Win flex-bison (/p/winflexbison/)

Win flex-bison is a port Flex & Bison tools to the Windows platform

Brought to you by: lexxmark (/u/lexxmark/)

Visual Studio custom build rules

Authors: (/u/lexxmark/)

Attachments

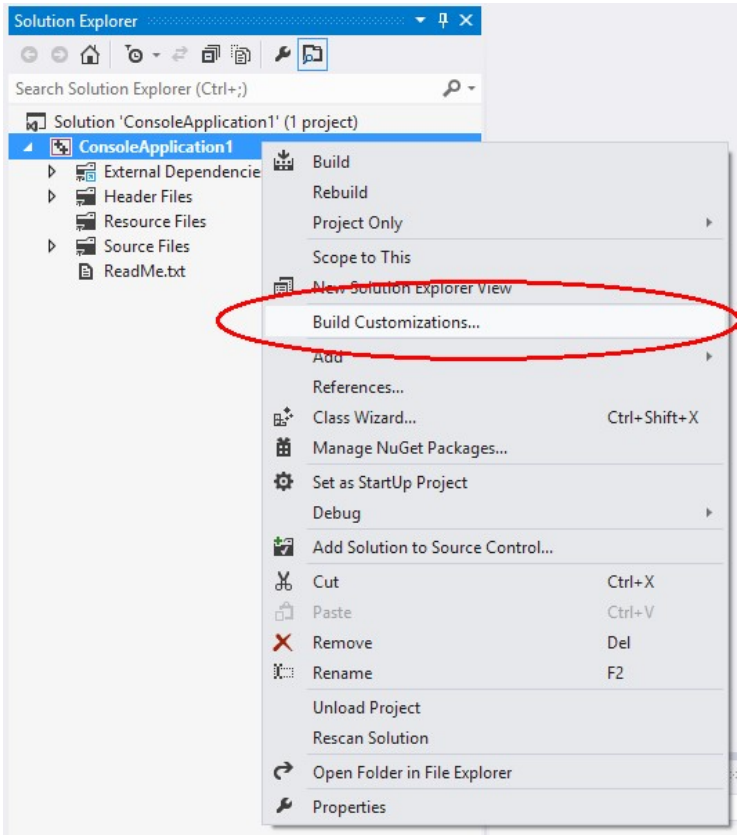
1.png (/p/winflexbison/wiki/Visual%20Studio%20custom%20build%20rules/attachment/1.png) (31654 bytes)
2.png (/p/winflexbison/wiki/Visual%20Studio%20custom%20build%20rules/attachment/2.png) (11445 bytes)
3.png (/p/winflexbison/wiki/Visual%20Studio%20custom%20build%20rules/attachment/3.png) (7316 bytes)
4.png (/p/winflexbison/wiki/Visual%20Studio%20custom%20build%20rules/attachment/4.png) (12213 bytes)
5.png (/p/winflexbison/wiki/Visual%20Studio%20custom%20build%20rules/attachment/5.png) (11534 bytes)
6.png (/p/winflexbison/wiki/Visual%20Studio%20custom%20build%20rules/attachment/6.png) (15611 bytes)
BisonProperties.png (/p/winflexbison/wiki/Visual%20Studio%20custom%20build%20rules/attachment/BisonProperties.png) (27186 bytes)
FlexProperties.png (/p/winflexbison/wiki/Visual%20Studio%20custom%20build%20rules/attachment/FlexProperties.png) (26767 bytes)
Flex_debuging.png (/p/winflexbison/wiki/Visual%20Studio%20custom%20build%20rules/attachment/Flex_debuging.png) (27219 bytes)
Properties.png (/p/winflexbison/wiki/Visual%20Studio%20custom%20build%20rules/attachment/Properties.png) (31446 bytes)
Verbosity.png (/p/winflexbison/wiki/Visual%20Studio%20custom%20build%20rules/attachment/Verbosity.png) (29722 bytes)

These steps help you setup custom build rules for Visual Studio 2010 and up.

