jnaerator-java 调用动态库的神器,JNA 代码自动生成工具

作者：centrexzj
原文链接：https://ld246.com/article/1536648376886
来源网站：链滴

jnaerator-java调用动态库的神器,JNA代码自动生成工具

众所周知，java程序如果要调用动态库(.so,.dll)的函数，最传统方式是使用JNI技术，用JNI写java调接口代码是非常痛苦的，调试也是比较麻烦的事儿，JNA推出后，大大降低了开发难度，java程序员要为对应的动态库定义java native方法代码，为对应的C数据结构写出java对象，就可以了，不需要为了调用动态库而写c/c++程序。

相比JNI,JNA是一个很大的进步，但java程序还是要写一些java代码才能正确调用动态库，动态库函涉及的所有结构类型都需要定义对应的java类型，如果结构类型比较多是个很大的工作量。

能不能更简单一些呢?

最近正为写调用动态库的事儿头痛，虽然我也会写JNI代码，但实在太麻烦，总想找个捷径，看了JN相关资料后，发现用JNA所需要写的相关java代码其实都是很有规则的。既然这样，会不会有提供JN代码生成的开源工具呢?

在google上七找八找的，居然找到了。这就是jnaerator

jnaerator是google贡献的一个开源项目，用于生成基于JNA/BridJ的调用C/Object-C语言动态库的ja a代码的代码生成工具。有了这个神器，你可以不需要为了调用动态库而手工写哪怕一行代码。

本文以实际举例的方式，介绍jnaerator的简单用法。

jnaerator可以命令行执行，也可以以maven插件方式运行，本文只介绍命令行执行方式。

下载jar

命令行执行jnaerator需要下载jnaerator的FatJar包。

maven中央仓库下载地址

http://central.maven.org/maven2/com/nativelibs4java/jnaerator/0.12/jnaerator-0.12-shaded.jr

JNA代码生成

如下图目录结构，有两个dll,及相对应的头文件，我们这两个动态库为例说明如何用jnaerator来生成套JNA代码。

```
├─bin
│     THFeature.dll
│     THFaceImage.dll
└─include
      THFaceImage_i.h

      THFeature_i.h
```

命令行执行如下

```
java -jar d:\download\jnaerator-0.12-shaded.jar \
    -runtime JNA \
    -mode Maven \
    -mavenGroupId net.gdface \
    -mavenArtifactId cassdk_jna \
```

```
-o jna_code
-package net.gdface.jna
-f \
-library THFaceImage \
bin\THFaceImage.dll include\THFaceImage_i.h
-library THFeature \
bin\THFeature.dll include\THFeature_i.h
```

参数说明：

- runtime JNA

指定目标运行库为JNA,

- runtime 可选的值有(区分大小写)：

```
JNA
JNAerator (based on JNA)
BridJ
支持 C++库
NodeJS
但是实测发现使用BridJ 是有问题的，NodeJS没试过
```

- mode Maven

指定输出模式为Maven

- mode可选的值(区分大小写):

```
Jar : JAR 生成jar包,可以使用-jar指定生成的jar包文件名
StandaloneJar : 生成包含所有依赖库的jar ,可以使用-jar指定生成的jar包文件名
Directory : 生成代码到文件夹
Maven : 生成maven格式的项目(pom.xml)
AutoGeneratedMaven : 生成maven格式的项目(pom.xml)，执行mave install 自动编译生成jar包
不生成源码
```

- mavenGroupId net.gdface

指定 maven项目的 groupId

- mavenArtifactId cassdk_jna

指定 maven项目的 artifactId

- o jna_code

指定输出文件夹 jna_code

- package net.gdface.jna

指定生成java代码的包名。如果不指定，则默认包名为 library name

- f

生成代码时强制覆盖已经存在的文件

- library THFaceImage -library THFeature

指定后面的动态库的名称(library name),在这里为”THFeature.dll”，如果不指定则library name 为头文件名称：‘THFeature_i’，

NOTE:-library就是个状态参数，只对其后面的文件名参数有效，所以这里用两次-library分别为THFeature_i.h和THFeature_i.h指定了不同的动态库名称

bin\THFeature.dll include\THFeature_i.h bin\THFeature.dll include\THFeature_i.h

指定要生成代码的动态库和对应头文件，前后顺序无关，可以不提供动态库文件名称，只需要.h文件可以生成JNA代码

生成的maven项目代码结构如下

```
J:\WORKSPACE.NEON\CASSDK54\CASSDK\CASSDK_WINDOWS_X86_64\JNA_CODE
│  pom.xml
│
└─src
   └─main
      ├─java
      │  └─net
      │     └─gdface
      │        └─jna
      │              EF_Param.java
      │              FaceAngle.java
      │              THFeatureLibrary.java
      │              THFaceImageLibrary.java
      │              THFI_FacePos.java
      │              THFI_Param.java
      │              THFI_Param_Ex.java
      │              TH_Image_Data.java
      │
      └─resources
         └─lib
            └─win64
                 THFeature.dll
                 THFaceImage.dll
```

NOTE:如果生成代码时不提供bin\THFeature.dll,THFaceImage.dll，则生成的文件夹中没有resourc文件夹

参考资料

《jnaerator命令行参数说明》