Compile the LM-X SDK on Windows

The information on this page refers to LM-X License Manager v4.7 and newer, which eliminated the need to use a separate mingw32 installer to compile the LM-X SDK using MinGW.

The following steps are designed to compile LM-X SDK on a Windows machine in 5 minutes or less from a command line, using Visual Studio, and MinGW.

Note: By default, when you install the SDK, Visual Studio opens to let you compile the SDK. We recommend that you use the project files and use the Build, Clean and Rebuild UI actions within the IDE to compile and recompile the SDK. Alternatively, you can compile the SDK from a command line using nmake, as described in "Compiling the LM-X SDK from a command line", below.

When compiling the LM-X SDK under Windows, one of the following is required:

- Visual Studio (Express, Standard, Pro or better).
- MinGW

Compiling the LM-X SDK using Visual Studio

To compile the LM-X SDK using Visual Studio:

Step 1. Copy your LM-X-SDK lmx.lic file to the config directory and specify a desired OPTION for the vendor.

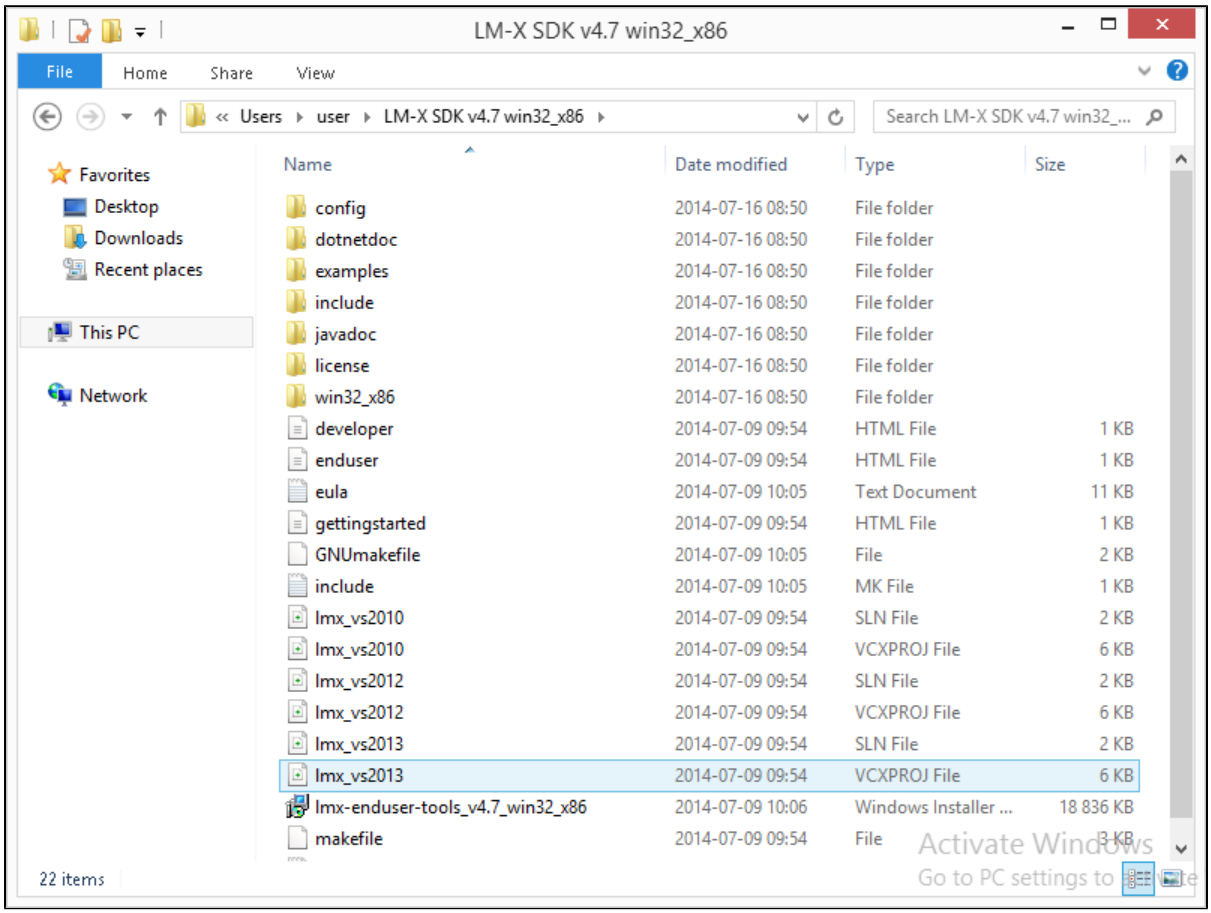
You can extend or modify the behavior of the license server during pre-compilation by editing the lmx_server_conf.c file.

