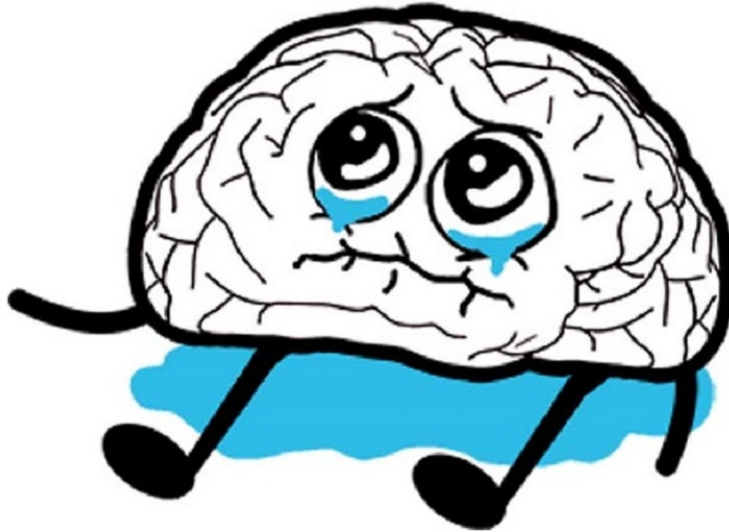


# YangYouji's WebSite

IT, VISION

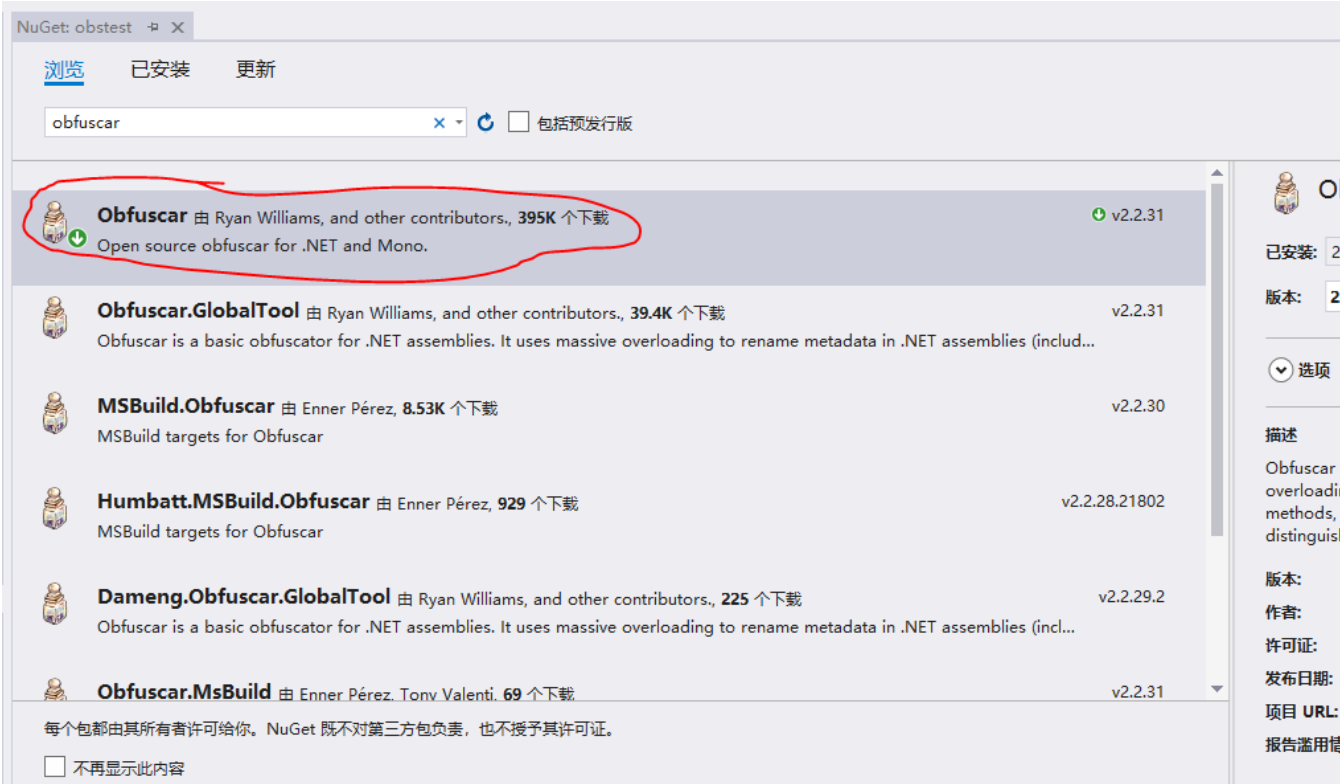


## C# 混淆工具 obfuscator

c# 编译生成的exe 与 dll 如果不进行混淆, 可以方便的反编译出源代码。虽然软件开源是趋势, 但这种非自愿, 非正式的公开还是给人不爽的感觉。c#混淆工具很多, 收费的也很多, 下面介绍一个开源混淆工具obfuscator, 简单混淆一下的使用场景足够了。使用方法参考了<https://www.qiufengblog.com/articles/csharp-code-encryption.html>.obfuscator源码放在github上<https://github.com/obfuscator/obfuscator>

### 一. 安装obfuscator

obfuscator可以通过Nuget工具方便的安装。在Nuget安装界面搜索obfuscator后点击安装即可



二.新建obfuscator配置文件

在项目的根目录新建obfuscator的配置文件，如果项目中有许多子项目，建议在主项目里配置obfuscator，后续会说明原因。新建的配置文件名为obfuscator.xml。建好后选择其文件属性 复制到输出目录 如果较新则复制。



