MyBox: Easy Tools Set
Development Guide

Author: Mara

Version: 2.0

2019-11-18

Contents

1 About MyBox

1.1 Resource Addresses

This is GUI(Graphic User Interface) program developed in JavaFx, whose target is to provide simple and easy tools . It is free and open sources, and its main page is following:

https://github.com/Mararsh/MyBox

Source codes, compilered packages, and documents are under Releases directory:

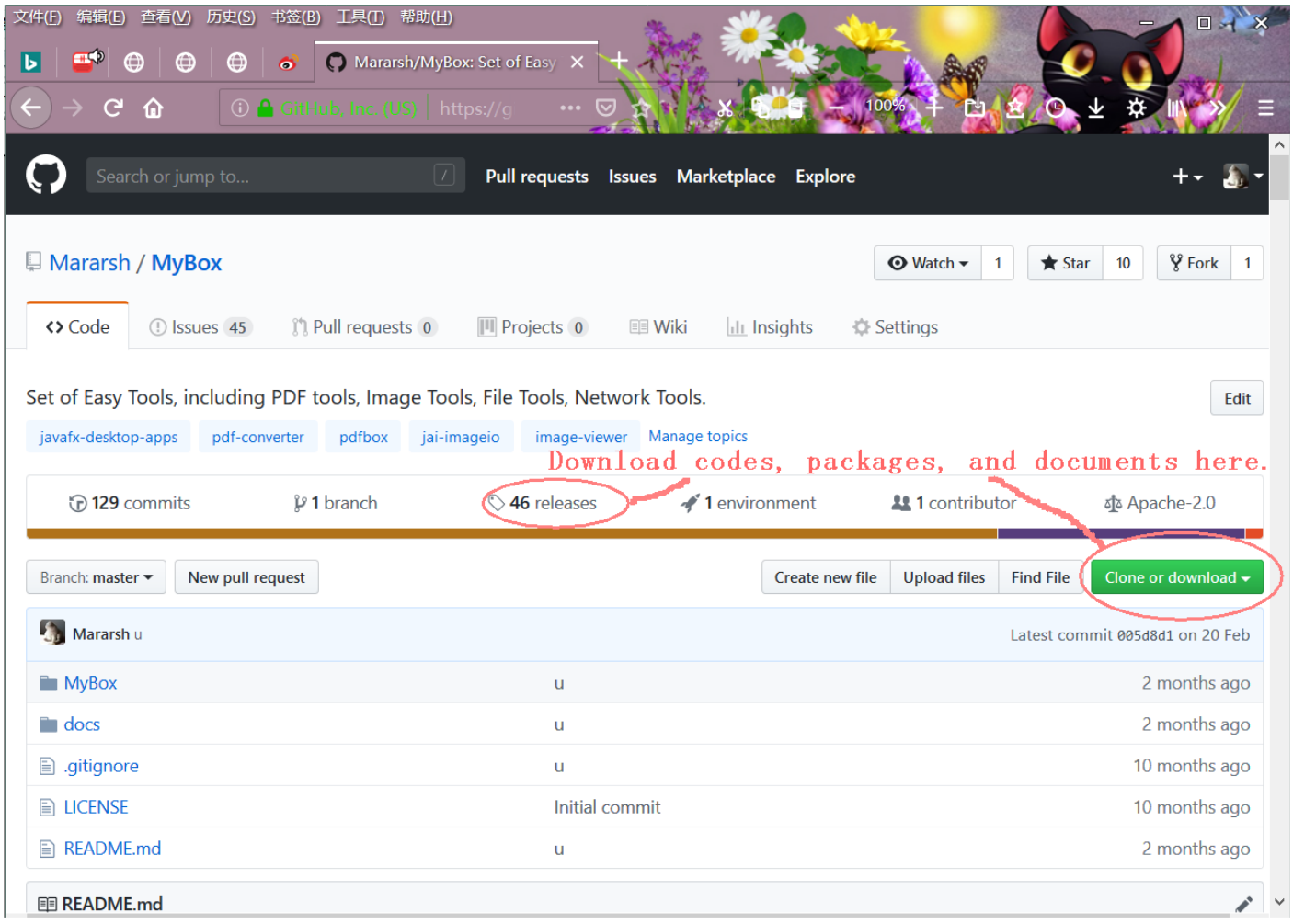
https://github.com/Mararsh/MyBox/releases

Welcome to submit software requirements and problem reports online:

https://github.com/Mararsh/MyBox/issues

Cloud storage:

https://pan.baidu.com/s/1fWMRzym_jh075OCX0D8y8A#list/path=%2F



1.2 Documents

This document introduces some summaries about MyBox development. It can be download from following address:

https://github.com/Mararsh/MyBox/releases/download/v5.8/MyBox-DevGuide-2.0-en.pdf

User Guides of MyBox include:

"MyBox User Guide – Overview"

https://github.com/Mararsh/MyBox/releases/download/v5.0/MyBox-UserGuide-5.0-Overview-en.pdf

"MyBox User Guide – PDF Tools"

https://github.com/Mararsh/MyBox/releases/download/v5.0/MyBox-UserGuide-5.0-PdfTools-en.pdf

"MyBox User Guide – Image Tools"

https://github.com/Mararsh/MyBox/releases/download/v5.0/MyBox-UserGuide-5.0-ImageTools-en.pdf

"MyBox User Guide – Desktop Tools"

https://github.com/Mararsh/MyBox/releases/download/v5.0/MyBox-UserGuide-5.0-DesktopTools-en.pdf

"MyBox User Guide – Network Tools"

https://github.com/Mararsh/MyBox/releases/download/v5.0/MyBox-UserGuide-5.0-NetworkTools-en.pdf

2 Source Codes

2.1 Develoment Resources

Both development and execution of MyBox depend only on open sources.

Current MyBox is based on Java 13 + openjfx 13 + Derby 15 + NetBean 11.

2.1.1 Softwares need download

To develop MyBox, following need be download:

Java 13 or higer (either Oracle JDK or openJDK)

https://www.oracle.com/technetwork/java/javase/downloads/index.html

http://openjdk.java.net/

NetBean 11.0 or higher

https://netbeans.org/

https://netbeans.apache.org/download/index.html

JavaFX Scene Builder 2.0

https://www.oracle.com/technetwork/java/javafxscenebuilder-1x-archive-2199384.html

If need modify/compile/build source codes, jpackage is need too. It is part of unrealesed jdk14:

https://jdk.java.net/jpackage/

2.1.2 Dependencies download automatically by Maven

MyBox depends on following too, but developers need not download them and Maven will fetch them automatically when sources are built:

JavaFx (It is not included in Java since Java 11, so JavaFx is added in project as dependencies)

https://gluonhq.com/products/javafx/

https://openjfx.io/

Derby:

http://db.apache.org/derby/

jai-imageio

https://github.com/jai-imageio/jai-imageio-core

PdfBox

https://pdfbox.apache.org/

Pdf2dom

http://cssbox.sourceforge.net/pdf2dom/

javazoom

http://www.javazoom.net/index.shtml

log4j

https://logging.apache.org/log4j/2.x/

tess4j

https://github.com/nguyenq/tess4j

tesseract

https://github.com/tesseract-ocr/tesseract

barcode4j

http://barcode4j.sourceforge.net

zxing

https://github.com/zxing/zxing

flexmark-java

https://github.com/vsch/flexmark-java

commons-compress

http://commons.apache.org/proper/commons-compress/index.html

2.1.3 Thirdparty codes referred directly

Following codes are copied as class files under source directory "MyBox\src\main\java\thridparty\".

GifDecoder:

https://github.com/DhyanB/Open-Imaging

EncodingDetect:

https://www.cnblogs.com/ChurchYim/p/8427373.html

2.1.4 Icons

Following website shares their resources with all of open sources projects:

https://icons8.com/icons/set/home

MyBox includes 5 sets of icons in different colors in which light-blue icons are download from above website and icons in other 4 colors are generated automatically by method "makeIcons()" of file "MainApp.java".

Icons	How to get	Color values
light-blue	download	#4788c7 #dff0fe
red	Reduce 215 of hue based on light-blue icons	#c94d58 #feb6be
pink	Increase 151 of red based on light-blue icons	#de88c7 #ffdcfe
blue	Increase 50% of saturation based on light-blue icons	#0066cc #92cbfe
orange	Reduce 171 of hue based on light-blue icons	#c8754b #fecdb6

