



# GOTC 2023

## 全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

---

# OPEN SOURCE, INTO THE FUTURE #

---

### 「OSPOCon」专场

本期议题：从国内GPL抗辩成功第一案看企业开源合规

陶冶 2023年5月28日

## 案情介绍

### 侵权指控

- 原告研发了“A投标文件制作工具”软件
- 该软件使用了开源软件基于GPL v2.0+ with Classpath Exception许可证发布的开源组件。
- 被告开发了“B投标文件制作工具”，经比对，约有1万余行代码实质性相似，原告提出侵权诉讼。



## 案情介绍

### 被告抗辩

- 原告软件使用GPL组件，整体被GPLv2协议约束。
- 即便认定原被告软件构成实质性相似，被告也因和原告存在GPL许可关系而不构成侵权。
- 原告未以GPL协议开源其代码，属于违反诚实信用原则，不应该支持其侵权诉讼请求。

## 法院判决

- 原告在**主程序**中使用了**GPL**组件，且其使用方式不满足Classpath Exception的要求。
- 原告负有将其软件代码以**GPL**协议开源的义务但未履行，不支持就相关代码的侵权诉讼请求。
- 原告**开发软件的预览程序**与主程序构成**GPL**中的独立模块，被告对该部分代码的使用侵犯了原告的著作权。

- 大胆地承认了GPL许可证的传染性。
- 引入了“不洁之手”规则，确立了“不遵守，则不保护”的规则，对开源生态的发展有重要的意义。

# 引出的问题

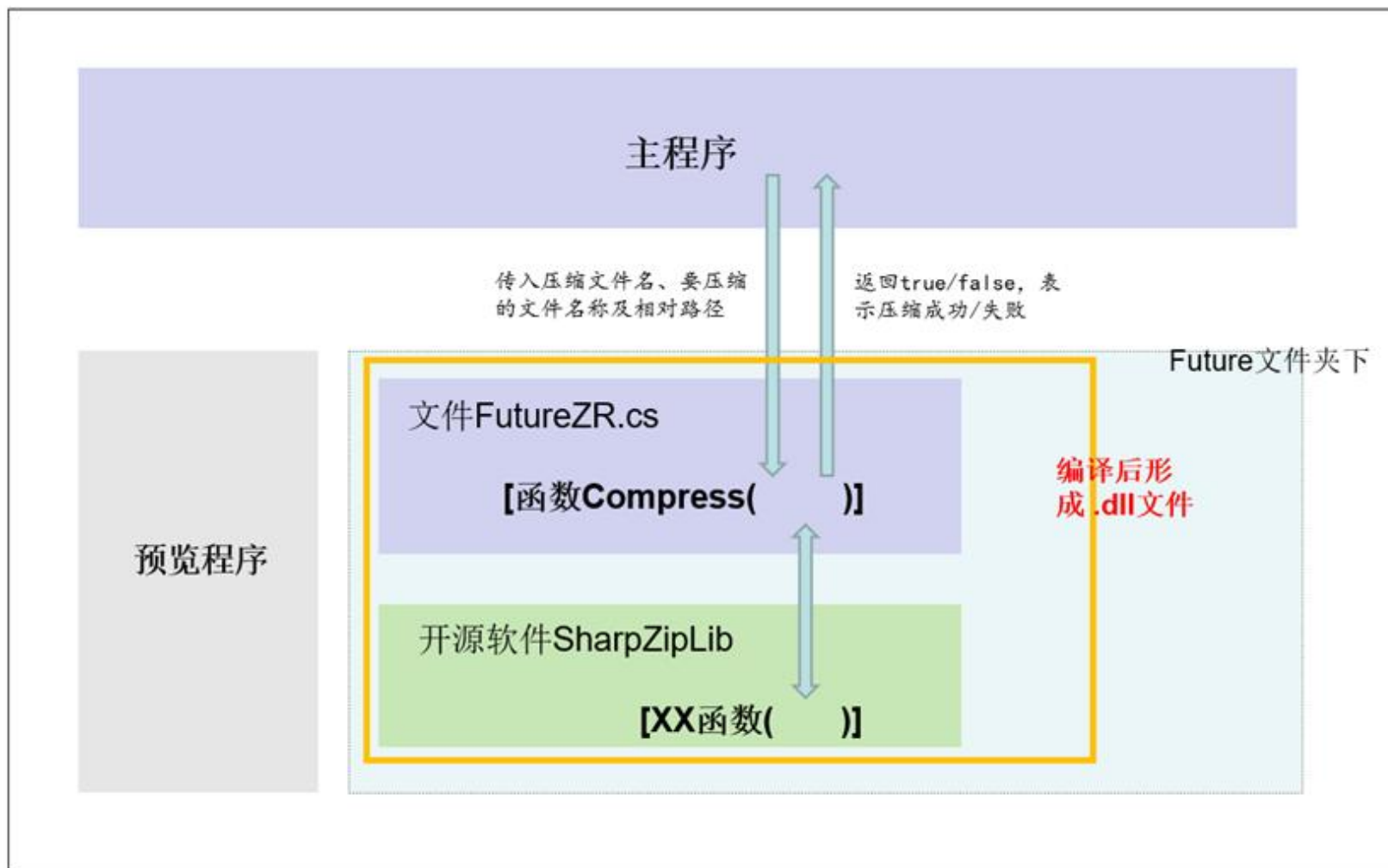
- 开源许可证的法律定性问题
- Classpath Exception 的适用问题
- GPL中独立模块的认定问题

