



链滴

手把手将你的 Java maven 项目通过 Graal VM 打包成 windows 可执行程序

作者: [MingGH](#)

原文链接: <https://ld246.com/article/1678602264854>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

原文发布于：[手把手将你的Java maven项目通过GraalVM打包成windows可执行程序](#)，欢迎使用 [RSS 订阅](#) 获取最新更新。

1. 背景相关

作为一个程序员，总是会在工作的时候开发一些方便自己工作的程序，但是作为一个Java程序员，给自己或者朋友开发的办公小程序总是使用一个Jar包并不是一个很好的体验，主要是：

- 启动不方便
- 需要Java环境
- 有时候打出来的Jar比较大，不便于分享传输

但是现在有了GraalVM，让这些都不是问题了，直接生成可执行程序，丢哪哪就能用

2. GraalVM

GraalVM是一款高性能的虚拟机，它能够直接将Java程序编译成本地可执行文件，可以在不安装JVM情况下运行程序。当然了它的特性远不止这些，比如支持多语言，更低的内存占用等等，但这些并不这篇博客所涉及的，有兴趣可以去他们的[官网](#)里了解一下。

安装GraalVM JDK

下载界面：<https://github.com/graalvm/graalvm-ce-builds/releases/tag/vm-22.3.1>

这里用到的是Java 17版本：

GraalVM Community Edition 22.3.1 Latest

 ezzarghili released this Jan 25  vm-22.3.1  78e9d6d

GraalVM is a high-performance Java Development Kit based on OpenJDK and provides the Native Image feature for ahead-of-time execution of JavaScript, LLVM bitcode, and other languages on the JVM. GraalVM includes the runtime (JVM and class libraries) as tools.

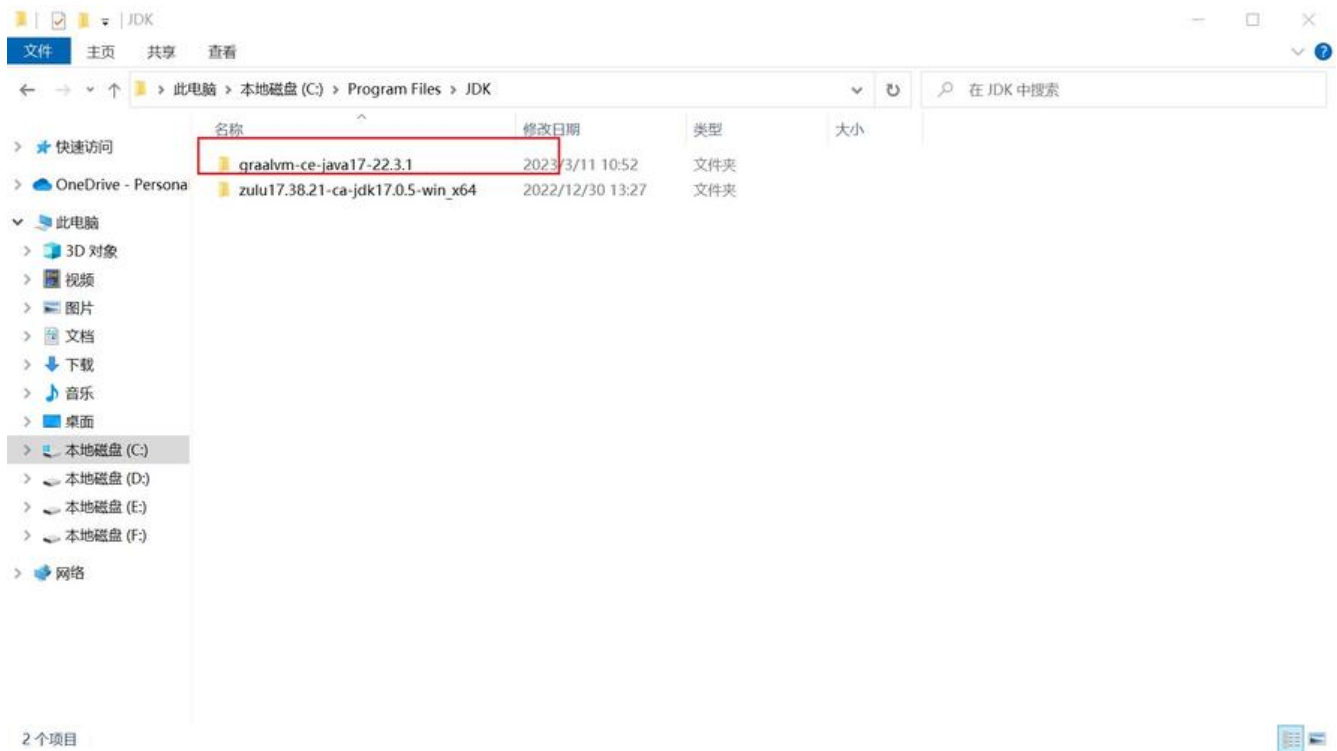
The release notes can be found on the website: https://www.graalvm.org/release-notes/22_3/#2231.

