如何在Visual Studio 2005編譯boost 1.34.1? (C/C++)

**1.下載boost source**  
到<http://www.boost.org>下載最新版本的boost，我目前下載的是1.34.1，將之解壓縮到c:\boost\_1\_34\_1\下。  
  
**2.編譯bjam**利用Visual Studio 2005 Command Prompt開啟DOS視窗，將目錄cd到C:\boost\_1\_34\_1\tools\jam\src下，執行build.bat，然後會在C:\boost\_1\_34\_1\tools\jam\src\bin.ntx86\產生bjam.exe，將bjam.exe複製到c:\boost\_1\_34\_1\下。  
  
**3.設定編譯環境**在VC8出現的warning，主要是以下幾類[4]  
1.C4819 : 代碼中有cp950無法顯示的字元。[1][4]  
2.VC8特有的的safe\_code技術。[3]  
  
修改user-config.jam (C:\boost\_1\_34\_1\tools\build\v2\user-config.jam) 的MSVC configuration

http://kb.cnblogs.com/Images/OutliningIndicators/None.gif#  MSVC configuration  
http://kb.cnblogs.com/Images/OutliningIndicators/None.gif  
http://kb.cnblogs.com/Images/OutliningIndicators/None.gif#  Configure msvc (default version, searched in standard location  
http://kb.cnblogs.com/Images/OutliningIndicators/None.gif#  and PATH).  
http://kb.cnblogs.com/Images/OutliningIndicators/None.gif#  using msvc ;  
http://kb.cnblogs.com/Images/OutliningIndicators/None.gifusing msvc : 8.0 : : <compileflags>/wd4819 <compileflags>/D\_CRT\_SECURE\_NO\_DEPRECATE <compileflags>/D\_SCL\_SECURE\_NO\_DEPRECATE <compileflags>/D\_SECURE\_SCL=0 ;

注：宏解释[3]  
    \_CRT\_SECURE\_NO\_DEPRECATE和\_SCL\_SECURE\_NO\_DEPRECATE用于关闭safe code代码警告；  
    \_SECURE\_SCL用于控制是否采用safe code对STL边界进行检查。   
  
或下載[user-config.jam](http://files.cnblogs.com/oomusou/user-config.zip)覆蓋之。  
  
**4.編譯boost**  
將目錄移至c:\boost\_1\_34\_1\下執行

http://kb.cnblogs.com/Images/OutliningIndicators/None.gifbjam --without-python --toolset=msvc-8.0 --prefix=c:\boost install

參數說明  
--without-python 表示不使用 python  
--toolset : 所使用compiler，Visual Studio 2005為msvc-8.0  
--prefix：指定編譯後library的安裝目錄  
  
**5.設定Visual Studio 2005環境**Tools -> Options -> Projects and Solutions -> VC++ Directories  
在Library files加上c:\boost\lib  
在Include files加上c:\boost\include\boost-1\_34\_1

**6.測試boost是否設定成功**

http://kb.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://kb.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gif/\*\*//\*   
http://kb.cnblogs.com/Images/OutliningIndicators/InBlock.gif(C) OOMusou 2007 http://oomusou.cnblogs.com  
http://kb.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://kb.cnblogs.com/Images/OutliningIndicators/InBlock.gifFilename    : boost\_StringTrim.cpp  
http://kb.cnblogs.com/Images/OutliningIndicators/InBlock.gifCompiler    : Visual C++ 8.0 / ISO C++ (boost)  
http://kb.cnblogs.com/Images/OutliningIndicators/InBlock.gifDescription : Demo how to boost to trim string  
http://kb.cnblogs.com/Images/OutliningIndicators/InBlock.gifRelease     : 02/22/2007 1.0  
http://kb.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif\*/  
http://kb.cnblogs.com/Images/OutliningIndicators/None.gif#include <iostream>  
http://kb.cnblogs.com/Images/OutliningIndicators/None.gif#include <string>  
http://kb.cnblogs.com/Images/OutliningIndicators/None.gif#include <boost/algorithm/string.hpp>  
http://kb.cnblogs.com/Images/OutliningIndicators/None.gif  
http://kb.cnblogs.com/Images/OutliningIndicators/None.gifusing namespace std;  
http://kb.cnblogs.com/Images/OutliningIndicators/None.gifusing namespace boost;  
http://kb.cnblogs.com/Images/OutliningIndicators/None.gif  
http://kb.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://kb.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gifint main() {  
http://kb.cnblogs.com/Images/OutliningIndicators/InBlock.gif  string s = "  Hello boost!! ";  
http://kb.cnblogs.com/Images/OutliningIndicators/InBlock.gif  trim(s);  
http://kb.cnblogs.com/Images/OutliningIndicators/InBlock.gif  cout << s << endl;  
http://kb.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}

執行結果

* Hello boost!!