- 合同的成立
  - 《民法典》第四百九十条后段：“当事人一方已经履行主要义务，对方接受时，该合同成立。”
- 格式条款
  - 《民法典》第四百九十六条第二款后段：“提供格式条款的一方未履行提示或者说明义务，致使对方没有注意或者理解与其有重大利害关系的条款的，对方可以主张该条款不成为合同的内容。”

```
14 //
15 // You should have received a copy of the GNU General Public License
16 // along with this program; if not, write to the Free Software
17 // Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
18 //
19 // Linking this library statically or dynamically with other modules is
20 // making a combined work based on this library. Thus, the terms and
21 // conditions of the GNU General Public License cover the whole
22 // combination.
23 //
24 // As a special exception, the copyright holders of this library give you
25 // permission to link this library with independent modules to produce an
26 // executable, regardless of the license terms of these independent
27 // modules, and to copy and distribute the resulting executable under
28 // terms of your choice, provided that you also meet, for each linked
29 // independent module, the terms and conditions of the license of that
30 // module. An independent module is a module which is not derived from
31 // or based on this library. If you modify this library, you may extend
32 // this exception to your version of the library, but you are not
33 // obligated to do so. If you do not wish to do so, delete this
34 // exception statement from your version.
35
36 // Suppress this in CF and 1.1, not needed. Static classes introduced in C# version 2.0
37 #if !NETCF_2_0 && !NET_1_1
```







# 独立模块的认定

Combining two modules means connecting them together so that they form a single larger program. If either part is covered by the GPL, the whole combination must also be released under the GPL—if you can't, or won't, do that, you may not combine them.

What constitutes combining two parts into one program? This is a legal question, which ultimately judges will decide. We believe that a proper criterion depends both on the mechanism of communication (exec, pipes, rpc, function calls within a shared address space, etc.) and the semantics of the communication (what kinds of information are interchanged).

If the modules are included in the same executable file, they are definitely combined in one program. If modules are designed to run linked together in a shared address space, that almost surely means combining them into one program.

By contrast, pipes, sockets and command-line arguments are communication mechanisms normally used between two separate programs. So when they are used for communication, the modules normally are separate programs. But if the semantics of the communication are intimate enough, exchanging complex internal data structures, that too could be a basis to consider the two parts as combined into a larger program.



THANKS