<http://blog.csdn.net/lishuai369/article/details/6455382>

一、Boost  
Boost库是一个可移植、提供源代码的C++库，作为标准库的后备，是C++标准化进程的发动机之一。其官方网站为<http://www.boost.org/>，目前最新版本为1.46.0，以下以此版本为准。  
Boost库大部分的子库都是C++模版提供，大部分情况下直接包含头文件就可以了。其中部分子库需要编译，如下图所示：  
  
上图中bjam.exe是boost提供的编译工具，下面会介绍如何获得。

二、Mingw: Minimalist GNU for Windows  
官方网站：<http://www.mingw.org/> 部分同学可能和我一样，使用自带Mingw环境的IDE，比如Code::Blocks，其官方网站：<http://www.codeblocks.org/>。请确保Mingw工具链目录添加到系统的环境变量PATH中，比如我路径为："D:/Program Files/CodeBlocks/MinGW/bin"，并将这个目录下的mingw32-make.exe复制并重命名为make.exe。然后验证gcc等能否正常使用：开始->运行->输入"cmd"回车->命令窗口输出"gcc -v"查看gcc版本信息，如果看到如下类似输出，那么就可以继续正是编译Boost了。  
  
  
三、bjam.exe  
bjam是boost自带的工具，用来方便地编译boost，当然也可以用来编译你自己的工程。bjam在boost源码中一起提供，需要编译才能使用，下载boost的时候会提示下载一个编译好的bjam，我不是很推荐，因为我用下载来的bjam编译boost的时候，出现一些错误，而用自己编译的bjam却顺利完成。  
在"E:/\_Libs\_/boost\_1\_46\_0/tools/build/v2/engine/src"下，使用命令"build mingw"，会在此目录下生成文件bin.ntx86/bjam.exe，将bjam.exe拷贝到boost源文件的根目录下，如"E:/\_Libs\_/boost\_1\_46\_0/"。在这个目录下的"./Jamroot"文件，详细描述了bjam的用法

四、jam配置  
bjam  --build-type=complete toolset=gcc stage  
上面的命令将编译所有需要编译的子库(除python)的各种版本，编译时间稍长，约40分钟~1小时。生成的文件在./stage/lib目录下。  
[bjam的详细配置boost管网上有介绍](http://www.boost.org/boost-build2/doc/html/bbv2/overview/invocation.html#bbv2.overview.invocation.examples)