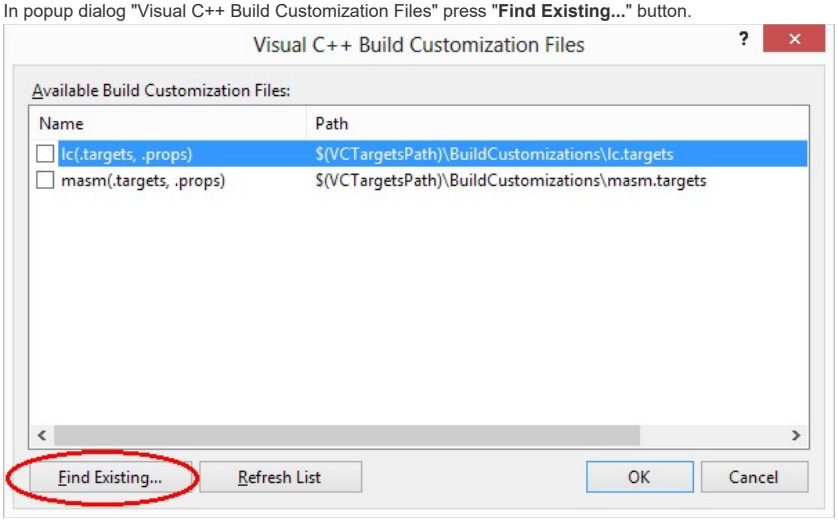
- First of all you should have 3 files:
  - **win_flex_bison_custom_build.props**
  - **win_flex_bison_custom_build.targets**
  - **win_flex_bison_custom_build.xml**

They are located in **custom_build_rules** sub-folder of the win_flex_bison archive. Or you can grab them here (http://sourceforge.net/projects/winflexbison/files/win_flex_bison_custom_build_rules.zip/download)

Launch Visual Studio and open an VC++ project. Open context menu for project item in Solution Explorer panel and select "**Build Customizations...**" menu item.

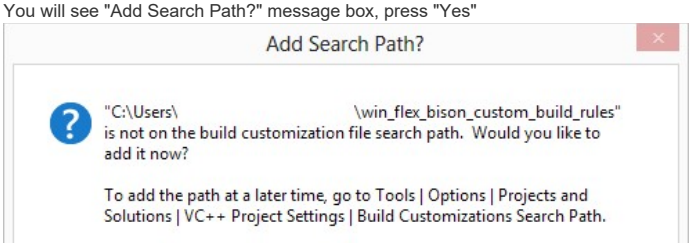


In popup dialog "Visual C++ Build Customization Files" press "**Find Existing...**" button.

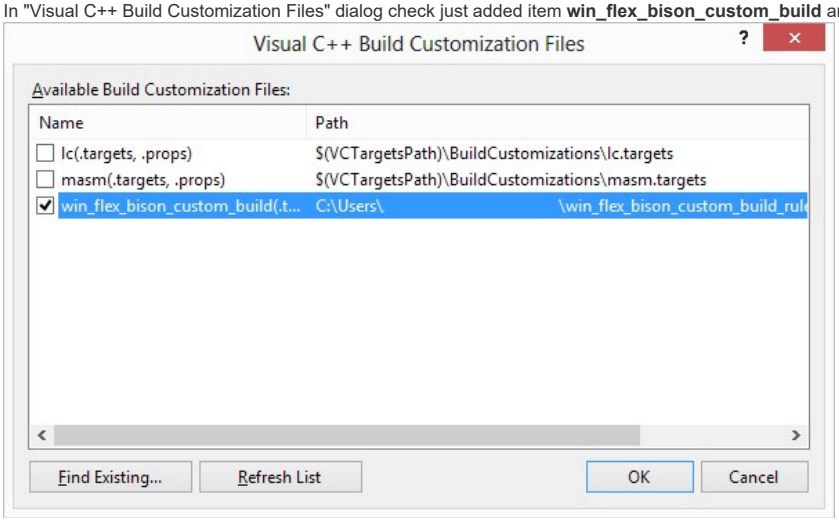


In Open File dialog select "**win_flex_bison_custom_build.targets**" file and press "Open".

