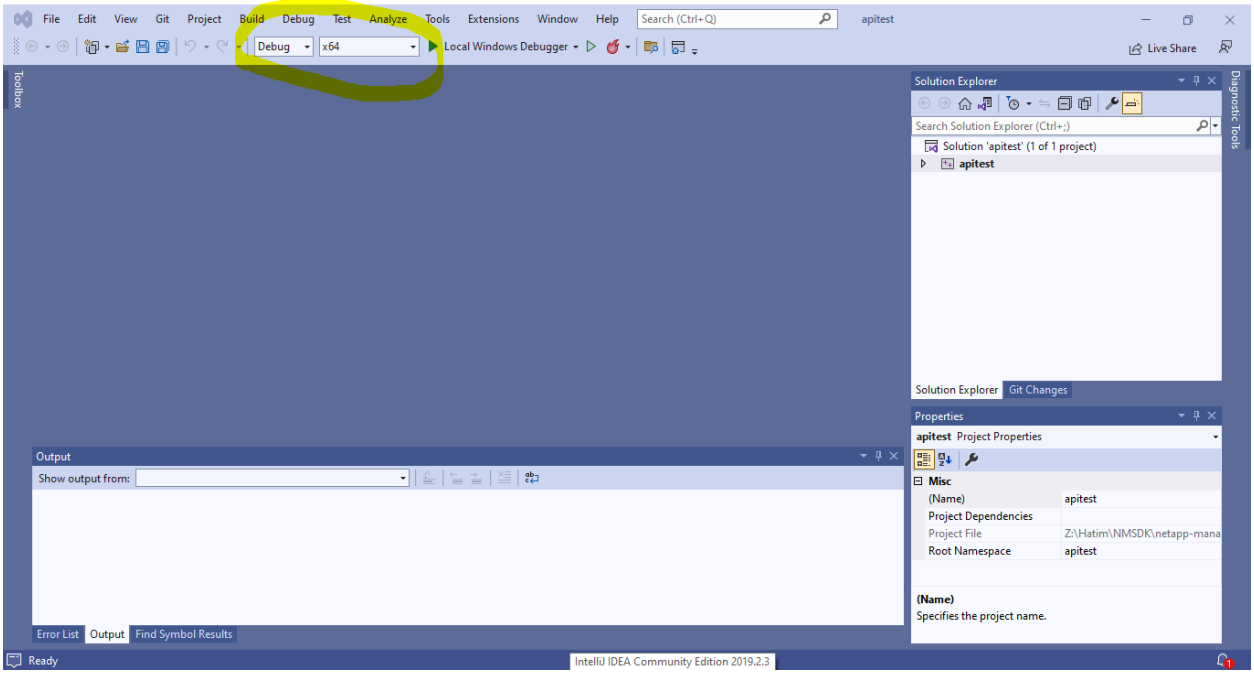
The following steps outline how to build and test the NMSDK binaries using OpenSSL 3.x on a Windows machine. In this scenario, it tests one of the C samples named “apitest”.

1. Define a system environment variable named “OPENSSL_PATH” with a value that contains the OpenSSL 3.x installation root folder. For example; C:\Program Files\OpenSSL-Win64.

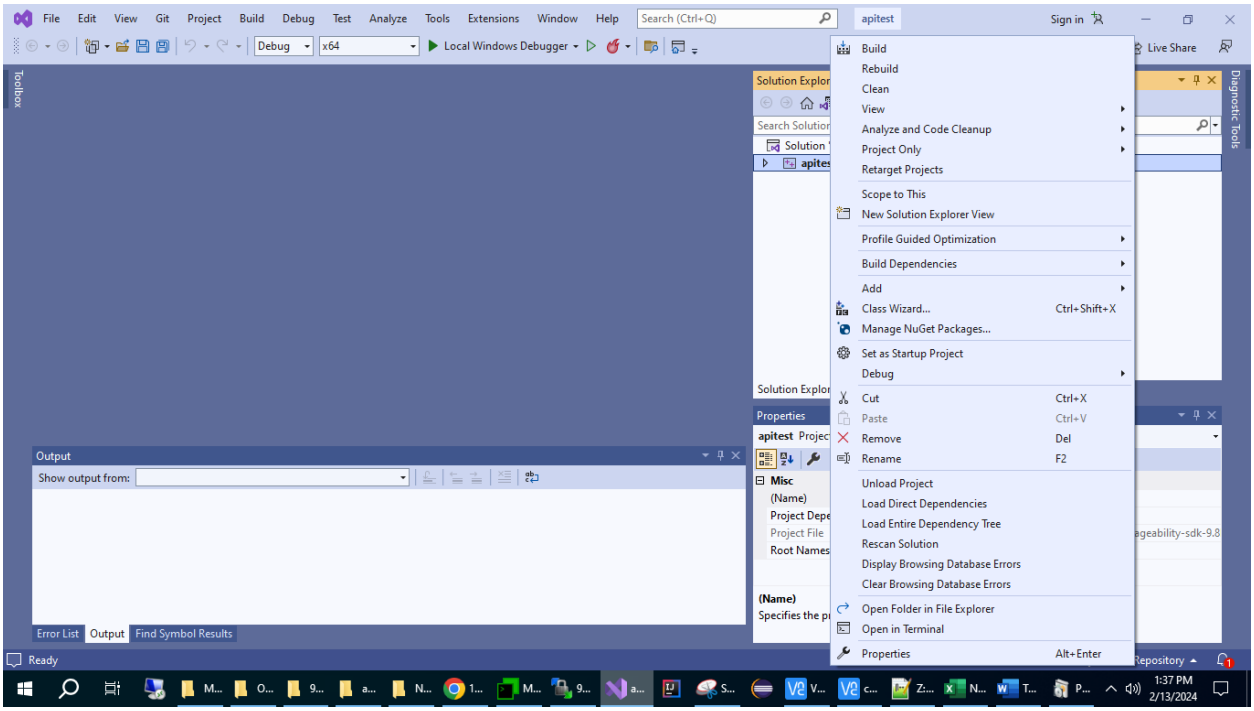


2. Copy files named “libssl.lib” and “libcrypto.lib” from C:\Program Files\OpenSSL-Win64\lib\VC\x64\MDd location to C:\Program Files\OpenSSL-Win64\lib.

