Java Environment

- Introduction
- Architecture
- Environment
  - No JVM
  - Current JVM
  - Update Alternatives
  - Update Java Alternatives
  - Known JVMs
- Java Layout
  - Java Installation Directory
  - The default-java symlink
  - The .jinfo file

Introduction

This page is based on experience of setting up Java 8 on Ubuntu 14.04. Much of it may be applicable to other Java versions and distros.

Architecture

First we check which architecture we are running on.

Open a terminal and type the command:

```
arch
```

The output could be `i686` or `i386` for a 32 bits system or `x86_64` for a 64 bits system.

Environment

No JVM

On a machine without any defined java, it will look like this. The `update-alternatives` doesn't display any defined alternatives:

```
user@machine:~$ update-alternatives --get-selections | grep java
lucli auto /usr/lib/liblucene2-java/lucli
```

If you try to run `java`, it doesn't run successfully:

```
user@machine:~$ java -version
The program 'java' can be found in the following packages:
  * default-jre
  * gcj-4.8-jre-headless
  * openjdk-7-jre-headless
  * gcj-4.6-jre-headless
  * openjdk-6-jre-headless
Try: sudo apt-get install <selected package>
```

The next test will be to use the `update-java-alternatives` command, to check if we have some installed JVM:

```
user@machine:~$ sudo update-java-alternatives --list
default-java 1073 /usr/lib/jvm/default-java
java-1.8.0-openjdk-1386 1069 /usr/lib/jvm/java-1.8.0-openjdk-1386
java-8-openjdk-1386 1069 /usr/lib/jvm/java-8-openjdk-1386
java-8-oracle 1070 /usr/lib/jvm/java-8-oracle
ramdisk-java-8-oracle 1073 /usr/lib/jvm/ramdisk-java-8-oracle
```

Hopefully in this example some existing JVM are installed however the JVM setup seems to be broken. Check your Java Environment then Install one.
If no existing JVM are installed, install Oracle JDK or Open JDK.

**Current JVM**

```bash
user@machine$ java -version
java version "1.8.0_91"
Java(TM) SE Runtime Environment (build 1.8.0_91-b14)
Java HotSpot(TM) Server VM (build 25.91-b14, mixed mode)
```

The previous command shows us that we have a running java on our system.

```bash
user@machine$ which java
/usr/bin/java
```

The previous command shows us that we are running a java command from /usr/bin.

Let's see where is the java command:

```bash
user@machine$ ls -al /usr/bin/java
lrwxrwxrwx 1 root root 22 Mar 12 2013 /usr/bin/java -> /etc/alternatives/java
```

As you can see the java command from /usr/bin is a symlink and will be resolved against the java command contained in /etc/alternatives.

```bash
user@machine$ file /etc/alternatives/java
/etc/alternatives/java: symbolic link to `/usr/lib/jvm/ramdisk-java-8-oracle/jre/bin/java'
```

The previous command shows us where lies the java command and hopefully the directory /usr/lib/jvm/ramdisk-java-8-oracle/jre/bin contains the real java command.

```bash
user@machine$ ls -al /usr/lib/jvm/ramdisk-java-8-oracle/jre/bin/java
-rwxr-xr-x 1 root root 5730 May 4 10:54 /usr/lib/jvm/ramdisk-java-8-oracle/jre/bin/java
```

**Update Alternatives**

On Ubuntu 14.04 the Java commands are under the /usr/bin and /etc/alternatives directories.

Both directories expose all the Java commands and tools as symlinks and as such when runned are resolved against the standard /usr/bin directory who usually populates the $PATH environment variable.

However in order to manage multiple version of Java the update-alternatives mechanism has been introduced. It allows multiple “providers” to be installed on the same machine, and configures which is the default.

This mechanism consists of two directories, /etc/alternatives who contains the symlinks and /var/lib/dpkg/alternatives who contains update-alternatives state information and a update-alternatives command.

Now take a look in the file /var/lib/dpkg/alternatives/java:

```bash
manual
/usr/bin/java
java.1.gz
/usr/share/man/man1/java.1.gz
/usr/lib/jvm/java-8-openjdk-i386/jre/bin/java
1069
/usr/lib/jvm/java-8-openjdk-i386/jre/man/man1/java.1.gz
/usr/lib/jvm/java-8-oracle/jre/bin/java
1070
/usr/lib/jvm/java-8-oracle/man/man1/java.1.gz
/usr/lib/jvm/ramdisk-java-8-oracle/jre/bin/java
1073
```