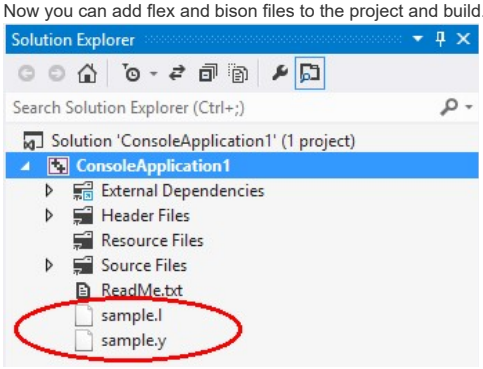
You will see "Add Search Path?" message box, press "Yes"



In "Visual C++ Build Customization Files" dialog check just added item **win_flex_bison_custom_build** and press "OK"



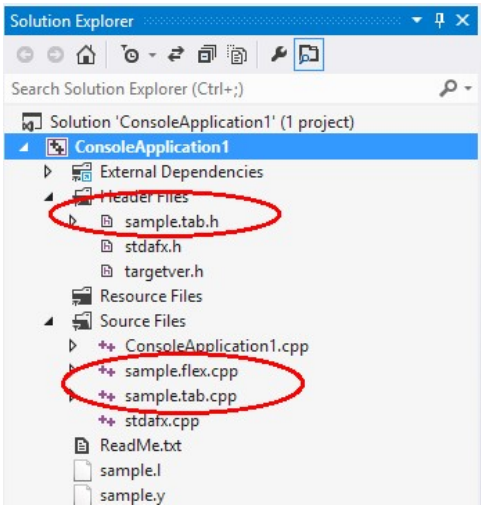
Now you can add flex and bison files to the project and build.



In build output you should see something like this:

```
1>------ Rebuild All started: Project: ConsoleApplication1, Configuration: Debug Win32 ------
1>  Process sample bison file
1>  Process sample flex file
1>  stdafx.cpp
1>  ConsoleApplication1.cpp
1>  Generating Code...
1>  ConsoleApplication1.vcxproj -> C:\Users\ConsoleApplication1\Debug\ConsoleApplication1.exe
========== Rebuild All: 1 succeeded, 0 failed, 0 skipped ==========
```

For **sample.y** bison file there are two output files: **sample.tab.h** and **sample.tab.cpp**. For **sample.l** flex file you'll got **sample.flex.cpp**. Now you can add them to the project and build. (*Don't forget to exclude cpp files from using precompiled headers*)

