

# Parser Generator 的安装与使用

潘宇

September 2019

## 目录

<b>1 Parser Generator 的安装</b>	<b>3</b>
<b>2 Parser Generator 的使用</b>	<b>3</b>
2.1 创建 parser generator 项目。 . . . . .	5
2.2 编写 Lex、Yacc 程序。 . . . . .	7
2.3 将 Lex、Yacc 程序编译为 C、C++ 程序。 . . . . .	7
2.4 创建 VS 工程项目，将 parser generator 2 生成的头文件和源 文件添加到项目中。 . . . . .	8
2.5 设置工程环境，编译运行。 . . . . .	9
2.6 PG 往年使用经验 . . . . .	14

## 1 Parser Generator 的安装

Parser Generator 的安装程序位于开发工具文件夹下的安装程序文件夹下，其中 ParGen.exe 为 Parser Generator 的安装程序，直接运行安装即可。

## 2 Parser Generator 的使用

在详细说明之前，先简单介绍 Parser Generator 2 的使用流程主要分为以下几步：

- a 创建 parser generator 项目。
- b 编写 Lex、Yacc 程序
- c 将 Lex、Yacc 程序编译为 C、C++ 或 java 程序
- d 创建 VS 工程项目，将 parser generator 2 生成的头文件和源文件添加到项目中
- e 设置工程环境，编译运行

其中前四步较为简单，最后一步较为复杂且根据 VS 版本，32/64 位，C/C++ 语言，Debug/Release 等情况的不同而方法不唯一，因此同学们需要认真处理。

王刚老师使用 Win10 专业版 Version 1067，Visual Studio 2015 的机器系统已尝试了各种组合，均成功，具体如下：

语言	调试?	位数	库目录	运行库设置	附加依赖项	其他设置
C	Debug	32	msvc2015	多线程调试 (/MTd)	ylmtd.lib	预处理器定义: YYDEBUG
C	Debug	64	msvc64	多线程调试 (/MTd)	ylmtd.lib	预处理器定义: YYDEBUG
C	Release	32	msvc2015	多线程(/MT)	ylmt.lib	
C	Release	64	msvc64	多线程(/MT)	ylmt.lib	
C++	Debug	32	msvc2015	多线程调试 (/MTd)	ylmtd.lib	预处理器定义: YYDEBUG
C++	Debug	64	msvc64	多线程调试 (/MTd)	ylmtd.lib	预处理器定义: YYDEBUG
C++	Release	32	msvc2015	多线程(/MT)	ylmt.lib	
C++	Release	64	msvc64	多线程(/MT)	ylmt.lib	
C	Debug	32	msvc32	多线程调试 (/MTd)	ylmtd.lib;legacy_stdio_definitio ns.lib; 后者是提供一些过时函数	预处理器定义: YYDEBUG 链接器-命令行-其他选项: /FORCE 后者解决 PG 单线程和 VC 多线程的重复定义
C	Release	32	msvc32	多线程(/MT)	ylmtd.lib;legacy_stdio_definitio ns.lib;	链接器-命令行-其他选项: /SAFESEH:NO

VS2015 下使用 PG 自带的库也能成功, 见表格最后两行。但只能是 C 程序, C++ 程序由于 VC 版本的问题, 无法使用 PG 自带库编译运行。也已有同学在 VS 2013、2012 上用 PG 自带的库编译成功, 也是 C 程序可以, C++ 程序不可以, 这一点大家注意。

其中 msvc32 为 PG 自带的库, msvc2015 和 msvc64 我们已提供, 位于开发工具文件夹下的 Parser Generator 编译库文件夹中, 将 msvc2015 和 msvc64 文件夹放置于 *DIR\CPP\Lib* (其中 *DIR* 为你安装 PG 的路径, 默认路径为 *C:\Program Files(x86)\Parser Generator 2*) 中, 一般均可成功使用 (即在你的机器上这两个编译好的库可以编译运行 Yacc 和 Lex 程序), 如不成功, 可参考开发工具文件夹下的 PG 库编译方法.docx 中的方法自己编译库, 其中所需要的 lbs 文件和 dll 动态链接库同样位于开发工具文件夹下的 Parser Generator 编译库文件夹中。

