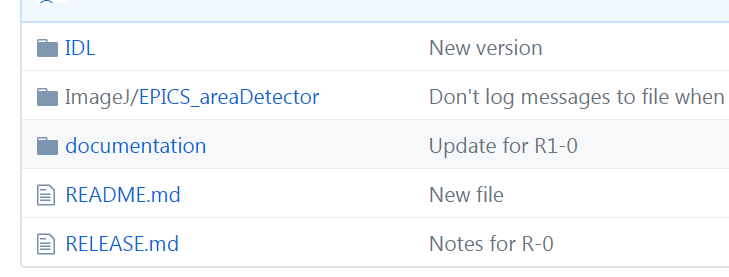
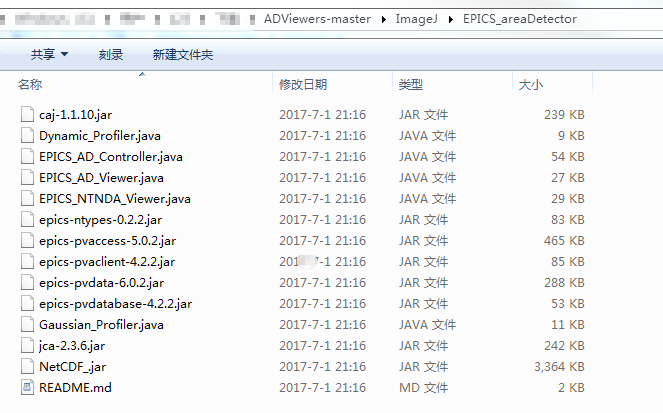
下载ADViewers，其结构入下：



README.md里面并未有太多有用信息。

打开ImageJ/EPICS\_areaDetector

其中，README.md里面有很多有趣的信息：

This directory contains the following ImageJ plugins

- EPICS\_AD\_Viewer.java

This plugin does real-time updates of images using the NDStdArrays plugin.

It supports taking a snapshot of the current frame into a new window and

collecting an image stack.

- EPICS\_AD\_Controller.java

This plugin supports the following operations:

- Defining the readout region of a detector from a rectangular ROI in ImageJ.

- Defining the ROI for NDPluginROI from a rectangular ROI in ImageJ.

- Defining a box or cross overlay in NDPluginOverlay using a rectangular or point

ROI in ImageJ.

- Defining a ellipse overlay in NDPluginOverlay using an oval or point ROI in ImageJ.

- Gaussian\_Profiler.java

This plugin does dynamic line profiles with real-time Gaussian profile

fitting. It is very useful for focusing and beam diagnostics. It was

written by Nouamane Laanait (previously at APS, currently at ORNL).

- Dynamic\_Profiler.java

This plugin does dynamic line profiles, i.e. line profiles where the plot

updates automatically when the image changes or when the line or rectangle

defining the profile is changed. This plugin was required in old versions

of ImageJ. However, in new versions of ImageJ the standard profile plot

tool has a "Live" option, which implements the capability of

Dynamic\_Profiler.java, so this plugin is no longer needed.

To uses these plugins in ImageJ do the following:

- Copy this folder into the ImageJ/plugins folder

- In ImageJ use the "Plugins/Compile and Run..." menu to compile the .java code

for the plugin. This produces .class files. This only needs to be done once.

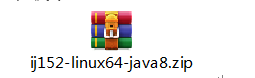
- The plugin will appear in the Plugins/EPICS\_areaDetector menu when ImageJ

is restarted.

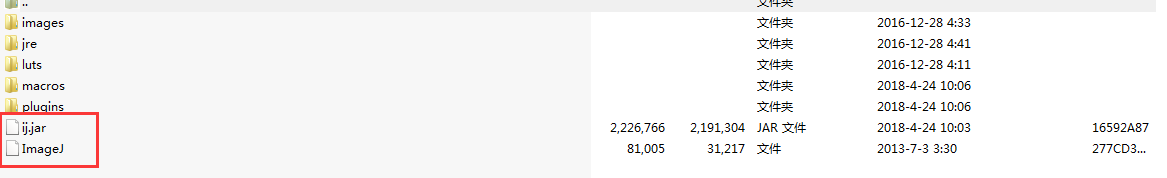
然而我并不会…..

…………………………………………………………………………………………

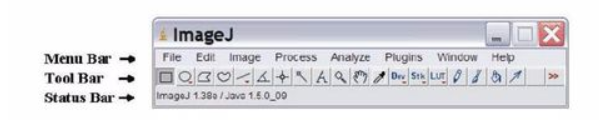
转头求助于Documentations对应网址：<http://cars.uchicago.edu/software/epics/areaDetectorViewers.html#ImageJController>其中给出ImageJ的下载链接<https://imagej.nih.gov/ij/>于是下载了



将其解压，它会生成ImageJ文件夹。



将ImageJ拷贝到ADViewers下面，解压，进入ImageJ文件夹，双击里面的ImageJ即可出现其界面，如下



可是按照说明使用.java：

To use these ImageJ plugins do the following:

* Install ImageJ from [ImageJ download site](https://imagej.nih.gov/ij/download.html). （已经下载，解压后生成ImageJ文件夹，将其放在ad/Appliaction下）
* Copy the entire directory ADViewers/ImageJ/EPICS\_areaDetector to the plugins/ directory in the ImageJ installation location. On OS X this can be done with the command:

cp -r Viewers/ImageJ/EPICS\_areaDetector /Applications/ImageJ/plugins

（已经将ADViewers/ImageJ/EPICS\_areaDetector中的文件全部拷贝至

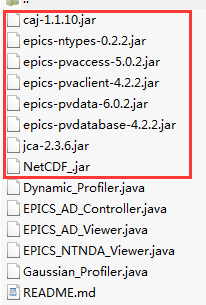
ad/Appliaction/ImageJ/plugins下面）

* The ImageJ plugin is supplied as Java source code, so you will need to compile the Java code. This can be done in the ImageJ Plugins/Compile and Run menu. Browse for the appropriate source file (EPICS\_AD\_Viewer.java, EPICS\_NTNDA\_Viewer.java, etc.) to compile and run it. The compilation step only needs to be done once, creating the required .class files.