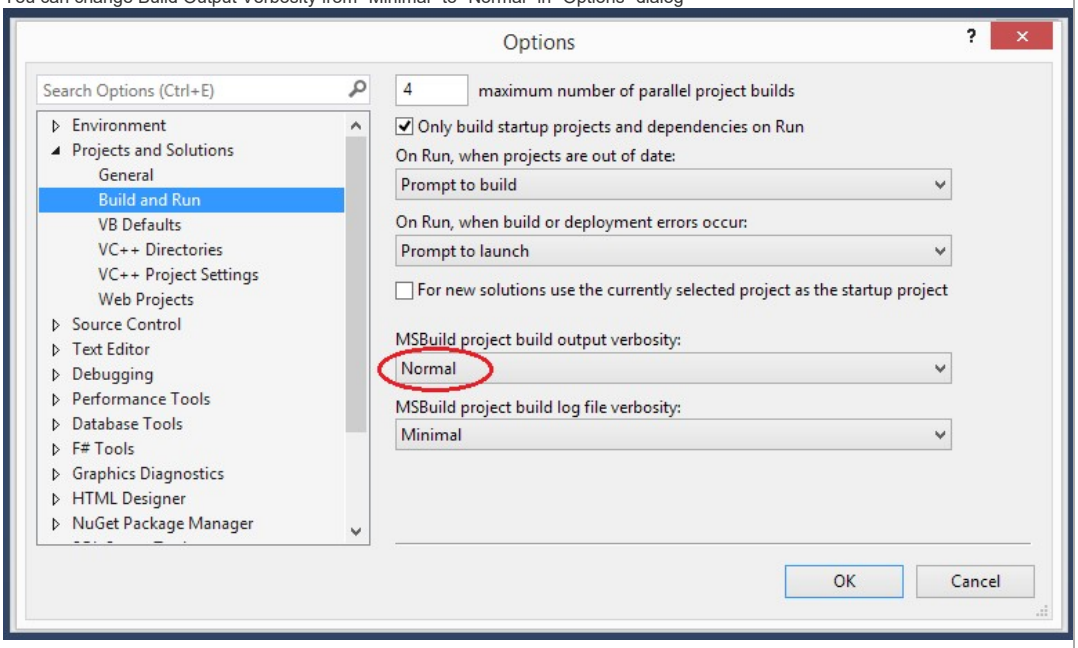


```
1>------ Build started: Project: ConsoleApplication1, Configuration: Debug Win32 ------
1>  Process sample bison file
1>  Process sample flex file
1>  sample.tab.cpp
1>  sample.flex.cpp
1>  Generating Code...
1>  ConsoleApplication1.vcxproj -> C:\Users\ConsoleApplication1\Debug\ConsoleApplication1.exe
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

If your flex/bison file is incorrect and you've got an error. But you don't see actual error message, something like this:

```
1>------ Build started: Project: ConsoleApplication2, Configuration: Debug Win32 ------
1>  Process "grammar.y" bison file
1>C:...\custom_build_rules\win_flex_bison_custom_build.targets(55,5): error MSB3721: The command "
1>C:...\custom_build_rules\win_flex_bison_custom_build.targets(55,5): error MSB3721: start /B /WAI
1>C:...\custom_build_rules\win_flex_bison_custom_build.targets(55,5): error MSB3721: exit /b %erro
========== Build: 0 succeeded, 1 failed, 0 up-to-date, 0 skipped ==========
```

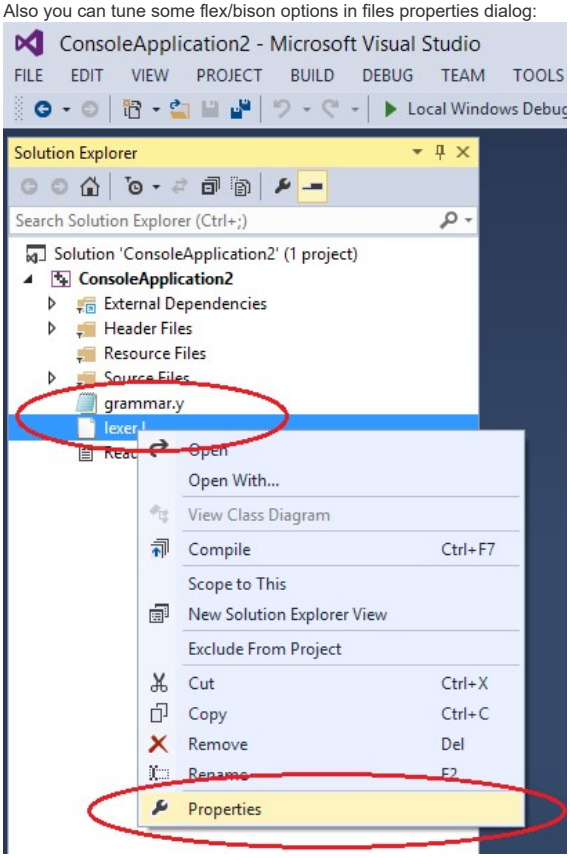
You can change Build Output Verbosity from "Minimal" to "Normal" in "Options" dialog

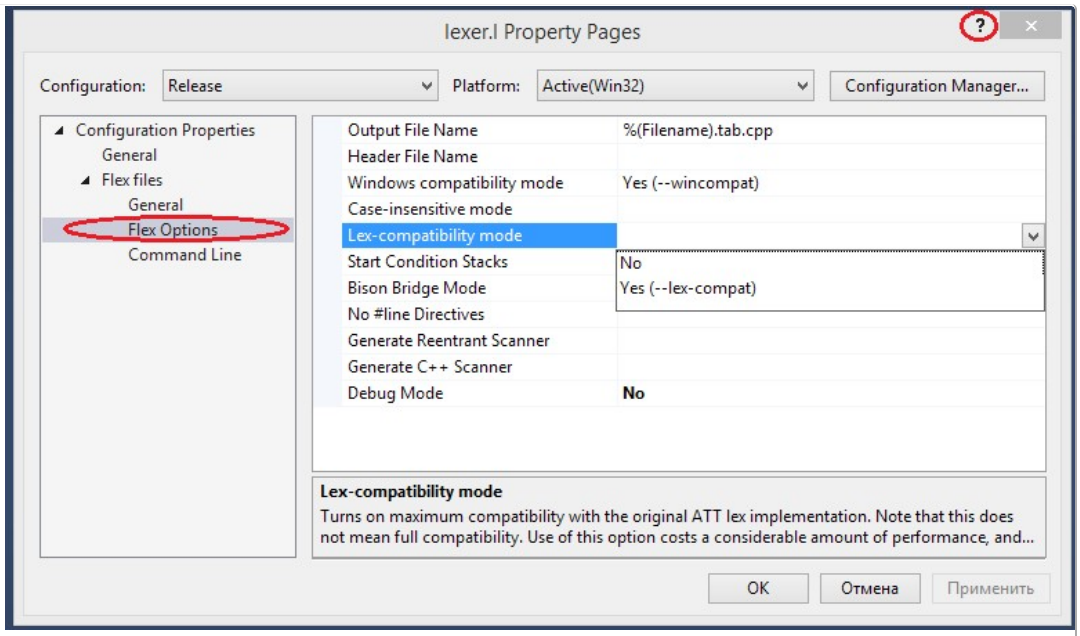


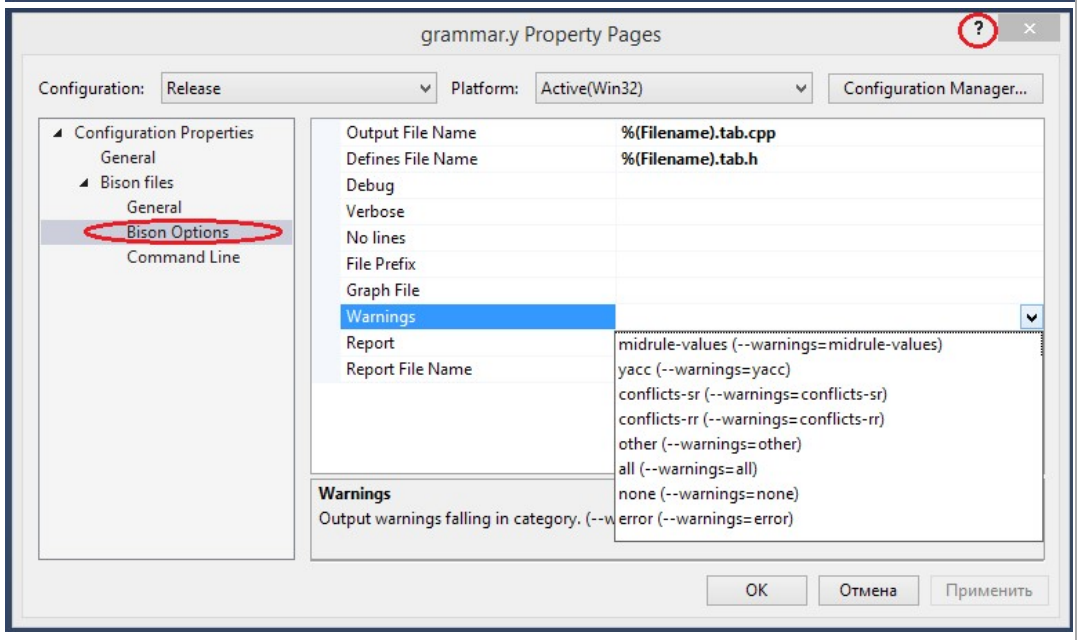
Then you will see more detailed output:

```
1>BisonTarget:
1>  Process "grammar.y" bison file
1>  grammar.y:51.1-4: error: invalid directive: '%sdw'
1>C:...\custom_build_rules\win_flex_bison_custom_build.targets(55,5): error MSB3721: The command "
1>C:...\custom_build_rules\win_flex_bison_custom_build.targets(55,5): error MSB3721: start /B /WAI
1>C:...\custom_build_rules\win_flex_bison_custom_build.targets(55,5): error MSB3721: exit /b %erro
1>
1>Build FAILED.
1>
1>Time Elapsed 00:00:01.21
========== Build: 0 succeeded, 1 failed, 0 up-to-date, 0 skipped ==========
```

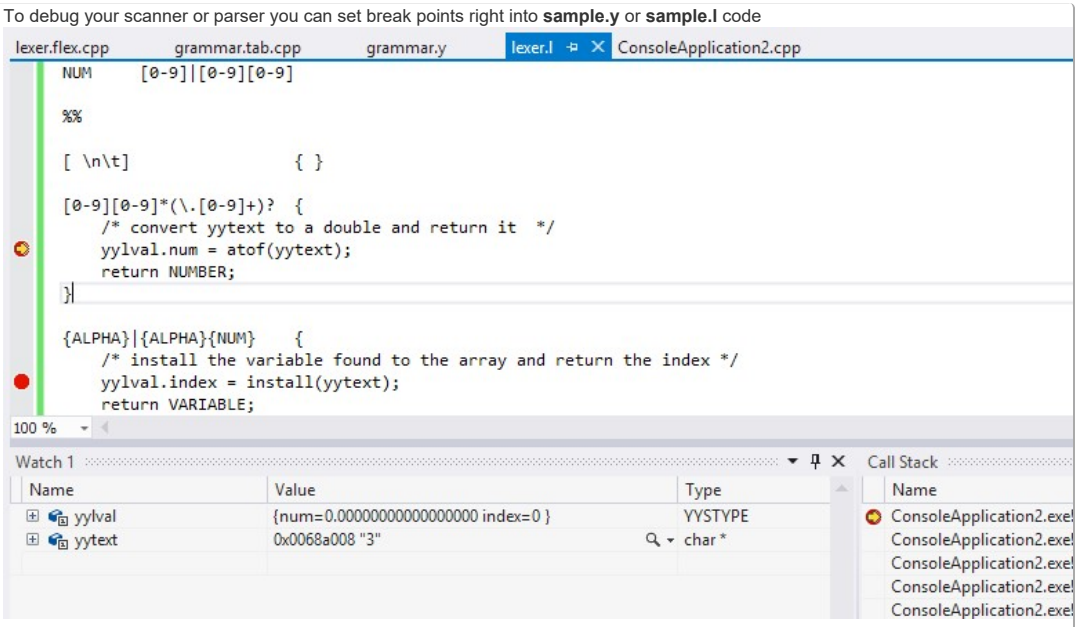
Also you can tune some flex/bison options in files properties dialog:




lexer.l Property Pages
Configuration: Release
Platform: Active(Win32)
Configuration Manager...
Configuration Properties
General
Flex files
General
Flex Options
Command Line
Output File Name
%(Filename).tab.cpp
Header File Name
Windows compatibility mode
Yes (--wincompat)
Case-insensitive mode
Lex-compatibility mode
Start Condition Stacks
Bison Bridge Mode
No #line Directives
Generate Reentrant Scanner
Generate C++ Scanner
Debug Mode
No
No
Yes (--lex-compat)
Lex-compatibility mode
Turns on maximum compatibility with the original ATT lex implementation. Note that this does not mean full compatibility. Use of this option costs a considerable amount of performance, and...
OK
Отмена
Применить


grammar.y Property Pages
Configuration: Release
Platform: Active(Win32)
Configuration Manager...
Configuration Properties
General
Bison files
General
Bison Options
Command Line
Output File Name
%(Filename).tab.cpp
Defines File Name
%(Filename).tab.h
Debug
Verbose
No lines
File Prefix
Graph File
Warnings
Report
Report File Name
midrule-values (--warnings=midrule-values)
yacc (--warnings=yacc)
conflicts-sr (--warnings=conflicts-sr)
conflicts-rr (--warnings=conflicts-rr)
other (--warnings=other)
all (--warnings=all)
none (--warnings=none)
error (--warnings=error)
Warnings
Output warnings falling in category. (--w
OK
Отмена
Применить

To debug your scanner or parser you can set break points right into **sample.y** or **sample.l** code



Enjoy!

Discussion

Tarek701 (/u/alphaboy/) - *2015-05-16*
(https://sourceforge.net/p/winflexbison/wiki/Visual%20Studio%20custom%20build%20rules/#a613)

Last edit: Tarek701 2015-05-16

gw1907 (/u/gw1907/) - *2015-06-26*
(https://sourceforge.net/p/winflexbison/wiki/Visual%20Studio%20custom%20build%20rules/#f354)

How can I change the custom build rules to build "C" parser?

Thanks

Alex Zhondin (/u/lexxmark/) - *2015-06-28*
(https://sourceforge.net/p/winflexbison/wiki/Visual%20Studio%20custom%20build%20rules/#d775)

It should generate "C" parser code now.

Do you mean generated files should be processed by "C" compiler (not "C++")?
In this case you should change extension of output files from .cpp to .c, so VS will recognize these files as pure "C" source code and will use "C" compiler.

Waheed Ejaz (/u/wejaz/) - *2017-02-16*
(https://sourceforge.net/p/winflexbison/wiki/Visual%20Studio%20custom%20build%20rules/#8157)

I generated lexer in a plain c file and let the bison generate a c++ code. I had to extern "C" the definitions of lexer functions i.e. yylex(), yylineno etc.
But still flex generated code gives linking error for yylval. I checked the generated file by parser, which is c++ file, and it has defined yylval.

