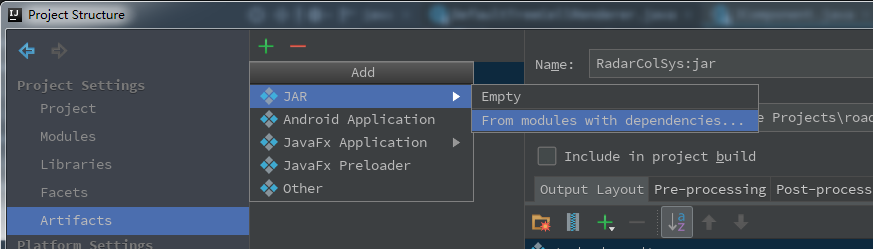
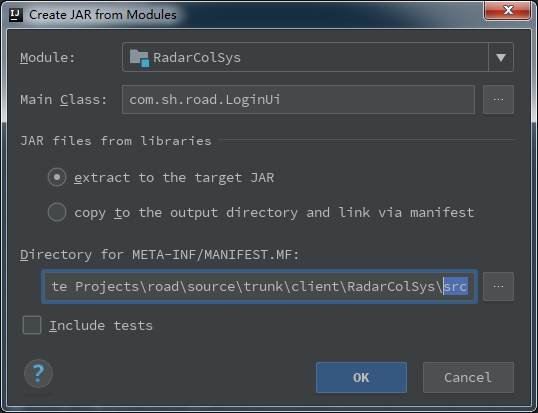
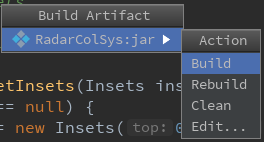
# 将java文件打包转成exe并整理成压缩包的操作步骤

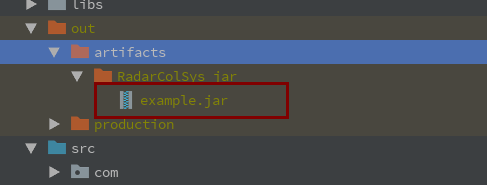
## **使用工具：**

* Intellij Idea community 2018.1 （开发工具）
* exe4j\_windows-x64\_6\_0 （exe转化工具）
* Inno Setup Compiler version 5.59 （安装包整理工具）

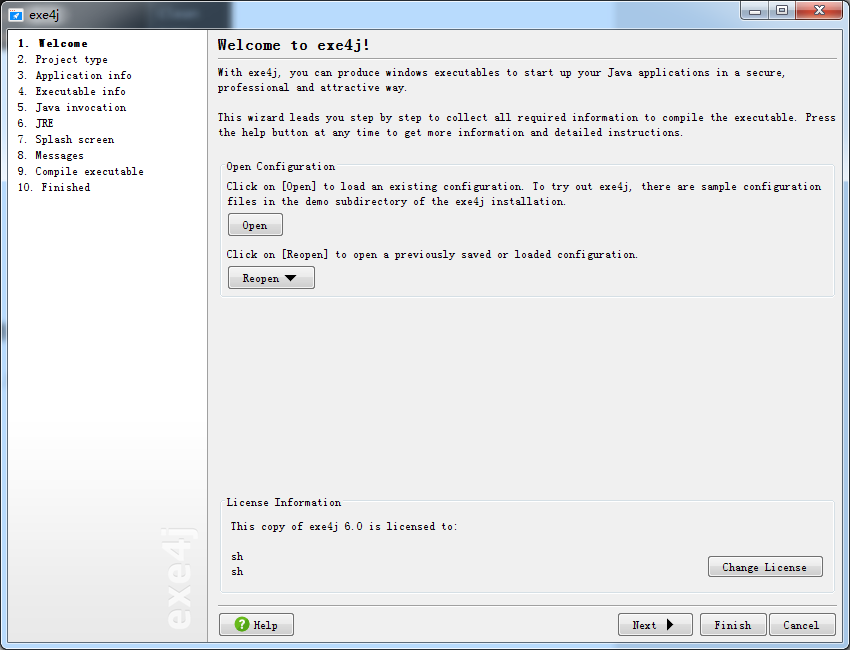
## 将java项目编译为可执行的jar包

1. 打开Project Structure，选择Artifacts，点击中间栏位上的‘+’号，在add弹窗中依次选择jar -> From modules with dependencies，如下图；
2. 在create Jar From modules弹窗中，在Main Class一行输入系统启动时执行的main方法，也可以通过点击右侧的小按钮在弹出窗中选择；然后在jar file from libraries 中选择extract to the target Jar，这个选项代表将jar文件提取到最后输出的可执行jar内部，而下一行的copy to output directory…则是将jar文件直接提取到可执行jar文件的同级目录；点击OK，进行下一步，如下图；
3. 在下一步的面板中直接点击apply -> OK即可；
4. 此时选择工具栏中的Build -> Build Artifacts; 弹出操作面板点击Build即可；