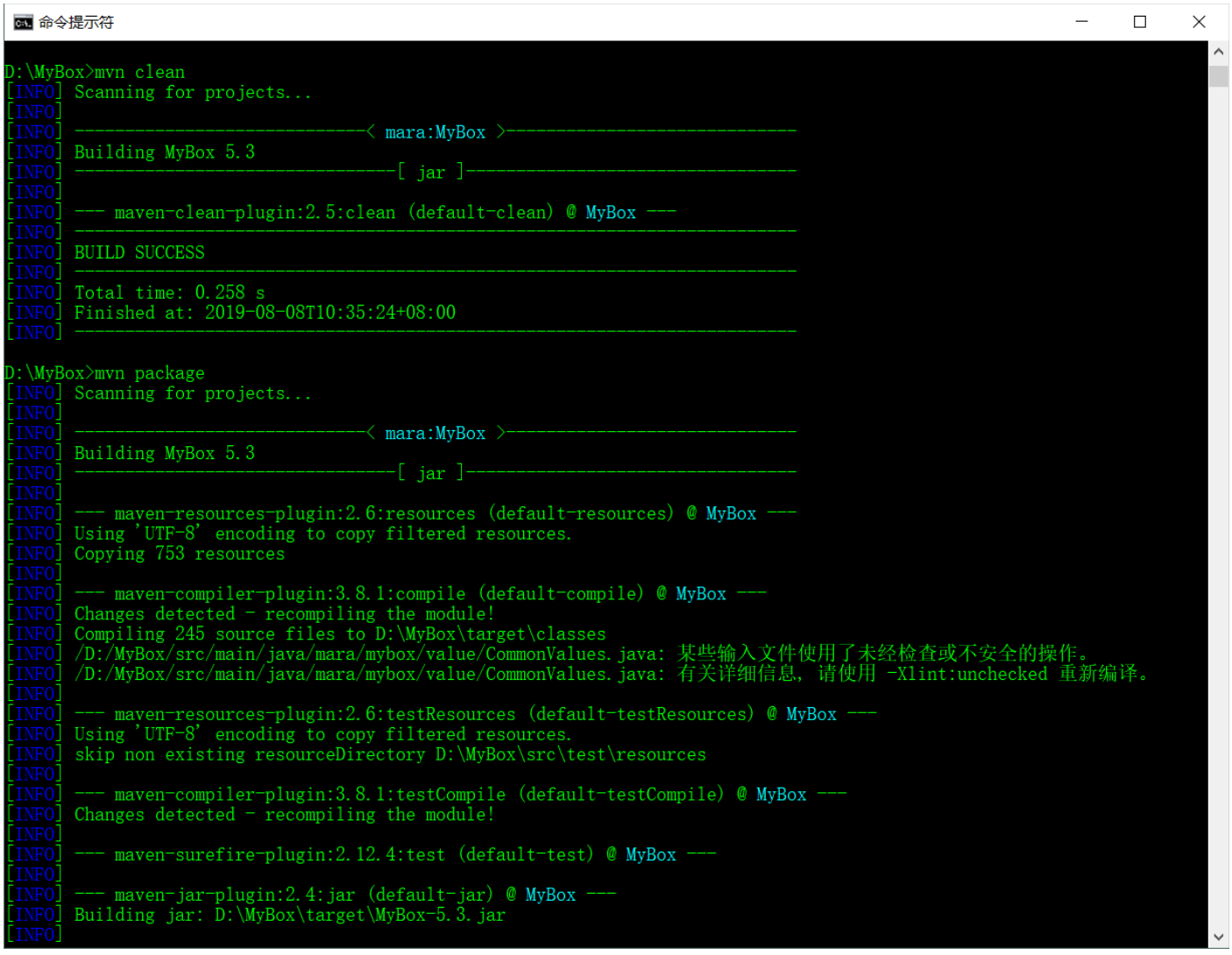
2.2 Build jar package by Maven

Project file “pom.xml” defines different profiles for different platforms, so package can be built against specified platform.

2.2.1 Build on win

Under root directory of MyBox source codes, execute following command: (Default profile is win)

```
mvn clean
mvn  package
```



2.2.2 Build on linux

Make sure JAVA_HOME is defined as jdk13.0.1, like "/usr/java/openjdk-13.0.1"

Under root directory of MyBox source codes, execute following command:

```
mvn clean
mvn -P linux package
```

2.2.3 Build on mac

Make sure JAVA_HOME is defined as jdk13.0.1, like "/usr/java/openjdk-13.0.1"

Under root directory of MyBox source codes, execute following command:

```
mvn clean
mvn -P mac package
```

2.2.4 Bulid cross-platform package

Cross-platform package can be built on any supported platform. It includes all resources for all platforms, so its size is larger.

Under root directory of MyBox source codes, execute following command:

```
mvn clean
mvn -P cross-platform package
```

2.2.5 Build package without Javafx

If user's env has openjfx 12.0.1 or higher installed, package without javafx can be built for them. The advantage is that the package is smaller, and the disadvantages are that javafx should be installed and that extra configuration is need for javafx.

Looks disadvantages are more, so no such package is made now.

2.3 Scripts for self-contain package

Self-contain package is that can run without extra requirements like Java env.

2.3.1 Tool “jpackage”

Tool “javapackager” was in Java 8 but removed since Java 9. A new tool “jpackage” is being developed by jdk14 team and they have provided early release:

https://jdk.java.net/jpackage/

It is really easy and need not installation. Just unpack it under any path.

Notice: jpackage itself is based on unreleased jdk14, and parameter “--runtime-image” can be used to define the jdk version for the target package.

2.3.2 Make exe package

There is a file named “**make-win-exe.bat**” under path “pack” of MyBox source codes. Double click it to generate self-contain package on Windows. Certainly, paths may need changed as developers' env.

The key statement of this script is:

```
D:\Programs\jdk-14\bin\jpackage  --package-type app-image  --app-version %version% --vendor Mara
--verbose --runtime-image  D:\Programs\Java\openjdk-13.0.1  --dest D:\tmp\make-mybox\out --name
MyBox --input D:\tmp\make-mybox\src --main-jar MyBox-%version%.jar  --icon D:\tmp\make-
mybox\res\MyBox.ico
```

The generated lanucher is “D:\tmp\make-mybox\out\MyBox\MyBox.exe”.

2.3.3 Make executable package on linux

There is a file named “**make-linux-bin.sh**” under path “pack” of MyBox source codes. Run it to generate self-contain package on Linux. Certainly, paths may need changed as developers' env.

The key statement of this script is:

```
../jdk-14/bin/jpackage  --package-type app-image --app-version $version --vendor Mara  --verbose
--runtime-image  /usr/java/openjdk-13.0.1  --dest  out  --name  MyBox  --input  src  --main-jar  MyBox-
$version.jar  --icon res/MyBox.png
```

The generated lanucher is “out/MyBox/bin/MyBox”.

2.3.4 Make dmg on mac

There is a file named “**make-mac-dmg.sh**” under path “pack” of MyBox source codes. Run it to generate self-contain package on Mac. Certainly, paths may need changed as developers' env.

The key statement of this script is:

```
../jdk-14/contents/Home/bin/jpackage  --package-type dmg --app-version $version --vendor Mara  --verbose
--runtime-image /Library/Java/JavaVirtualMachines/jdk-13.0.1.jdk/Contents/Home  --dest  out  --name
MyBox  --input  src  --main-jar  MyBox-$version.jar  --icon res/MyBox.icns
```

The generated lanucher is “out/MyBox-$version.dmg”.

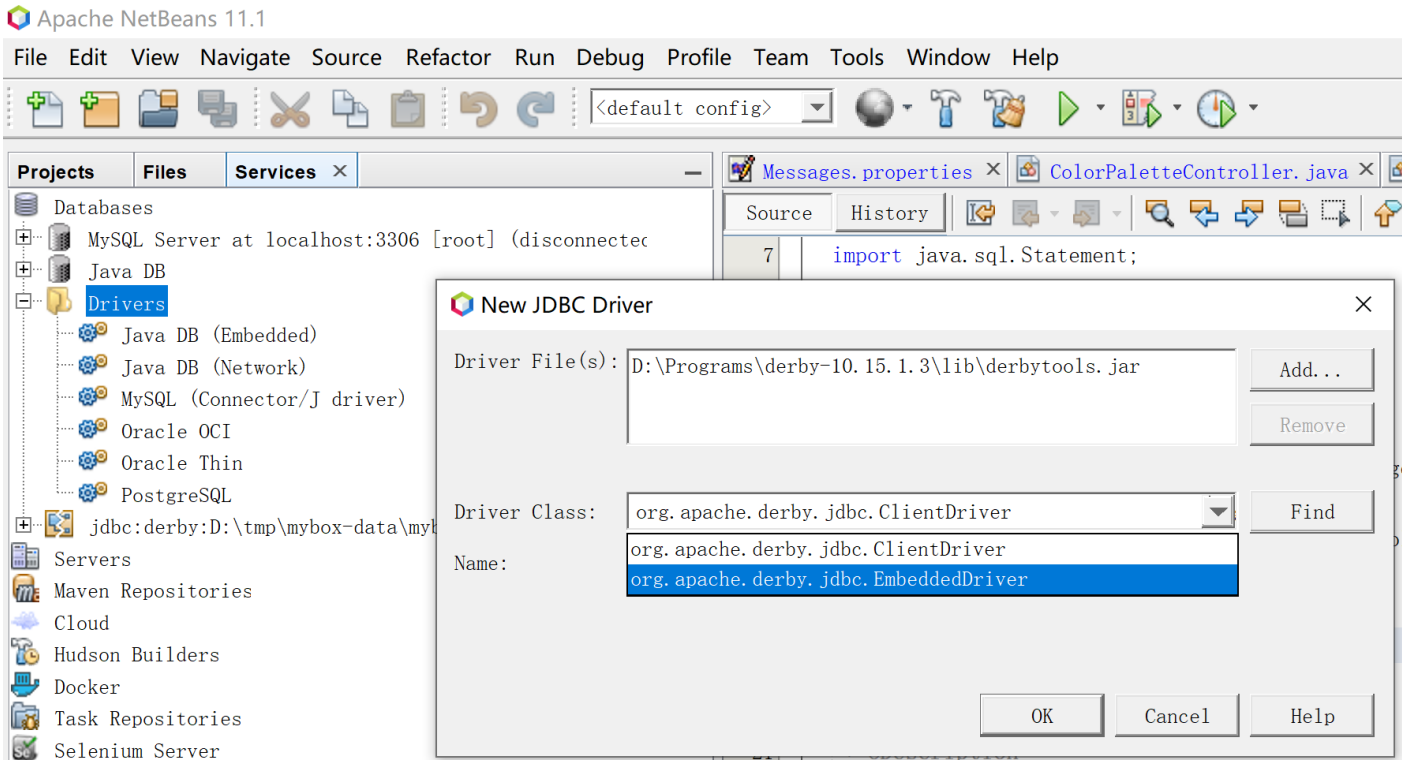
Notice: icns is need for packing on Mac. Script “**generate-mac-icns.sh**” under path “pack” can help to create it automatically. Certainly, paths may need changed as developers' env.