利用命令$javac EPICS\_AD\_Viewer.java编译EPICS\_AD\_Viewer.java，会提示错误：程序包ij不存在….等等52个错误。

解决方法：我将ad/Appliaction/ImageJ/ij.jar拷贝至ad/Appliaction/ImageJ/plugins下，并利用$jar –xvf ij.jar将其解压缩

还有错。继续解压.jar，将由ADViewers/ImageJ/EPICS\_areaDetector拷贝过来的所有的.jar （如下图）都解压了



然后在输入$javac EPICS\_AD\_Viewer.java就不报错了。编译结束，生成很多EPICS\_AD\_Viewer$xxxxxxxx.class

接着，我继续输入

$javac EPICS\_NTNDA\_Viewer.java 无报错

$javac Dynamic\_Profiler.java 无报错

$javac Gaussian\_Profiler.java 无报错

$javac EPICS\_AD\_Controller.java 提示使用了不安全的错误，建议利用-Xlint：unchecked重新编译，因此输入：

$javac -Xlint：unchecked EPICS\_AD\_Controller.java有一个警告，我忽略了它，继续下面的步骤。

输入$java EPICS\_NTNDA\_Viewer错误: 在类 EPICS\_NTNDA\_Viewer 中找不到 main 方法, 请将 main 方法定义为: public static void main(String[] args) 否则 JavaFX 应用程序类必须扩展javafx.application.Application

解决方法：事实证明，我理解错了。

以上都是错误的操作。正确的方法：

ADViewers/ImageJ/EPICS\_areaDetector里面都是插件，将EPICS\_areaDetector整个文件夹拷贝至ad/Application/ImageJ/plugins下，运行ImageJ在Plugins按钮下选择Compile and Run，至此就编译好了。

弯路收获：

jar文件是java中的[class文件](https://www.baidu.com/s?wd=class%E6%96%87%E4%BB%B6&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)打包而成的，相当于类库

jar -xvf xxx.jar解压缩某个.jar文件

JAVA 一种编程语言，带有这样的扩展名的文件是 JAVA编译出来的源文件（就是代码文件）。

JAVA一般存在两种文件格式，如下：

1.\*.java文件是保存源代码的文本文件 (\*代表类名）

使用 javac \*.java可以编译该文件

使用 java \*可以运行该类

2.\*.class是用于保存 Java类的 二进制编码以及Class对象，每一个 Java类都有一个解释该类特征的 Class对象。\*.jar文件 是一种压缩文件格式

* The ImageJ viewers use the pure-Java libraries for EPICS Channel Access. This means that unlike the IDL Viewer, no C-based shareable-libraries or DLLs are needed. Prior to areaDetector R1-9 the ImageJ plugin used the standard Java mechanism for setting the EPICS Channel Access settings using a JCALibrary.properties file. This was often confusing because it uses a different mechanism than all C-based Channel Access clients, and because multiple JCALibrary.Properties files might be found in the Java search path, making it hard to figure out where a setting was coming from. Starting with areaDetector R1-9 the ImageJ plugin uses the same EPICS environment variables as Channel Access clients that use the C Channel Access library. Note that for EPICS\_AD\_Viewer the environment variable EPICS\_CA\_MAX\_ARRAY\_BYTES almost always needs to be set, because the default value of 16KB is rarely large enough for images. EPICS\_CA\_MAX\_ARRAY\_BYTES must be at least as large as the largest image size in bytes that you want to display. However, it is important not to set EPICS\_CA\_MAX\_ARRAY\_BYTES to an unnecessarily large value like 100 MB, because the EPICS CA library allocates buffers of size EPICS\_CA\_MAX\_ARRAY\_BYTES whenever the required buffer size is larger than 16KB. Remember also that EPICS\_CA\_MAX\_ARRAY\_BYTES must be set for both the IOC process and for the ImageJ client process. When using the V4 EPICS\_NTNDA\_Viewer it is not necessary to set EPICS\_CA\_MAX\_ARRAY\_BYTES on either the ImageJ client or the IOC processes.
* Start ImageJ and go to the Plugins/EPICS\_areaDetector/EPICS\_AD\_Viewer or EPICS\_NTNDA\_Viewer to run the plugin.
* For the EPICS\_AD\_Viewer type in PV prefix for the NDStdArrays plugin for the detector to be viewed (e.g. 13SIM1:image1:).
* For the EPICS\_NTNDA\_Viewer type in the PV name for the NDPluginPva plugin for the detector to be viewed (e.g. 13SIM1:Pva1:Image).
* The background color of the PV prefix or PV name will change to green and you should see message saying that the PVs have connected. If you don't the most likely problem is a firewall.
* Press the Start button to begin displaying images.

更多关于ImageJ的说明见<https://imagej.nih.gov/ij/docs/index.html>

备注：应用开源软件太少，对其工作原理没有整体认识。