JRE Installation

Installation of the Java Platform, Standard Edition Runtime Environment (JRE), allows you to run Java programs on your PC and as you browse the Internet. The `rJava` package used by `loadeR` and `loadeR.ECOMS` packages needs JRE in order to use the powerful capabilities of the ?netCDF Java Library.

It is very likely that JRE is already installed on your computer, but just in case it is not, here are some instructions on how to get it.

Linux users

To find out if JRE is installed in your machine, and in negative case to get instructions on how to install it, please refer to ?this link

Windows / Mac users

Click on the link below to download the JRE installer and perform the installation procedure by keeping the default installer settings. For Windows users, please be sure that you have downloaded the 32-bit JRE installer as indicated below.

?JRE installer download link (version 7 update 60):

Java SE Runtime Environment 7u60

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

○ **Accept License Agreement** ◉ **Decline License Agreement**

Product / File Description	File Size	Download
Linux x86	31.55 MB	jre-7u60-linux-i586.rpm
Linux x86	46.18 MB	jre-7u60-linux-i586.tar.gz
Linux x64	32.06 MB	jre-7u60-linux-x64.rpm
Linux x64	44.81 MB	jre-7u60-linux-x64.tar.gz
Mac OS X x64	48.52 MB	jre-7u60-macosx-x64.dmg
Mac OS X x64	44.5 MB	jre-7u60-macosx-x64.tar.gz
Solaris x86	52.17 MB	jre-7u60-solaris-i586.tar.gz
Solaris x64	16.12 MB	jre-7u60-solaris-x64.tar.gz
Solaris SPARC	54.92 MB	jre-7u60-solaris-sparc.tar.gz
Solaris SPARC 64-bit	18.16 MB	jre-7u60-solaris-sparcv9.tar.gz
Windows x86 Online	0.88 MB	jre-7u60-windows-i586-iftw.exe
Windows x86 Offline	28.04 MB	jre-7u60-windows-i586.exe
Windows x86	39.94 MB	jre-7u60-windows-i586.tar.gz
Windows x64	29.55 MB	jre-7u60-windows-x64.exe
Windows x64	41.64 MB	jre-7u60-windows-x64.tar.gz

Installing the `rJava` package

?rJava is a R package providing a low-level interface to Java from R. If Java is installed and adequately configured in your computer, the `rJava` package will be automatically downloaded if not present when installing the `loadR` package, as any other package dependency. `rJava` can be also installed directly by:

```
> install.packages("rJava")
```

Once `rJava` is installed, you may want to know the characteristics of the java version used by R:

```
library(rJava)
.jinit()
```

```
[1] 0
.jcall("java/lang/System", "S", "getProperty", "java.vendor")
[1] "Oracle Corporation"
.jcall("java/lang/System", "S", "getProperty", "java.runtime.version")
[1] "1.7.0-b147"
.jcall("java/lang/System", "S", "getProperty", "os.arch")
[1] "x86"
```

Therefore, in principle you should not worry about this and could skip this section. But just in case something related to `rJava` goes wrong during the installation, please bear in mind the following information:

Linux

Linux users can alternatively use the *apt-get* choice from the terminal:

```
~$ apt-get install r-cran-rjava
```

(In case of doubts, there is a recent discussion on both choices ?at this link that you may find helpful).

You can also find useful information about the installation and configuration of openJDK in ?this thread of the Ubuntu forum.

Mac OS X

Mac OS X comes with Java pre-installed, but sometimes an old version (~1.6) is used by the system, even if a new version is installed as described in the previous section. In this case, the problem is changing the default Java version to be used by R (or RStudio) to a newer one. You can check the version which is used in an R session by

```
> library(rJava)
> .jinit()
> .jcall("java/lang/System", "S", "getProperty", "java.runtime.version")
```

If you are having problems with newly installed Java version please take a look to this:
?http://stackoverflow.com/questions/26948777/how-can-i-make-rjava-use-the-newer-version-of-java-on-osx

In some cases, the problem is fixed for R but not for RStudio. In this situation, opening RStudio from the terminal may solve the problem.

```
user$ /Applications/RStudio.app/Contents/MacOS/RStudio
```

Take a look also at the following link:
?https://support.rstudio.com/hc/communities/public/questions/200650933-rJava-fails-to-load-in-RStudio-Desktop-OS-X

Windows

For Windows 7, you can find some quick advice on how to get up and running with R + rJava ?at this link

If an error likes this one appears on your `R` console:

```
Error : .onLoad failed in loadNamespace() for 'rJava', details:
 call: inDL(x, as.logical(local), as.logical(now), ...)
 error: unable to load shared object 'C:/Users/antonio/Documents/R/win-library/3.2/rJava/libs/x64/rJava.dll':
 LoadLibrary failure:  %1 no es una aplicación Win32 válida.

Error : package 'rJava' could not be loaded
```

it can be due to a misconfiguration of `JAVA_HOME` environment variable.

Please check that the corresponfing value points to the **correct Java version and bit architecture** (32 or 64).

```
> Sys.getenv("JAVA_HOME")
```

You can remove from system environment and restart `R` or remove from `R` sesssion:

```
> if (Sys.getenv("JAVA_HOME")!="")
+     Sys.setenv(JAVA_HOME="")
```

or you can force the JVM ():

```
> Sys.setenv(JAVA_HOME="C:\\Program Files (x86)\\Java\jdk1.7.0\\jre")
```

and check which JVM is using rJava:

```
>.jinit()
> .jcall("java/lang/System", "S", "getProperty", "java.vendor")
[1] "Oracle Corporation"
> .jcall("java/lang/System", "S", "getProperty", "java.runtime.version")
[1] "1.7.0-b147"
> .jcall("java/lang/System", "S", "getProperty", "os.arch")
[1] "x86"
```