Note: If you are upgrading LM-X, remember to copy your LM-X security configuration file from a previous LM-X installation to the config directory. (In LM-X SDK versions older than 4.2, security_config.lmx file was named after your vendorname.lmx.)

Step 2. Run Visual Studio.

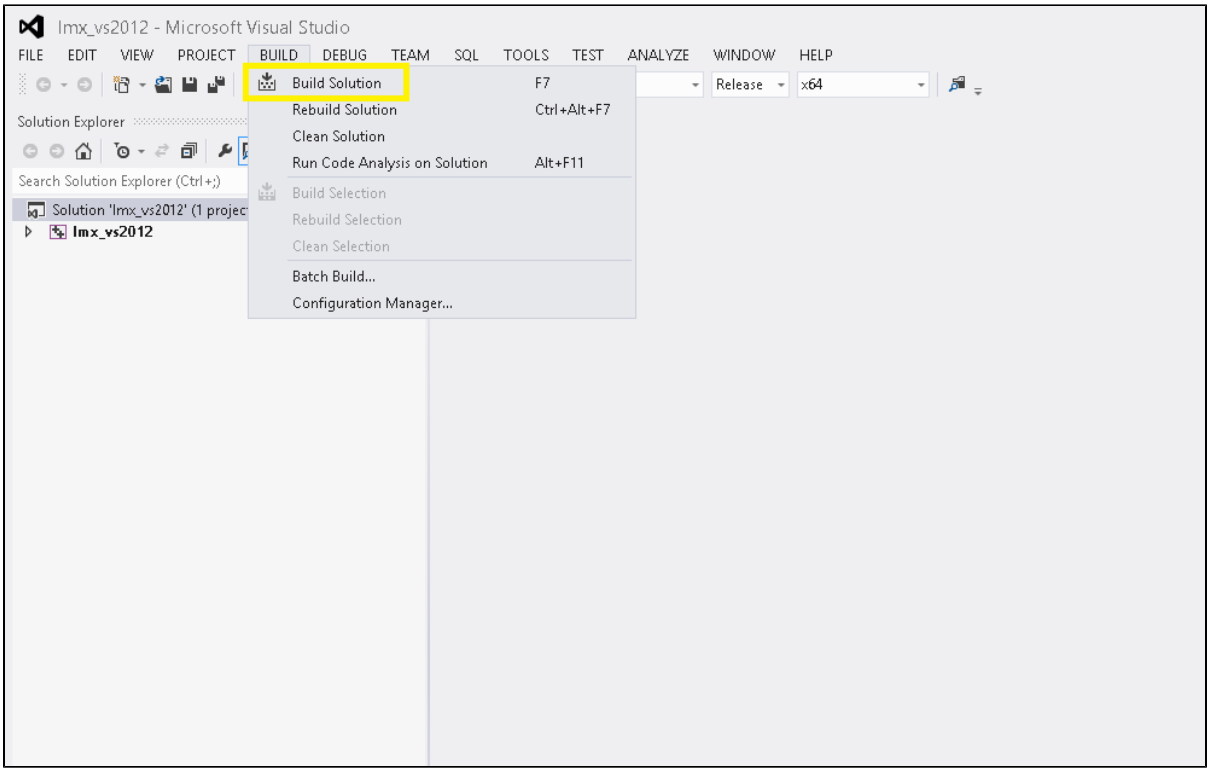
To run Visual Studio, either:

- Run Visual Studio by default by clicking **Finish** at the end of installing the LM-X SDK on Windows
  or
- Double-click on the proper Visual Studio solution file, as shown below:

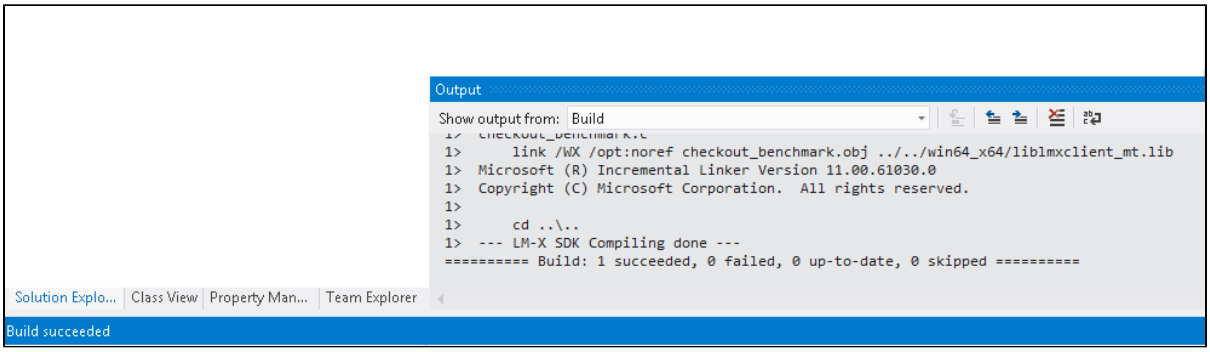


Note: Make sure the config directory includes the license file.

Step 3. Click Build tab and from the list of options that appears, select "Build Solution".



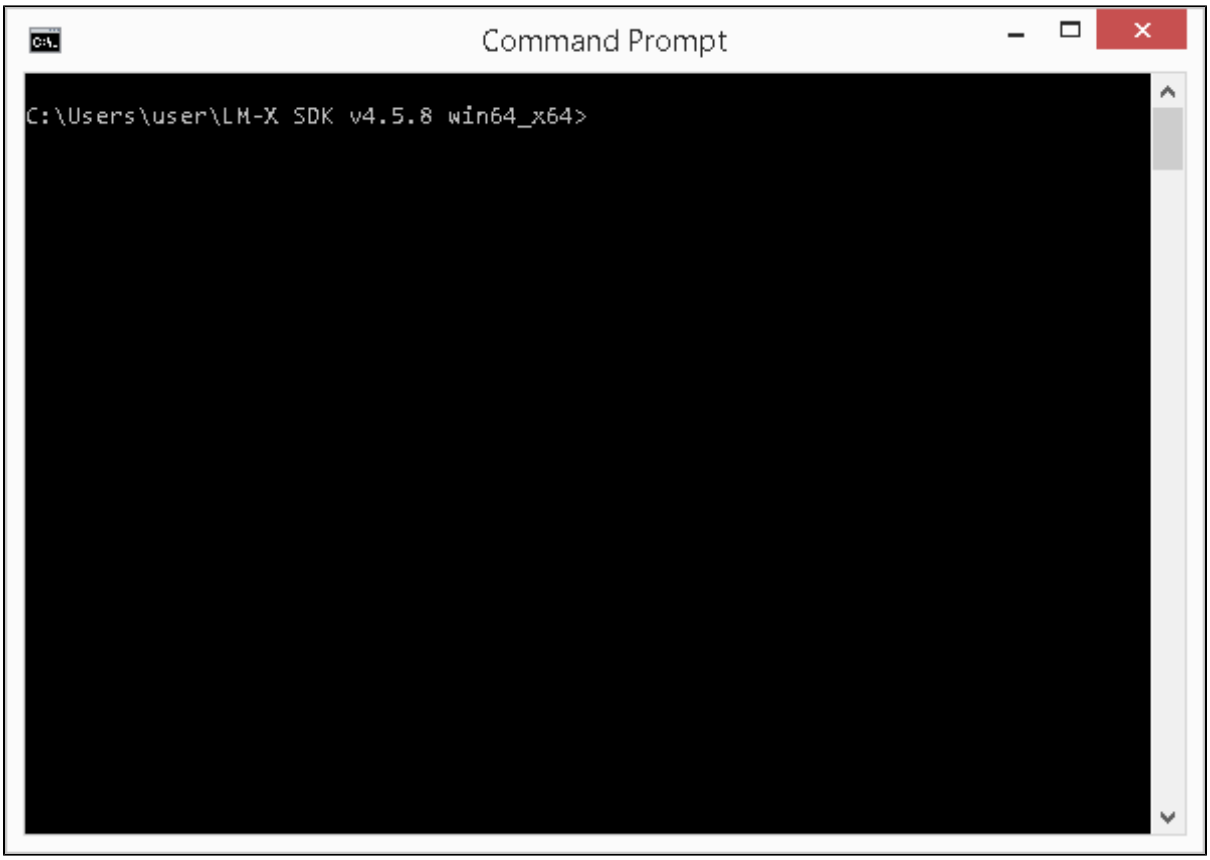
When the compilation is successfully completed, you will see a "Build succeeded" message on the Visual Studio status bar, as shown below.