Here are the convenience links for the GraalVM base downloads:

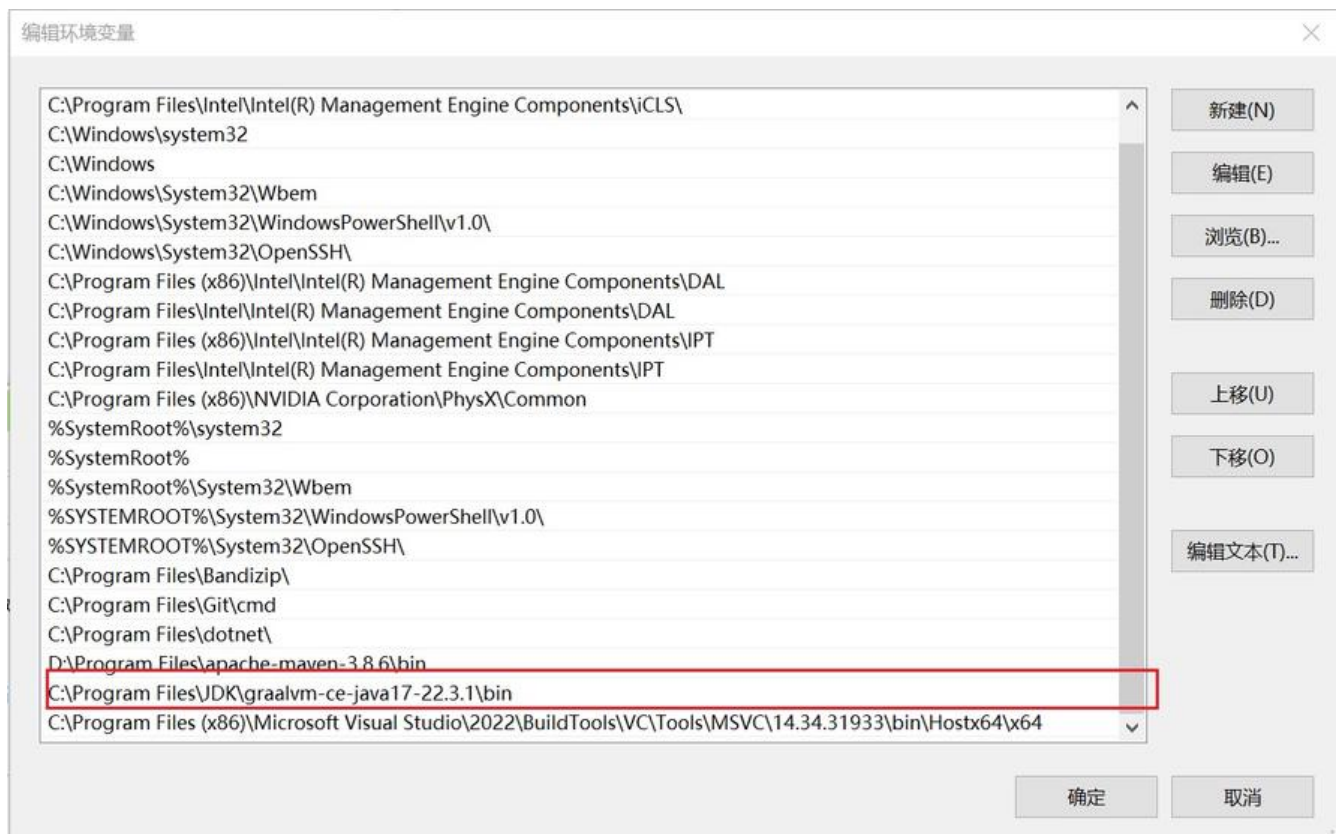
Platform	Java 11	Java 17	Java 19	
Linux (amd64)	 download	 download	 download	instructions
Linux (aarch64)	 download	 download	 download	instructions
macOS (amd64) †	 download	 download	 download	instructions
macOS (aarch64) †	 download	 download	 download	instructions
Windows (amd64)	 download	 download	 download	instructions

Based on the commit: [oracle/graal@vm-22.3.1](#).

下载完成之后，放入你本地的JDK目录，例如：



接着配置环境变量



简单的通过cmd测试一下

```
Microsoft Windows [版本 10.0.19044.2604]
(c) Microsoft Corporation。保留所有权利。

C:\Users\tangy>java -version
openjdk version "17.0.6" 2023-01-17
OpenJDK Runtime Environment GraalVM CE 22.3.1 (build 17.0.6+10-jvmci-22.3-b13)
OpenJDK 64-Bit Server VM GraalVM CE 22.3.1 (build 17.0.6+10-jvmci-22.3-b13, mixed mode, sharing)

C:\Users\tangy>
```

安装native-image

Native Image是一种将Java代码提前编译为独立可执行文件的技术，此刻执行文件包括应用程序类依赖、运行时库以及JDK静态连接的本地代码。Graalvm通过子模块SubstrateVM来支持Native Image，相比JVM其生成的程序具有更快的启动时间和更低的运行时开销。

