# 张武生

博客园 :: 首页 :: 新随笔 :: 联系 :: 订阅 Ⅲ :: 管理

posts - 13, comments - 2, trackbacks - 0, articles - 0

日	_	=	Ξ	四	五	六
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

2020年10月

# Windows下面编译libiconv

Posted on 2011-12-22 09:46 张武生 阅读(1875) 评论(0) 编辑 收藏

2011-12-22 09:43:18

http://www.codeproject.com/KB/library/Buildlibiconv.aspx

# 金 公告

昵称: 张武生 园龄: 9年1个月

粉丝: 0 关注: 3 +加<del>关</del>注

# Introduction

Libiconv is one of the most useful and famous libraries that converts texts between encodings and has been developed under GNU projects. However, now it does not provide the way to build it with Microsoft Visual Studio but with gcc. Before, it had provided but now stopped explaining how to build it with Microsoft Visual Studio. However, many of the computer programmers are using Microsoft Visual Studio and want to build libiconv with Microsoft Visual Studio. So, this article is about how to build libiconv with Microsoft Visual Studio. If you can build libiconv with Microsoft Visual Studio, you will be able to change various compilation options and build it in order to optimize libiconv according to your preference and needs.

#### 搜索



# 簡 常用链接

我的随笔

我的评论

我的参与

# **Background**

If you have no idea about <code>libiconv</code>, you can refer to it here or you can Google for it. You may be confused with <code>iconv</code> and <code>libiconv</code>. <code>Iconv</code> is the executable program that uses <code>libiconv</code> while <code>libiconv</code> is the library for converting text strings between two different encodings. This article is related to <code>libiconv</code>. However, this article is not about <code>libiconv</code> but about how to build <code>libiconv</code> with Microsoft Visual Studio. The GNU library <code>libiconv</code> is under LGPL license. So, you have to be really careful when you use<code>libiconv</code> for your commercial use or when you don't want to publicize your source code. About LPGL license, you can refer here.

最新评论 我的标签

更多链接

前 我的标签

Vs2010 设置路径(1)

🛅 随笔档案

2011年12月(1)

2011年11月(2)

2011年10月(5)

2011年9月(5)

#### 最新评论

1. Re:开通博客啦,老家伙跟上新时代 啦

欢迎欢迎,我也是昨天刚开的.哈哈
--DotNetProgramer

2. Re:开通博客啦,老家伙跟上新时代 啦

欢迎!

--dudu

#### 阅读排行榜

- 1. Windows下面编译libiconv(1875)
- 2. 设置VS2010的默认路径(1842)
- 3. 使用Boost.Flyweight的好处(1186)
- 4. 关于boost.msm和boost.statechart 的讨论(943)
- 5. NYF修复工具(787)

#### 评论排行榜

## **How to Get the Source Code of libicony**

At the moment, the most recent version of libiconv is 1.14. You can download the source code
oflibiconv 1.14 here and here. Of course, you can get the older versions of libiconv here.

### **How to Build**

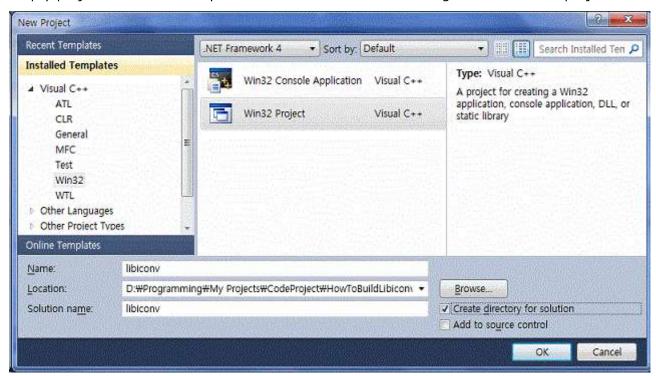
I am going to explain the case of Microsoft Visual Studio 2010 professional edition but the explanation can be applied to the earlier versions of Microsoft Visual Studio and to the express edition of each of them.

