Chapter 1

Pharo Zero Conf

with the participation of:
Camillo Bruni

Weren't you fed up not be able to install Pharo from a single command line or to pass it arguments? Using a nice debugger and an interactive environment development does not mean that Pharo developers do not value automatic scripts and love command line. Yes we do and we want the best of both worlds! Since Pharo 2.0, Pharo supports a way to define and handle command line argument. We will present that in this chapter.

We really wanted it to free our mind of retaining arbitrary information. A zero configuration is then a script that automatically download everything you need to get started.

In this chapter we will show how to get the zeroConf scripts family for Pharo as well as how you can pass argument to the environment from the command-line.

1.1 Getting the latest VM and the Image

First here is a way to download a zero configuration script that downloads a script to download the latest 2.0 Pharo image and vm.

```
wget http://files.pharo.org/script/ciPharo20PharoVM.sh
```

Note that files.pharo.org is an alias for pharo.gforge.inria.fr/ci. So we suggest to use wget instead of curl because curl does not deal well with redirections.

To execute the script that we just downloaded, you should change its permissions using chmod a+x or invoke it via bash as follows. We suggest you also to have a look at the script help.

```
bash ./ciPharo20PharoVM.sh --help
```

What the help mentions is that the script will download the current vm and put it into the vm folder, vm.sh a script to launch the system, vm-ui.sh a script to launch the image in ui mode, the image and its associated changes.

```
Result in the current directory:
   vm                    VM directory
   vm.sh                 Script forwarding to the VM in vm
   Pharo.image           The latest pharo image
   Pharo.changes         The corresponding pharo changes
```

Stéf ►*update to last version*◄

Grabbing and executing it. If you just want to directly execute the script you can also do the following

```
wget http://files.pharo.org/ci/script/ciPharo20PharoVM.sh | bash
```

If you do not like the log of web use --quiet -O.

```
wget --quiet -O - http://files.pharo.org/ci/script/ciPharo20PharoVM.sh | bash
```

Note for the believers in automated tasks. The scripts are fetched automatically from our jenkins server (https://ci.inria.fr/pharo/job/Scripts-download/) from the gitorious server https://gitorious.org/pharo-build/pharo-build. Yes we believe in automated tasks that free our energy.

1.2 Getting the latest VM only

You can also use different scripts. For example ciPharoVM.sh only downloads the latest vm.

```
wget http://pharo.gforge.inria.fr/ci/script/ciPharoVM.sh | bash
```

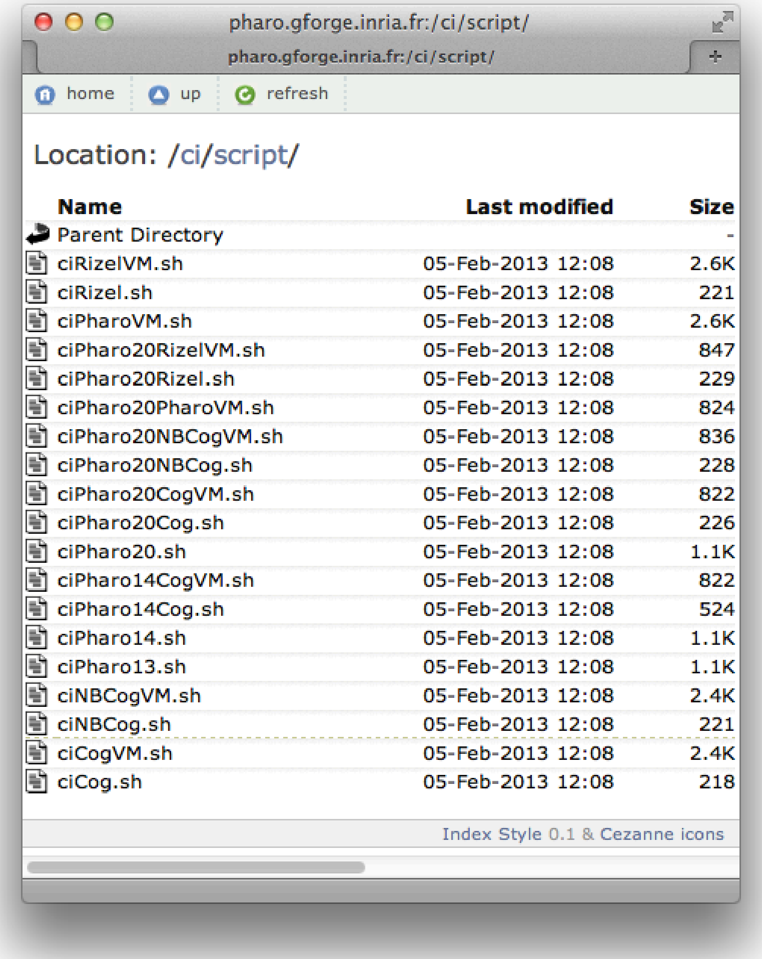
Again as any script you can always check its help message.

```
This script will download the latest Pharo VM

Result in the current directory:
   vm              directory containing the VM
   vm.sh           script forwarding to the VM inside vm
```