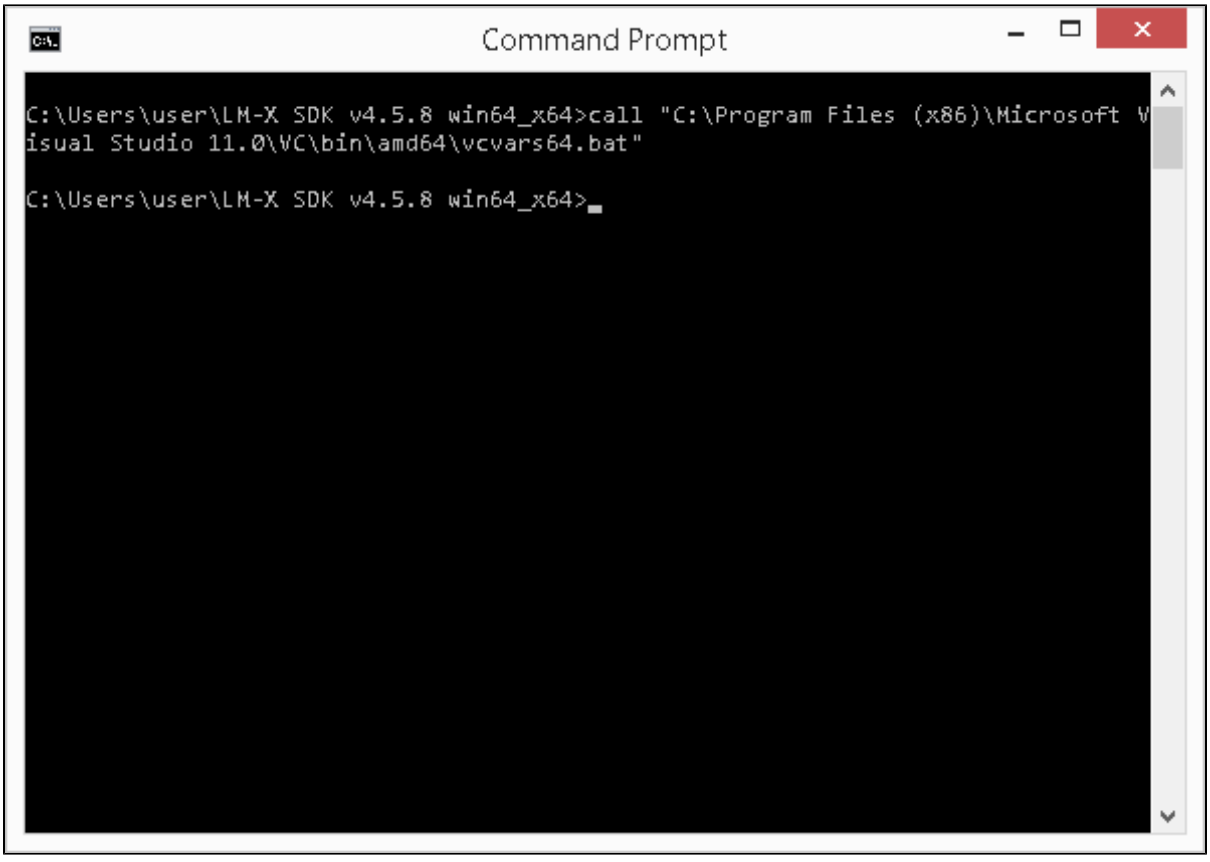
Compiling the LM-X SDK from a command line using Visual Studio

To compile the LM-X SDK from a command line using Visual Studio:

Step 1. Run cmd.exe.