[**vs2005下编译基于stlport的boost**](http://www.cppblog.com/thisisbin/archive/2008/12/09/68929.html)

1、安装stlport，参考文章：<http://www.cnblogs.com/DonLiang/archive/2007/10/13/923143.html>  
如果是DEBUG模式请在项目的属性=>配置属性=>c/c++=>命令行的附加选项增加/D\_STLP\_DEBUG  
  
2、boost\_1\_37\_0\tools\jam\stage下运行build.bat编译生成bjam.exe并拷贝到boost根目录  
  
3、修改配置文件boost\_1\_37\_0\tools\build\v2\user-config.jam，将两个stlport选项打开  
using stlport : 5.2.1 : "D:\\work\\VS\\Projects\\STLport-5.2.1\\stlport" : "D:\\work\\VS\\Projects\\STLport-5.2.1\\lib" ;  
4、vs2005命令提示下编译命令：bjam stdlib=stlport --build-type=complete msvc-8.0 stage，接下来是漫长的等待，大概两个小时以上吧

注意编译debug 版本设置工程-〉c++ -> 命令行附加选项加入 /GX  /D\_STLP\_DEBUG。代码生成-〉运行时库必须选择多线程版本的

编译BOOST和STLPORT

对于vs2003.net，如果只安装boost 相对比较简单，可以跳到步骤2，这里要安装boost 的stlport 版本。即同时安装stlport 和 以stlport 为标准库编译 boost .  
1、先安装stlport  
进入stlport\src 目录  
运行nmake –f vc71.mak install  
这样默认stlport头文件放入 ....\Microsoft Visual Studio .NET 2003\Vc7\include\stlport 目录  
在vc 的工具-〉选项-〉项目-〉vc++目录 中设置stlport头文件目录。  
注意目录位置要放在vc 头文件链接的上面，这样#include 标准库是优先选择了stlport 而不是 vc 自带的stl .  
lib 自动拷贝到了vc7/lib 中可以不用设置。  
  
对于5.02 版本，src目录中没有makefile 文件, 有cmd控制台进入build/ lib 目录,执行完 configure.bat 然后再对vc71.mak 进行 make  
  
2、编译boost,  先编译bjam：  
a 运行tools\build\jam\_src\build.bat，然后会看见 bin.ntx86目录，里面有bjam.exe  
  
b设置环境变量 如:  PATH= e:\boost1.32.0\boost\tools\build\jam\_src\bin.ntx86  
 或者把bjam 拷贝到boost 根目录  
  
SET VC71\_ROOT="C:\Program Files\Microsoft Visual Studio .NET 2003\Vc7"  
这个也可以不用设置。  
  
使用vc提供的工具 visual studio.net 2003 命令提示控制台,进入boost目录：执行  
bjam  "-sTOOLS=vc-7\_1 " install   
  
便可把相关的头文件和编译后的库文件默认安装到c:\boost。如果不使用stlport 对于vc2003安装就这么简单，  
  
  
对于支持stlport 的boost 安装执行  
bjam "-sTOOLS=vc-7\_1-stlport" "-sSTLPORT\_PATH=E:\STLPort

"-sSTLPORT\_VERSION=4.6.2" stage

我安装的stlport是4.6.2版本的，注意上面指定的stlport目录和版本，在编译过程中会出错但能编译完。  
如：F:\boost\boost\_1\_32\_0\boost\lexical\_cast.hpp(150) : error C2679:  
还有其他一些原因不能编译出来的库。  
大部分安装都会有这个问题，可以添加 "-sBUILD=<native-wchar\_t>off" 选项关闭编译时的/zc:wchar\_t选项  
bjam "-sTOOLS=vc-7\_1-stlport" "-sBUILD=<native-wchar\_t>off"  "-sSTLPORT\_PATH=E:\STLPort" "-sSTLPORT\_VERSION=4.6.2"  stage  
注意按照上面 stlport  lib 的目录是在 E:\STLPort\stlport-4.6.2\lib ,后面数字是版本号指定的目录中的数字  
  