Stéf ►*update the script to the latest version which should include vm-ui.sh*◄

Figure 1.1 shows the list of scripts available that you can get at http://pharo.gforge.inria.fr/ci/script/.


pharo.gforge.inria.fr:/ci/script/
pharo.gforge.inria.fr:/ci/script/
+
home
up
refresh

Location: /ci/script/

Name	Last modified	Size
Parent Directory		-
ciRizelVM.sh	05-Feb-2013 12:08	2.6K
ciRizel.sh	05-Feb-2013 12:08	221
ciPharoVM.sh	05-Feb-2013 12:08	2.6K
ciPharo20RizelVM.sh	05-Feb-2013 12:08	847
ciPharo20Rizel.sh	05-Feb-2013 12:08	229
ciPharo20PharoVM.sh	05-Feb-2013 12:08	824
ciPharo20NBCogVM.sh	05-Feb-2013 12:08	836
ciPharo20NBCog.sh	05-Feb-2013 12:08	228
ciPharo20CogVM.sh	05-Feb-2013 12:08	822
ciPharo20Cog.sh	05-Feb-2013 12:08	226
ciPharo20.sh	05-Feb-2013 12:08	1.1K
ciPharo14CogVM.sh	05-Feb-2013 12:08	822
ciPharo14Cog.sh	05-Feb-2013 12:08	524
ciPharo14.sh	05-Feb-2013 12:08	1.1K
ciPharo13.sh	05-Feb-2013 12:08	1.1K
ciNBCogVM.sh	05-Feb-2013 12:08	2.4K
ciNBCog.sh	05-Feb-2013 12:08	221
ciCogVM.sh	05-Feb-2013 12:08	2.4K
ciCog.sh	05-Feb-2013 12:08	218

Index Style 0.1 & Cezanne icons

Figure 1.1: All the scripts are available at http://pharo.gforge.inria.fr/ci/script/ .

Note that scripts

1.3 The scripts

Here is a typical script. We list it here for archival purpose and the definition may change in the future but you should get the idea.

ciPharo20PharoVM.sh.

```
#!/bin/bash

# stop the script if a single command fails
set -e

# ARHUMENT HANDLING ===========================================================

if { [ "$1" = "-h" ] || [ "$1" = "--help" ]; }; then
    echo "This script will download the latest Pharo 2.0 image and VM

Result in the current directory:
    vm               VM directory
    vm.sh            Script forwarding to the VM in vm
    Pharo.image      The latest pharo image
    Pharo.changes    The corresponding pharo changes"
    exit 0
elif [ $# -gt 0 ]; then
    echo "--help is the only argument allowed"
    exit 1
fi

# FETCH DATA ==================================================================
wget --quiet -qO - http://pharo.gforge.inria.fr/ci/script/ciPharoVM.sh | bash
wget --quiet -qO - http://pharo.gforge.inria.fr/ci/script/ciPharo20.sh | bash
```

Now that you get ready to launch Pharo nearly everywhere, let us look at the way you can always script the image from the command-line.