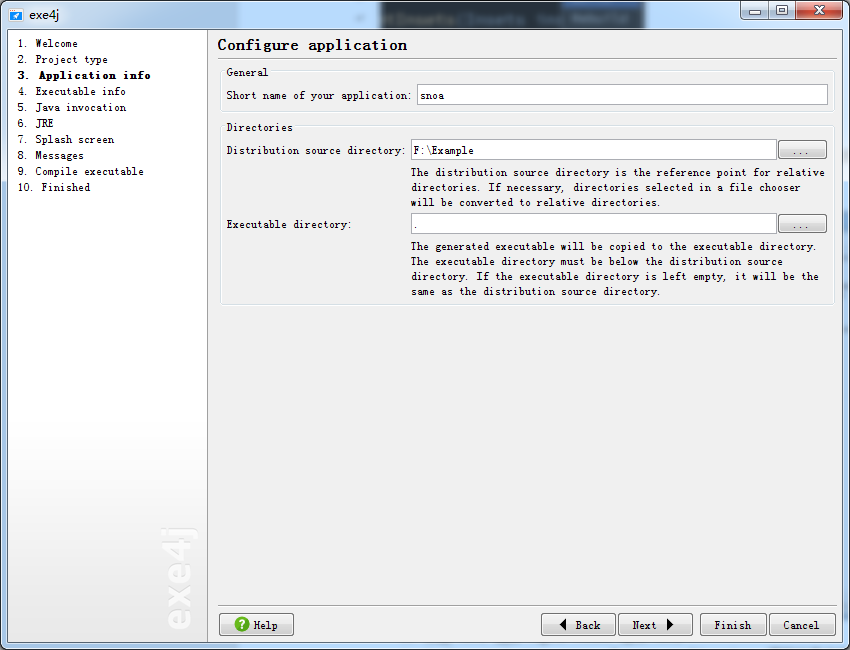
Build结束以后再根目录的 out-> artifacts 中可以看到我们生成的可执行jar文件



## 将可执行jar转化为exe文件

1. 创建一个文件夹Example，把项目中的resourse资源以及第一步得到的jar一起拷贝到文件夹目录中；
2. 打开exe.4j ；
3. 点击next，选择Regular mode，继续next；
4. Configure Application：name of your application 填写任意名字；

distribution source directory 点击右侧按钮，在文件选择器里选择刚刚创建好的文件夹，确定。

executable directory，可执行文件的路径，在文件选择器里选择刚刚创建好的文件夹，确定。

1. Configure executable

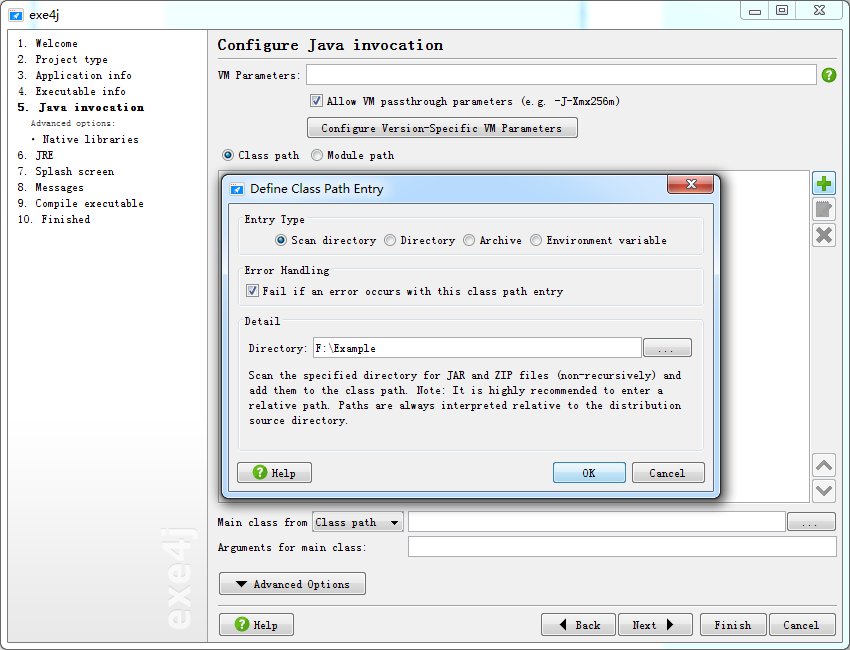
在executable name后填充exe文件的名字

一般要对exe设置图标，则选中icon File 并在文件选择器中选择\*.ico文件即可；

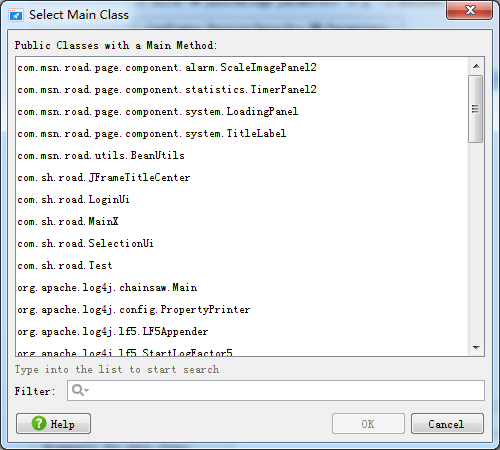
选中Allow only a single running instance of the application，表示这个应用最多只能有一个运行中的程序；