```
.ccp bison generated file

extern YYSTYPE yylval;

#ifndef YY_INITIAL_VALUE
# define YY_INITIAL_VALUE(Value) /* Nothing. */
#endif

/* The semantic value of the lookahead symbol.  */
YYSTYPE yylval YY_INITIAL_VALUE(yyval_default);

.c lex generated file
extern YYSTYPE yylval;
yylval = "SomeValue";
```

There is a linking error. The lex generated file can't find the definition of yylval. Both the lex and bison generated files are in the same project (MS Visual Studio 2010).

Any ideas, what is wrong here.

Alex Zhondin (/u/lexxmark/) - *2017-02-17*

(https://sourceforge.net/p/winflexbison/wiki/Visual%20Studio%20custom%20build%20rules/#46f4)

I have not been working with VS 2010 for a long time.
I can only guess if YYSTYPE is different for lexer and for parser.
Please show all errors as they appear in VS Studio output window.

Waheed Ejaz (/u/wejaz/) - *2017-02-17*

(https://sourceforge.net/p/winflexbison/wiki/Visual%20Studio%20custom%20build%20rules/#a1cc)

It is just this one.

```
Error   11  error LNK1120: 1 unresolved externals   \asn1cc.exe 1
Error   10  error LNK2001: unresolved external symbol _yylval    asnlex_win.obj
```

Anyway, I have moved on to using .cpp files. They work fine.

Log in (/auth/) to post a comment.