1.4 Scripting the Image from the commandLine

Now let us have a look at how the command line is handled in Pharo (2.0). As usual we will start to show you how to find your way alone.

Stéf ►*here*◄

How to find our way

```
./vm.sh Pharo.image --help
```

```
Usage: [<subcommand>] [--help] [--copyright] [--version] [--list]
    --help       print this help message
    --copyright  print the copyrights
    --version    print the version for the image and the vm
    --list       list a description of all active command line handlers
```

```
   <subcommand> a valid subcommand in −−list

Documentation:
A DefaultCommandLineHandler handles default command line arguments and options.
The DefaultCommandLineHandler is activated before all other handlers.
It first checks if another handler is available. If so it will activate the found handler.
```

```
./vm.sh Pharo.image −−version
M:    NBCoInterpreter NativeBoost−CogPlugin−IgorStasenko.15 uuid: 44b6b681−38f1−
      4a9e−b6ee−8769b499576a Dec 18 2012
NBCogit NativeBoost−CogPlugin−IgorStasenko.15 uuid: 44b6b681−38f1−4a9e−b6ee−
      8769b499576a Dec 18 2012
git://gitorious.org/cogvm/blessed.git Commit: 452863
      bdfba2ba0b188e7b172e9bc597a2caa928 Date: 2012−12−07 16:49:46 +0100 By:
      Esteban Lorenzano <estebanlm@gmail.com> Jenkins build #5922
```

```
./vm.sh Pharo.image −−list
Currently installed Command Line Handlers:
   st            handles .st source files
   Fuel          Handles fuel files
   config        Install and inspect Metacello Configurations from the command line
   save          Rename the image and changes file
   test          A command line test runner
   update        load updates
   printVersion  Print image version
   eval          directly evaluates passed in one line scripts
```

Evaluating Pharo Expressions

```
./vm.sh Pharo.image eval '1+2'
```

```
./vm.sh Pharo.image eval −−help
Usage: eval [−−help] <smalltalk expression>
   −−help    list this help message
   <smallltalk expression>  a valid Smalltalk expression which is evaluated and
                      the result is printed on stdout

Documentation:
A CommandLineHandler that reads a string from the command line, outputs the
      evaluated result and quits the image.

This handler either evaluates the arguments passed to the image:
   $PHARO_VM my.image eval  1 + 2

or it can read directly from stdin:
   echo "1+2" | $PHARO_VM my.image eval
```

Loading Metacello Configuration

```
./vm.sh Pharo.image config −−help
Usage: config [−−help] <repository url> [<configuration>] [−−install[=<version>]] [−−
      group=<group>] [−−username=<username>] [−−password=<password>]
   −−help              show this help message
   <repository url>    A Monticello repository name
   <configuration>     A valid Metacello Configuration name
   <version>           A valid version for the given configuration
   <group>             A valid Metacello group name
   <username>          An optional username to access the configuration's repository
   <password>          An optional password to access the configuration's repository

Examples:
   # display this help message
   $PharoVM My.image config

   # list all configurations of a repository
   $PharoVM My.image config $MC_REPOS_URL

   # list all the available versions of a confgurtation
   $PharoVM My.image config $MC_REPOS_URL ConfigurationOfFoo

   # install the stable version
   $PharoVM My.image config $MC_REPOS_URL ConfigurationOfFoo −−install

   #install a specific version '1.5'
   $PharoVM My.image config $MC_REPOS_URL ConfigurationOfFoo −−install=1.5

   #install a specific version '1.5' and only a specific group 'Tests'
   $PharoVM My.image config $MC_REPOS_URL ConfigurationOfFoo −−install=1.5 −−
      group=Tests
```

1.5 Using Zero conf on Jenkins

For example here is the configuration for Athens.