3 Coding with Netbeans

3.1 Config Derby

3.1.1 Add Derby Drivers

Find file “Derbytools.jar” in Derby installation package and use Netbeans “Services” - “Databases” - “Drivers” to add 2 Derby JDBC drivers “Client Driver” and “Embeded Driver”, like following graph:

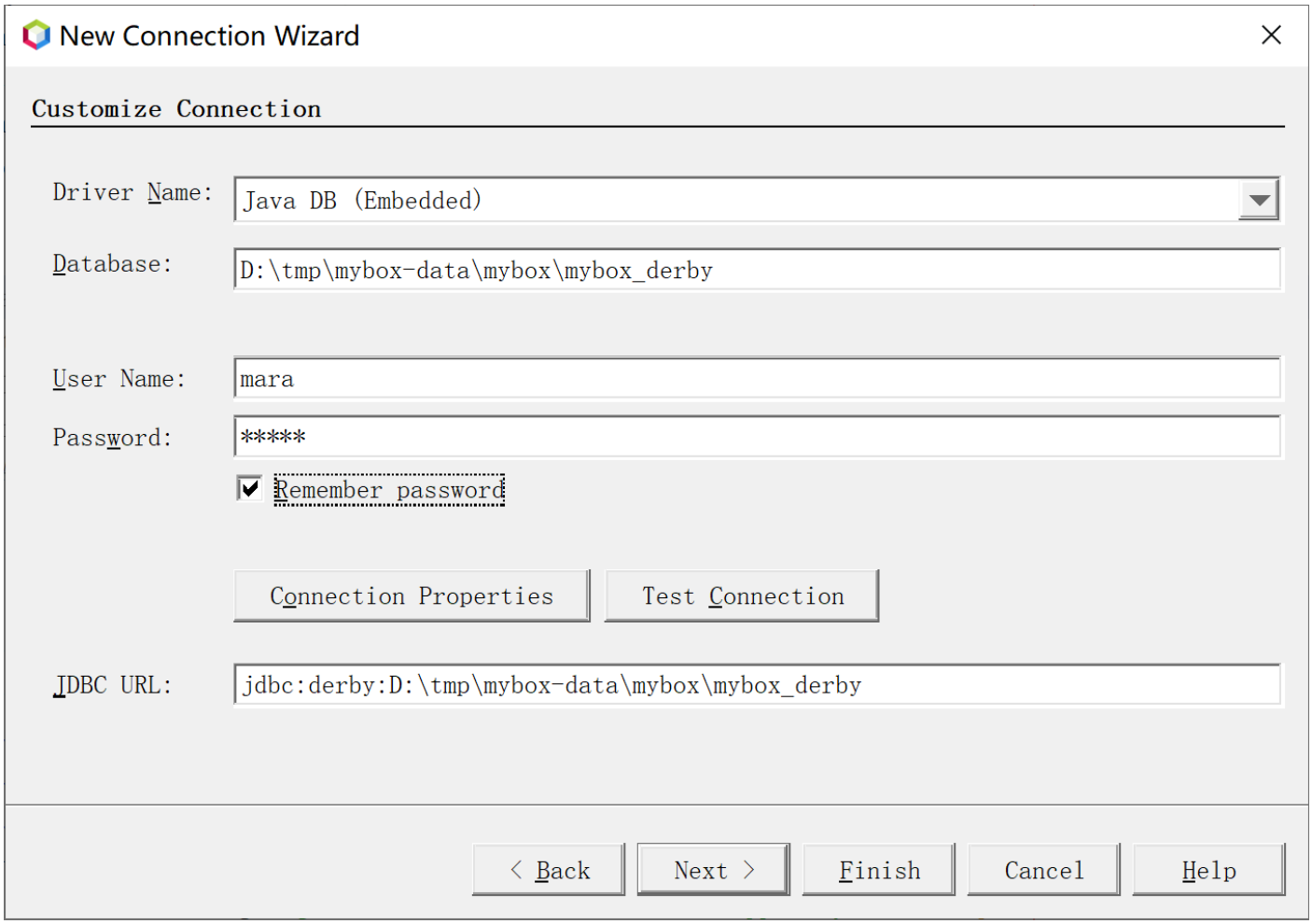


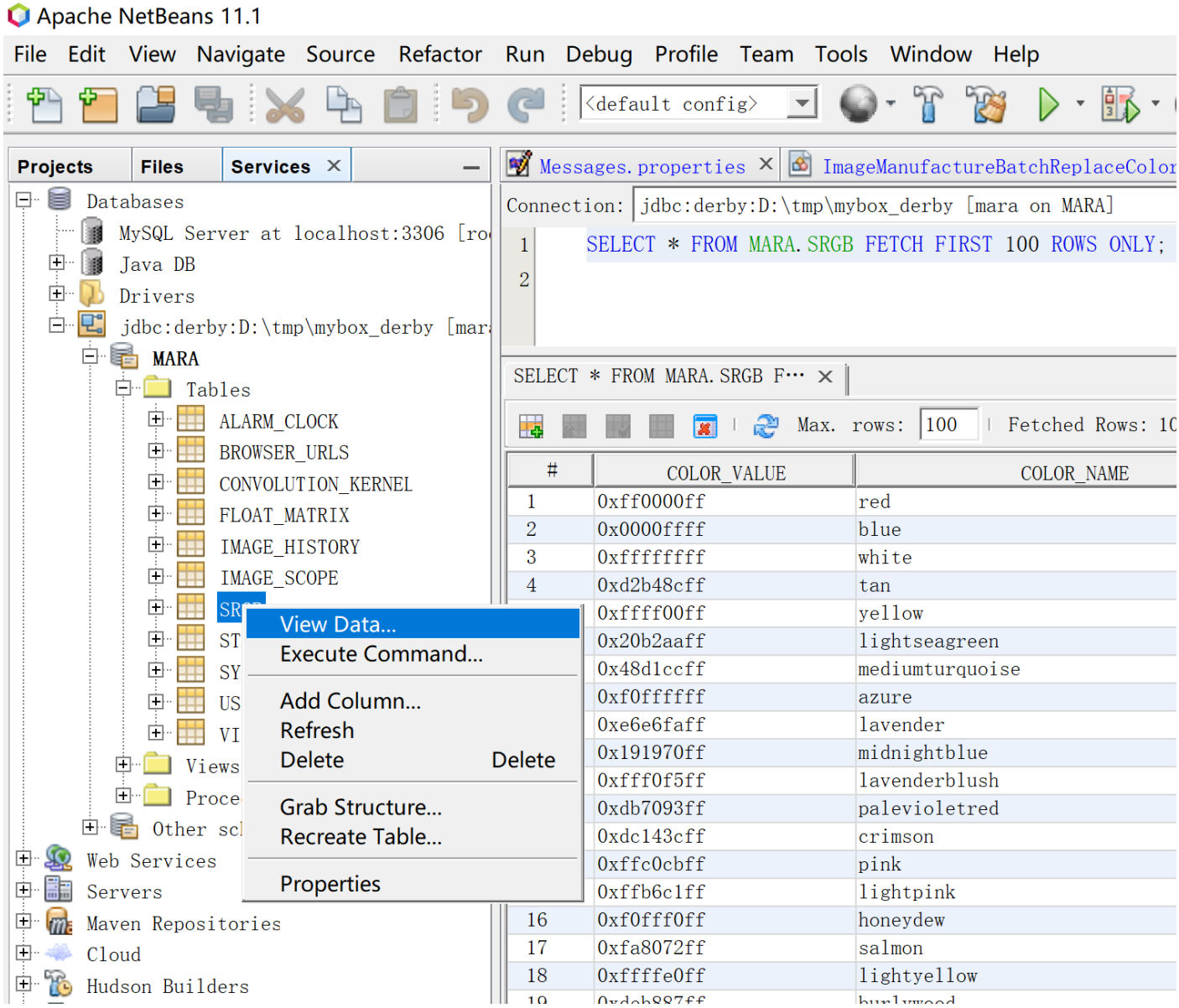
3.1.2 Start/Connect to Derby as emdedded mode

Notice: Only one application can accesse derby in this mode. NetBeans can not visit derby when other JVM or application has connected it. And others can not visit derby when NetBeans has connected it.

When MyBox is not started, developer can use this mode to start derby and check data.

