

2.6. Licensing the Library

First of all, the big disclaimer:

Warning: We are *not* lawyers; we cannot help you decide if you should use the library or how to apply the license, only your lawyer can advise you concerning legal matters.

That said, it should also be noted that we have neither the resources (time, cash, etc) nor the patience to enforce the license (at all); the reality is that all of this is left to your discretion.

If you prefer to leave the lawyers out of it, the rest of this section will help you apply the license to your application.

2.6.1. Licensing FAQ

Can I use this library with my proprietary (closed source) application?

Yes, that is our intent and we do not believe there is any problem regarding the license.

Will I need to open source my entire app?

No, it is not required by the license.

Will I need to distribute all of my “object files” on the App Store?

No, this is also not required by the license, although you should offer to provide them upon request. See below for more detail.

But I read somewhere that “iPhone apps may not use LGPL code”?

That statement is an over-generalization that does not apply in this case. Most likely your source is either:

- referring to the GPL, which is significantly different from the LGPL
- referring to a different version of the LGPL; we intentionally use version 2.1, which has specific static linking exceptions.
- not a lawyer either and too lazy to read the whole license

Basically, if you leverage the appropriate sections of the license, it should be fully compatible with the App Store restrictions and requirements.

This is too complicated, can I just pay for an easier license?

No, it is not possible. There are multiple problems with this approach, some brief highlights:

- Most open source projects (including this one) do not have a single author.

Tracking down every contributor and getting their approval could be quite a challenge.

- The license is meant to protect users' rights to the software. Giving you special treatment bypasses the protection we offered them, effectively revoking their rights. This would be a violation of their trust and completely defeats the purpose of the license.

You may think of this as the “price” you pay for using our open source library. If you want to make your life easier, you should be petitioning Apple for shared library support...

What if you add a clause that lets me do whatever I want?

No, also not possible. In addition to the problems mentioned above, there are even more problems with this:

- Sourceforge requires an OSI approved license for hosting our project; an altered license would no longer be approved.
- Again we are not lawyers and therefore not qualified to change the license, we would have to pay one of those slimy buggers to do it.

Do I need to add an “about” dialog to display your copyright/license?

No, not as such. We won't discourage you from plugging the library if you like, but it is not a requirement. You should think of our license as a supplement to your own software license, therefore it is appropriate to display it where (and only where) you display your own:

- If you follow Apple's recommendation, the App Store is the only place that the user accesses your license, so it should also be the only place where the library supplement is available.
- If your app already has some kind of “about” view that displays your copyright/license information, it is also appropriate to display the same information for the library.

Do I need to include the entire library in my application bundle?

No, it is not necessary:

- If you have not modified the library, it is sufficient to provide a link to the project and the version information.
- If you are using a modified version, you may provide a link to download that instead of including it in the bundle.

2.6.2. Modifications

What is a “modification”? Again, we leave it to your discretion with this guidance:

- If you use the distributed binary SDK you have certainly not modified the library.

- If you are working from Mercurial, *any* change you have made to the “source code” of the library is a modification, it does not matter how trivial. You can see what changes have been made by running `hg status -mard`; if this command outputs anything, you have modified the library.

If you find that you have made changes to the library, you should carefully consider how far you want to proceed down that path. Once you publish your changes you have created a “fork” of the project which you now need to maintain. Are you prepared to merge every time the library is updated?

If your change adds a useful feature to the library, we absolutely encourage you to submit patches. Assuming you can get your patch into the project, then you will no longer need to use a modified version! When submitting patches, ensure that your changes are appropriate for all of our users. Specifically, we are not interested in patches that simply hack up the library to work the way you want. Compare a patch that changes the default behavior to your preference (probably not acceptable), to a patch that adds a new configuration to support the feature you have added (probably fine).

2.6.3. Object File Distribution

Section 6 of the LGPL v2.1 specifically permits static linking with the library. If your project is not open source, this section does require that you make your object files available to your users. The intent, as indicated in the license, is that a user who has obtained your software may exercise their right to modify the library and then re-link their modified version into your application.

We recommend that you apply Subsection 6c, which only requires that you make a written offer to provide the object files. Now...if you consider the actual utility of this mechanism - that it is only applicable to developers, and only those with in depth knowledge of the tools, the time required for development - all to have a new barcode reader in a specific version of your application that only they can use, the reality is that no one is going to request this. You probably should not even waste time preparing for it until a request is made.

Additionally, to avoid “casual requests” from nefarious types that just want to inconvenience you, also consider charging a fee for the distribution of this material (as permitted by the license); just add up the cost of burning and shipping a disk. If this cost is “large” compared to the price of your app, the likelihood of a request is reduced even further.

2.6.4. Using the Unmodified Library

Applying the license in this case is somewhat simpler. These are the basic steps you can follow:

1. Verify that the rest of your software license is compatible with the LGPL. You cannot use the library if they are incompatible.

For those using the default App Store license, we have reviewed this and believe it is compatible with the LGPL.

2. At the end of your license text, in an annex or supplement, start by declaring your use of the library and offering a link to the project. Something like this:

This software uses the open source ZBar Barcode Reader library, version 1.2, which is available from <http://zbar.sourceforge.net/iphone>

If you built your own version of the library, replace the version callout with eg, “cloned from Mercurial revision xxxxxxxx”

3. Then append the contents of the text file COPYING, included with the library. This is all of the copyright information for the library.
4. Then append the contents of the text file LICENSE, also included with the library. This is just the LGPL version 2.1 which you may also obtain from <http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html>
5. You may choose to make the written offer for the object files explicit. Provide some text and whatever link or email address is appropriate.

2.6.5. Using a Modified Library

We intentionally leave this option vague and force you to refer to the license as an underhanded way of encouraging you to contribute back to the project ;)