需要在管理员模式下打开一个cmd窗口，然后执行：

```
gu install native-image
```

3.安装Visual studio

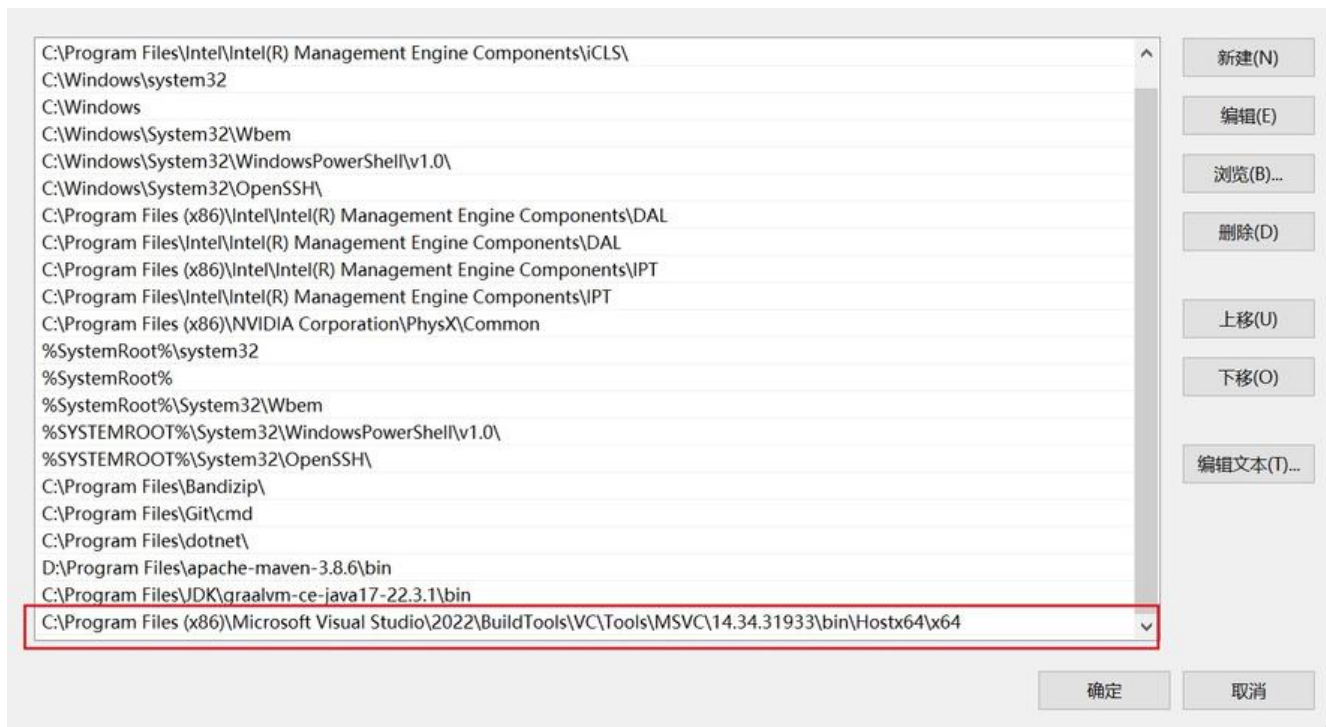
为了打包成exe可执行程序，这一步也是不可避免需要做的，这里推荐安装2019版本之后的，我安装的是visual studio 2022，下载地址：<https://visualstudio.microsoft.com/zh-hans/vs/>

安装完成以后我们还需要一些配置，不然直接进行打包会失败

配置cl.exe到系统的环境变量

安装完Visual studio后，你可以通过我的路径匹配找到你的 **cl.exe** 的路径，并配置cl.exe到系统的环境变量**。**

```
C:\Program Files (x86)\Microsoft Visual Studio\2022\BuildTools\VC\Tools\MSVC\14.34.31933
bin\Hostx64\x64
```



如果有问题：可以参考我之前踩过的坑：

【已解决】Default native-compiler executable 'cl.exe' not found via environment variable PATH

4. 将一个简单的Java应用打包成可执行程序

这里以一个简单的Java应用举例，它的功能是将剪切板上的一大串的xml文本解析为排序后的xml，后重新放回xml，方便我们平时在工作的时候对xml进行对比。

代码已经放在github，有需要的可以自取：<https://github.com/MingGH/clip-sort-xml>

这里贴一下主要代码部分，你可能不信，但它是ChatGPT写出来的，我只做了获取剪切板内容和放入切板这两个方法。。

添加graal-sdk依赖

```
<dependency>
  <groupId>org.graalvm.sdk</groupId>
  <artifactId>graal-sdk</artifactId>
  <version>21.3.0</version>
  <scope>provided</scope>
</dependency>
```

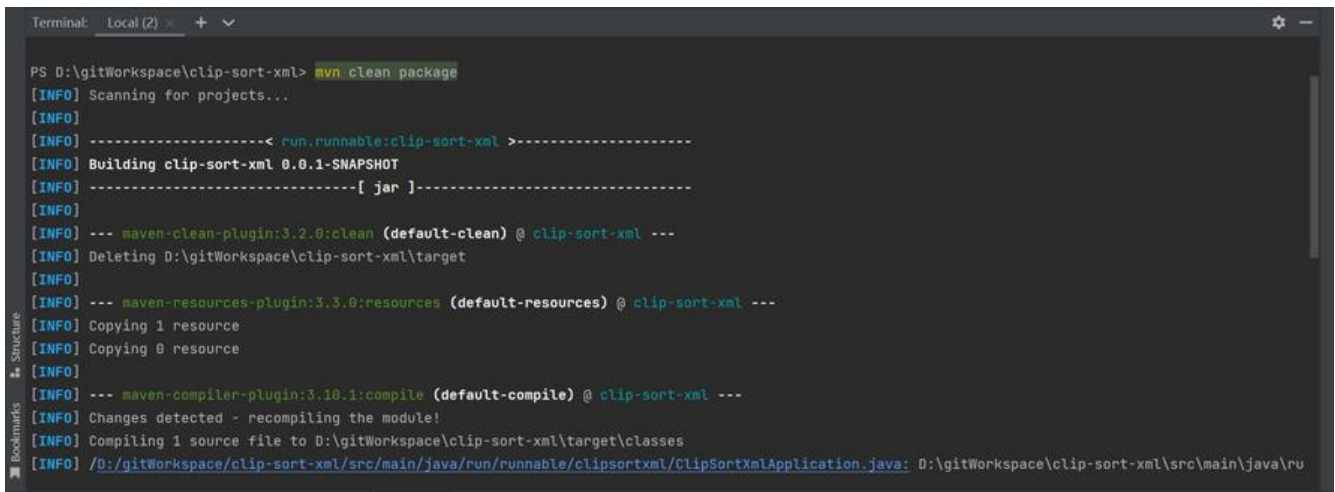
对maven项目使用跟踪代理生成反射配置

有了代码之后，我们需要对maven项目使用跟踪代理生成反射配置

需要先把你的代码打成jar包，虽然这里并不是SpringBoot项目，但是我使用的是spring-boot-maven plugin对项目进行打包。


```
<plugin>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-maven-plugin</artifactId>
  <version>3.0.4</version>
</plugin>
```

