Installing Docker

Check target

C:\Users\admin\AppData\Local\Programs\Git\bin\bash.exe --login -i "C:\Program Files\Docker Toolbox\start.sh"

Installing, Updating and Uninstalling Repository Plugins in IntelliJ

<https://www.jetbrains.com/help/idea/installing-updating-and-uninstalling-repository-plugins.html>

* ssh remote\_username@remote\_host
* Le caractère"**~**" représente le répertoire HOME de l'utilisateur.

$ **cd ~**  
$ **pwd**  
/home/toto

* tail -f logs/Scheduler.log
* cat Scheduler.log | grep "WARN"
* $ tail -n 500 Scheduler.log.1

#### liste dynamique des processus pour surveiller l’activité du system top

Example of selection script :

// the IP mask of the host to be selected

var ipMask = “192.168.0.\*“;

/\*

\* Check if the IP of the host that holds the node is equal to the IP given in argument

\* The IP can be given as x.x.x.x or using the token \* to match a network for example. (ie x.x.x.\*)

\*/

if (org.ow2.proactive.scripting.helper.selection.SelectionUtils.checkIp(ipMask)) {

selected = true;

} else {

selected = false;

}

3 important files (lors d’un ajout d’un sous-module):

Build.gradle 🡪 PROJECT NAME : scheduling

Settings.gradle

Gradle.properties

$ find . -type f | xargs grep -Eni "PA\_TASK\_REPLICATION"

$ find . -type f | xargs grep -Eni "hsql"

find ~ type f -name "hsqldb.jar"

strings scheduler.lobs

xargs scheduler.lobs

$ find . -name '\*.c' | grep 'stdlib.h'

This pipes the output (stdout)\* from find to (stdin of)\* grep 'stdlib.h' *as text* (ie the filesnames are treated as text). grep does its usual thing and finds the matching lines in this text (any file names which themselves contain the pattern). The contents of the files are never read.

$ find . -name '\*.c' | xargs grep 'stdlib.h'

This constructs a **command** grep 'stdlib.h' to which each result from find is an argument - so this will look for matches **inside** each file found by find (xargs can be thought of as turning its stdin into arguments to the given commands)\*

Use -type f in your find command, or you will get errors from grep for matching directories. Also, if the filenames have spaces, xargs will screw up badly, so use the null separator by adding -print0 and xargs -0 for more reliable results:

find . -type f -name '\*.c' -print0 | xargs -0 grep 'stdlib.h'

<https://www.cyberciti.biz/faq/howto-use-grep-command-in-linux-unix/>

<http://www.sqlpac.com/referentiel/docs/unix-find-grep-commandes.html#.WUjjeOvyipo>

<http://www.linuxcertif.com/doc/keyword/grep/>

<https://www.lifewire.com/uses-of-linux-command-find-2201100>

Pour lancer tous les tests functional:

./gradlew spotlessApply clean build functionalTest

## Pour lancer un test fonctionnel particulier:

./gradlew spotlessApply clean build rm:rm-server:functionalTest

Tu vas dans le répertoire rm/rm-server

puis tu fais ../../gradlew funcTest -DfunctionalTest.single="functionaltests/nodesource/TestLocalInfrastructureRestartDownNodesPolicy\*"

../../gradlew funcTest -DfunctionalTest.single="functionaltests/scripts/TestScriptEngines\*"

../../gradlew funcTest -DfunctionalTest.single="functionaltests/scripts/TestScriptEngines\*"

## Connaître la version de gradle et gradle wrapper

gradle -v

./gradlew -v

## execution de cette commande dans le repo pour acquerir gradle wrapper après avoir installé gradle (X.Y est la version similaire à celle du projet de base)

gradle wrapper --gradle-version X.Y