### **First Step**

You download the most recent version of <u>libiconv</u> which is 1.14 from here or here. And unzip the downloaded file "*libiconv-1.14.tar.qz*" on your certain folder.

## **Second Step**

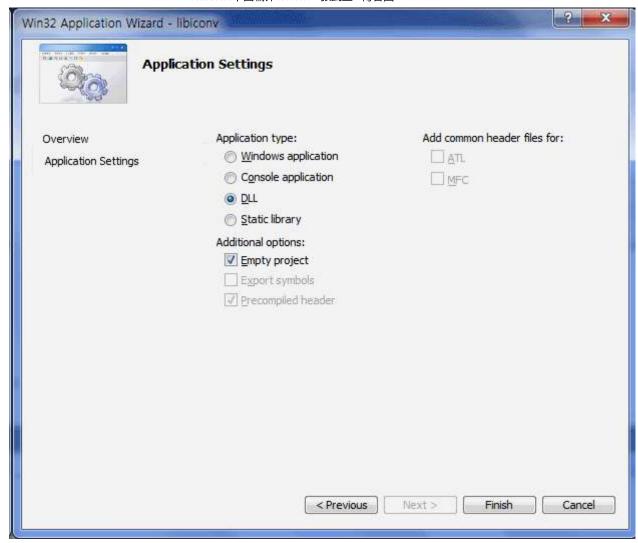
You run your Microsoft Visual Studio. Then, you make a new WIN32 project. Let's say "libiconv" as the project name. Check to create directory for solution. Choose DLL as the application type and check Empty project for additional options. Click the button "finish" to generate the new project.



1. 开通博客啦, 老家伙跟上新时代啦(2)



- 1. Windows下面编译libiconv(1)
- 2. 修改 nginx的POST大小设置(1)
- 3. 设置VS2010的默认路径(1)
- 4. 使用Boost.Flyweight的好处(1)



### **Third Step**

Now, you have to copy some files from the folders generated by unzipping "libiconv-1.14.tar.gz" to the project folders. To build "libiconv", you need to compile three files "localcharset.c", "relocatable.c" and "iconv.c". That is the key idea!

Copy three files "relocatable.h", "relocatable.c" and "iconv.c" in the folder "...\libiconv-1.14\lib\" to the project folder "...\libiconv\".

Copy "...\libiconv-1.14\libcharset\lib\localcharset.c" to the project folder "...\libiconv\".

Copy "...\libiconv-1.14\libcharset\include\localcharset.h.build.in" to the project folder

"...\libiconv\libiconv\" and then, rename the copied "localcharset.h.build.in" to "localcharset.h. Copy "...\libiconv-1.14\windows\libiconv.rc" to the project folder "...\libiconv\libiconv\".

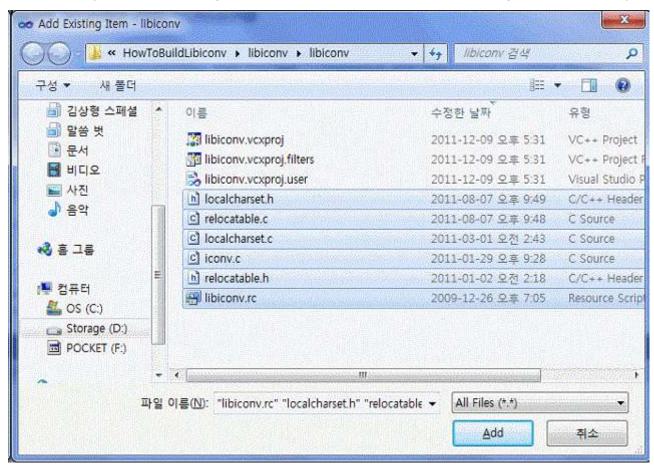
Make folder "include" under the project folder "...\libiconv\" so to make "...\libiconv\include". Copy "...\libiconv-1.14\include\iconv.h.build.in\" to the project include folder "...\libiconv\include\" and then, rename the copied "iconv.h.build.in" to "iconv.h".

Copy "...\libiconv-1.14\config.h.in" to the project include folder "...\libiconv\include" and then, rename the copied "config.h.in" to "config.h".