以上所说的整体五个步骤的大致流程可参看李鹏师兄的 CSDN 博客:

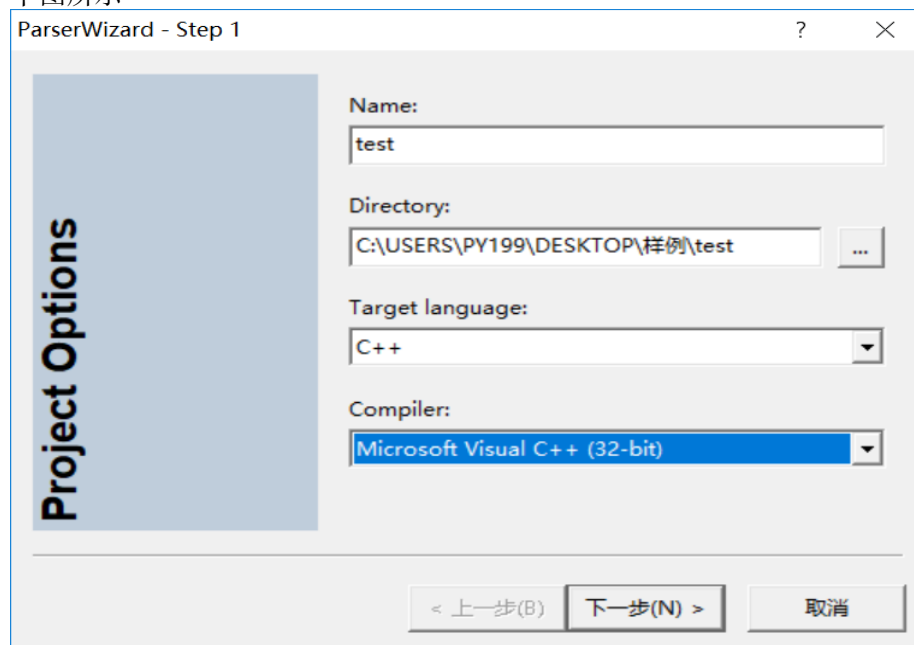
<http://blog.csdn.net/lpstudy/article/details/51330063>. 但需要注意该方案不支持 C++ 语言的文件流输入输出, 因此我这里给出一种用于 C++ 语言, VS2015/2017, 32 位, Release 模式下 (即表格第七行) 支持 c++ 语言的文件流读入写出方案作为例子进行 Parser generator 2 使用的讲解, 打算选用其他模式的各位同学可举一反三进行配置。

## 2.1 创建 parser generator 项目。

打开 Parser generator 2, 选 Project/ParserWizard 菜单项, 开始项目的创建。

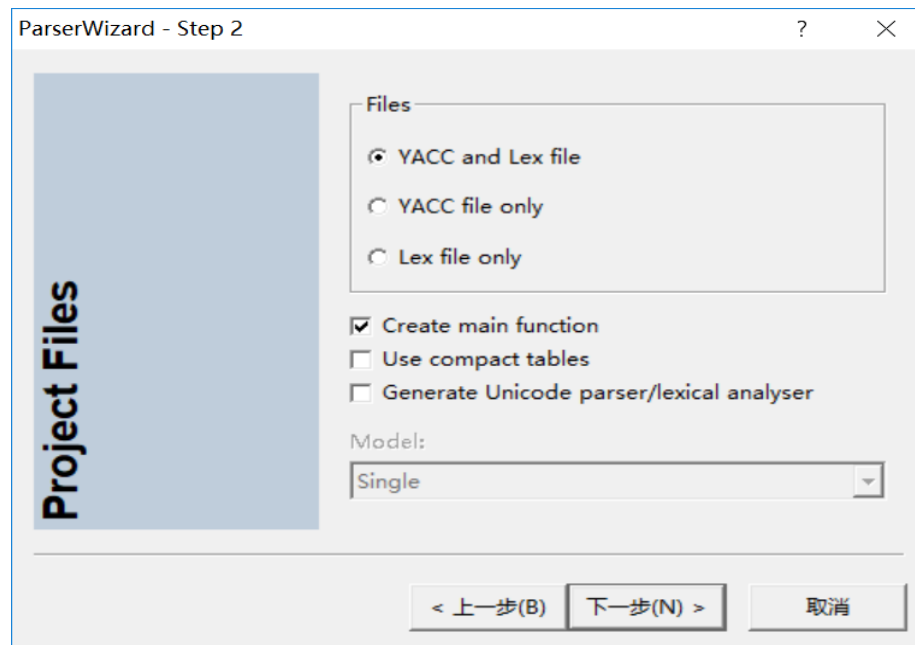
第一步: 确定项目名称、保存目录、生成分析程序语言 (C、C++...), 开发环境 (VC, BC...).

