*These are the bare necessary instructions for installation and testing the C library contained in the ExportDCT folder. All instructions concern 64-bit systems. Attempt 32-bit installation at your own peril.*

*This project was mainly built to be run on Linux (Ubuntu). I have managed to compile it in Windows 7 x64 with libjpeg-8d and include the dll file in the folder, but cannot promise it will work, or that I can reproduce the exact process.*

**Ubuntu installation instructions:**

**libJPEG8d:**

The libjpeg8 (libjpeg8-dev) package is needed for various Ubuntu applications and is most likely already included in your distribution. If not, install using apt-get. Make sure LD\_LIBRARY\_PATH includes libjpeg8. You can always install libjpeg8 from source using the folder provided, if you are confident you know what you are doing.

**ExportDCT:**

Build **libExportDCT.so** by running:

gcc -c -Wall -fpic ExportDCT.c

gcc -shared -o libExportDCT.so ExportDCT.o -l:libjpeg.so.8

and copy it somewhere in your LD\_LIBRARY\_PATH. Don't forget to check the read permissions.

You can test native C functionality independently from trying Java, by compiling the provided TestDCT.c code. First run:

gcc -o TestDCT TestDCT.c -lExportDCT -ljpeg

make ./TestDCT executable and then run it, having **demo.jpg** in the folder. The output should be:

40

30

1016

*If yes, then the C code is working correctly and you are ready to proceed to Java/JNA integration.*

**Windows installation instructions:**

Before trying anything else, test if the precompiled DLL file works. Be always wary that another version of libjpeg may be somewhere in the path, as certain programs carry their own version of the library.

If you need to compile from scratch, proceed as follows:

**libJPEG8d:**

-Download libjpeg8d from the Independent JPEG group (<http://www.ijg.org/files/>)

-Use an 64-bit version of MSYS (MSYS64) to compile libjpeg:

a) run ./configure from within the **jpeg-8d\_distribution** folder.

b) edit the generated **jconfig.h** file by replacing the line:

#undef HAVE\_PROTOTYPES

with

#define HAVE\_PROTOTYPES

(this file is replaced during configuration, do the manual change after completing the configuration)

c) ./make and then ./make install

d) enter the **/.libs** subdirectory and take **libjpeg-8.dll**

**ExportDCT:**

-Copy **libjpeg-8.dll** and **jconfig.h** to the **Auxiliary\_Sources\ExportDCT\** subdirectory.

-Go to the **ExportDCT**\ subdirectory and run (in MSYS64):

gcc -shared -o ExportDCT.dll ExportDCT.c libjpeg-8.dll -Wl,--output-def,ExportDCT.def,--out-implib,libExportDCT.a

For testing that the native C code works independently from Java, a test file is included in the distribution. You can build **TestDCT.c** with:

gcc –o TestDCT.exe TestDCT.c ExportDCT.dll libjpeg-8.dll (in MSYS64)

Then run **TestDCT.exe**, having **demo.jpg** in the folder. The output should be:

40

30

1016

If yes, then the C code is working correctly and you are ready to proceed to Java/JNA integration.