可以通过IDEA快速打包，或者通过命令 `mvn clean package`进行打包



```
Terminal: Local (2) x + v
PS D:\gitWorkspace\clip-sort-xml> mvn clean package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< run.runnable:clip-sort-xml >-----
[INFO] Building clip-sort-xml 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-clean-plugin:3.2.0:clean (default-clean) @ clip-sort-xml ---
[INFO] Deleting D:\gitWorkspace\clip-sort-xml\target
[INFO]
[INFO] --- maven-resources-plugin:3.3.0:resources (default-resources) @ clip-sort-xml ---
[INFO] Copying 1 resource
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.10.1:compile (default-compile) @ clip-sort-xml ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to D:\gitWorkspace\clip-sort-xml\target\classes
[INFO] /D:/gitWorkspace/clip-sort-xml/src/main/java/run/runnable/clipSortXml/ClipSortXmlApplication.java: D:\gitWorkspace\clip-sort-xml\src\main\java\ru
```

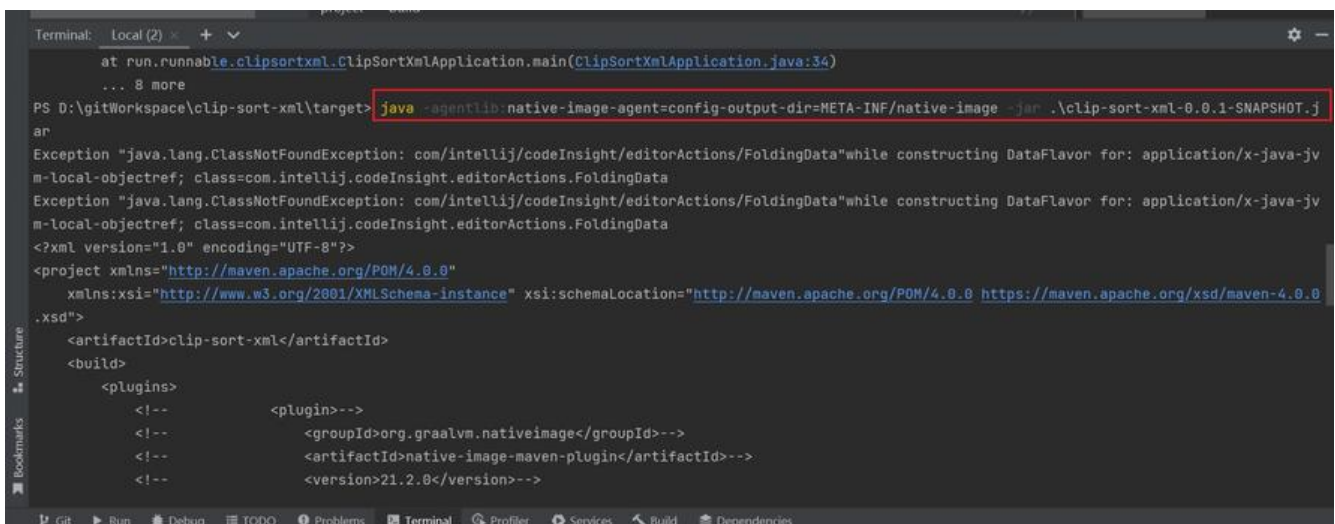
构建成功之后，我们进入target目录下： `cd target`

然后使用跟踪代理运行应用程序：

执行此命令前注意你的idea配置的JDK也需要是GraalVM JDK

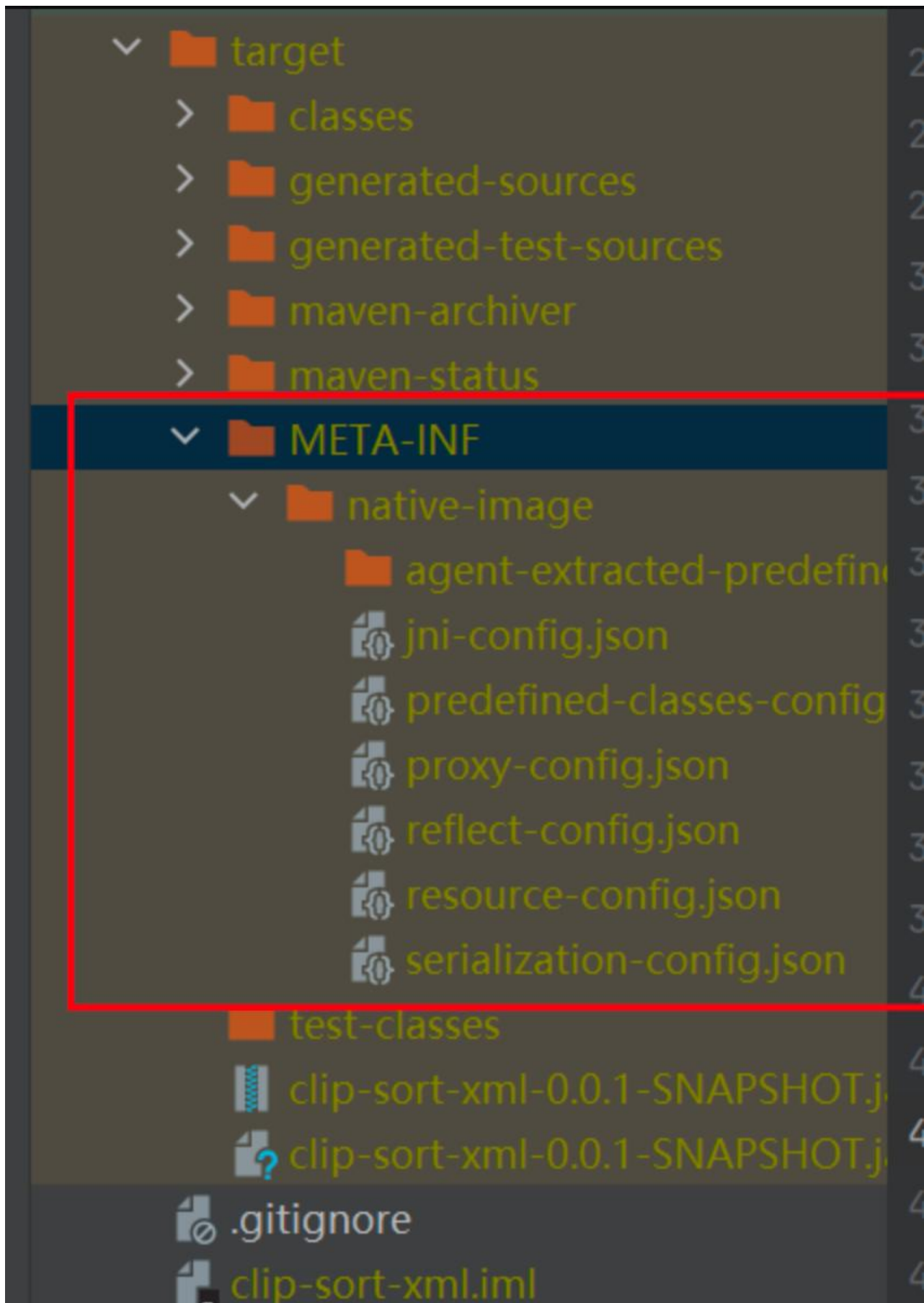
```
java -agentlib:native-image-agent=config-output-dir=META-INF/native-image -jar ./app.jar
```