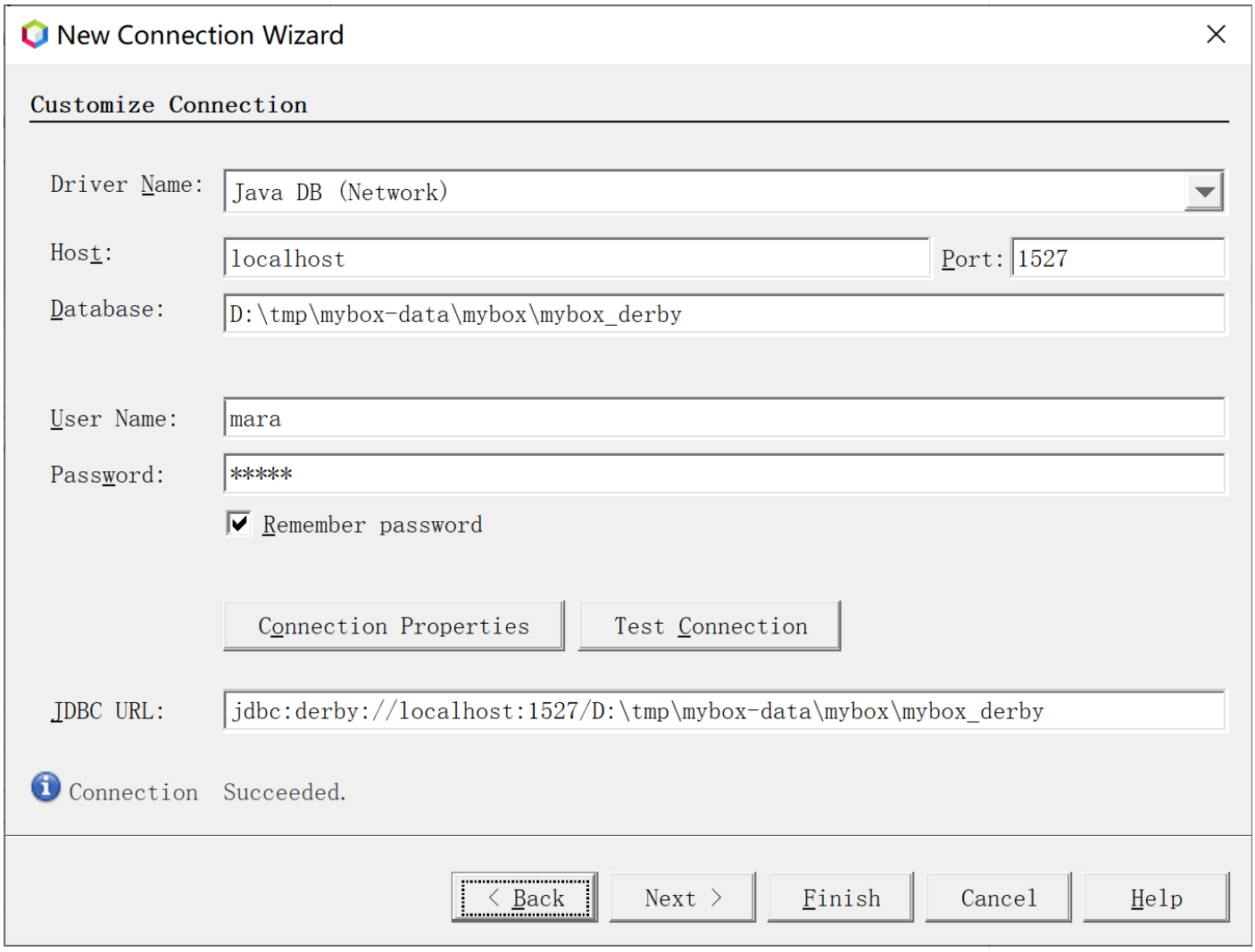
Before debug MyBox, make sure connection in embedded mode has been closed to avoid MyBox fails to visit db.




Apache NetBeans 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Projects Files Services
Databases
MySQL Server at localhost:3306 [ro
Java DB
Drivers
jdbc:derby:D:\tmp\mybox_derby [mar
MARA
Tables
ALARM_CLOCK
BROWSER_URLS
CONVOLUTION_KERNEL
FLOAT_MATRIX
IMAGE_HISTORY
IMAGE_SCOPE
SR
ST
SY
US
VI
Views
Proce
Other sc
Web Services
Servers
Maven Repositories
Cloud
Hudson Builders
View Data...
Execute Command...
Add Column...
Refresh
Delete Delete
Grab Structure...
Recreate Table...
Properties
Messages.properties
ImageManufactureBatchReplaceColor
Connection: jdbc:derby:D:\tmp\mybox_derby [mara on MARA]
1 SELECT * FROM MARA.SRGB FETCH FIRST 100 ROWS ONLY;
2
SELECT * FROM MARA.SRGB F···
Max. rows: 100
Fetched Rows: 10
COLOR_VALUE COLOR_NAME
1 0xff0000ff red
2 0x0000ffff blue
3 0xffffffff white
4 0xd2b48cff tan
0xffff00ff yellow
0x20b2aaff lightseagreen
0x48d1ccff mediumturquoise
0xf0ffffff azure
0xe6e6faff lavender
0x191970ff midnightblue
0xfff0f5ff lavenderblush
0xdb7093ff palevioletred
0xdc143cff crimson
0xffc0cbff pink
0xffb6c1ff lightpink
16 0xf0fff0ff honeydew
17 0xfa8072ff salmon
18 0xfffe0ff lightyellow

3.1.3 Connect to derby in client mode

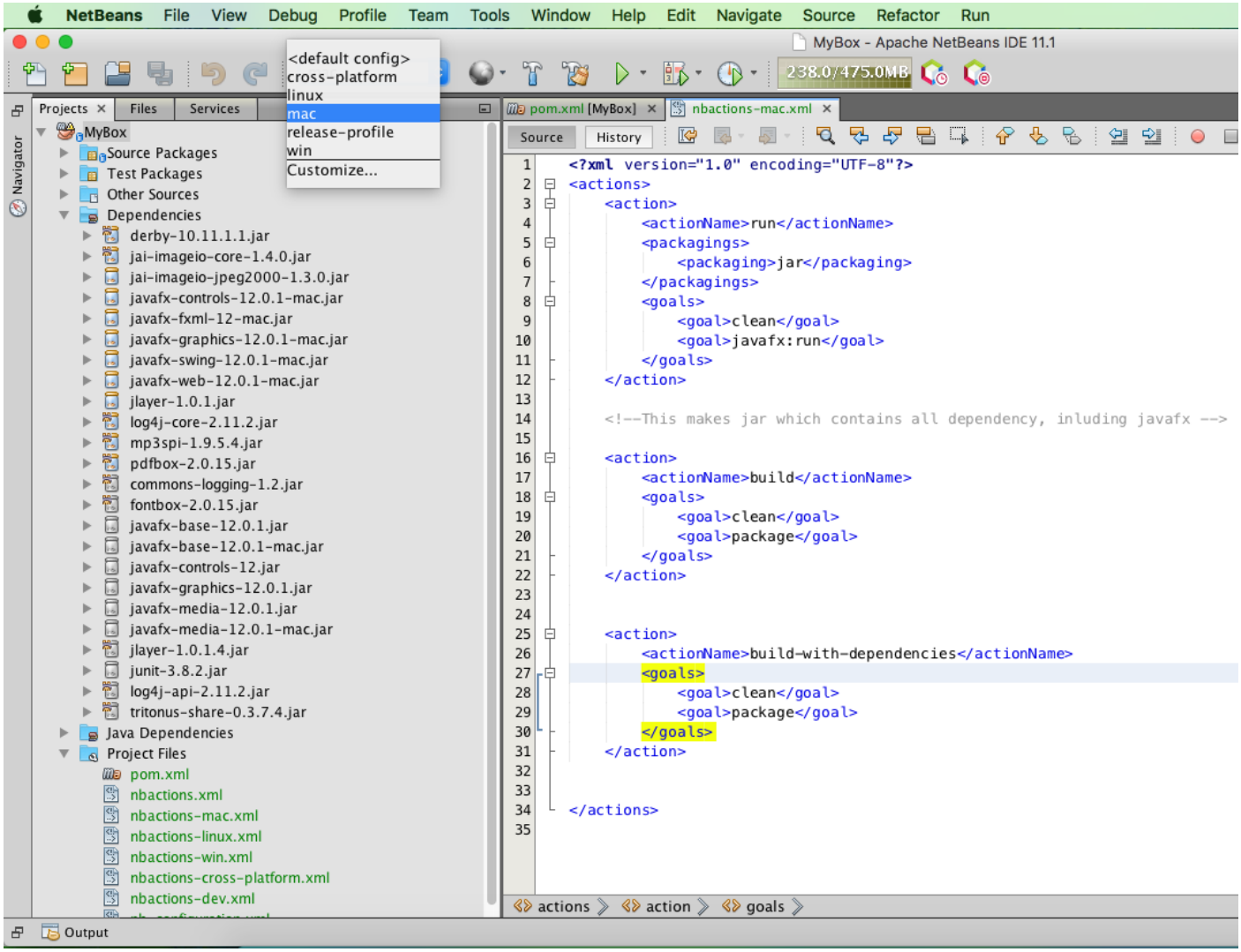
When Derby has been started in network(server) mode, NetBeans can connect to it by client mode. Developer can use this mode to check data when MyBox has been started.



3.2 Manage project

3.2.1 Switch configration profile

Project files “nbactions*.xml” are netbeans configration profiles for different maven profiles(different platforms). They can be switch easily in netbeans to build package for current plarform:



3.2.2 Quick debug

According to guides in offical network, action “run” is defined as “javafx:run” in files “nbactions*.xml”. Thus when click button “run”, MyBox will run directly in IDE javafx env and no jar package is generated. Time is saved much.