这里命名为 test, 选择目标语言为 C++, 编译器为 vc++(32-bit), 如下图所示:



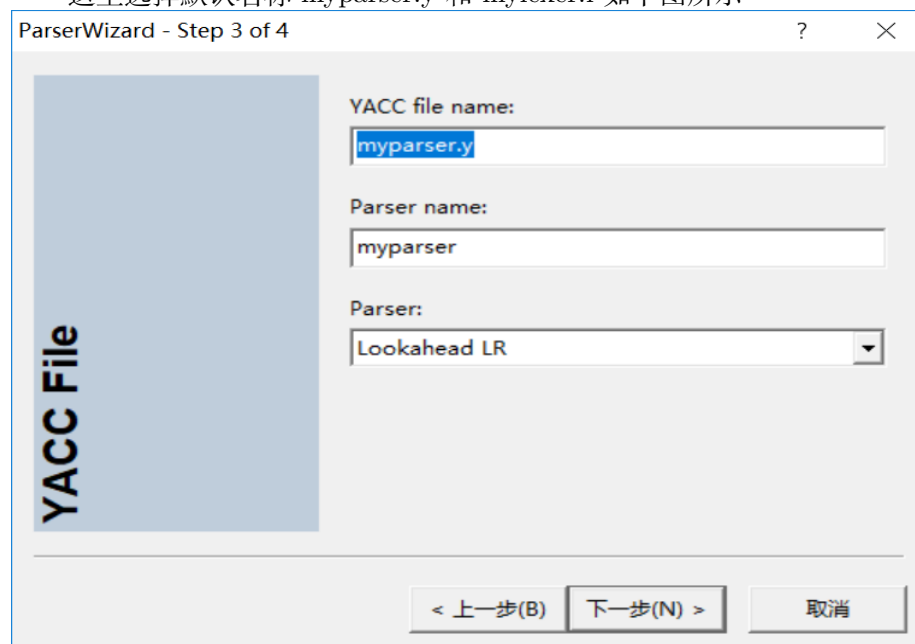
第二步: 决定是否生成 Lex 和 (或) Yacc 文件? 是否创建 main 函数? 是否使用压缩的 LR 分析表? 是否生成 Unicode 程序? single/multiple 模式?

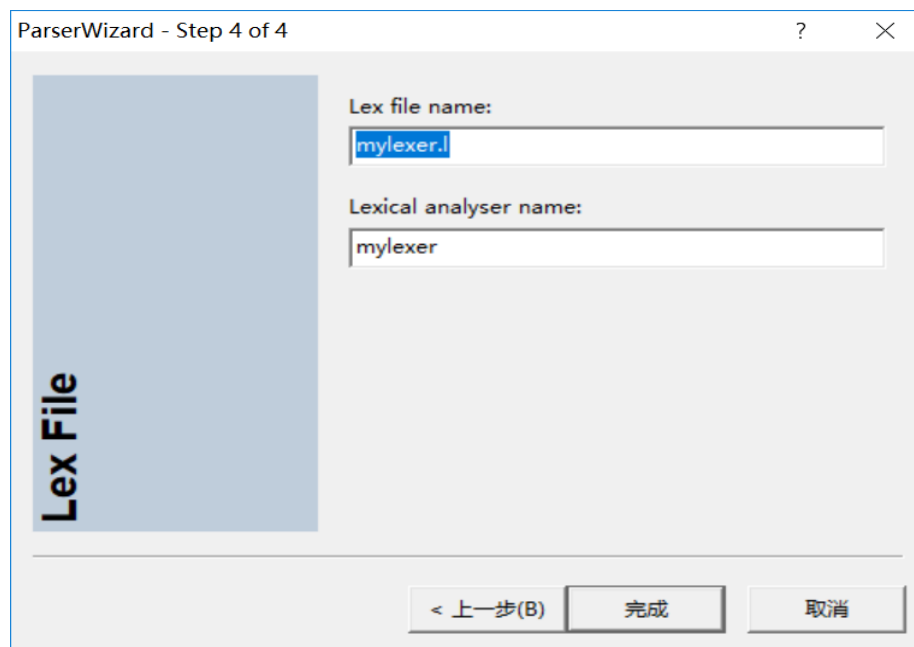
这里同时测试 Lex 和 Yacc 文件, 创建 main 函数, 如下图所示:



第三步、第四步：设置 lex、yacc 文件名——两者前缀不能相同！否则生成同名 C 文件，会产生冲突。

这里选择默认名称 myparser.y 和 mylexer.l 如下图所示：





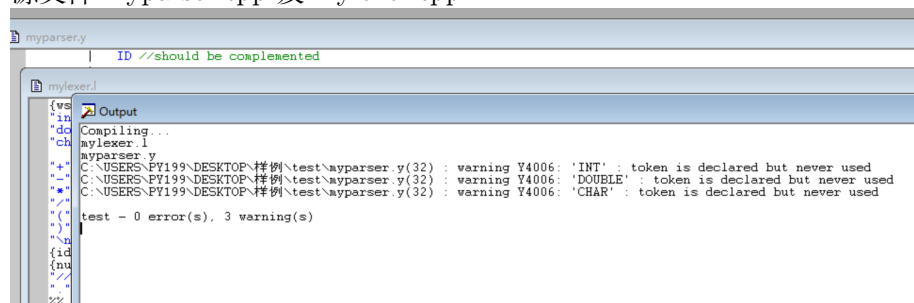
点击完成，完成项目的创建。

## 2.2 编写 Lex、Yacc 程序。










这里使用李鹏师兄的测试代码进行测试，代码已保存在开发工具文件夹下的 test 文件夹中的 myparser.y 和 mylexer.l 中，复制粘贴内容即可。

### 2.3 将 Lex、Yacc 程序编译为 C、C++ 程序。

选 Project/build 菜单项, 生成对应头文件 myparser.h 及 mylexer.h 和源文件 myparser.cpp 及 mylexer.cpp

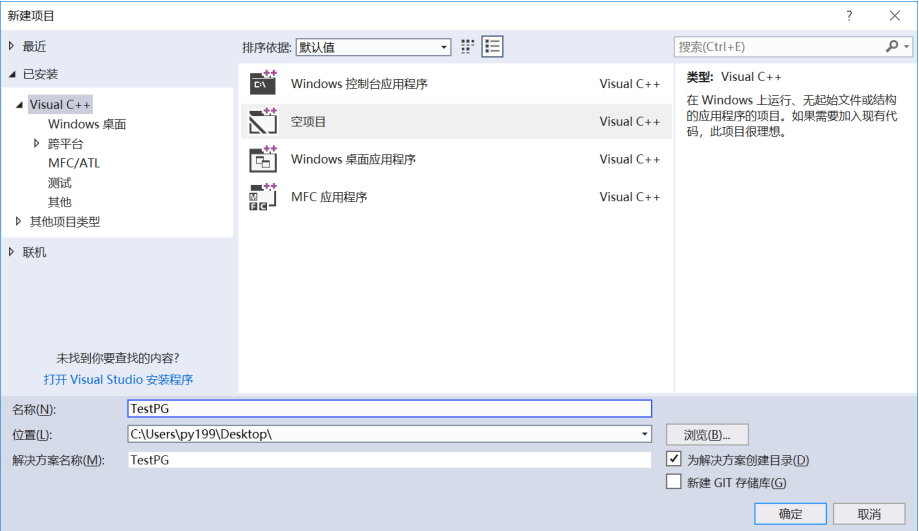


2.4 创建 VS 工程项目，将 parser generator 2 生成的头文件和源文件添加到编译原理实验指导项目中。

	mylexer.cpp	2019/9/1 23:20	C++ source file	9 KB
	myparser.cpp	2019/9/1 23:20	C++ source file	17 KB
	mylexer.l	2019/9/1 22:54	Dictionary File	2 KB
	mylexer.h	2019/9/1 23:20	Header file	2 KB
	myparser.h	2019/9/1 23:20	Header file	3 KB
	test.pgp	2019/9/1 22:53	Parser Generator Pr...	1 KB
	mylexer.v	2019/9/1 23:20	Verbose File	6 KB
	myparser.v	2019/9/1 23:20	Verbose File	5 KB
	myparser.y	2019/9/1 22:55	YACC Source File	2 KB