That file teaches us that there are several java command installed on this machine.
The following command shows the current alternatives configuration. This command helps to ensure that the default-java we defined will resolve the correct commands, tools and libraries.

```
user@machine:~$ update-alternatives --get-selections | grep java
```

```
appletviewer manual /usr/lib/jvm/default-java/bin/appletviewer
extcheck manual /usr/lib/jvm/default-java/bin/extcheck
firefox-javaplugin.so auto /usr/lib/jvm/default-java/jre/lib/i386/libnpjp2.so
idlj manual /usr/lib/jvm/default-java/bin/idlj
itweb-settings auto /usr/lib/jvm/default-java/jre/bin/itweb-settings
jar manual /usr/lib/jvm/default-java/bin/jar
jarsigner manual /usr/lib/jvm/default-java/bin/jarsigner
java manual /usr/lib/jvm/default-java/jre/bin/java
javac manual /usr/lib/jvm/default-java/bin/javac
javadoc manual /usr/lib/jvm/default-java/bin/javadoc
javah manual /usr/lib/jvm/default-java/bin/javah
javap manual /usr/lib/jvm/default-java/bin/javap
javaws auto /usr/lib/jvm/java-6-openjdk-i386/jre/bin/javaws
jcmd manual /usr/lib/jvm/default-java/bin/jcmd
jconsole manual /usr/lib/jvm/default-java/bin/jconsole
jdb manual /usr/lib/jvm/default-java/bin/jdb
jdeps manual /usr/lib/jvm/default-java/bin/jdeps
jexec manual /usr/lib/jvm/default-java/jre/lib/jexec
javaws auto /usr/lib/jvm/java-6-openjdk-i386/jre/bin/javaws
jstat manual /usr/lib/jvm/default-java/bin/jstat
jstatd manual /usr/lib/jvm/default-java/bin/jstatd
jmap manual /usr/lib/jvm/default-java/bin/jmap
jsadebugd manual /usr/lib/jvm/default-java/bin/jsadebugd
jstat manual /usr/lib/jvm/default-java/bin/jstat
jstatd manual /usr/lib/jvm/default-java/bin/jstatd
keytool manual /usr/lib/jvm/default-java/jre/bin/keytool
lucli auto /usr/lib/liblucene2-java/lucli
native2ascii manual /usr/lib/jvm/default-java/bin/native2ascii
orbd manual /usr/lib/jvm/default-java/jre/bin/orbd
pack200 manual /usr/lib/jvm/default-java/jre/bin/pack200
policytool manual /usr/lib/jvm/default-java/jre/bin/policytool
rmic manual /usr/lib/jvm/default-java/bin/rmic
rmid manual /usr/lib/jvm/default-java/jre/bin/rmid
rmiregistry manual /usr/lib/jvm/default-java/jre/bin/rmiregistry
schemagen manual /usr/lib/jvm/default-java/bin/schemagen
serialver manual /usr/lib/jvm/default-java/bin/serialver
servertool manual /usr/lib/jvm/default-java/jre/bin/servertool
rmnmserver manual /usr/lib/jvm/default-java/jre/bin/rmnmserver
unpack200 manual /usr/lib/jvm/default-java/jre/bin/unpack200
wagen manual /usr/lib/jvm/default-java/bin/wagen
wsimport manual /usr/lib/jvm/default-java/bin/wsimport
xjc manual /usr/lib/jvm/default-java/bin/xjc
```

**Update Java Alternatives**

**Known JVMs**

The `update-alternatives` command shows the installed JVM on a command name basis:

```
user@machine:~$ update-alternatives --list java
/usr/lib/jvm/default-java/jre/bin/java
```

The `update-java-alternatives` command shows the installed JVMs and their respective priority number.
Java Layout

Java Installation Directory

Either the Open JDK or the Oracle JDK install their environment below `/usr/lib/jvm`:

```bash
user@machine:/usr/lib/jvm$ ls -al
```

Beyond the various directories, files and links in the `/usr/lib/jvm` directory several remarks need to be done.

The default-java symlink

First of all we use the debian layout by setting up the `default-java` symlink. Tomcat or Eclipse are tools who use the `default-java` symlink.

We see in the previous output that the `default-java` point to a `ramdisk-java-8-oracle` who is a local symlink who points to another directory.

Then setup the `default-java` symlink to your default JVM

```bash
cd /usr/lib/jvm
sudo ln -s ramdisk-java-8-oracle default-java
```

The `.jinfo` file

Second we see `.jinfo` files, one per JDK installed. In our example we have three different JVM and we see their respective `.jinfo` files.

Let's see the content of the `java-8-oracle.jinfo`:
As you can see this file contains the description of each java tool, either jre, jdk or plugin tools are associated with a particular command or library.

In case the jinfo file doesn't exist, create one with the previous content and adjust paths accordingly.

The priority value needs to be unique among the JVM installations. Choose a priority number who do not collapse an existing one.

Then setup the default-java.jinfo symlink to your default JVM.
cd /usr/lib/jvm
sudo ln -s .ramdisk-java-8-oracle.jinfo .default-java.jinfo