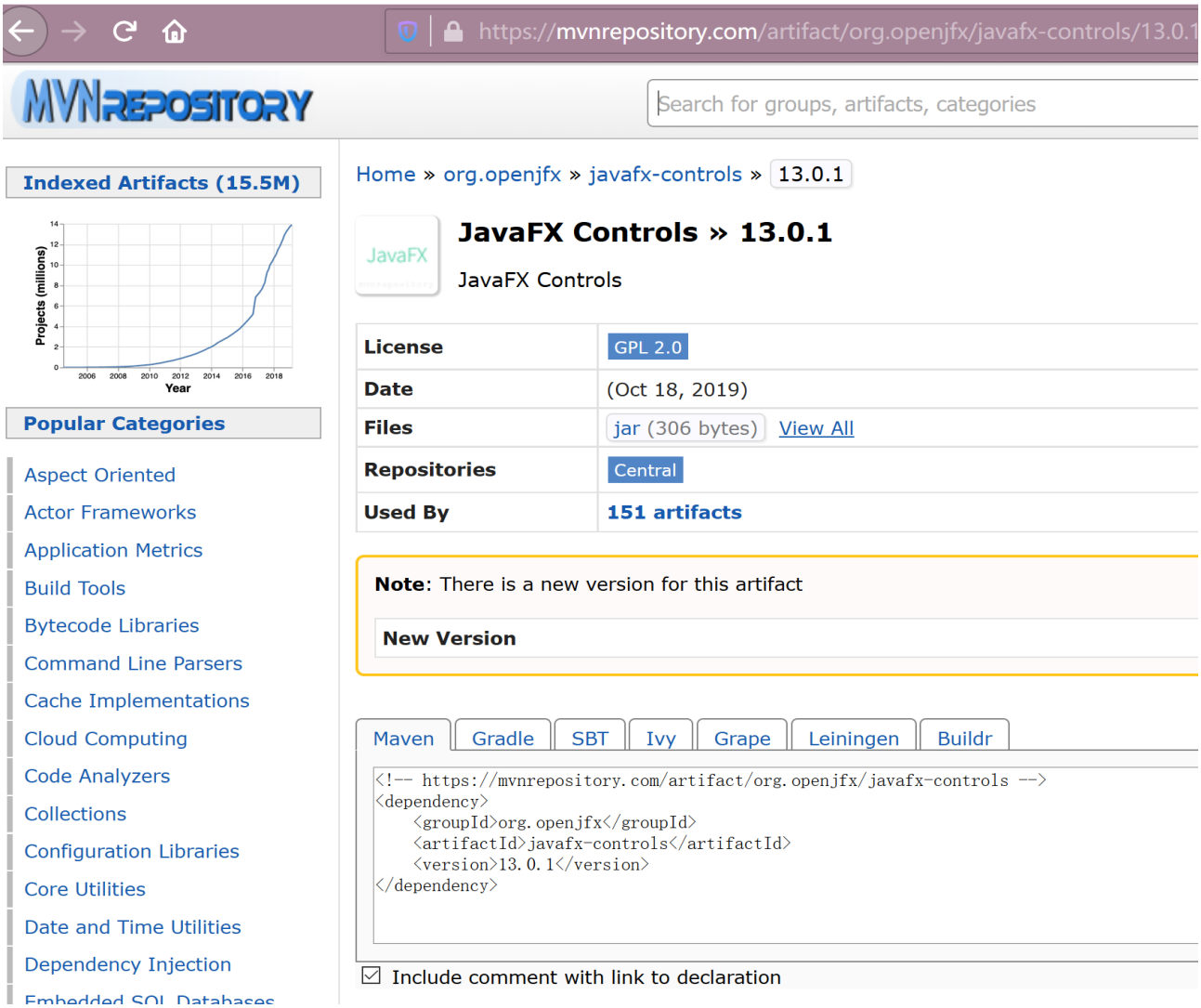
3.2.3 Build package with all dependencies

All “nbactions*.xml” define action “build” as “package”, so jar with all dependencies against selected profile(platform) will be generated when execute action “build”. More time is need compared with quick debug.

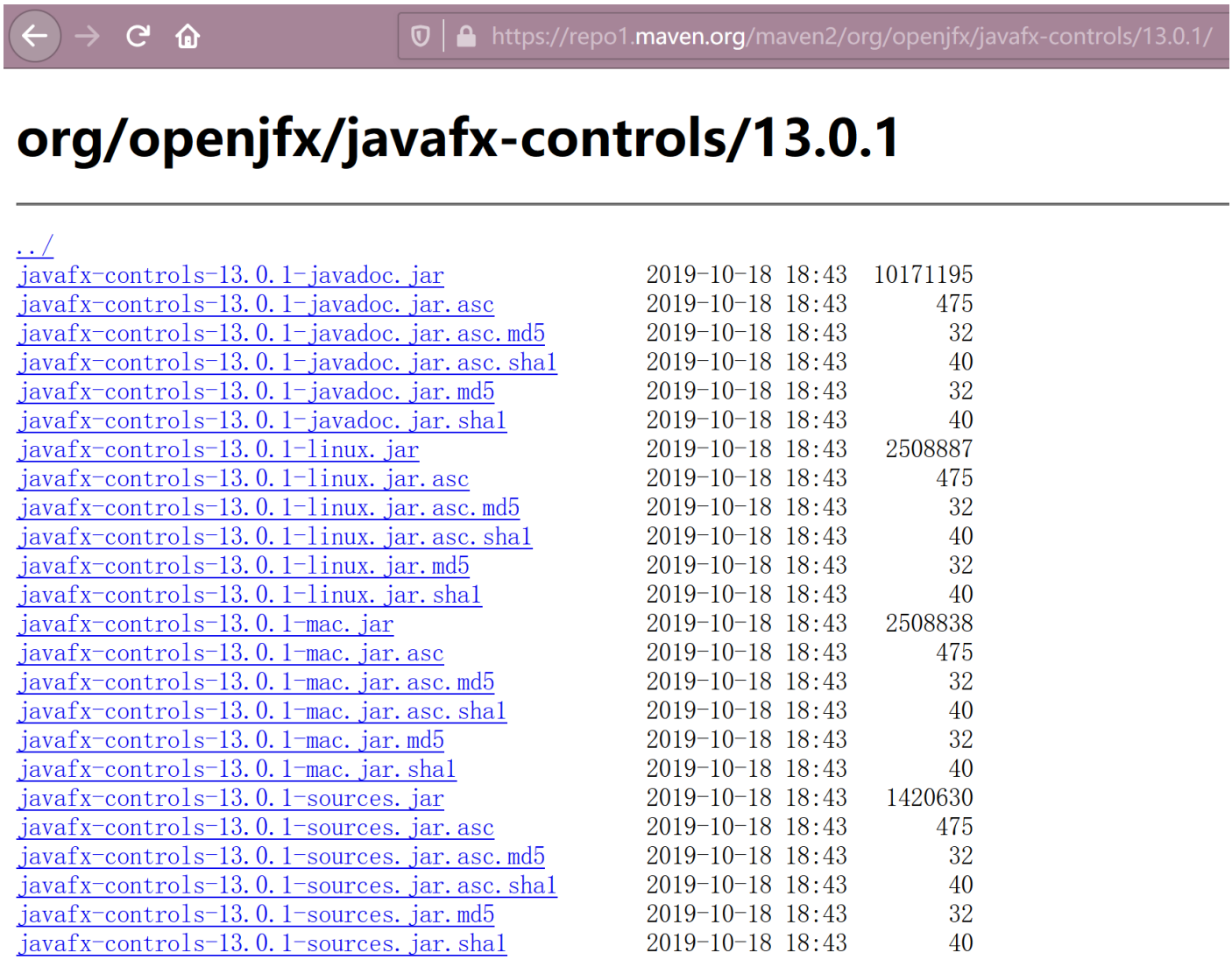
3.3 Find and download codes

Generally, it is easy and direct to search and down latest source codes by Maven in NetBeans. But sometimes some versions may be older or download is slow by IDE(Example in my env), then following address may be useful to find codes more directly:

https://mvnrepository.com/



Click “View All” to enter maven library:



3.4 Exceptions

3.4.1 Remove modules

Problem: Add module decriptor “module-info.java” to some project, then remove it. Then all source files are in wrong status in NetBeans.

Solution: Delete NetBeans caches like “C:\Users\mara\AppData\Local\NetBeans\Cache\11.1”.