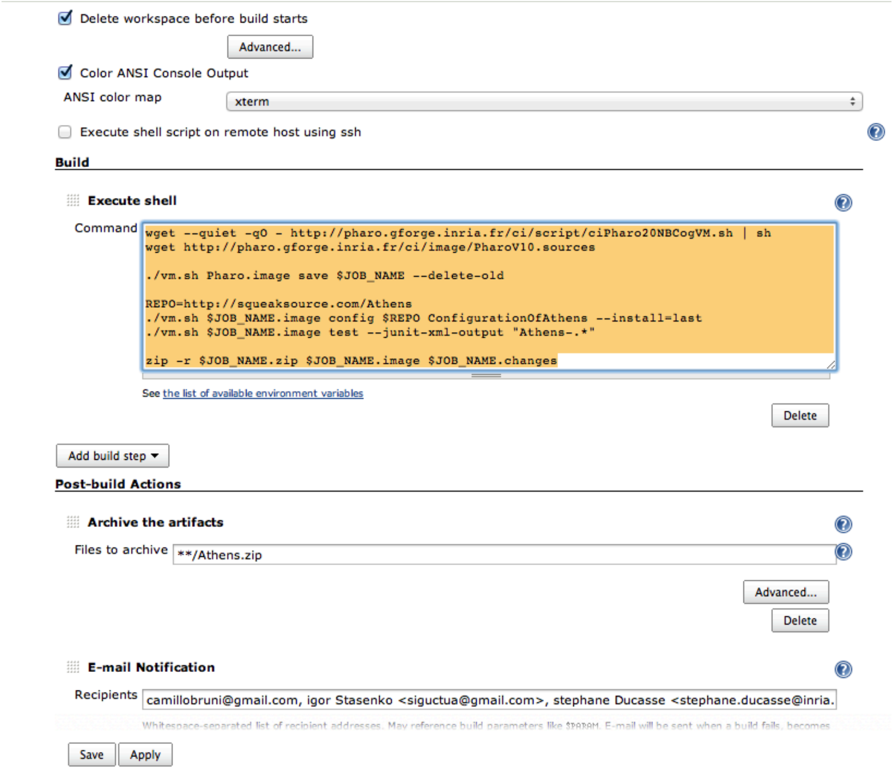
```
wget −−quiet −qO − http://pharo.gforge.inria.fr/ci/script/ciPharo20NBCogVM.sh | sh
wget http://pharo.gforge.inria.fr/ci/image/PharoV10.sources

./vm.sh Pharo.image save $JOB_NAME −−delete−old

REPO=http://squeaksource.com/Athens
```

```
./vm.sh $JOB_NAME.image config $REPO ConfigurationOfAthens --install=last
./vm.sh $JOB_NAME.image test --junit-xml-output "Athens-.*"

zip -r $JOB_NAME.zip $JOB_NAME.image $JOB_NAME.changes
```



☑ Delete workspace before build starts

Advanced...

☑ Color ANSI Console Output

ANSI color map: xterm

□ Execute shell script on remote host using ssh

Build

Execute shell

Command

```
wget --quiet -qO - http://pharo.gforge.inria.fr/ci/script/ciPharo20NBCogVM.sh | sh
wget http://pharo.gforge.inria.fr/ci/image/PharoV10.sources

./vm.sh Pharo.image save $JOB_NAME --delete-old

REPO=http://squeaksource.com/Athens
./vm.sh $JOB_NAME.image config $REPO ConfigurationOfAthens --install=last
./vm.sh $JOB_NAME.image test --junit-xml-output "Athens-.*"

zip -r $JOB_NAME.zip $JOB_NAME.image $JOB_NAME.changes
```

See the list of available environment variables

Delete

Add build step

Post-build Actions

Archive the artifacts

Files to archive: **/Athens.zip

Advanced...

Delete

E-mail Notification

Recipients: camillobruni@gmail.com, igor Stasenko <siguctua@gmail.com>, stephane Ducasse <stephane.ducasse@inria.

Whitespace-separated list of recipient addresses. May reference build parameters like $PARAM. E-mail will be sent when a build fails, becomes

Save Apply

Figure 1.2: ZeroConf .

1.6 Conclusion

As conclusion, you have now tools to quickly create UIs and to reuse them. Since the reuse is really a strong value for specs, keep in mind that your tools can be reused. So do not forget to provide a proper API for it.