之后在advanced options 里点击’32-bit or 64-bit’,在该页面选中gererate 64-bit executable点击Next,并持续一直到Configure Java invocation页面。

1. Configure Java invocation：点击右侧+号，在弹出窗中选择第一步新建的文件夹路径拷贝到输入框中，同时要记得把Error Handling的单选框选中，点击OK；

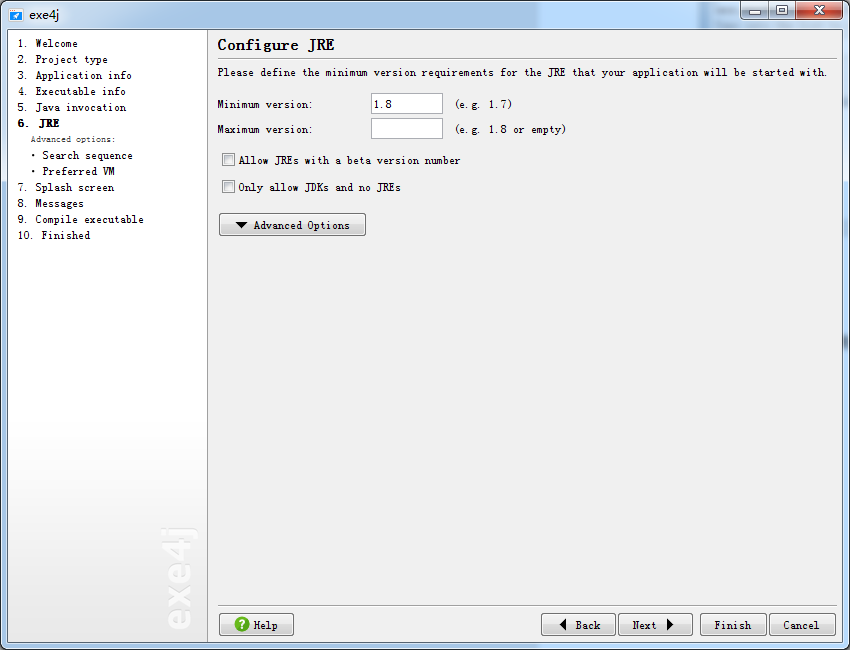


在Main class from后点击按钮，选择系统需要执行的main方法所在java文件

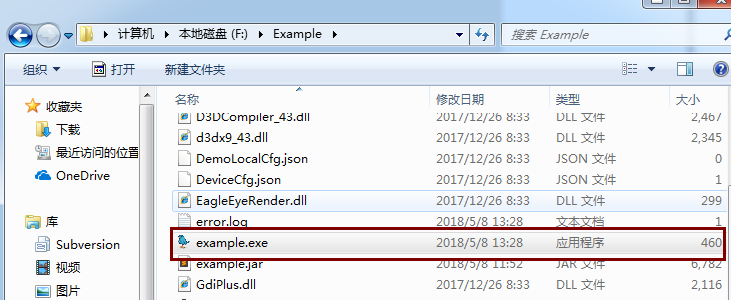


选中以后点击ok关闭弹窗；

点击next；

1. Configure JRE：在Minimum 后的输入框输入1.8，这里是最低的java版本号，鉴于项目中对java 8的使用，故选择最低1.8；点击Next
2. 一直点击next直至出现exe4j has finished页，点击click here tostart the application，可启动我们生成的exe。