3.4.2 OS broken

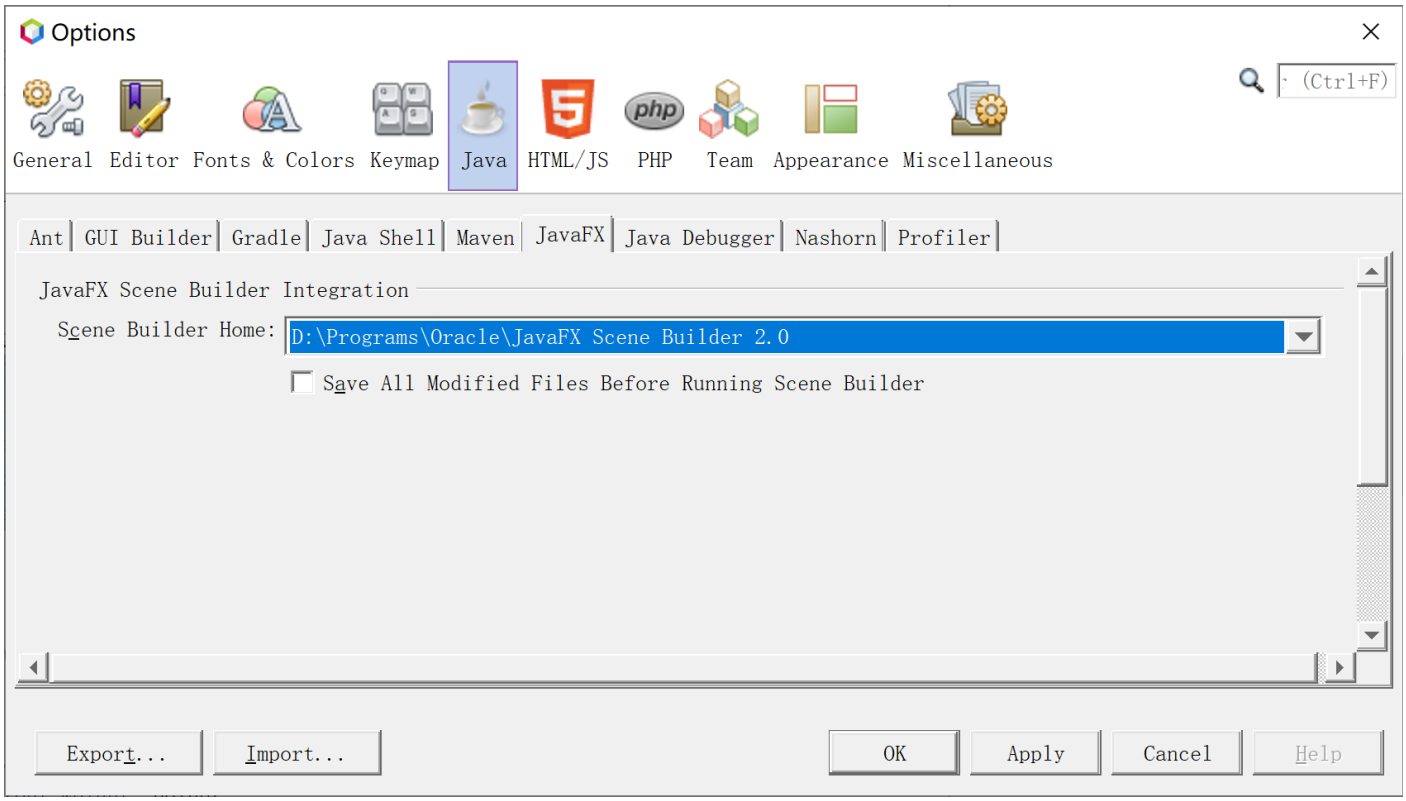
Problem: When system is down suddenly, some data in memory have not been updated to cache and may cause NetBeans abnormal. Example, left project tree can never be shown in my env once time.

Solution: Delete Configure data, like “C:\Users\mara\AppData\Roaming\NetBeans\11.1”. Reinstall NetBeans may be another way.

3.4.3 Editor instead of JavaFX Scene Builder is opened when double click fxml

Check “Tools” - “Options” - “Java” - “JavaFX”, and make sure path of JavaFX Scene Builder is there.

This option may become empty after quit and re-enter NetBeans.



4 Coding JavaFx

4.1 Offical Guides

https://docs.oracle.com/javafx/2/

4.2 Configure of latest JavaFx

https://openjfx.io/openjfx-docs/

4.3 Interface in CSS style

Offical guides of JavaFX CSS:

https://docs.oracle.com/javafx/2/api/javafx/scene/doc-files/cssref.html

Development guides:

https://docs.oracle.com/javafx/2/get_started/css.htm

https://docs.oracle.com/javafx/2/css_tutorial/jfxpub-css_tutorial.htm

4.4 Enable/Disable HiDPI

JavaFX has implemented dpi-aware since Java 9. It is enabled by default.

Following setting can enbale/disable hidpi:

```
System.setProperty("prism.allowhidpi", "false");
```

Notice: It must be run before JavaFx is lanuched.

4.5 Resolution of Control' snapshot

Method snapshot() can be used to make snapshot of control's current status. Snapshot in Java 8 is always WYSIWYG, while snapshot since Java 9 is in low resolution.

Following article explains this problem:

http://news.kynosarges.org/2017/02/01/javafx-snapshot-scaling/

In my env:

“Toolkit.getDefaultToolkit().getScreenResolution()" returns 216

"Screen.getPrimary().getDpi()" returns 72

"Screen.getPrimary().getOutputScaleX()" returns 2.25

72 * 2.25 = 162

96 * 2.25 = 216

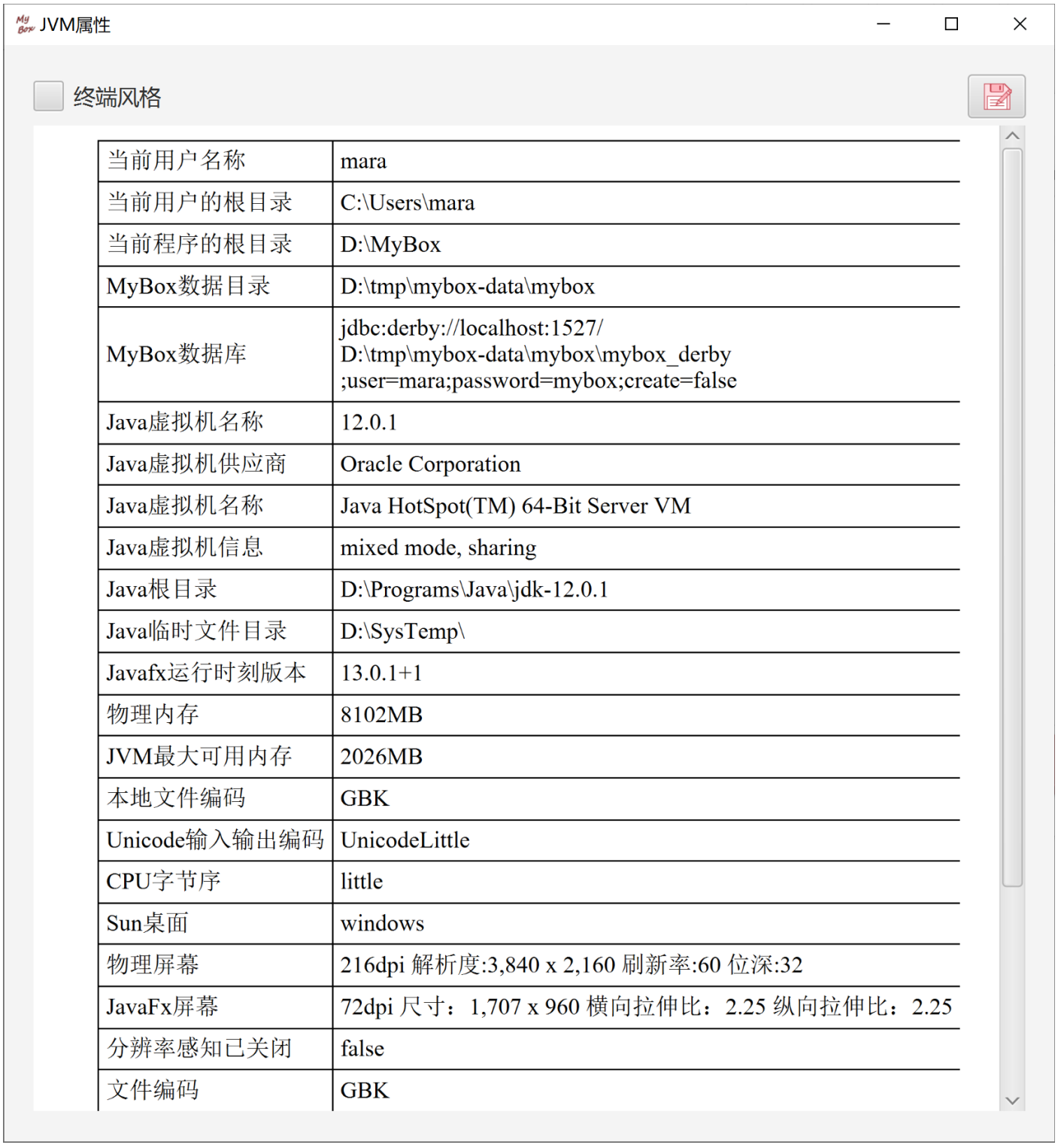
216 / 72 = 3

In order to get clear snapshots, MyBox takes the way mentioned in above article: scale control in 3 times and then make snapshot. The bad thing is that the result is in very large size.