Step 2. Set the environment variable.



The following are examples of environment variables:

Visual Studio 2010 32-bit

```
C:\Users\user\LM-X SDK v4.5.8 win64_x64> call %PROGRAM_FILES%\Microsoft Visual Studio 10.0\VC\bin\vcvars32.bat
```

Visual Studio 2012 32-bit

```
C:\Users\user\LM-X SDK v4.5.8 win64_x64> call %PROGRAM_FILES%\Microsoft Visual Studio 11.0\VC\bin\vcvars32.bat
```

Visual Studio 2013 32-bit

```
C:\Users\user\LM-X SDK v4.5.8 win64_x64> call %PROGRAM_FILES%\Microsoft Visual Studio 12.0\VC\bin\vcvars32.bat
```

Visual Studio 2010 64-bit

```
C:\Users\user\LM-X SDK v4.5.8 win64_x64> call %PROGRAM_FILES%\Microsoft Visual Studio 10.0
\VC\bin\amd64\vcvars64.bat
```

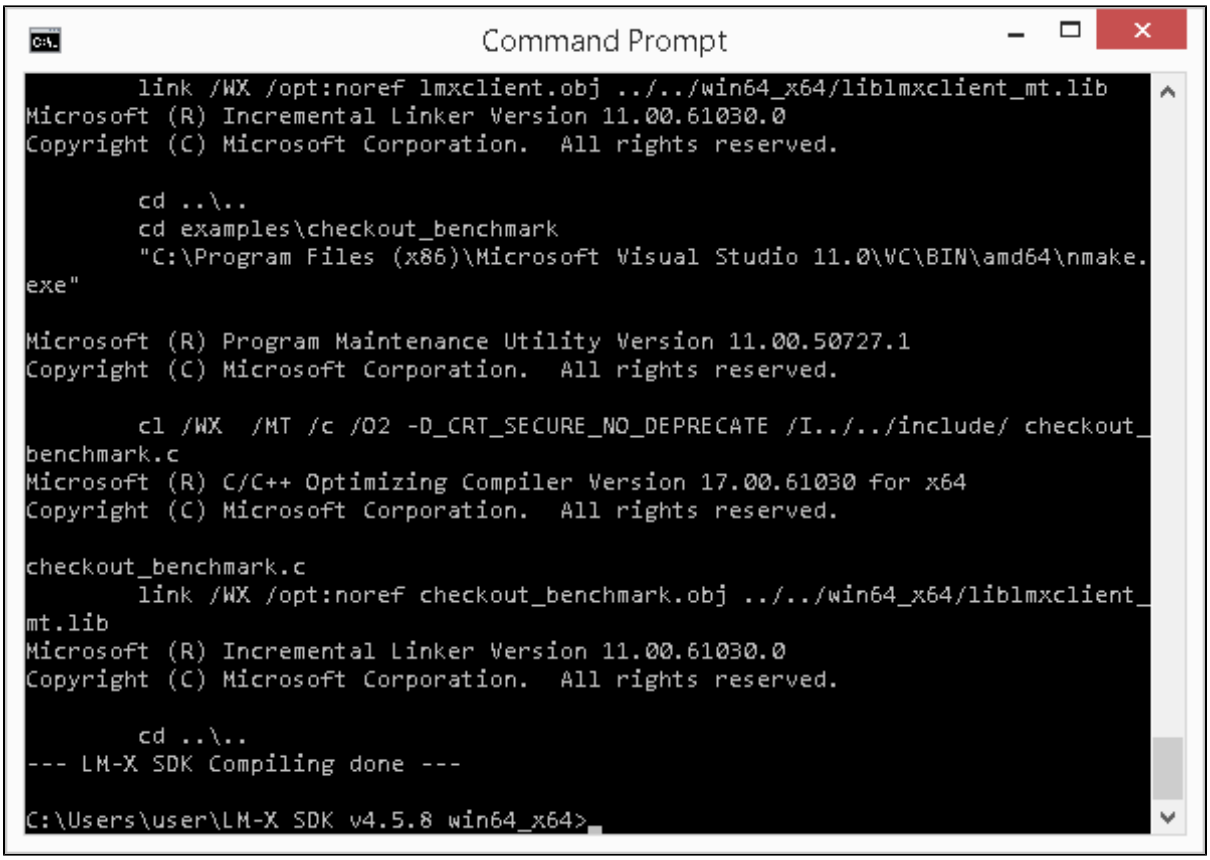
Visual Studio 2012 64-bit

```
C:\Users\user\LM-X SDK v4.5.8 win64_x64> call %PROGRAM_FILES%\Microsoft Visual Studio 11.0
\VC\bin\amd64\vcvars64.bat
```