编译后在boost 根目录出现一个stage 目录，里面是所有编译过的库。我的stlport 版本有158个文件，vc 200多一些，可见stlport 有些库没出来，多半是跟wchar\_t有关的。  
  
完成后可以测试下面的程序  
#include "stdafx.h"  
#include <iostream>  
#include <boost/regex.hpp>  
#include <boost/thread.hpp>  
  
int main()   
{    
    // 3 digits, a word, any character, 2 digits or "N/A",    
    // a space, then the first word again   
    boost::regex reg("\\d{3}([a-zA-Z]+).(\\d{2}|N/A)\\s\\1");      
    std::string correct="123Hello N/A Hello";    
    std::string incorrect="123Hello 12 hello";      
    assert(boost::regex\_match(correct,reg)==true);    
    assert(boost::regex\_match(incorrect,reg)==false);  
    boost::regex reg1("(new)|(delete)");  
    boost::smatch m;std::string s=  "Calls to new must be followed by delete. \  Calling simply new results in a leak!";  
    if (boost::regex\_search(s,m,reg1))   
    {  // Did new match?    
        if (m[1].matched)      
            std::cout << "The expression (new) matched!\n";    
        if (m[2].matched)      
            std::cout << "The expression (delete) matched!\n";  
    }  
    return 0;  
}  
  
注意编译debug 版本设置工程-〉c++ -> 命令行附加选项加入 /GX  /D\_STLP\_DEBUG。代码生成-〉运行时库必须选择多线程版本的, 没有 /D\_STLP\_DEBUG  编译时回停留在这里  
elif defined(\_DEBUG)  
#            define BOOST\_LIB\_RT\_OPT "-gdp"  
#            pragma message("warning: STLPort debug versions are built with /D\_STLP\_DEBUG=1")  
#            error "Build options aren't compatible with pre-built libraries"  
#        else  
#            define BOOST\_LIB\_RT\_OPT "-p"  
  
如果没有指定使用多线程版本，编译会提示你缺少libboost\_regex-vc71-sgdp-1\_xx\_1.lib 的库，其实默认编译没有这个库。可能指定一些编译参数会有吧. 如果不指定多线程的话。stlport 的库也有链接问题。这些库基本都是多线程版本的（不代表线程安全）。最好指定为多线程动态库 /md 或 /mdd 版本。静态库编译出来太大了  
  
对于test 库，程序能编译但不能链接，对于vc自带stl 与stl port 都一样，不知道哪里需要设置。里面有些文件直接在vc工程中都不能编译，可能版本还不是太好，错误如下 test\_tools::tt\_detail::check\_impl  , 搜搜网上存在这样问题的不少。这样BOOST\_CHECK 之类的宏就不能使用了  
error LNK2019: 无法解析的外部符号 "void \_\_cdecl boost::test\_tools::tt\_detail::check\_impl(class boost::test\_tools::predicate\_result const &,class boost::basic\_wrap\_stringstream<char> &,class boost::unit\_test::basic\_cstring<char const >,unsigned int,enum boost::test\_tools::tt\_detail::tool\_level,enum boost::test\_tools::tt\_detail::check\_type,unsigned int,...)" (?check\_impl@tt\_detail@test\_tools@boost@@YAXABVpredicate\_result@23@AAV?$basic\_wrap\_stringstream  
@D@3@V?$basic\_cstring@$$CBD@unit\_test@3@IW4tool\_level@123@W4check\_type@123@IZZ)

**1.下载boost source**  
到[http://www.boost.org](http://www.boost.org/)下载最新版本的boost，我目前下载的是1.34.1，将之解压到c:\boost\_1\_34\_1\  
  
**2.编译bjam**C:\boost\_1\_34\_1\tools\jam\src下，执行build.bat，然后会在C:\boost\_1\_34\_1\tools\jam\src\bin.ntx86\生成一个bjam.exe，将bjam.exe复制到c:\boost\_1\_34\_1\下。  我这不能编译。也可以下载一个  
  