2.4 创建 VS 工程项目，将 parser generator 2 生成的头文件和源文件添加到项目中。

创建空项目，这里我创建的项目名称为 TestPG.



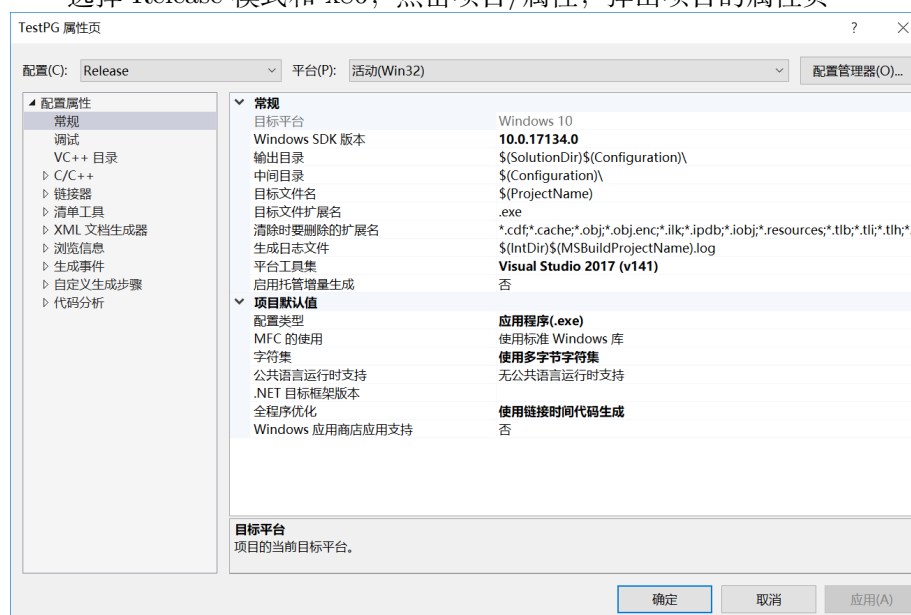
点击头文件/添加/现有项将 myparser.h 及 mylexer.h 添加入头文件, 点击源文件/添加/现有项将 myparser.cpp 及 mylexer.cpp 添加入源文件。如下图所示：