And then, copy all the header files (\*.h) and definition files (\*.def) in the folder "...\libiconv-1.14\lib" to the project include folder "...\libiconv\include".

### **Fourth Step**

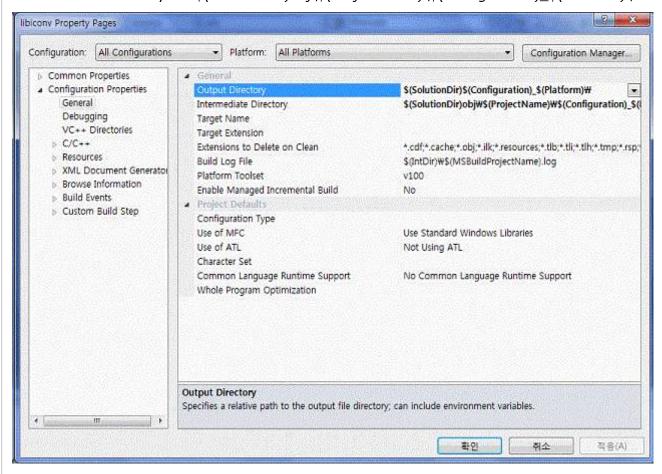
Now, you have to add existing items to the project "libiconv" that we made at the second step. Execute "project > Add Existing items..." at the main menu to add existing items to the project.



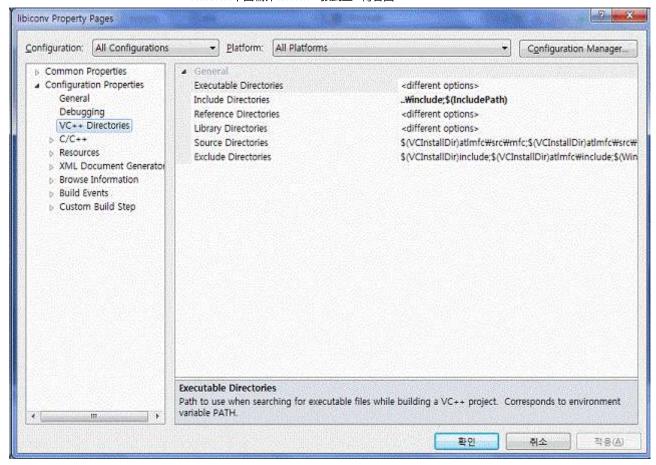
## **Fifth Step**

You can make 64-bit platform through configuration manager in order to generate *libiconv.dll* for 64-bit system. You can also make two other configurations "ReleaseStatic" and "DebugStatic" in order to generate <code>libiconvStatic.lib</code> as a static link library.

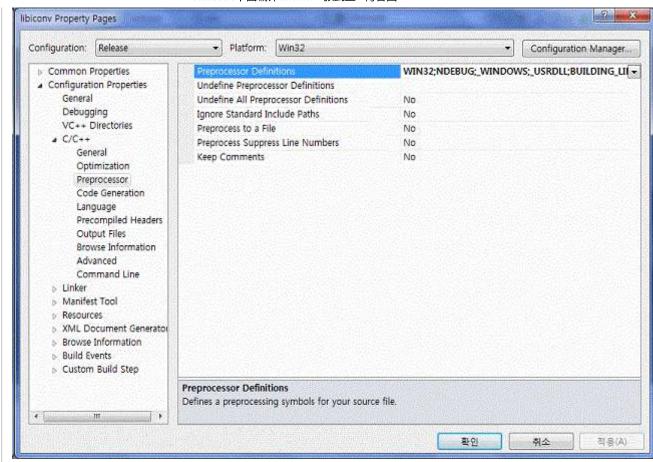
At the project properties, change Output Directory as "\$(SolutionDir)\$(Configuration)\_\$(Platform)\" and Intermediate Directory as "\$(SolutionDir)obj\\$(ProjectName)\\$(Configuration) \$(Platform)\".



