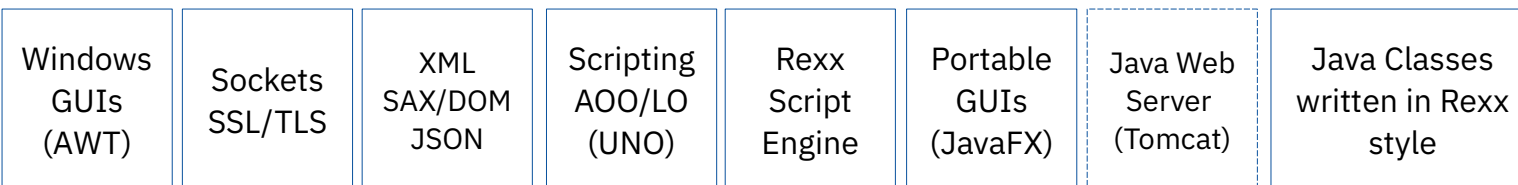


BSF4ooRexx

Setup Environment for Java/OpenJDK (2023-01-19)

Business Programming 2



Agenda



- Operating System
 - Process and environment
- Java/OpenJDK
 - Startup of Java programs
 - Defining switches for Java/OpenJDK (JVM)
 - BSF4ooRexx
 - Environment variable "[BSF4Rexx_JavaStartupOptions](#)"
 - URLs for downloading some Java/OpenJDKs
 - Directions to create a JRE with Java and JavaFX on your own



- Manages each program in a separate "process"
 - Manages access rights
 - Manages memory, resources like pipes, sockets, ...
 - Manages priorities
 - Creates an execution "environment"
 - Sets up standard files "stdin" (0), "stdout" (1), "stderr" (2)
 - Defines "environment variables", e.g.
 - "PATH": determines which directories will be searched for a program
 - "CLASSPATH": determines which directories and Java archives Java searches for Java classes

- Environment variables

- Shell (Terminal, command line)

- Windows: "cmd" (cmd.exe)

```
set
echo %PATH%
set MYVAR=%PATH%;c:\my\dir
set CLASSPATH=%CLASSPATH%;c:\path\to\my.jar
```

- Unix: "sh" – runs one of the many shell programs ("echo \$SHELL")

```
env | sort
echo $PATH
export MYVAR=$PATH:/my/dir
export CLASSPATH=$CLASSPATH:/path/to/my.jar
```

- Java or OpenJDK ?
 - Currently jointly developed by Oracle and non-Oracle programmers as an opensource project named "OpenJDK"
 - "Java"
 - Programming language
 - Also, commercial version owned by Oracle (after buying Sun)
 - "JDK" (Java development kit)
 - Includes the Java runtime environment (JRE)
 - Includes compiler and development tools
 - New major version currently every six months: OpenJDK 18 (2023-01-12)
 - "OpenJDK"
 - Free, open-source Java runtime and Java development environment
 - Home: <<http://openjdk.java.net/>> (2023-01-12)

- Start a Java program on the command line

```
java SomeCompiledJavaClass
```

- `java[.exe]` creates the "Java virtual machine (JVM)", loads the class `SomeCompiledJavaClass.class` and executes its static method `main()`
- Possible to supply switches, e.g.
 - Set maximum heapsize to 4 GB (on 64-bit operating systems)

```
java -Xms4GB SomeCompiledJavaClass
```
 - Set `CLASSPATH` to use
 - Windows

```
java -cp "%CLASSPATH%;c:\path\to\my.jar" SomeCompiledJavaClass
```
 - Unix

```
java -cp "$CLASSPATH:/path/to/my.jar" SomeCompiledJavaClass
```

- Define command line options for the JVM
 - Environment variable `BSF4Rexx_JavaStartupOptions`
 - Used by `rexx[.exe]`, `rexxj.cmd` (Windows), `rexxj.sh` (Unix)
 - Optional, allows configuring the JVM, if needed!
 - Set maximum heap size to 4 GB, add the jar files "`c:\one\a.jar`" and "`c:\two\b.jar`" and directory "`c:\xyz`" to the classpath ("`-cp`")
 - **Windows**

```
set BSF4Rexx_JavaStartupOptions=-Xms4GB -cp "%CLASSPATH%;c:\one\a.jar;c:\two\b.jar;c:\xyz"
rexxj.cmd myRexxProgram.rex
rexx myRexxProgram.rex
```
 - **Unix**

```
export BSF4Rexx_JavaStartupOptions=-Xms4GB -cp "$CLASSPATH:/one/a.jar:/two/b.jar:/xyz"
rexxj.sh myRexxProgram.rex
rexx myRexxProgram.rex
```

- BSF4ooRexx850 honors *all* Java class libraries in the
 - lib subdirectory of the BSF4ooRexx850 installation subdirectory
 - Unix, e.g.: /opt/BSF4ooRexx850/lib
 - Windows: %ProgramFiles%\BSF4ooRexx850\lib or %ProgramFiles(x86)%\BSF4ooRexx850\lib
 - lib subdirectory in the user's home directory (preferred!)
 - Unix: \$HOME/BSF4ooRexx/lib
 - Windows: %UserProfile%\BSF4ooRexx\lib
- Copying external Java class libraries (stored in jar or zip archives) to any of these lib directories makes them immediately available!
 - No need to change the CLASSPATH environment in this case