## 2.5 设置工程环境，编译运行。

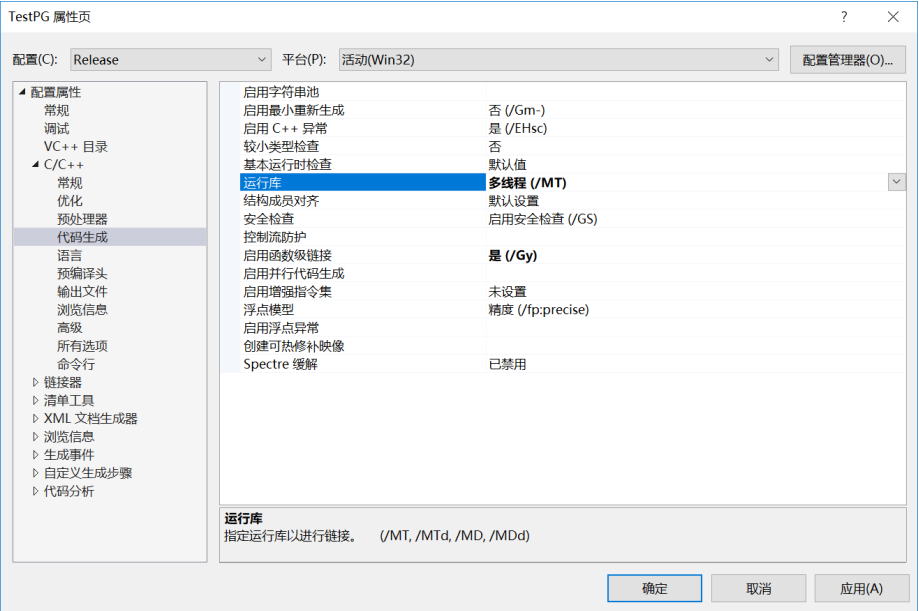
选择 Release 模式和 x86，点击项目/属性，弹出项目的属性页



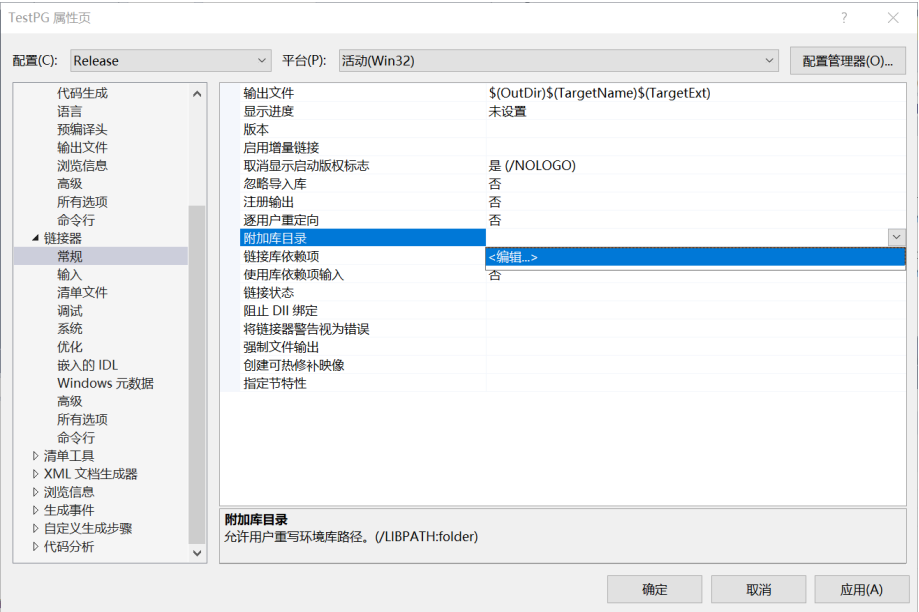
点选 C/C++ 的常规页下的附加包含目录，添加 `DIR\CPP\include` (与之前说明相同，其中 DIR 为你安装 Parser Generator 2 的路径，默认路径为 `C:\Program Files (x86)\Parser Generator 2`)