Visual Studio 2013 64-bit

```
C:\Users\user\LM-X SDK v4.5.8 win64_x64> call %PROGRAM_FILES%\Microsoft Visual Studio 12.0
\VC\bin\amd64\vcvars64.bat
```

Step 3. From the root directory of the LM-X distribution, run nmake**.**



Note: You must run nmake from the SDK root directory. Running nmake from a subdirectory may produce error messages and fail.

Cleaning the LM-X SDK using Visual Studio

```
C:\Users\user\LM-X SDK v4.5.8 win64_x64> nmake clean
```

Note: You may want to clean previously compiled files when rebuilding the SDK with a different license or security key.

See Installation issues for information about problems and workarounds related to compiling the LM-X SDK.

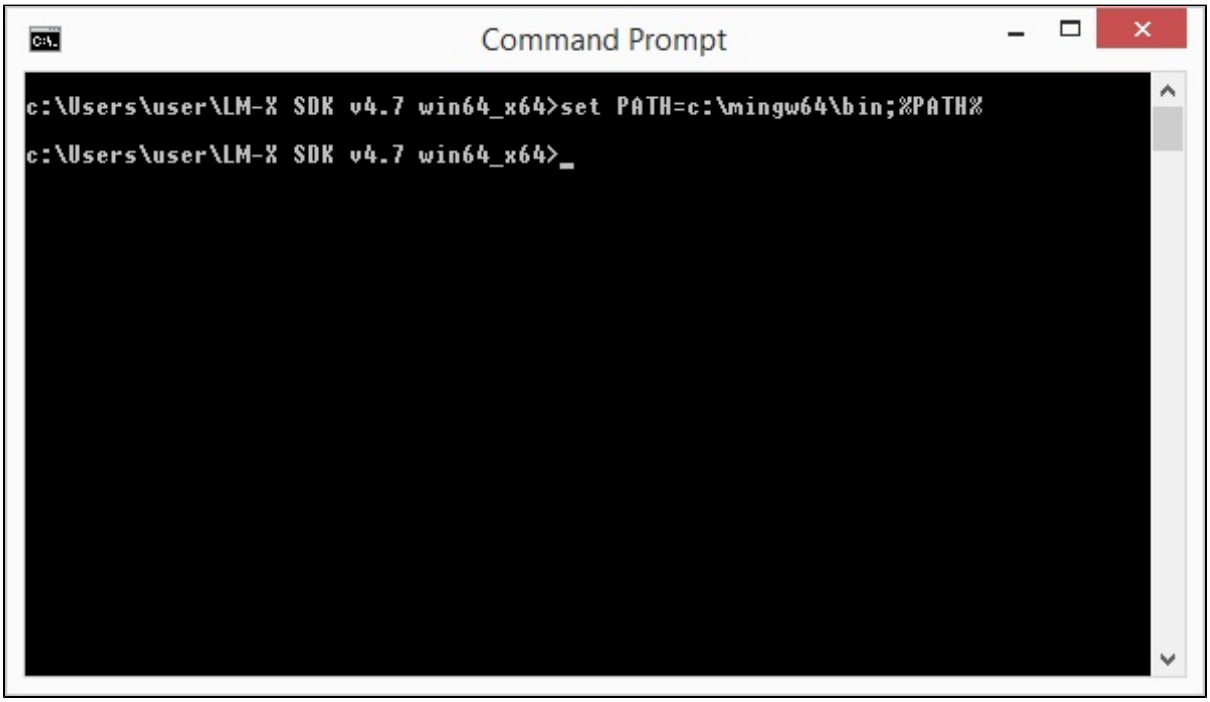
Compiling the LM-X SDK using MinGW