**3.设定编译环境**

修改user-config.jam (C:\boost\_1\_34\_1\tools\build\v2\user-config.jam) 的MSVC configuration

A:vs2003.net

http://images.csdn.net/syntaxhighlighting/OutliningIndicators/None.gifusing msvc : 7.1 ;

 B:vs2005.net

http://images.csdn.net/syntaxhighlighting/OutliningIndicators/None.gifusing msvc : 8.0 : : <compileflags>/wd4819 <compileflags>/D\_CRT\_SECURE\_NO\_DEPRECATE <compileflags>/D\_SCL\_SECURE\_NO\_DEPRECATE <compileflags>/D\_SECURE\_SCL=0 ;

 在VC8.0出现的warning，主要是以下2类

a.C4819 : 代码中cp950无法显示的字元.

b.VC8.0特有的的safe\_code技术.  
  
  
宏解释:  
    \_CRT\_SECURE\_NO\_DEPRECATE和\_SCL\_SECURE\_NO\_DEPRECATE用于关闭safe code代码警告，    \_SECURE\_SCL用于控制是否用safe code对STL边界进行检查。

**C: 加上　python , 目前我用的是2.5版**

http://images.csdn.net/syntaxhighlighting/OutliningIndicators/None.gifusing python : 2.5 ;  
http://images.csdn.net/syntaxhighlighting/OutliningIndicators/None.gif

**4:下载bzip2-1.0.4    zlib-1.2.3,   icu4c-3.6:**

bzip2-1.0.4  下载:<http://www.bzip.org/>

zlib-1.2.3 下载: <http://www.zlib.net/>

icu4c-3.6下载:  <http://www.icu-project.org/>

**5:下载python2.5 , 安装到c:\**

python2.5 下载: [**http://www.python.org/**](http://www.python.org/)

**6:写一个批处理文件，内容是：**

http://images.csdn.net/syntaxhighlighting/OutliningIndicators/None.gifSET BZIP2\_SOURCE="D:/bzip2-1.0.4"  
http://images.csdn.net/syntaxhighlighting/OutliningIndicators/None.gifSET ZLIB\_SOURCE="D:/zlib-1.2.3"  
http://images.csdn.net/syntaxhighlighting/OutliningIndicators/None.gifSET ICU\_PATH="D:/icu4c-3.6"  
http://images.csdn.net/syntaxhighlighting/OutliningIndicators/None.gifbjam --toolset=msvc-8.0 --stagedir=./lib\_x86 --builddir=./ address-model=32 link=static runtime-link=static threading=multi stage debug release  
http://images.csdn.net/syntaxhighlighting/OutliningIndicators/None.gifbjam --toolset=msvc-8.0 --stagedir=./lib\_x64 --builddir=./ address-model=64 link=static runtime-link=static threading=multi stage debug release

**或**

http://images.csdn.net/syntaxhighlighting/OutliningIndicators/None.gifSET BZIP2\_SOURCE="D:/bzip2-1.0.4"  
http://images.csdn.net/syntaxhighlighting/OutliningIndicators/None.gifSET ZLIB\_SOURCE="D:/zlib-1.2.3"  
http://images.csdn.net/syntaxhighlighting/OutliningIndicators/None.gifSET ICU\_PATH="D:/icu4c-3.6"  
http://images.csdn.net/syntaxhighlighting/OutliningIndicators/None.gifbjam --toolset=msvc-8.0 --stagedir=./lib\_x86 --builddir=./ address-model=32 link=shared runtime-link=shared threading=multi stage debug release  
http://images.csdn.net/syntaxhighlighting/OutliningIndicators/None.gifbjam --toolset=msvc-8.0 --stagedir=./lib\_x64 --builddir=./ address-model=64 link=shared runtime-link=shared threading=multi stage debug release

**7.将批处理文件放到C:\boost\_1\_34\_1,  执行批处理文件**

boost\_1\_38\_0 编辑 bjam

必须先在环境变量path中指定要编译的bjam目录，不然不能使用build\_xx.bat进行编译，并提示一个诡异的错误。