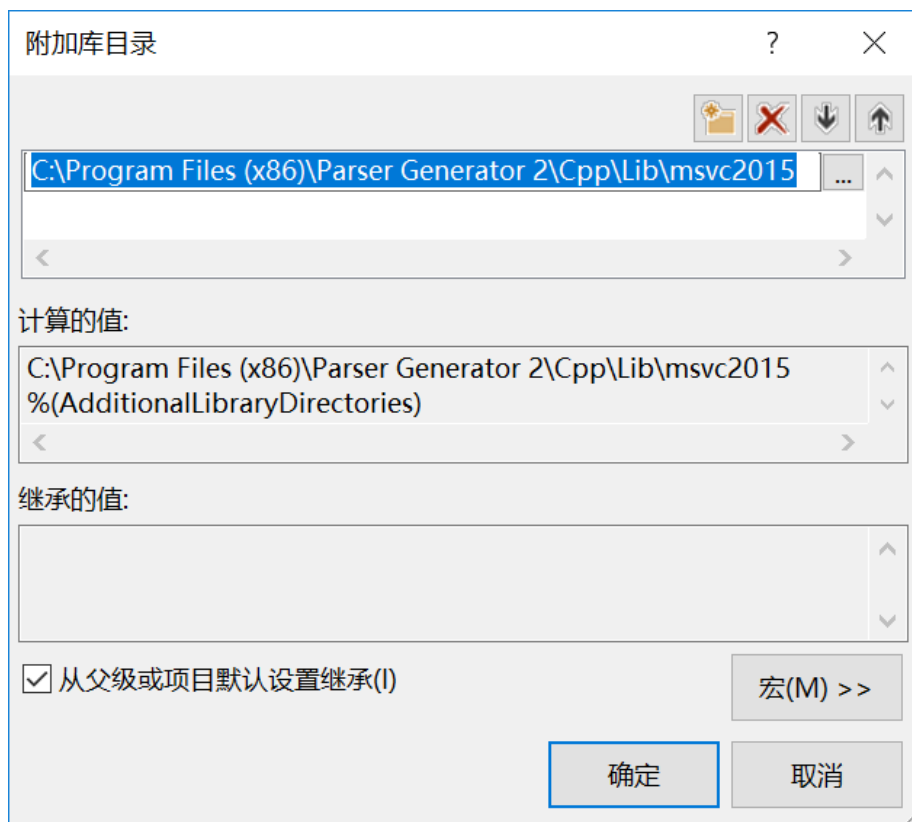


点选 C/C++ 的代码生成页，运行库选项修改为多线程 (/MT)

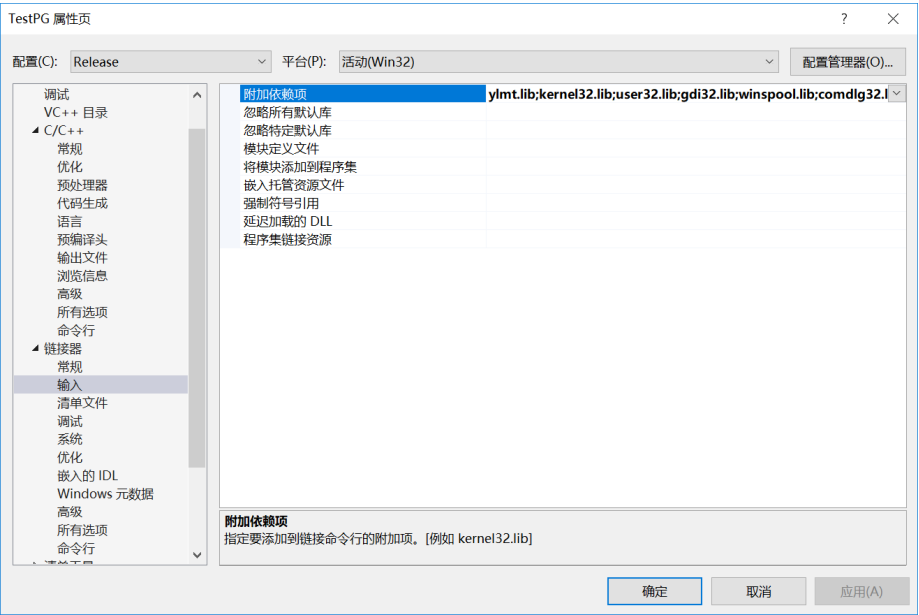


点选链接器的常规页的附加库目录，添加 *DIR\CPP\Lib\msvc2015* (与之前说明相同，其中 DIR 为你安装 Parser Generator 2 的路径，默认路径为 *C : \Program Files (x86)\Parser Generator 2*，注意不要忘记将 *msvc2015* 文件夹放置于 *DIR\CPP\Lib* 路径中)

