

## Qt 的開源版本與商業版區別及LGPL 與閉源程序

十月的天空 Qt 學習 2019-02-12

原文地址：

<http://blog.51cto.com/devbean/313477>

最近一直在學習Qt，Qt 有兩個許可證：LGPL 和商業協議。這兩個協議在現在的Qt 版本中的代碼是完全一致的（潛在含義是，Qt 的早期版本，商業版的Qt 通常包含有一些開源版本所沒有的庫，比如QtSingleApplication 這個庫）。所以現在對於普通開發人員和部分商業公司來說，使用LGPL 版本的Qt 可以節省很大的開銷。這兩個版本最大的區別在於，前者是免費的，後者是收費的。既然代碼都是一致的，所以費用就要是用來購買Qt 的售後服務和培訓等等相關服務。現在我們是來說一下版權的問題。LGPL 是一個開源協議，因此，有人會擔心LGPL 能否用於開發閉源程序，能夠拿來賣錢。儘管現在國內有些公司不是很重視這方面的問題，不過，如果你違反了協議，某一天被別人發來一紙律師函的時候，真的是欲哭無淚了哦。所以，我們還是先來研究一下這個協議，LGPL 究竟能不能用於開發閉源程序。以下內容是我查找了N 多網站總結出來的，因為豆子不是律師，所以LGPL 協議基本看不懂。究竟怎樣去理解這個協議，還是希望能夠有專業人士說出來。這裡就算做是一種拋磚引玉吧！儘管沒有十分的確定，但是這裡所說的理解基本也是八九不離十的了。至於什麼是LGPL 協議，這裡就不再多說了，我們關心的是，如果使用LGPL 協議開發商業程序。請注意，這裡所說的閉源程序，是指不以某種形式開放源代碼，也就是說，用戶（包括其他開發者）不能獲取其源代碼的程序。首先說明一點，這裡的含義是，你可以用LGPL協議開發商業程序，當然也可以是非商業的閉源程序。但是，它是有一些限制的。這就是我們要討論的重點。既然我們已經對其定性，那麼我們直接進入主題：

**使用LGPL 協議開發閉源程序，如果你使用動態鏈接的形式，那麼，你可以以任何形式發布你的應用程序 如果你因某種原因必須靜態鏈接一個基於LGPL 協議發布的庫（一下我們簡稱為LGPL 庫），那麼，你有義務進行下面的工作：**

- 1. 你必須在你的文檔中說明，你的程序中使用了LGPL 庫，並且說明這個庫是基於LGPL 發布的；**
- 2. 你必須在你的應用程序發布中包含一份LGPL協議，通常就是那個文本文件；**
- 3. 你必須開放使用了LGPL 庫代碼的所有代碼，例如某些封裝器。但是，其他使用這些封裝器的代碼就不需要開放了；**
- 4. 你必須包含你的應用程序的餘下部分的目標文件（通常就是我們所說的.o 等等），或者是其他等價的文件。源代碼並不是必須的。**

是不是很難理解呢？我們詳細的說一下。第一條很容易理解；第二條也很容易理解，你可以在這裡找到LGPL 協議的內容，複製下來隨你的程序一起發布就可以了。第三條就不那麼好理解了。簡單來說，LGPL 協議要求，如果你的類使用了LGPL 庫的代碼，那麼必須把這個類開源。例如，如果你的程序app.exe 每個源文件都使用了LGPL 庫的代碼，那麼你的所有源代碼都要開源。為了避免這種情況，我們通常編寫一個封裝器，把LGPL 庫的代碼封裝起來，這樣就只需要開放這個封裝器的代碼，而其他使用了這個封裝器的代碼就不需要開放。第四條是對第三條的一種補充：那些使用了封裝器的程序不需要開源，但是你必須把你編譯的那些中間文件開放出來，Windows 下就是那些.o 文件。你很奇怪，為什麼LGPL 協議要這樣規定呢？LGPL 所做的工作是，它保證了用戶能夠有這樣一種能力：修改你使用LGPL 庫函數的方式（那些封裝器就是你使用LGPL 庫的方式，那些已經開源了），重新編譯這些代碼，然後重新對程序進行連接（連接所需要的目標文件也是包含了的，這是第四條規定的），就可以得到一個新的可執程序。好了，如果你還不明白如何使用，我們來看一個例子。假設我們使用一個名為Lib 的庫，這個庫是基於LGPL 協議發布的。如果你使用Lib.dll 做動態鏈接（Windows 下），好，一切OK。無論你的程序怎麼樣，你都可以做你所做的事情。我們主要是來看，如果你要使用靜態鏈接，那麼你需要如何組織你的代碼。如果你有一個main.cpp（我們假設所有Lib 庫的函數都是用了lib\_ 前綴）：

```
1. // main.cpp
2. int main() {
3.     lib_init();
4.     lib_do_something();
5.     lib_done();
6.     lib_close();
7.
8.     return 0;
9. }
```

現在你已經完成了main.cpp，但是你必須把它開源！因為它使用了LGPL 庫的代碼。這是上面第三條規定的。我不想把它開源，怎麼辦呢？好，我們建一個新的文件lib\_wrapper.cpp:

```
1. void my_lib_init()
2. {
3.     lib_init();
4. }
5.
6. void my_lib_do_something()
7. {
8.     lib_do_something();
```

```
9. }  
10.  
11. void my_lib_done()  
12. {  
13.     lib_done();  
14. }  
15.  
16. void my_lib_close()  
17. {  
18.     lib_close();  
19. }
```

在main.cpp 中，我們做相應的修改：

```
1. int main() {  
2.     my_lib_init();  
3.     my_lib_do_something();  
4.     my_lib_done();  
5.     my_lib_close();  
6.  
7.     return 0;  
8. }
```

現在，main.cpp 不再是直接使用了LGPL 庫的代碼了，因此它不需要開源，而我們的封裝器lib\_wrapper.cpp 必須開源。

好，編譯一下我們的程序，你會得到main.o（Windows 下）這個目標文件。

在最終程序的發布中，你需要包含一下文件：

1. 一份文檔，其中聲明：這個程序使用了Lib庫，這個庫是基於LGPL 協議發布的;
2. LGPL.txt;
3. lib\_wrapper.cpp
4. main.o

這樣，用戶可以通過修改lib\_wrapper.cpp 的內容改變你使用LGPL 庫的方式，例如：

```
1. void my_lib_done()  
2. {  
3.     lib_done();  
4.     lib_close();  
5. }  
6.  
7. void my_lib_close()  
8. {  
9.     // lib_close();  
10. }
```

然後編譯這個lib\_wrapper.cpp，最終重新鏈接。一個新的可执行程序誕生啦！

好了，這就是在使用LGPL庫開發閉源程序所需要遵守的東西了。還是建議大家能夠遵守協議，尊重作者的勞動成果哦~

附件中是LGPL協議的文本文件。

附件：

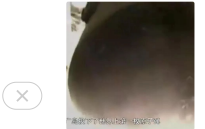
<http://down.51cto.com/data/2355754>

[閱讀原文](#)

喜歡此內容的人還喜歡

俄羅斯和德國做錯了什麼，中國也生氣了，有的國家就是瘋狗！

新財迷



隨筆| 教育資本化，惡魔的凝視，深淵的入口！

一個壞土豆