执行时，会真实的启动你的项目并且执行，所以最好正常流程和异常流程都要测试到。

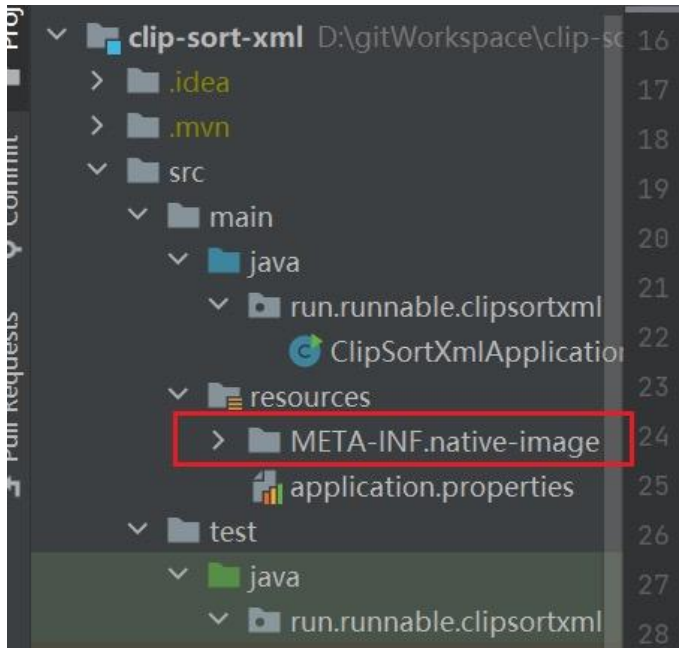


```
Terminal: Local (2) x + v
at run.runnable.clipSortXml.ClipSortXmlApplication.main(ClipSortXmlApplication.java:34)
... 8 more
PS D:\gitWorkspace\clip-sort-xml\target> java -agentlib:native-image-agent=config-output-dir=META-INF/native-image -jar ./clip-sort-xml-0.0.1-SNAPSHOT.jar
Exception "java.lang.ClassNotFoundException: com/intellij/codeInsight/editorActions/FoldingData"while constructing DataFlavor for: application/x-java-jvm-local-objectref; class=com.intellij.codeInsight.editorActions.FoldingData
Exception "java.lang.ClassNotFoundException: com/intellij/codeInsight/editorActions/FoldingData"while constructing DataFlavor for: application/x-java-jvm-local-objectref; class=com.intellij.codeInsight.editorActions.FoldingData
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <artifactId>clip-sort-xml</artifactId>
  <build>
    <plugins>
      <!--
      <plugin-->
      <!--
      <groupId>org.graalvm.nativeimage</groupId-->
      <!--
      <artifactId>native-image-maven-plugin</artifactId-->
      <!--
      <version>21.2.0</version-->
    </plugins>
  </build>
</project>
```