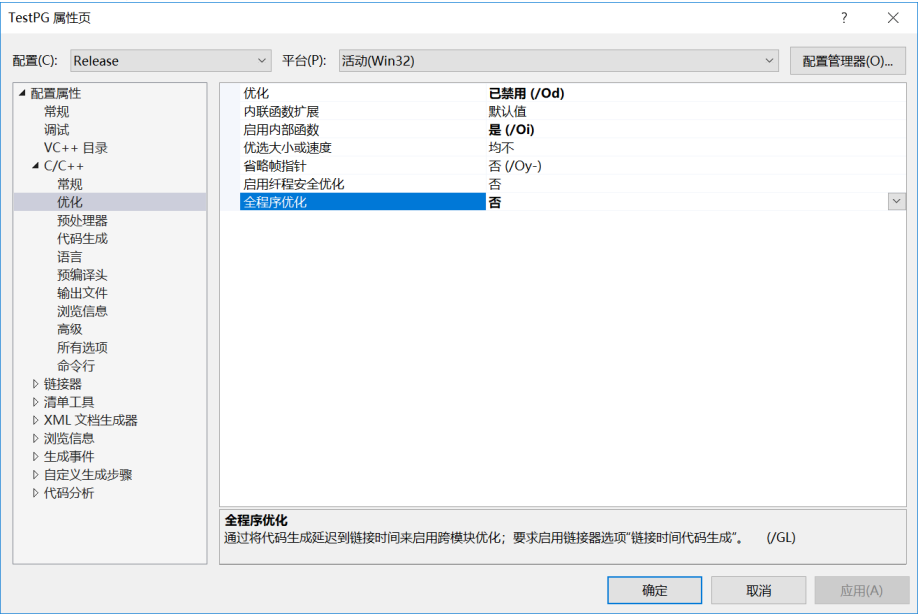




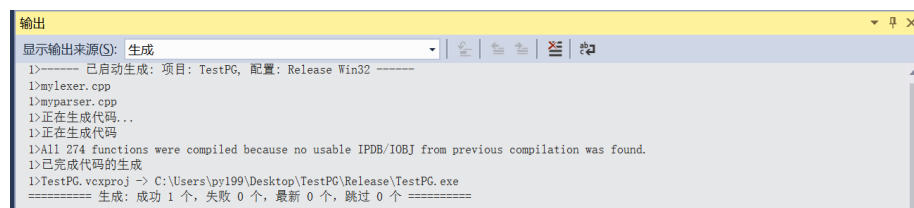
点击链接器的输入页，添加 ylmt.lib:



若遇到 Error：对象或库文件 release\...\lib 是使用比创建其他对象所用编译器旧的编译器创建的：请重新生成旧的对象和库，点击 C/C++ 的优化页将全程序优化关闭即可。

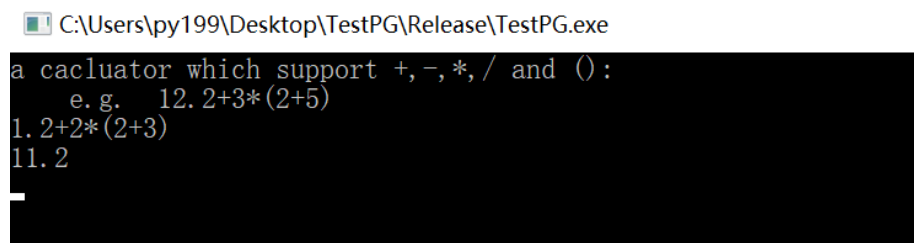


全部完成后生成解决方案：



```
输出
显示输出来源(S): 生成
1>----- 已启动生成: 项目: TestPG, 配置: Release Win32 -----
1>mylexer.cpp
1>myparser.cpp
1>正在生成代码...
1>正在生成代码
1>All 274 functions were compiled because no usable IPDB/IOBJ from previous compilation was found.
1>已完成代码的生成
1>TestPG.vcxproj -> C:\Users\py199\Desktop\TestPG\Release\TestPG.exe
====生成: 成功 1 个, 失败 0 个, 最新 0 个, 跳过 0 个====
```

成功后 ctrl+F5 运行即可, 结果成功演示如下:



```
C:\Users\py199\Desktop\TestPG\Release>TestPG.exe
a caculator which support +,-,*,/ and ():
e. g. 12. 2+3*(2+5)
1. 2+2*(2+3)
11. 2
```

## 2.6 PG 往年使用经验

### a Parser Generator 注意事项 (王艺霖):

注意代码生成-> 运行库与 Parser Generator 的 lib 一致。对于 vs2015, 如果使用 msvc32 的 lib, 则必须使用 ylmtrid.lib 和 ylmtri.lib, 而且必须使用 C++ 语言, 运行时可能会提示找不到 vc6 运行库, 从网上下载即可。对于更低版本的编译器, 有可能与 vs2015 不兼容, 这时有两个选择, 自己编译 lib, 或者使用 msvc32 的 lib 并且使用 C++ 语言。

### b 建议第一个 yacc 的.y 就选 c 32 位, 先别拷贝老师的代码, 生成.h.c 文件, 然后去 vs 里配置, 看看能不能运行, 能运行就是.y 不对, 否则就是 vs 配置的问题, 这样比较快 (冯永琦)