- Starting with Java 9: "module system", **very** important!
 - "Cheat sheet":
 - <<https://zeroturnaround.com/rebellabs/java-9-modules-cheat-sheet/>> (2023-01-12)
 - Java/JDK 11 (fall 2018) removed JavaFX modules
 - Now separate modules, home of JavaFX: <<https://openjfx.io>> (2023-01-12)
 - Setting up environment for Java 11 to find JavaFX 11
 - cf. <<https://openjfx.io/openjfx-docs/#install-javafx>> (2023-01-12)
 - Windows, e.g. (for Java 11, adjust path)

```
set FXMODULES=-p c:\javafx-sdk-11.0.1\lib -add-modules=javafx.controls,javafx.fxml
set BSF4Rexx_JavaStartupOptions=-cp %CLASSPATH% %FXMODULES%
```
 - Unix, e.g. (for Java 11, adjust path)

```
export FXMODULES=-p /javafx-sdk-11.0.1/lib -add-modules=javafx.controls,javafx.fxml
export BSF4Rexx_JavaStartupOptions=-cp $CLASSPATH $FXMODULES
```

- Some Java installation packages
 - Oracle, URL: <<https://www.java.com>> (2023-01-12)
 - Java license for commercial use changed
- From the same, free OpenJDK sources
 - Liberica, URL: <<https://bell-sw.com/>> (2023-01-12)
 - Also has downloads with JavaFX, look for "full Liberica"
 - Zulu, URL: <<https://www.azul.com/downloads/>> (2023-01-12)
 - Also has downloads with JavaFX, look for „FX“ (Filter: „JDK FX“)
 - OpenJDK, URL: <<https://adoptOpenJDK.net>> (2023-01-12)
- Hint: look for installation packages with JavaFX included
 - Usually the word "full" or "fx" in the package name

Create Your Own JRE (with JavaFX)



- Java module system (since Java 9, 2017-09-21)
 - Meant for stand-alone Java applications
 - Allows to remove unneeded Java modules to save space
- Java runtime environment (JRE)
 - There are many deployments of applications that need a JRE
 - A JRE makes all Java classes available to
 - Non-Java and scripting languages
 - Server applications
- Need to create one own's JRE

Create Your Own JRE (with JavaFX), 1

- Java modules introduced with Java 9 (2017-09-21)
 - Only distributed as "JDK" Java development kit
 - Environment needs to be adjusted to modules
 - Needs may be different at compile and runtime!
 - Idea: create smallest possible footprint for Java applications by using only the needed modules!
 - Maybe relevant for standalone Java applications on very small hardware
 - JavaFX donated to the opensource community by Oracle
 - Java 11 (2018-09-25) removed the JavaFX modules!
 - Download opensource JavaFX modules from Gluon
 - <https://gluonhq.com/products/javafx/> (2023-01-12)

Create Your Own JRE (with JavaFX), 2

- Need for a full Java runtime environment (JRE), e.g.,
 - Server configurations where many different servlets need many different Java modules
 - Scripting, ad-hoc programs
 - Unforeseeable need for Java modules
- JDK comes with a tool named [jlink](#)
 - Allows to create a tailored Java runtime environment
 - Can be used to create a full JRE from any modular JDK

Create Your Own JRE (with JavaFX), 3



- Steps

- Download JDK (e.g. <<https://adoptopenjdk.net>>)
 - Locate JDK home directory and assign it to the `JAVA_HOME` environment variable
 - All JDK modules in: `$JAVA_HOME/jmods` (Unix),
`%JAVA_HOME%\jmods` (Windows)
- Download Open JavaFX (<<https://gluonhq.com/products/javafx/>>)
 - Locate JavaFX directory and assign it to the environment `FX_DIR` variable
 - All JavaFX modules in: `$FX_DIR/jmods` (Unix),
`%FX_DIR%\jmods` (Windows)



Create Your Own JRE (with JavaFX), 4



- Steps

- Open a command line/terminal window

- Define environment variables

```
set JAVA_HOME=path-to-JDK-home
```

```
set FX_DIR=path-to-JavaFX-directory
```

- Issue the `jlink` command (Windows)

```
%JAVA_HOME%\bin\jlink -p %JAVA_HOME%\jmods,%FX_DIR%\jmods --add-modules ALL-MODULE-PATH --output tgtdir
```

- Issue the `jlink` command (Unix)

```
$JAVA_HOME/bin/jlink -p $JAVA_HOME/jmods,$FX_DIR/jmods --add-modules ALL-MODULE-PATH --output tgtdir
```

- '`tgtdir`' will contain the appropriate JRE with all modules from JDK and from JavaFX!

Windows: `tgtdir\bin\java --list-modules`

Unix: `tgtdir/bin/java --list-modules`