执行完成之后，会在target目录下生成一个 `META-INF`的文件夹，这个文件夹中包含的内容就是使用跟踪代理生成的反射配置。



把这个文件夹的内容复制到项目 `resources` 目录下

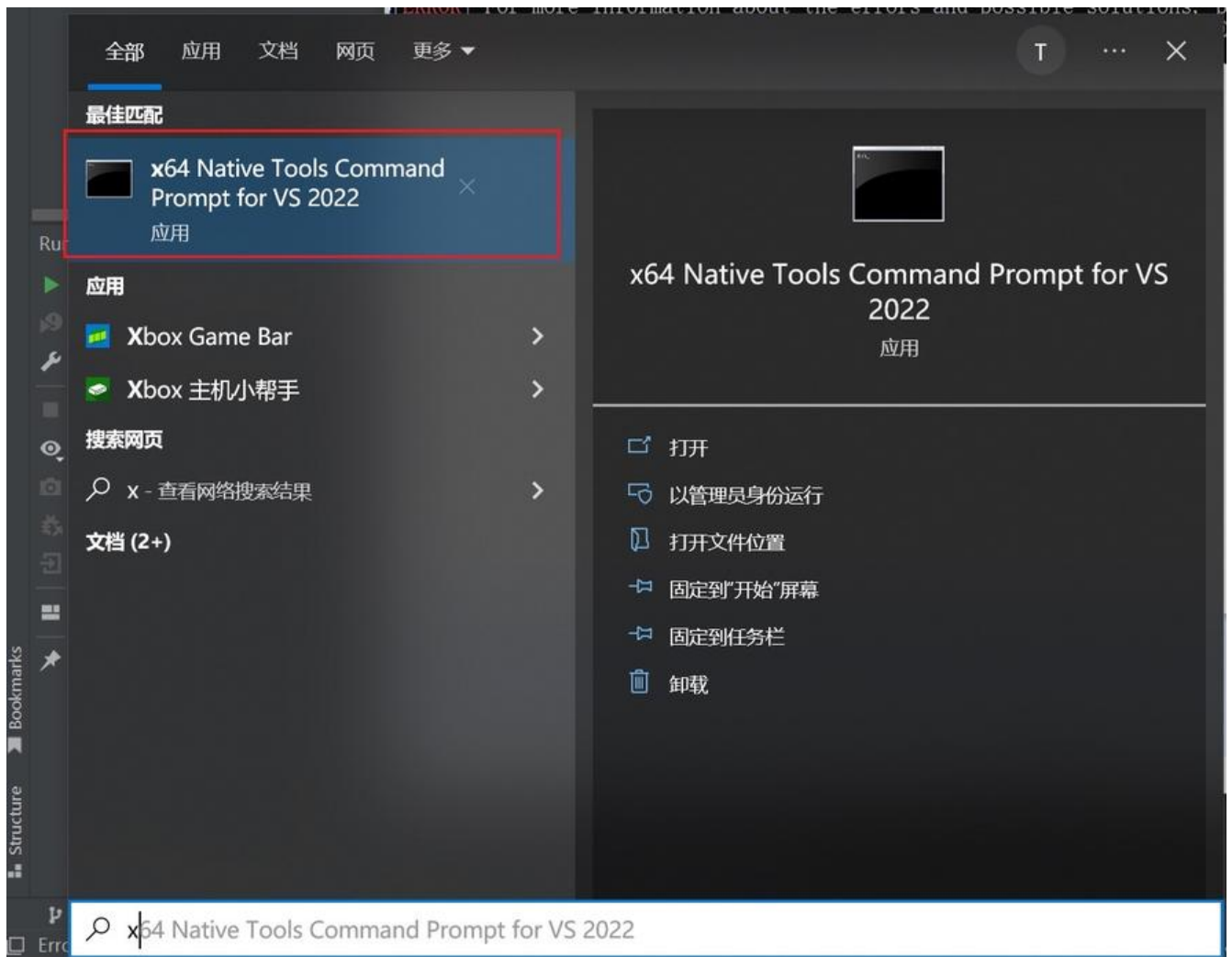


然后回到 `pom.xml` 文件，注释掉 `spring-boot-maven-plugin` 添加 `native-image-maven-plugin` 如下：

```
<plugin>
  <groupId>org.graalvm.nativeimage</groupId>
  <artifactId>native-image-maven-plugin</artifactId>
  <version>21.2.0</version>
  <executions>
    <execution>
      <goals>
        <goal>native-image</goal>
      </goals>
      <phase>package</phase>
    </execution>
  </executions>
  <configuration>
    <skip>false</skip>
    <imageName>graalvmMaven</imageName>
    <mainClass>run.runnable.clipsortxml.ClipSortXmlApplication</mainClass>
    <buildArgs>
      --no-fallback
    </buildArgs>
  </configuration>
</plugin>
```