Following is an example:

```
Bounds bounds = webView.getLayoutBounds();
```

```
double scale = Toolkit.getDefaultToolkit().getScreenResolution()
                    / Screen.getPrimary().getDpi();
if  (scale < 1) scale = 1;
int imageWidth = (int) Math.round(bounds.getWidth() * scale);
int imageHeight = (int) Math.round(bounds.getHeight() * scale);
SnapshotParameters snapPara = new SnapshotParameters();
snapPara.setFill(Color.TRANSPARENT);
snapPara.setTransform(javafx.scene.transform.Transform.scale(scale, scale));
WritableImage snapshot = new WritableImage(imageWidth, imageHeight);
snapshot = webView.snapshot(snapPara, snapshot);
```



当前用户名称	mara
当前用户的根目录	C:\Users\mara
当前程序的根目录	D:\MyBox
MyBox数据目录	D:\tmp\mybox-data\mybox
MyBox数据库	jdbc:derby://localhost:1527/ D:\tmp\mybox-data\mybox\mybox_derby ;user=mara;password=mybox;create=false
Java虚拟机名称	12.0.1
Java虚拟机供应商	Oracle Corporation
Java虚拟机名称	Java HotSpot(TM) 64-Bit Server VM
Java虚拟机信息	mixed mode, sharing
Java根目录	D:\Programs\Java\jdk-12.0.1
Java临时文件目录	D:\SysTemp\
Javafx运行时刻版本	13.0.1+1
物理内存	8102MB
JVM最大可用内存	2026MB
本地文件编码	GBK
Unicode输入输出编码	UnicodeLittle
CPU字节序	little
Sun桌面	windows
物理屏幕	216dpi 解析度:3,840 x 2,160 刷新率:60 位深:32
JavaFx屏幕	72dpi 尺寸：1,707 x 960 横向拉伸比：2.25 纵向拉伸比：2.25
分辨率感知已关闭	false
文件编码	GBK

4.6 Set controller for fxml dynamically

Refers to:

https://stackoverflow.com/questions/23132302/invocationtargetexception-when-running-a-javafx-program/53129147?r=SearchResults#53129147

Example:

```
FXMLLoader loader = new FXMLLoader(getClass().getResource("main.fxml"));
loader.setController(new MainController(path));
Pane mainPane = loader.load();
```

By this way parameters can be set for the controller.

But actions of all controls in this fxml can not be defined directlly.

4.7 Update legend in chart dynamically

Refers to:

https://stackoverflow.com/questions/37634769/dynamically-change-chart-colors-using-colorpicker/37646943?r=SearchResults#37646943

Example in MyBox:

```
Set<Node> legendItems = histogramChart.lookupAll("Label.chart-legend-item");
if (legendItems.isEmpty()) {
    return;
}
for (Node legendItem : legendItems) {
    Label legendLabel = (Label) legendItem;
    Node legend = legendLabel.getGraphic();
    if (legend != null) {
        String colorString = FxmlColor.rgb2Hex(colorTable.get(legendLabel.getText()));
        legend.setStyle("-fx-background-color: " + colorString);
    }
}
```

4.8 Danger of “setAlwaysOnTop”

Consider following scenario:

1) Stage A is set as "setAlwaysOnTop(true)".

2) Stage B pops an Alert to have user select some buttons.

3) Stage A is on top and covering buttons of the Alert.

Then app is blocking: Stage A can not get focus before user click buttons on Alert but Alert is unavaliable before stage A is moved away from top of it.

The way to avoid this confliction referred to following article: program should always bring alert in front

https://stackoverflow.com/questions/38799220/javafx-how-to-bring-dialog-alert-to-the-front-of-the-screen?r=SearchResults

```
Stage stage = (Stage) alert.getDialogPane().getScene().getWindow();
stage.setAlwaysOnTop(true);
stage.toFront();                    <----- This is key point
```

Following report has been submitted to jdk dev team:

https://bugs.java.com/bugdatabase/view_bug.do?bug_id=JDK-8230127

4.9 Ghost data in TableView/ListView

Method “UpdateItem” is often changed when define Cell of TableView/ListView, and ghost data, which are non-existed rows shown in bottom, will appear if not handle condition of “empty”

https://stackoverflow.com/questions/26821319/javafx-listview-and-treeview-controls-are-not-repainted-correctly

Solution is always writing codes for empty data:

```
@Override
protected void updateItem(final T persistentObject, final boolean empty) {
    super.updateItem(persistentObject, empty);
    if (empty) {                    <----  Must have this part
        setText(null);
        setGraphic(null);
    } else {
        // ... rest of your code.
    }
}
```

4.10 Listen events in embedded page

The special thing of listening events in embedded page is that the page can not receive events' notice when focus is not in it and both it and its parent will receive events' notice when focus is in it.

This will make 2 types of toubles: 1) The events can be handled twice if embedded page and its parent are inherited by same class and have same logic to handle and listen events. 2) Embedded page will miss some events when focus is not in it.

One solution is to put all logic in main page and causes it very complex. Better solution is embedded page does not listen events directly anymore and main page dispatchs events information.

4.11 Refresh ListView

ListView always refreshs itself when data is added or deleted. But it may be unaware of data updates and can not refresh itself when data attributes have been changed.

Program need refresh ListView.

https://stackoverflow.com/questions/13906139/javafx-update-of-listview-if-an-element-of-observablelist-changes?r=SearchResults

```
for (int i = 0; i < listView.getItems().size(); i++) {
    listView.getItems().set(i, listView.getItems().get(i));
}
```

4.12 Items in one sentence

1. Always assume user knows nothing about programming.
2. Do not expect user will open user guide. Someone, including me, even will not look at or think about prompts on the screen.
3. Necessary prompts can help user understand intention of the program.
4. Ambiguous prompt is worse than no prompt.
5. Always response for user, like pop information, make sound, or update interface.
6. Provide most options and set default choice.
7. Remember user's inputs or seletions and fill in or set values automatically in next time.
8. Less inputs or selections meanwhile keep interfaces clean.
9. Use radio or check buttons instead of drop-down list if there is enough place in interface.
10. No more than 3 levels to select.
11. When too many controls in interface, consider popular layout: Left-right areas like curtain, vertical accordion menus, tabs to switch targets
12. User' screen resolution may be from 1k to 4k, so the design size of most MyBox interfaces is smaller than 1100 * 720.
13. Make use of auto-wrap container like FlowPane, to work for both small screen and large screen.
14. Controls can be updated only in JavaFx thread. Use Platform.runLater to change control when handle event in its listener.
15. Progress should be shown when operation costs much time.
16. Time-consume operation should not be run in JavaFx thread to avoid blocking interface.
17. Backgournd task should always check whether task is canceled, especially in loop or before/after long-run statements.
18. When listen mouse events, event.getX() and event.getY() return the coordinate of event.getSource() which is the control owns the events. Example, when listen on ImageView, the coordinate is related to the image.
19. Iterative calls may happen when listen both onMouseEntered and onMouseExited and change focus in handling of the events(like pop menu) or change control's size.
20. When double click mouse, single-click event is tiggerred at first and then double-click event is triggerred. So need avoid duplicated or conflicted handlings against same event.

5 Image Manufacture

5.1 Java Image I/O Technology

Please refer following:

https://docs.oracle.com/javase/8/docs/technotes/guides/imageio/

https://docs.oracle.com/javase/8/docs/technotes/guides/imageio/spec/imageio_guideTOC.fm.html

https://docs.oracle.com/javase/tutorial/2d/index.html

https://www.javaworld.com/article/2076764/java-se/image-processing-with-java-2d.html

5.2 Image Meta Data

Please refer following:

https://docs.oracle.com/javase/10/docs/api/javax/imageio/metadata/doc-files/standard_metadata.html

https://docs.oracle.com/javase/10/docs/api/javax/imageio/metadata/doc-files/gif_metadata.html

https://docs.oracle.com/javase/10/docs/api/javax/imageio/metadata/doc-files/jpeg_metadata.html

https://docs.oracle.com/javase/10/docs/api/javax/imageio/metadata/doc-files/png_metadata.html

https://docs.oracle.com/javase/10/docs/api/javax/imageio/metadata/doc-files/tiff_metadata.html

5.3 Multiple Frames Image File

It is a file which includes multiple independant images.

Currently MyBox supports multiple frames image files in formats of animated gif and tiff/tif.

5.4 Tiff/Tif file

Please refer following:

https://en.wikipedia.org/wiki/TIFF

https://en.wikipedia.org/wiki/GeoTIFF

https://www.adobe.io/open/standards/TIFF.html

5.5 Animated Gif

Please refer following:

http://giflib.sourceforge.net/whatsinagif/index.html

https://www.jianshu.com/p/df52f1511cf8

https://stackoverflow.com/questions/22259714/arrayindexoutofboundsexception-4096-while-reading-gif-file

https://github.com/DhyanB/Open-Imaging

https://programtalk.com/python-examples/com.sun.media.imageioimpl.plugins.gif.GIFImageWriterSpi/

5.6 Big Image

Big image is a picture that includes too many pixels to be loaded and displayed under limitation of current memory usage. For all operations which use image as input, big image should be concerned.

The pressure agianst memory is the pixels number of the image, instead of the bytes number of image file.

Example, a jpg file of 42M includes 65500x4504 pixels. Each pixel occupies 3 bytes, so the image data requires 844M bytes memory. If need load and display the whole image in interface of JavaFx, data will be tranferred between file, BufferedImage, and WritableImage, and at least 2.6G memory will be occupied. In my practice, at least 5G should be defined in "-Xmx" to display this image although about 2.3G is shown as the memory usage.

Another example, a png file of 52M includes 8101x4557 pixels, and it needs only 750M memory to load and display whole image in memory.

The principles of handling big images by MyBox are following:

1) Evaluate the required memory for whole image, and judge whether load all data in memory. (About 5 times of pixels data plus 200M)
2) If enough memory is available to load whole image, read all data for next operations. Try best to operate in memory and avoid file I/O.
3) If memory may be out, subsample the image for next operations.
4) The sample ratio is determined by following rule: Make sure the sampled image is good enough while the sampled data occupy limited memory.
5) The sampled image is mainly to display the image, and not suitable for operations against whole image and images merging.
6) Some operations, like splitting and subsampling, can be done by reading part of image data and writing-while-reading, so they are suitable for big images.
7) Image which can not be loaded wholely may be suitable to be handled batchly. Example, an image of 500M pixels can not be displayed under limitation of 1.8G, thus can not be cropped, zoomed, or color-adjusted interactively. But when crop, zoom, or color-adjust the image in batch way, the operations may be successful under same memory limitation, since the data need not transferred as interface pixels.