配置文件内容如下：

```
<?xml version="1.0" encoding="utf-8" ?>
<Obfuscator>
  <!--输入路径-->
  <Var name="InPath" value="." />
  <!--输出路径:加密混淆过的路径-->
  <Var name="OutPath" value=".\Obfuscator_Output" />
</Obfuscator>
```

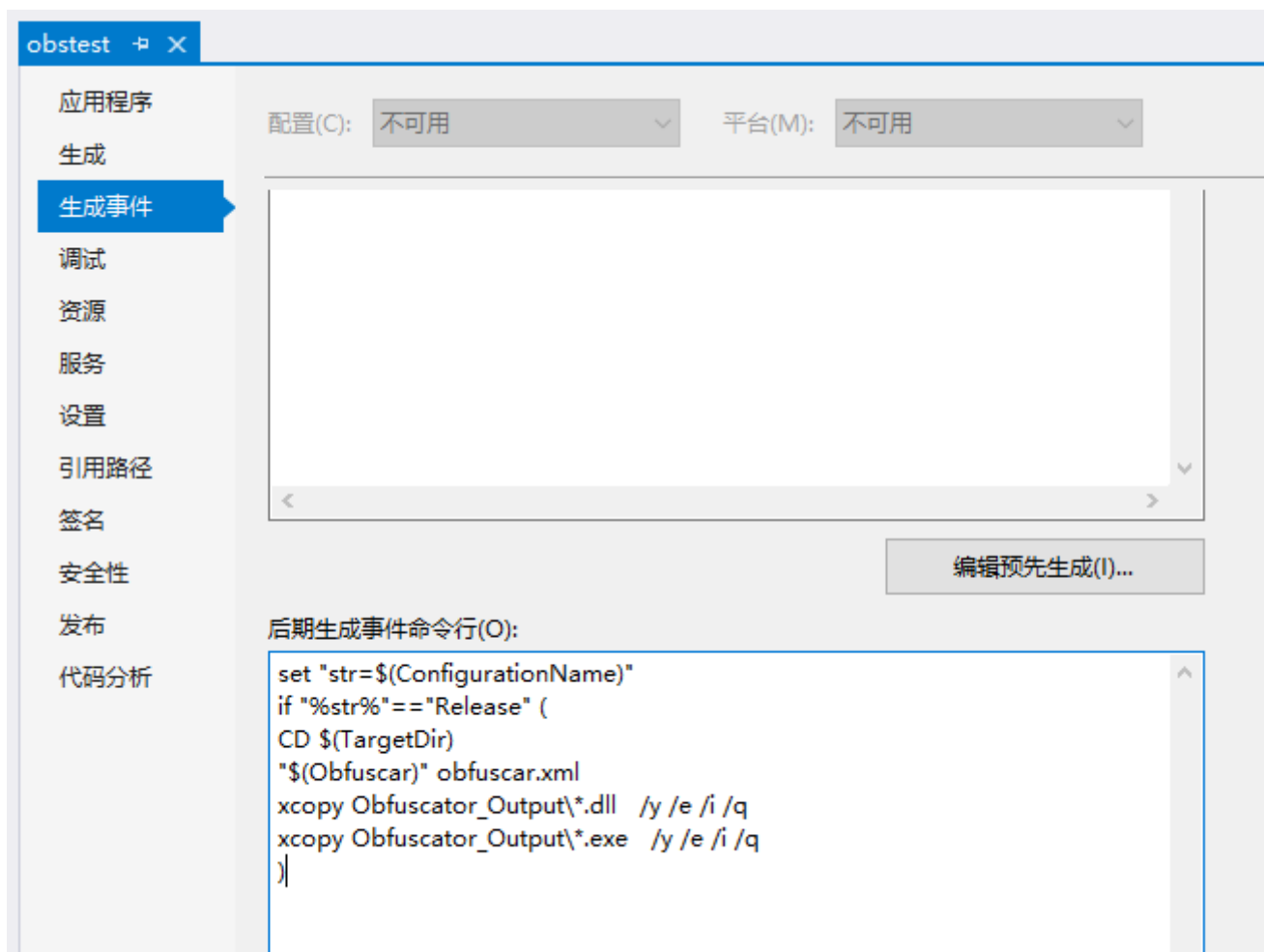
```
<!--混淆代码的参数-->
<Var name="ReuseNames" value="false" />
<Var name="HideStrings" value="true" />
<Var name="KeepPublicApi" value="false" />
<Var name="HidePrivateApi" value="true" />
<Var name="UseKoreanNames" value="true" />
<!--要混淆的模块-->
<Module file="$(InPath)\obstest.exe" />
</Obfuscator>
```