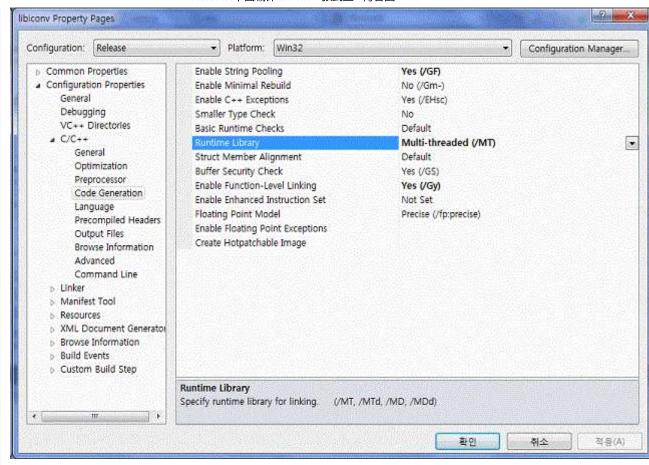
Change Include Directories as "..\include;\$(IncludePath)":



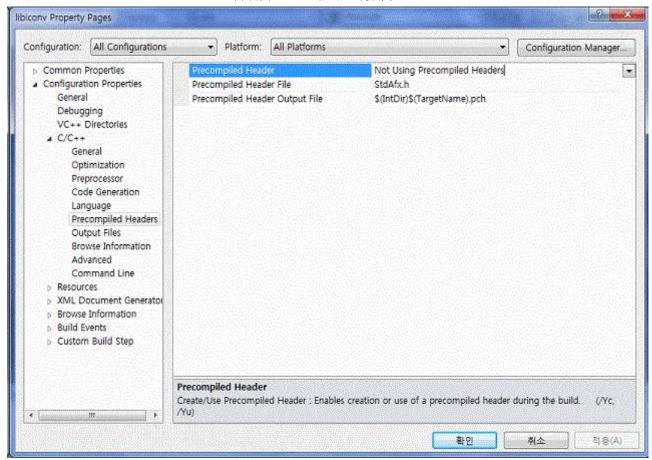
You have to add "BUILDING\_LIBICONV" and "BUILDING\_LIBCHARSET" to Peprocessor Definitions of all Platforms and of all configurations.



You'd better set Runtime Library to "Multi-threaded" when building dynamic link library *libiconv.dll*. Then, the dependency on VC Runtime library can be controlled by the applications that will be built and dynamically linked with *libiconv.dll* because *libiconv.dll* does not need VC Runtime library but only the application that uses *libiconv.dll* may or may not need VC Runtime library. However, when building the static link library/libiconvStatic.lib, you can choose Runtime Library option for *libiconvStatic.lib* depending on the application that uses *libiconvStatic.lib*.



You have to change Precompiled Header option to "Not Using Precompiled Headers".



## **Sixth Step**

Now, it is the time to tweak the source code of the libiconv.

#### libiconv.rc

Open *libiconv.rc* with text editor or the source code editor of Visual Studio IDE by double-clicking *libiconv.rc* in the Solution explorer and insert some code at line 4 as follows:

You may be asked to change Line endings to "Windows (CR LF)". Then, let it do so. It will be more convenient for you if you mainly use Windows.

#### localcharset.c

Open localcharset.c and delete or comment the lines 76 - 79 as follows:

#### iconv.c

Open iconv.c and delete or comment the lines 246 - 248 and add three lines there as follows:

#### localcharset.h

Open localcharset.h and delete or comment the lines 23 - 27 and add 7 lines there as follows:

#### config.h

Open config.h in the project include folder "...\libiconv\include" and delete or comment the lines 29 - 30 as follows:

☐ Collapse | Copy Code

```
///* Define as good substitute value for EILSEQ. */
//#undef EILSEQ
```

Otherwise you can redefine **EILSEQ** as good substitute value.