通过 Native Tools Command Prompt 进行打包

现在接近尾声了，从你的windows搜索窗口，打开 `Native Tools Command Prompt`，然后进入到项目路径。



这个控制台我在执行的时候有点奇怪，只能在C盘执行，所以我把整个项目复制到了C盘的临时目录。

进入对应目录，然后执行：`mvn clean package` 你就会听到你的电脑起飞了，至少我的电脑是风扇转准备起飞了。

```

C:\Users\tangy\tempWorkspace\clip-sort-xml>mvn clean package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< run.runnable:clip-sort-xml >-----
[INFO] Building clip-sort-xml 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-clean-plugin:3.2.0:clean (default-clean) @ clip-sort-xml ---
[INFO] Deleting C:\Users\tangy\tempWorkspace\clip-sort-xml\target
[INFO]
[INFO] --- maven-resources-plugin:3.3.0:resources (default-resources) @ clip-sort-xml ---
[INFO] Copying 1 resource
[INFO] Copying 6 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.10.1:compile (default-compile) @ clip-sort-xml ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to C:\Users\tangy\tempWorkspace\clip-sort-xml\target\classes
[INFO] /C:/Users/tangy/tempWorkspace/clip-sort-xml/src/main/java/run/runnable/clipSortxml/ClipSortXmlApplication.java: C:\Users\tangy\tempWorkspace\clip-sort-xml/src/main/java/run/runnable/clipSortxml/ClipSortXmlApplication.java使用或覆盖了已过时的 API。
[INFO] /C:/Users/tangy/tempWorkspace/clip-sort-xml/src/main/java/run/runnable/clipSortxml/ClipSortXmlApplication.java: 有关详细信息, 请使用 -Xlint:deprecation 重新编译。
[INFO]
[INFO] --- maven-resources-plugin:3.3.0:testResources (default-testResources) @ clip-sort-xml ---
[INFO] skip non existing resourceDirectory C:\Users\tangy\tempWorkspace\clip-sort-xml\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.10.1:testCompile (default-testCompile) @ clip-sort-xml ---
[INFO] Changes detected - recompiling the module!
[INFO]
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @ clip-sort-xml ---
[INFO]
[INFO] --- maven-jar-plugin:3.3.0:jar (default-jar) @ clip-sort-xml ---
[INFO] Building jar: C:\Users\tangy\tempWorkspace\clip-sort-xml\target\clip-sort-xml-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- native-image-maven-plugin:21.2.0:native-image (default) @ clip-sort-xml ---
[INFO] ImageClasspath Entry: xalan:xalan:jar:2.7.2:compile (file:///C:/Users/tangy/.m2/repository/xalan/xalan/2.7.2/xalan-2.7.2.jar)
[INFO] ImageClasspath Entry: xalan:serializer:jar:2.7.2:compile (file:///C:/Users/tangy/.m2/repository/xalan/serializer/2.7.2/serializer-2.7.2.jar)
[INFO] ImageClasspath Entry: xerces:xercesImpl:jar:2.9.1:compile (file:///C:/Users/tangy/.m2/repository/xerces/xercesImpl/2.9.1/xercesImpl-2.9.1.jar)
[INFO] ImageClasspath Entry: xml-apis:xml-apis:jar:1.3.04:compile (file:///C:/Users/tangy/.m2/repository/xml-apis/xml-apis/1.3.04/xml-apis-1.3.04.jar)
[INFO] ImageClasspath Entry: run.runnable:clip-sort-xml:jar:0.0.1-SNAPSHOT (file:///C:/Users/tangy/tempWorkspace/clip-sort-xml/target/clip-sort-xml-0.0.1-SNAPSHOT.jar)
[WARNING] Major.Minor version mismatch between native-image-maven-plugin (21.2.0) and native-image executable (Unknown)
[INFO] Executing: C:\Program Files\JDK\graalvm-ce-java17-22.3.1\lib\svm\bin\native-image.exe -cp C:\Users\tangy\.m2\repository\xalan\xalan\2.7.2\xalan-2.7.2.jar;C:\Users\tangy\.m2\repository\xalan\serializer\2.7.2\serializer-2.7.2.jar;C:\Users\tangy\.m2\repository\xerces\xercesImpl\2.9.1\xercesImpl-2.9.1.jar;C:\Users\tangy\.m2\repository\xml-apis\xml-apis\1.3.04\xml-apis-1.3.04.jar;C:\Users\tangy\tempWorkspace\clip-sort-xml\target\clip-sort-xml-0.0.1-SNAPSHOT.jar --no-fallback -H:Class=run.runnable.clipSortxml.ClipSortXmlApplication -H:Name=graalvmMaven
=====
GraalVM Native Image: Generating 'graalvmMaven' (executable)...
=====
[1/7] Initializing... (6.8s @ 0.22GB)
Version info: 'GraalVM 22.3.1 Java 17 CE'
Java version info: '17.0.6+10-jvmci-22.3-b13'
C compiler: cl.exe (microsoft, x64, 19.34.31933)
Garbage collector: Serial GC
[2/7] Performing analysis... [*****] (24.4s @ 3.25GB)
6,942 (82.37%) of 8,428 classes reachable
13,548 (63.39%) of 21,371 fields reachable
33,991 (51.39%) of 66,148 methods reachable
208 classes, 93 fields, and 777 methods registered for reflection
94 classes, 116 fields, and 83 methods registered for JNI access
3 native libraries: crypt32, ncrypt, version
[3/7] Building universe... (3.2s @ 1.84GB)
[4/7] Parsing methods... [**] (2.7s @ 0.66GB)
[5/7] Inlining methods... [***] (1.2s @ 1.79GB)
[6/7] Compiling methods... [****] (19.6s @ 1.29GB)
[7/7] Creating image... (2.9s @ 2.27GB)
15.75MB (49.26%) for code area: 20,477 compilation units
15.95MB (49.88%) for image heap: 188,315 objects and 74 resources
280.34KB (0.86%) for other data
31.98MB in total
=====
Top 10 packages in code area:
788.22KB java.util
524.22KB java.lang.invoke
469.28KB c.s.org.apache.xerces.internal.impl.xs.traversers
457.89KB org.apache.xerces.impl
433.94KB com.sun.org.apache.xerces.internal.impl
418.55KB org.apache.xerces.impl.xs.traversers
418.46KB com.sun.crypto.provider
388.04KB java.io
377.12KB java.lang
323.31KB org.apache.xerces.impl.xs
11.11MB for 278 more packages

Top 10 object types in image heap:
3.20MB byte[] for code metadata
1.79MB java.lang.String
1.60MB java.lang.Class
1.55MB byte[] for general heap data
1.16MB byte[] for java.lang.String
766.75KB byte[] for embedded resources
551.72KB java.util.HashMap$Node
542.34KB com.oracle.svm.core.hub.DynamicHubCompanion
513.05KB int[]
360.30KB char[]
3.57MB for 1245 more object types
=====
1.4s (2.2% of total time) in 27 GCs | Peak RSS: 5.54GB | CPU load: 5.63
=====
Produced artifacts:
C:\Users\tangy\tempWorkspace\clip-sort-xml\target\awt.dll (jdk_lib)
C:\Users\tangy\tempWorkspace\clip-sort-xml\target\fontmanager.dll (jdk_lib)
C:\Users\tangy\tempWorkspace\clip-sort-xml\target\freetype.dll (jdk_lib)

