Here is the correct procedure for setting up the current (as of 7/2/13) GNU bignum libraries with Eclipse CDT, MinGW, and msys for C++. To get through this, you should have used Unix or Linux before, as well as Windows, and you should have a vague recollection of programming and compiling programs. This is the culmination of over a week of research and hardcore frustration, so if I messed something up note it politely or I will blow you up with the power of my mind!

1) I assume you have already downloaded and installed Eclipse and MinGW and have installed msys into MinGW. You must install MinGW before msys!

2) Download the tarball for the GMP libraries from gmplib.org to ${gmp\_download}. I downloaded the gmp-5.1.2.tar.xz because I have never used lzip and didn't know if it was available in msys.

3) Open up an msys window (essentially a bash shell). cd ${gmp\_build} and tar -Jxvf ${gmp\_download}/gmp-x.x.x.tar.xz

Those tar options are different from what you may find elsewhere on the web! -Jxvf is right for xz (and I think lzip), but for gzip you use -xzvf.

4) cd gmp-x.x.x and run ./config.guess. Write down the output. You will need it next.

5) Run ./configure --prefix=${gmp\_build} --build=<output of config.guess> --enable-cxx --with-gnu-ld

According to a helpful person online, if you don't explicitly tell GMP to build for your platform it builds everything, which is bad. The cxx option builds the C++ libraries and --with-gnu-ld allows it to work with ld. Pretty straightforward.

6) make

7) make install

EX: suppose you installed to C:/gmp. You should have gmp/include/gmp.h and gmpxx.h. You should also have gmp/lib/libgmp.a, libgmp.la, libgmpxx.a, libgmpxx.la. You should also have a share directory with stuff in it.

8) Set up eclipse:

* Go to project --> properties
* Under C/C++ build --> Environment edit the PATH variable and add ${gmp\_build}/include;${gmp\_build}/lib
* Under C/C++ build --> settings --> tool settings --> GCC Assembler --> general add ${gmp\_build}/include as an include path.
* Same place but --> GCC C++ compiler --> Includes add ${gmp\_build}/include as an include path.
* Same place --> GCC C++ compiler --> Miscellaneous add -lgmp -lgmpxx to the END of the line. THE END OF THE LINE!
* Same place --> GCC C compiler Add the same include paths and miscellaneous options as before.
* Same place --> MinGW C++ linker --> Libraries Add to the "Libraries (-l)" both **gmpxx** and **gmp** IN THAT ORDER! Now add ${gmp\_build}/lib to "Library Search Path (-L)"
* Under C/C++ General --> Paths & Symbols --> Incudes Tab check that you have ${gmp\_build}/include in your include directories for Assembly, C, and C++. If they aren't there add them in.
* Same place --> Libraries Tab check that you have gmp and gmpxx IN THAT ORDER. It should already be populated.
* Same Place --> Library Paths Tab Check for ${gmp\_build}/lib which should already be there.
* Hit "Apply" and make sure you rebuild the index or the changes won't take. Hit OK to close out.

9) Run this short program to verify your setup:

 #include <math.h>

 #include <stdio.h>

 #include <stdlib.h>

 #include <iostream>

 #include <gmp.h>

 #include <gmpxx.h>

 using namespace std;

 int main ()

 {

 mpz\_t p;

 mpz\_init\_set\_ui (p,3);

 return 0;

 }

Your compile commands should look similar to this:

g++ "-IC:\gmp\include" -O0 -g3 -Wall -c -fmessage-length=0 -lgmp -lgmpxx -o main.o "..\main.cpp" g++ "-LC:\gmp\lib" -o GMPDebug.exe main.o -lgmp -lgmpxx

Notes:

1) The order of the options is important. I don't know all of the whys, but if the second command line (which links the program) has the -lgmp -lgmpxx flags before the -o option, the linking will fail miserably.

2) The -l flag is a tricky one. It actually says "Go look in -L for liblibrary.a". In this case "Go look in C:\gmp\lib for libgmp.a and libgmpxx.a".

3) I have heard of bugs involving cout and the 64 bit version of eclipse, so I am using the 32 bit version, where I am seeing the same bug. ☹

4) If you put gmp before gmpxx in the linker, the overloaded operators won’t work.