#### iconv.h

Open iconv.h in the project include folder "...\libiconv\include" and delete or comment the line 175 and add 1 line as follows:

```
□ Collapse | Copy Code
//#if @HAVE WCHAR T@
#if HAVE WCHAR T
```

Delete or comment the line 128 and add 1 line as follows:

//#if @USE MBSTATE T@

#if USE MBSTATE T

```
☐ Collapse | Copy Code
```

Delete or comment the lines 107-108 and add 2 lines as follows:

```
☐ Collapse | Copy Code
//#if @USE MBSTATE T@
//#if @BROKEN WCHAR H@
#if USE MBSTATE T
#if BROKEN_WCHAR_H
```

Delete or comment the line 89 and add 2 lines as follows:

```
☐ Collapse | Copy Code
```

```
//extern LIBICONV DLL EXPORTED size t iconv (iconv t cd, @ICONV CONST@ char* * inbuf,
```

Delete or comment the lines 26 - 31 and add 8 lines as follows:

```
☐ Collapse | Copy Code
//#if @HAVE VISIBILITY@ && BUILDING LIBICONV
//#define LIBICONV DLL EXPORTED attribute (( visibility ("default")))
//#else
//#define LIBICONV DLL EXPORTED
//#endif
//extern LIBICONV DLL EXPORTED @DLL VARIABLE@ int libiconv version; /* Likewise */
#if BUILDING LIBICONV
#define LIBICONV DLL EXPORTED declspec(dllexport)
#elif USING STATIC LIBICONV
#define LIBICONV DLL EXPORTED
#define LIBICONV DLL EXPORTED declspec(dllimport)
#endif
extern LIBICONV DLL EXPORTED int libiconv version; /* Likewise */
```

### **How to Use**

When you use newly built libiconv, the only header file that you need is *iconv.h*. You will need to link either the import library *libiconv.lib* or the static library *libiconvStatic.lib* in your project property or write the code in one of your source file as follows:

```
#pragma comment (lib, "libiconv.lib")
or
#pragma comment (lib, "libiconvStatic.lib")
```

In the source of the application that uses this library either *libiconv.dll* or *libiconvStatic.lib*, if you don't define anything but only include *iconv.h*, your application will use *libiconv.dll* while it will use *libiconvStatic.lib* if you define USING\_STATIC\_LIBICONV before you include *iconv.h* in your application as follows:

```
//#define USING_STATIC_LIBICONV
#include <iconv.h>
Collapse | Copy Code
```

# **Copyright Issues**

The original libiconv is under LGPL licence. So, the tweaked libiconv has also to be under LGPL. If you develop any application that uses libiconv, you have to be careful about copyright issues. If your application is linked with *libiconv.dll* dynamically, you don't have to publicize your source code of your application. However, if your application is linked statically with *libiconvStatic.lib*, you have to publicize either the source code of the application or the object files (\*.obj) of the application so that others can build the application that is linked statically with the library under LGPL license.

### License

This article, along with any associated source code and files, is licensed under The GNU Lesser General Public License (LGPLv3)



0

+加关注

« 上一篇: 修改 nginx的POST大小设置

刷新评论 刷新页面 返回顶部

#### (评论功能已被禁用)

【推荐】华为HCIE直播密训: VRRP+MSTP典型组网案例分析、 一步玩转HCIE认证

【推荐】超50万行C++/C#:大型组态工控、电力仿真CAD与GIS源码库

【推荐】未知数的距离,毫秒间的传递,声网与你实时互动

#### 相关博文:

- 如何使用VisualStudio构建libiconv
- · cygwin编译libiconv库的方法(网络收藏)
- · 利用VS编译libiconv库
- libIconv.lib编码库的生成和使用
- · 使用MinGW 编译 iconv 库
- » 更多推荐...

#### 最新 IT 新闻:

- 《英雄联盟》手游开启公测:支持iOS/Android双平台
- · 武汉弘芯抵押光刻机: 芯片行业到底出现了什么问题?
- · 美国发生大火:即将烧至暴雪总部员工现已撤离
- ·特斯拉车辆安全报告发布:开AP平均740万KM才有一起事故
- ·最新测试: 阿联酋电信移动网速 98.78Mb/s 全球最快,中国第九
- » 更多新闻...

Powered by:

博客园

Copyright © 2020 张武生

Powered by .NET 5.0.0-rc.2.20475.5 on Kubernetes