Save Configuration 可以保存我们以上步骤的操作配置到本地，可以重复使用。

1. 退出exe4j，并打开我们创建的目录，（此处我的为F:\Example），可以看到我们的exe文件

双击效果图



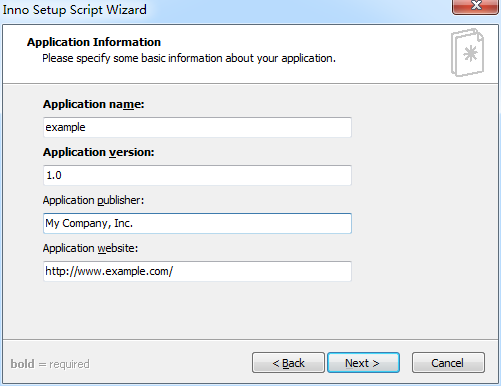
则exe文件转化执行成功。

## 将exe文件整理为安装包

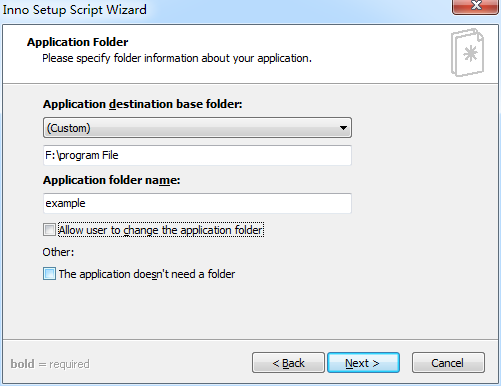
1. 首先打开我们的Inno setup compiler
2. Ctrl + N

点击Next；

1. 输入Application name 和 application version，应用名称和应用的版本信息



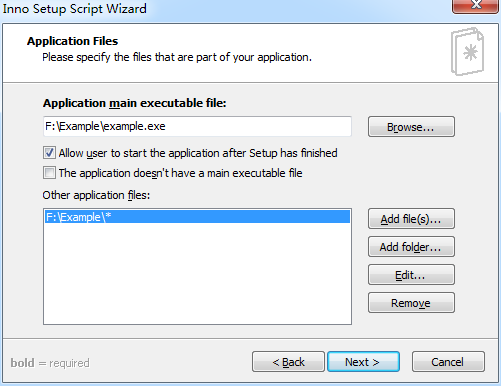
输入结束以后点击Next；



首先将base folder选择为custome，然后在下面的输入框填入安装包安装时的安装路径；

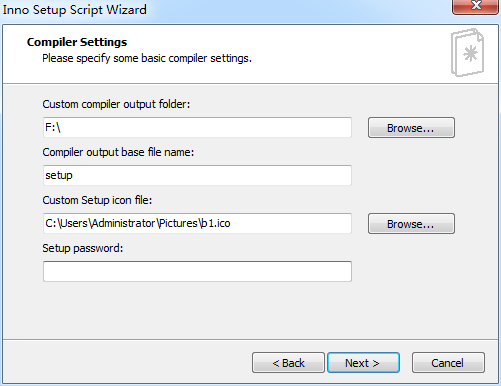
Application folder name填入安装路径中此项目的所在文件夹，点击Next；

5、



在Application main executable file下，点击browse，选择在第二步生成的exe文件；由于项目有一系列的\*.dll、\*.properties等依赖，故在add folder里选择最开始创建的文件夹（此文件夹之前已经把所有需要的外部环境导入），点击next；

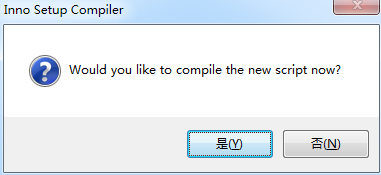
6、持续next直至以下页面，



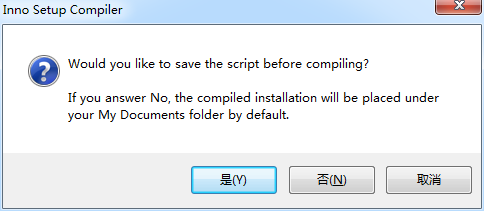
Custom compiler output folder填入安装包的输出路径，base file name是安装包的名称，

Setup icon file 是安装包的图标，其中名称和图标可以不填入，点击next，最后finish；

7、finish以后会有如下弹窗，点击是开始编译脚本

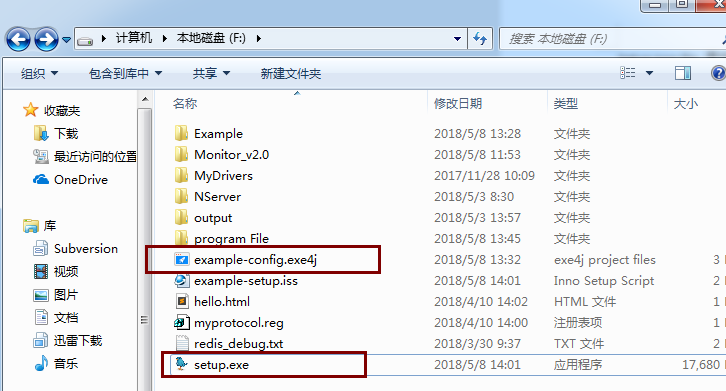


下图提示在编译之前保存脚本文件，选择否将会默认保存到计算机的文档目录下；



编译成功后关闭即可。

8、测试到我们设定的目录中寻找安装文件



可以看到安装文件已经被保存了。

结束。

后续：

此时我们打开安装包一直继续直到结束，程序正确运行；

以下为安装后的文件：