When compiling the LM-X SDK using MinGW, make sure Cygwin is installed on your machine. Also, please consider MinGW-specific requirements and limitations, as described in Supported platforms.

To compile the LM-X SDK using MinGW:

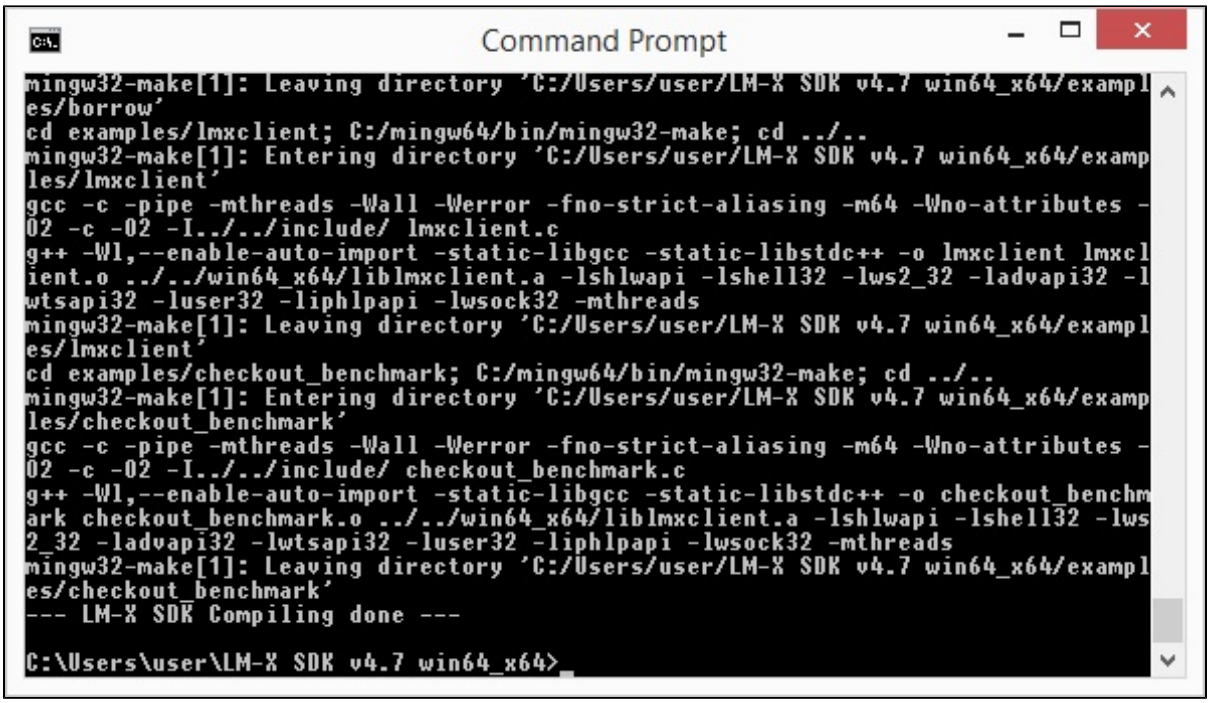
Step 1. Run cmd.exe.

Step 2. Set the environment variable.



Note: The path for mingw32-make may vary depending on the machine and architecture being used.

Step 3. From the root directory of the LM-X distribution, run mingw32-make.



Cleaning the LM-X SDK using MinGW

```
C:\Users\user\LM-X SDK v4.5.8 win64_x64> mingw32-make clean
```

Note: You may want to clean previously compiled files when rebuilding the SDK with a different license or security key.

See Installation issues for information about problems and workarounds related to compiling the LM-X SDK.