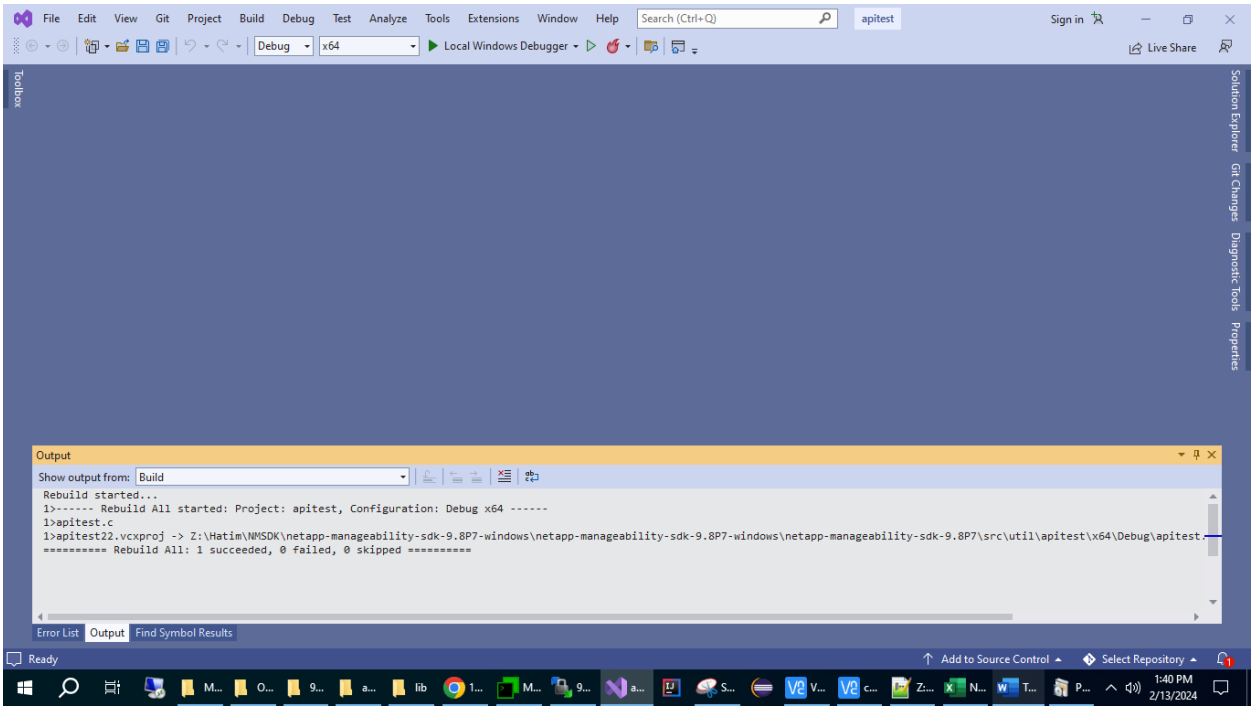
3. Copy the NMSDK binaries netapp-manageability-sdk- XXXXX-windows.exe zip file to the <UserLocation> folder.
4. Unzip the netapp-manageability-sdk-XXXXX-windows.exe file, so it can extract all the NMSDK binaries into the <UserLocation>/netapp-manageability-sdk- XXXXX-windows/netapp-manageability-sdk- XXXXX location. Refer to this path as <NMSDKBinariesRoot>.
5. Change the directory to the sample application using command **cd <NMSDKBinariesRoot>/src/util/apitest**
6. Open the Visual Studio project file named apitest22.vcxproj in Visual Studio 2022. This opens the apitest sample in the Visual Studio integrated development environment (IDE). Select **Debug** and **x64** targets from the drop-down menus.



7. Right click on the **apitest** project in the Solution Explorer Window and select **Rebuild**.



This compiles the sample and creates an executable named “apitest.exe” for a Debug 64 bit target in the location shown in the output window.



8. Go to the executable location and copy the debug version of NMSDK binaries named “libxmld.dll” from the <NMSDKBinariesRoot>\lib\nt\x64\VS22 location to the apitest executable file location.
9. Copy the “libssl-3-x64.dll” and “libcrypto-3-x64.dll” dll files from the C:\Program Files\OpenSSL-Win64\bin location to the apitest executable file location.

10. Go to the apitest executable file location, take the command prompt for that location, and run the **apitest.exe <FilerIP> <user> <password> system-get-version** command. It should return the same result as the following example output:

```
<results status="passed">
        <build-timestamp>1686036536</build-timestamp>
        <is-clustered>true</is-clustered>
        <version>NetApp Release Clawhammer__9.14.1: Tue Jun 06 07:28:56 UTC
2023</version>
        <version-tuple>
                <system-version-tuple>
                        <generation>9</generation>
                        <major>14</major>
                        <minor>1</minor>
                </system-version-tuple>
        </version-tuple>
</results>
```

Note

Replace the highlighted values which appear in the commands throughout the process with their actual values.

- <UserLocation>
- <NMSDKBinariesRoot>
- <FilerIP>
- <user>
- <password>