```

当出现 **build Success**的时候就表示打包成可执行程序已经完成。

当中可能会碰到一些错误，我已经替你踩过一次坑了，所以可以直接参考我的解决办法：

【已解决】native-image - fatal error C1034: stdio.h: 不包括路径集

【已解决】native-image 抛错fatal error 1083

【已解决】Default native-compiler executable 'cl.exe' not found via environment variable PATH

完成上面的步骤之后，你会看到在target目录，哇，真的有一个.exe文件，迫不及待去点开试试你的能。

但是把，如果你是直接使用的我的这个项目，那么快乐还没有来的这么快。

双击之后你会发现一个黑色窗口飘过，然后啥也没发生，其实是因为代码执行出错了

解决Default flavor mapping not found错误

我们先按照正常方式进行排查，任意cmd窗口执行刚刚打包的那个.exe文件，就可以看到具体的错误信息：

```
C:\Users\tangy\tempWorkspace\clip-sort-xml\target>clip-sort-xml.exe
Exception in thread "main" java.lang.InternalError: Default flavor mapping not found
    at java.datatransfer@17.0.6/java.awt.datatransfer.SystemFlavorMap.initSystemFlavorMap(SystemFlavorMap.java:214)
    at java.datatransfer@17.0.6/java.awt.datatransfer.SystemFlavorMap.getNativeToFlavor(SystemFlavorMap.java:109)
    at java.datatransfer@17.0.6/java.awt.datatransfer.SystemFlavorMap.nativeToFlavorLookup(SystemFlavorMap.java:378)
    at java.datatransfer@17.0.6/java.awt.datatransfer.SystemFlavorMap.getFlavorsForNative(SystemFlavorMap.java:611)
    at java.desktop@17.0.6/sun.awt.datatransfer.DataTransferer.getFlavorsForFormats(DataTransferer.java:424)
    at java.desktop@17.0.6/sun.awt.datatransfer.ClipboardTransferable.<init>(ClipboardTransferable.java:88)
    at java.desktop@17.0.6/sun.awt.datatransfer.SunClipboard.getContents(SunClipboard.java:138)
    at run.runnable.clipsortxml.ClipSortXmlApplication.getFromClip(ClipSortXmlApplication.java:67)
    at run.runnable.clipsortxml.ClipSortXmlApplication.main(ClipSortXmlApplication.java:29)

C:\Users\tangy\tempWorkspace\clip-sort-xml\target>
```

这个错误也已经解决：

【已解决】native-image: Default flavor mapping not found

这里再重复一下：

这是 **agentlib native-image-agent.exe**

的bug，已经有人反馈在GraalVM的issues中，你可以在这个链接中发现其他人也碰到了这个问题：<https://github.com/oracle/graal/issues/5369>

解决办法也很简单，在你通过**跟踪代理**

生成的 **META-INF/native-image/resource-config.json**中的 **includes**

添加一行：**{ "pattern": "\\Qsun/datatransfer/resources/flavormap.properties\\E" }**


```

1  {
2  "resources":{
3  "includes":[
4  {
5  "pattern":
6  "\\QMETA-INF/services/javax.xml.parsers.DocumentBuilderFactory\\E"
7  },
8  {
9  "pattern":
10 "\\QMETA-INF/services/javax.xml.transform.TransformerFactory\\E"
11 },
12 {
13 "pattern":"\\Qorg/apache/xml/serializer/Encodings.properties\\E"
14 },
15 {
16 "pattern":"\\Qorg/apache/xml/serializer/output_xml.properties\\E"
17 },
18 {
19 "pattern":"\\Qsun/datatransfer/resources/flavormap.properties\\E" }
20 ]},
21 "bundles":[
22 {
23 "name":"org.apache.xml.serializer.XMLEntities",

```

保存然后重新执行 `mvn clean package`

这时打包出来的 .exe 文件才是真正可用的，这样一来一个 exe 文件，更方便丢给不懂代码的朋友了。

5. 参考内容

[\[native-image\] Swing application cannot be compiled on Windows](#)

[\[native-image\] makeShimDLL for java.dll is missing exports and fails](#)

[\[native-image\] java.home property not set](#)

[swing JTextField ctrl+v Throw Exception: java.lang.InternalError: Default flavor mapping not found](#)

[GraalVM Native Image 快速入门](#)

[【已解决】native-image - fatal error C1034: stdio.h: 不包括路径集](#)

[GraalVM与Spring Native初体验，一个让你的应用在100ms内启动的神器](#)

[【已解决】native-image: Default flavor mapping not found](#)

[【已解决】native-image 抛错fatal error 1083](#)

[【已解决】Default native-compiler executable 'cl.exe' not found via environment variable PATH](#)

[【已解决】native-image: Default flavor mapping not found](#)

[【已解决】Could not find agent library native-image-agent on the library path](#)

[ChatGPT](#)