说明:

- 1.混淆的参数有很多，可以参考<https://docs.obfuscator.com/getting-started/configuration.html#settings> 进行添加或删除
- 2.需要混淆模块可以是exe 也可以是dll，可以一个也可以多个。实际使用时，需要改成自己的exe名字或dll名字

### 三.配置后期生成事件命令行

右击项目选择属性，在生成事件里填入如下内容，



```
set "str=$(ConfigurationName)"
if "%str%"=="Release" (
```

```

CD $(TargetDir)
"$$(Obfuscator)" obfuscator.xml
xcopy Obfuscator_Output\*.dll /y /e /i /q
xcopy Obfuscator_Output\*.exe /y /e /i /q
)

```

说明:

1. if "%str%"=="Release" 这句是限制了只release版本进行混淆。如果都有混淆，删掉即可。
2. CD \$(TargetDir) "\$\$(Obfuscator)" obfuscator.xml 这个两句是cd到生成目录。执行xml 进行配置好的混淆
3. xcopy Obfuscator\_Output\\*.dll /y /e /i /q xcopy是将混淆后的文件复制到生成目录 覆盖掉原来的问题。注意如果在子项目中运行这两句话，会导致主项目找不到dll 而生成失败。Obfuscator\_Output 这个文件夹不会主动创建，需要自己手动在生成目录创建一下。

#### 四.直接点击运行即可

如果运行成功，在生成的输出框里可以看到如下类似的内容

```

1>----- 已启动全部重新生成: 项目: obstest, 配置: Release Any CPU -----
1> obstest -> D:\vsproject\obtest\obtest\bin\Release\obtest.exe
1> Note that Rollbar API is enabled by default to collect crashes. If you want to opt out, please
run with -s switch
1> Loading project obfuscator.xml...Processing assembly: obstest, Version=1.0.0.0, Culture=neutral,
PublicKeyToken=null
1> Loading assemblies...Extra framework folders: Done.
1> Hiding strings...
1> Renaming: fields...Parameters...Properties...Events...Methods...Types...Done.
1> Saving assemblies...Done.
1> Writing log file...Done.
1> Completed, 0.61 secs.
1> 找不到文件 - *.dll
1> 复制了 0 个文件
1> 复制了 1 个文件
===== 全部重新生成: 成功 1 个, 失败 0 个, 跳过 0 个 =====

```

#### 五.工作原理和注意点

Obfuscator混淆功能实现的核心方法是 "\$\$(Obfuscator)" obfuscator.xml 这句话。调用命令行，运行 Obfuscator.Console.exe 这个程序，使用 obfuscator.xml配置文件。运行这个程序后，Obfuscator 会修改exe 和 dll中的函数名和变量名修改为没有意义的字符，达到简单混淆的目的。

但是由于修改了函数名，因此如果是给第三方用的dll，第三方就不能使用了。除非设置 `<Var name="KeepPublicApi" value="true" />`

另外，如果自己的项目里一个子项目被多个项目引用，混淆只能统一运行一次，即将所有需要混淆的exe 和 dll 放在一个xml里运行。不然会导致自己的项目找不到dll函数入口。

本条目发布于2021年11月10日 [<https://www.yangyouji.info/archives/772>]。属于csharp分类，被贴了 obfuscator 标签。

---