2.8 Build

To build CEAL use the command *make*. Arguments are presented in the table below:

Name	Description
CELLDEF=n	defines the type of memory cell to build in (see 2.1, Memcell)
PLAT=msc	use: MS cl C++ compiler, Windows
PLAT=unx	use: GCC g++ C++ compiler, Windows and Unix
MEMDEF=n	defines the type of memory (see 2.1, Memory)
GMP=1	See 2.9

2.9 Building with GMP bignum library

CEAL comes with its own portable bignum library set by default to 4096-bit precision arithmetic. This built-in library is portable and simple, at the same time may not be efficient. To get more efficiency, especially working with larger N's, one can connect defacto standard bignum library GMP. There are two downsides:

1) A dependency on a 3rd party tool; and
2) GMP can be used only in Unix-like systems (Cygwin with GCC also works)

To build CEAL with GMP. Do the following.

First, try to build test programs linking GMP – goto 'unumber\test_gmp' and run 'run.sh'. If it builds 2 executables, then GMP is installed. If not, then:

1) Test that 'm4' is installed: 'm4 –help'. If not:
    a. Unpack 'm4', then 'sh configure', then 'make'
    b. 'make' may give some error, but it may still have done the job
    c. copy executable from './src/m4' to '/bin'
    d. test that m4 works
2) Unpack GMP and configure with C++ 'sh configure --enable-cxx'
3) 'make'
4) Copy 'gmp.h' and 'gmpxx.h' to include directory (can be /usr/include)
5) Directory '.libs' must have 'libgmp.a' and 'libgmpxx.a'.
6) Copy content of '.libs' to lib directory (can be '/lib' or '/usr/lib')

If you have build GMP as described above try to run 'run.sh' again and make sure it builds. If not, seek professional help.

Next, we build CEAL with GMP:

```
'make PLAT=unx GMP=1'
```