Install or Uninstall Java

- Introduction
- Install
  - Command Line
  - Scripting
- Uninstall
  - Command Line
  - Scripting

Introduction

After checking your Java Environment, downloaded and installed either an Oracle JDK or an Open JDK, now it's time to pack everything together.

At this point, you have a JVM installed below your `/usr/lib/jvm` directory and you've a `jinfo` file tuned to your needs with a unique priority number.

Install

Now it's time to link `/etc/alternatives` with our new installed JVM.

Command Line

Usually one has to install each java commands, tools or libraries one by one with the following interactive command:

```
user@machine:$ /usr/lib/jvm$ sudo update-alternatives --config java
There are 4 choices for the alternative java (providing /usr/bin/java).

Selection  Path                                                  Priority  Status
--------------------------------------------------------------
 0  /usr/lib/jvm/ramdisk-java-8-oracle/jre/bin/java  1073  auto mode
 1  /usr/lib/jvm/default-java/jre/bin/java            1073  manual mode
 2  /usr/lib/jvm/java-8-openjdk-i386/jre/bin/java     1069  manual mode
 3  /usr/lib/jvm/java-8-oracle/jre/bin/java           1070  manual mode
 * 4  /usr/lib/jvm/ramdisk-java-8-oracle/jre/bin/java  1073  manual mode
```

If your new installed JVM doesn't appear in this list you need to add it manually with:

```
sudo update-alternatives --install "~/bin/java" "java" "~/usr/lib/jvm/default-java/jre/bin/java" 1073;
```

Check if your added JVM java command has been processed by running:

```
user@machine:$ update-alternatives --list java
/usr/lib/jvm/default-java/jre/bin/java
/usr/lib/jvm/java-8-openjdk-i386/jre/bin/java
/usr/lib/jvm/java-8-oracle/jre/bin/java
```

Scripting

Doing this task one by one is pretty tedious, we provide a shell script ito facilitate the work.

The trick here is to always increment the priority number for a new installed JVM and define a unique alternative name. Here is an excerpt of the man page:

```
If the alternative name specified exists already in the alternatives system's records, the information supplied will be added as a new set of alternatives for the group. Otherwise, a new group, set to automatic mode, will be added with this information. If the group is in automatic mode, and the newly added alternatives' priority is higher than any other installed alternatives for this group, the symlinks will be updated to point to the newly added alternatives.
```
This shell script mirrors the content of the `jinfo` file described in the Java Environment Tutorial.

This shell script file contains various known plugins. However, at this time, we only describe the Firefox Java plugin support. It is why we suggest the other plugin support as comments.

Usually, we store those shell scripts under `/usr/lib/jvm/tools`. This one is called `installDefaultJavaAlternatives.sh`.

Don’t forget to `chmod` your shell script.

```bash
user@machine:~/$ cd /usr/lib/jvm/tools
user@machine:/usr/lib/jvm/tools$ sudo chmod 744 installDefaultJavaAlternatives.sh
user@machine:/usr/lib/jvm/tools$ ls -al installDefaultJavaAlternatives.sh
-rwxr--r-- 1 root root 6814 Jul 12 11:53 installDefaultJavaAlternatives.sh
```

Then run it:

```bash
user@machine:/usr/lib/jvm/tools$ sudo ./installDefaultJavaAlternatives.sh
```

Then run:

```bash
user@machine:/usr/lib/jvm/tools/sudo update-java-alternatives -s default-java
```

Sometimes the previous command fails because `default-java` refers to an existing JVM:
In our configuration `.default-java.jinfo` refers to `.ramdisk-java-8-oracle.jinfo`:

```bash
user@machine:/usr/lib/jvm/tools$ ls -al .default-java.jinfo
lrwxrwxrwx 1 root root 30 Jul 12 14:29 .default-java.jinfo -> ./ramdisk-java-8-oracle.jinfo
```

The `.default-java.jinfo` is linked to `.ramdisk-java-8-oracle.jinfo` who use `ramdisk-java-8-oracle` as its name. This is the reason why the previous command failed.

It means that `ramdisk-java-8-oracle` should be installed through the Update Alternatives mechanism before we can set the alternatives to `default-java` or you define a `jinfo` file with a different priority number.

Then check again that everything is correct with your Java Update Alternatives Environment.

**Uninstall**

Now it’s time to unlink `/etc/alternatives` with the JVM we want to uninstall.

**Command Line**
Usually one has to uninstall each Java command, tool or library one by one with the following interactive command:

```bash
sudo update-alternatives --remove "java" /usr/lib/jvm/default-java/jre/bin/java;
```

**Scripting**

Doing this task one by one is pretty tedious, we provide a shell script to facilitate the work.
This shell script mirror the content of the jinfo file described in the Java Environment Tutorial.

This shell script file contains various known plugins. However at this time we only describe the Firefox Java plugin support. It is why we suggest the other plugin support as comments.

Usually we store those shell scripts under /usr/lib/jvm/tools. This one is called removeDefaultJavaAlternatives.sh.

Don't forget to chmod your shell script.
Then run it:

```
user@machine:~$ cd /usr/lib/jvm/tools
user@machine:/usr/lib/jvm/tools$ sudo chmod 744 removeDefaultJavaAlternatives.sh
user@machine:/usr/lib/jvm/tools$ ls -al removeDefaultJavaAlternatives.sh
-rwxr--r-- 1 root root 6814 Jul 12 11:53 removeDefaultJavaAlternatives.sh
```

As you’ve uninstalled a JVM, it depends whether or not you have other installed JVMs. In case you have more than one, the `update-alternatives` will update your configuration with the highest priority JVM.

Then check again that everything is correct with your Java Update Alternatives Environment.