PI Machine Learning Installation Guide

Learn how to set up PI Machine Learning for your installation.

- General security considerations
- Installation Windows
- Installation Linux
- Install Rserve remotely on a Linux server
- Advanced configuration

General security considerations

Never execute the R-Server as a root user. It's best to run the R-Server as a restricted user with limited rights. Be aware that a user with access to the R-Server can potentially delete all files for which he has write permissions. Under Linux and new versions of Windows Server 2016 it might be advisable to run the R-Server inside of a Docker container.

Installation Windows

For windows we recommend using the Microsoft R Open interpreter, which is available for download at: https://mran.microsoft.com/download/ . Next, you need to install the deployr-rserve library, which takes care of the communication between CPM and R. Deployr-rserve is available at https://github.com/Microsoft/deployr-rserve/releases . Download the Windows zip file (at the time of writing deployrRserve_8.0.5.1.zip).

Afterwards, navigate to the Download folder in a command line. Then execute

```
R CMD INSTALL deployrRserve_8.0.5.1.zip
```

where deployrRserve_8.0.5.1.zip stands for the filename of the file you just downloaded. You might have to give the full path to R for this to work, so something like:

```
"C:\Program Files\R\R-3.3.2\bin\x64\R" CMD INSTALL "C:\Program Files\R\deployrRserve_8.0.5.1.zip"
```

Now open the R interpreter (as a restricted user, see section "Security considerations"!) and execute

```
library(deployrRserve);
rservePath <- system.file(package="deployrRserve", "Rserve.exe");
cmd <- paste(file.path(R.home(),"bin","R"),"CMD",rservePath);
system(cmd);
```

The Rserve instance should now be ready to use. Note that on some Windows systems the pathnames don't get constructed properly. There you might have to convert the Rserve Path to have a format like "C:\PROG~\R" instead of "C:\Program Files\R".
All that remains is to edit your config.properties for CPM. Simply add the following lines to your config.properties:

```
rserve.enabled = true
rserve.port = 7004
```

Now you should be ready to use the R-Integration. Note that you will have to restart the Rserve server manually after a system restart.

Installation Linux

Most distributions should provide the R interpreter in their package system.

For example in Arch Linux: pacman -S R

In Ubuntu: apt-get install r-base

After you have performed the R-Setup, start up the R-Interpreter (as a restricted user, see section "Security considerations"!).
On the command line type:

```
install.packages('Rserve')
```

A window with a list of mirrors will open, just select one close to you and Rserve is installed. The most likely reason for any error messages here is that your firewall blocks your connection. In such a case please consult your system administrator.
Next, load the Rserve library and start the server:

```
library(Rserve);
Rserve();
```

All that remains is to edit your config.properties for CPM. Simply add the following lines to your config.properties:

```
rserve.enabled = true
rserve.port = 6311
```

Now you should be ready to use the R-Integration. Note that Rserve will be started as a daemon and therefore keep running after you exit the interpreter.

You will have to restart the Rserve server manually after a system restart.

Install Rserve remotely on a Linux server

tested system: Ubuntu

```
$ sudo apt-get install r-base
$ R
> install.packages("Rserve")

// run the Rserve instance and make it available remotely

> library(Rserve)
> Rserve(args='--vanilla --RS-enable-remote')
>q()
```

Run Rserve daemon from console without running R:

```
$ Rscript -e "library(Rserve); Rserve(args='--vanilla --RS-workdir /home/ruser/workdir --RS-enable-remote')"
```

Test remote server access:
$ telnet *remoteServerIP* 6331

Advanced configuration

There are three other configuration options in the config.properties:

- rserve.hostname: Sets the address where the Rserve instance is running. The default value is 127.0.0.1 (i.e. localhost)

- rserve.timeout: Sets a timeout for an Rserve query. After the end of the timeout no further attempt to recieve the requested data is made and an error will be displayed.
- rserve.terminate: If set to true, processes running queries that passed the timeout will be sent a SIGKILL signal. This should terminate the process completely, but might leave some resources hanging.