Continually handling images may affect memory usage. Example, memory occupied by previous operation has not been collected by GC and then less memory can be required by current operation. So it is better to restart MyBox to handle big image because it can occupy most of available memory.

In order to handle big image interactively, user can extend the maximum memory for JVM.

5.7 Image Sampling

There are 2 types of image sampling: Downsampling(Also called Subsampling) and Upsampling(Also called interpolating).

When the pixels number of image is very big, subsampling is helpful to load and display the image under limited memory.

The rule of subsampling is very simple: given the sampling ratio, the image pixels are selected to read.

Example, when ratio is 3, only one pixel is read in adjacent matrix of width 3 and height 3.

Image subsampling is mainly used to handle big image. When sample ratio is 1, it acts same as "Crop". The difference between subsmapling and functions of "Crop"/"Size" in tool "Image Manufacture" is that subsampling only reads the required part of data in memroy and is writing while reading for big image. For small image, subsampling does same things as tool "Image Manufacture" and they both load all data and do operations in memory.

5.8 Image Grayscale

Please refer following:

https://en.wikipedia.org/wiki/HSL_and_HSV

https://en.wikipedia.org/wiki/Grayscale

5.9 Color Distance

Please refer following:

https://en.wikipedia.org/wiki/Color_difference

5.10 How to get sepia image

Please refer following:

https://stackoverflow.com/questions/21899824/java-convert-a-greyscale-and-sepia-version-of-an-image-with-bufferedimage/21900125#21900125

5.11 Image Blending

Please refer following:

https://en.wikipedia.org/wiki/Blend_modes

https://baike.baidu.com/item/混合模式/6700481?fr=aladdin

https://blog.csdn.net/bravebean/article/details/51392440

https://www.cnblogs.com/bigdream6/p/8385886.html

5.12 Convolution

Please refer following:

https://en.wikipedia.org/wiki/Convolution

https://en.wikipedia.org/wiki/Kernel_(image_processing)

http://colah.github.io/posts/2014-07-Understanding-Convolutions/

5.13 Flood-Fill

https://en.wikipedia.org/wiki/Flood_fill

5.14 Image Size

The following concepts are different:

“Orginal Size”: pixels number saved in image file.

“Loaded Size”: pixels number in memory which can be changed by Load/Crop/Transform.

“Display Size”: pixels number in screen after user zooms image in interface.

“Selected Size”: screen area selected by user, which will be mapped to memory area and calculated according to scale ratio of orginal size.

Example, an image has original size 1000x500, loaded size 800x400, display size 600x300, and selected size 700x200.

By setting loaded size, large image can be read in memory while small image can be scaled to view.

Loaded size is different from zooming in interface. Loaded pixels determine memory usage, while interface zooming only affect pixels in screen.

5.15 Image quantization

Technique of reducing distinct colors in image is called "Color Quantization". "Dithering" is to optimize the result of Color Quantization.

Different algorithms can be selected for different purposes:

RGB Uniform is fastest and good enough for 256 colors quantization of most images.

K-Means Clustering is good for calculating mostly different colors in image.

Popularity is good for calculating mostly occurred colors in image.

For some quantization algorithms, color space will be divided into some regions to reduce the data set. Parameter "Regions Bit Depth" determines the size of data scope to be used for preprocessing.

Example, 4 bit depth means total 16*16*16 color values mapped from 256*256*256 color space.

After image pixels are mapped into new regions, quantization runs against this smaller values collection.

So actually twice quantization happens cause regions mapping is also a quantization.

Notice: Larger bit depth does not mean better results. It always cost much longer running time and may give worse outputs. Result quality depends on image attributes, algorithm, and parameters. Example, when image includes less different colors, small bit depth is better.

Refer：

http://web.cs.wpi.edu/~matt/courses/cs563/talks/color_quant/CQindex.html

http://tpgit.github.io/UnOfficialLeptDocs/leptonica/color-quantization.html

https://www.researchgate.net/publication/220502178_On_spatial_quantization_of_color_images

http://rosettacode.org/wiki/Color_quantization

http://www.imagemagick.org/Usage/quantize/

5.16 Filters

Refer:

http://www.jhlabs.com/ip/index.html

6 Coding in Java

6.1 Data precision

About whether use BigDecimal to keep precision, following link is useful:

https://stackoverflow.com/questions/322749/retain-precision-with-double-in-java

"Do not waste your efford using BigDecimal. In 99.99999% cases you don't need it"

"BigDecimal is much slower than double"

"The solution depends on what exactly your problem is:

- If it's that you just don't like seeing all those noise digits, then fix your string formatting.

  Don't display more than 15 significant digits (or 7 for float).

- If it's that the inexactness of your numbers is breaking things like "if" statements,

  then you should write if (abs(x - 7.3) < TOLERANCE) instead of if (x == 7.3).

- If you're working with money, then what you probably really want is decimal fixed point.

  Store an integer number of cents or whatever the smallest unit of your currency is.

- (VERY UNLIKELY) If you need more than 53 significant bits (15-16 significant digits) of precision, then use a high-precision floating-point type, like BigDecimal."

6.2 Charset and encoding

Charset is the mapping between characters(for printing and displaying) and bytes(in memory or files). Encoding is the way about how to save bytes in memory or file.

https://en.wikipedia.org/wiki/UTF-8

https://www.cnblogs.com/ChurchYim/p/8427373.html

https://www.cnblogs.com/maohuidong/p/8044568.html

6.3 Problems

6.3.1 “cannot access class … because module ... does not export ...”

I met following error under netbean11.1 + openjdk 12.0.1 + openjfx12.0.1:

Exception in thread "JavaFX Application Thread" java.lang.IllegalAccessError: class mara.mybox.controller.ImageViewerController (in unnamed module @0x1ec2591a) cannot access class com.sun.javafx.charts.Legend (in module javafx.controls) because module javafx.controls does not export com.sun.javafx.charts to unnamed module @0x1ec2591a

This happened due to a reference of internal API.

Java codes are packed in different modules since Java 9 and env need configured when they refer to each other:

https://github.com/javafxports/openjdk-jfx/issues/236

http://mail.openjdk.java.net/pipermail/openjfx-dev/2018-June/021977.html

https://stackoverflow.com/questions/53237287/module-error-when-running-javafx-media-application

https://github.com/openjfx/javafx-maven-plugin

https://openjfx.io/openjfx-docs/#modular

https://community.oracle.com/blogs/vincentvauban/2018/07

Some classes are public but they are not in the export list of module, so module can not visit it without confiration.

Following article mentioned this problem and its solution:

https://stackoverflow.com/questions/56459093/openjfx-custom-runtime-image-using-maven-and-jlink-module-exports-or-command-l/56467911?r=SearchResults#56467911

MyBox is not moduled. Its solution was adding following lines in pom.xml:

```
<plugin>
    <groupId>org.openjfx</groupId>
    <artifactId>javafx-maven-plugin</artifactId>
    <version>0.0.3</version>
    <configuration>
        <mainClass>mara.mybox.MyBox</mainClass>
        <options>
            <option>--add-opens</option>
            <option>javafx.controls/com.sun.javafx.charts=ALL-UNNAMED</option>
        </options>
    </configuration>
</plugin>
```

And the final solution is only using public API.

6.3.2 “xxx uses unchecked or unsafe operations.”

Refer:

https://stackoverflow.com/questions/8215781/how-do-i-compile-with-xlintunchecked

https://stackoverflow.com/questions/262367/type-safety-unchecked-cast?r=SearchResults

Example, codes in problem：

```
List<String> names = new ArrayList();
Map<String, ImageInformation> oldList = new HashMap();
```

Change it as:

```
List<String> names = new ArrayList<>();
Map<String, ImageInformation> oldList = new HashMap<>();
```

Replacement in batch can solve this problem very easily.

Add compile option to check invalid lines:

```
<artifactId>maven-compiler-plugin</artifactId>
<configuration>
    <compilerArgument>-Xlint:unchecked</compilerArgument>
</configuration>
```

Supress this checking by add annotation:

```
@SuppressWarnings("unchecked")
public Object getValue() {
    return value;
}
```

6.4 Items in one sentence

1. Always assume most extreme inputs and most weird operations.
2. Assume user' hardware configuration(CPU/memory/disk) is lower than average level.
3. Use BufferedInputStream and BufferedOutputStream for file I/O.
4. Use try-with-resource for file I/O.
5. Notice charset when read/write text file.
6. Notice charset when convert between bytes and string.
7. When class has many parameters, do not write many constructors to initialized different sets of parameters. Better coding is static creator and chain assignment by return “this” in all “set” methods.
8. Anonymous class is very useful for scenario like lots of duplicated codes with only difference that several methods have difference implements.
9. Generic is very useful for scenario like lots of duplicated codes with only differernce of data type.
10. As less float-point calculations as possible in loop. If can, change float-point calculations as integer calculations.
11. When listen data changing, need judge whether change trully happens, and avoid too frequently manufactures by setting minimum threshold of changing.
12. Reboot application when change JVM parameters or need clean env in simple way.
13. Notice whether the index is 0-based or 1-based, especially for thrid-party codes. 0-based is default. If 1-based, should write comment for it.

<End of Document>