SWISH Installer Documentation

Franco Masotti

Copyright © 2016-2017 Franco Masotti `franco.masotti@student.unife.it`

Copying and distribution of this file, with or without modification, are permitted in any medium without royalty provided the copyright notice and this notice are preserved.

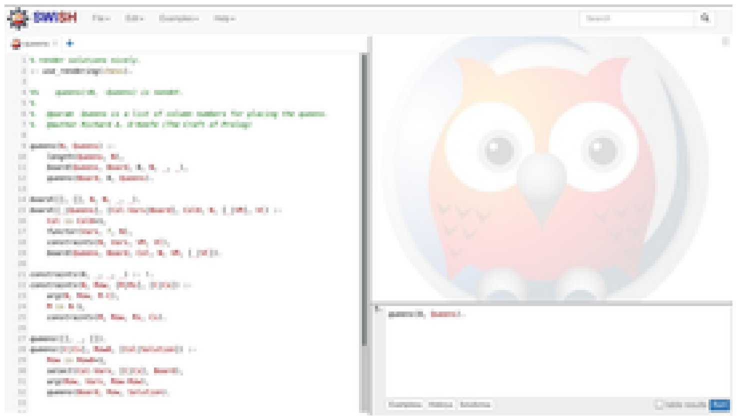
Table of Contents

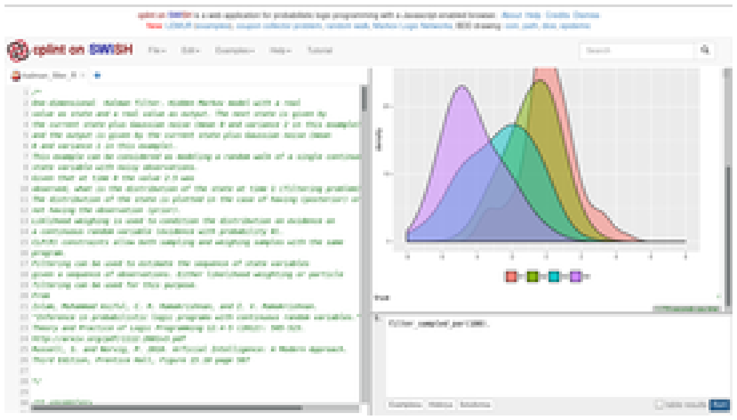
1 About

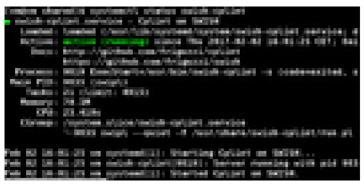
1.1 About

The purpose of this repository is to handle the creation of serveral packages and helpers in order to install Cplint[1] on SWISH[2] with an Rserve[3] environment, as well as the "vanilla" version of SWISH.

SWISH relies on SWI Prolog[4], a prolog interpteter.







1 See item [**Cplint**] in Chapter 10 [References], page 20.
2 See item [**SWISH**] in Chapter 10 [References], page 20.
3 See item [**Rserve**] in Chapter 10 [References], page 20.
4 See item [**Swipl**] in Chapter 10 [References], page 20.

1.1.1 Terminology

In order to understand this documentation correctly some terminology used here must be explained.

`<` and `>` means pseudocode.

`#` means that the command must be executed by `root`.

`$` means that the command must be executed by the current user.

1.1.2 Directory listings

When you see something like the following, it represents a directory listing of the first named (head) directory. These representations are generated from `$ tree --charset=ascii -d <dirname>`.

```
common/
|-- rserve-sandbox
|-- swish
`-- swish-cplint
```

2 Demonstrations

If you want to see how the final result might look like, have a look at `http://cplint.lamping.unife.it/` for Cplint on SWISH and `http://swish.swi-prolog.org/` for the `"`vanilla`"` version of SWISH.

3 Available packages and descriptions

3.1 Available packages

Distributions	**SWISH**	**Cplint on SWISH**	**Cplint on SWISH binary**	**Rserve sandbox**	**Rserve sandbox binary**
Arch Linux	`swish`	`swish-cplint`	`swish-cplint-bin`	`rserve-sandbox-docker`	`rserve-sandbox-docker-bin`

3.2 Descriptions

The following terms will be used throughout the document to identify the packages in a generic manner (non-distribution specific).

-	**SWISH**	**Cplint on SWISH**	**Cplint on SWISH binary**	**Rserve sandbox**	**Rserve sandbox binary**
Description	the vanilla version of SWISH	SWISH with the Cplint suite. Uses Rserve sandbox as a main dependency	precompiled version of Cplint on SWISH. Uses Rserve sandbox binary as well as precompiled web components	an R environment running inside a docker container	precompiled version of Rserve sandbox

4 Installation

4.1 Arch Linux

There at least two possibilities to install the packages.

4.1.1 Using an AUR helper

Yaourt[1] is among the most popular AUR[2] helpers available. The following command[3] will install all the dependencies automatically. Use the package names described in the Chapter 3 [Available packages], page 4, section.

```
$ yaourt -Sa s <package name> --noconfirm
```

> **Note:** When you install Yaourt on Parabola GNU/Linux-libre[4] you will be asked to remove `your-freedom`.

4.1.2 Without using an AUR helper

Adapt this command to match the to be installed package:

```
$ wget "https://aur.archlinux.org/cgit/\
aur.git/snapshot/<package name>.tar.gz"
$ tar -zxvf <package name>.tar.gz
$ cd <package name>
$ makepkg -sri --noconfirm
```

4.1.2.1 Cplint on SWISH

- Install `swi-prolog-devel`
- Install `rserve-sandbox-docker`
- Install `swish-cplint`

4.1.2.2 Cplint on SWISH binary

- Install `swi-prolog-devel`
- Install `rserve-sandbox-docker-bin`
- Install `swish-cplint-bin`

4.1.2.3 SWISH

- Install `swish`

4.1.3 Daemons management

Arch Linux and derivative distros use Systemd[5] as the init system. First see Chapter 5 [Components download], page 7, and then see Chapter 6 [Daemons management], page 8.

[1] See item [**Yaourt**] in Chapter 10 [References], page 20.
[2] See item [**AUR**] in Chapter 10 [References], page 20.
[3] See item [**Packages on the AUR**] in Chapter 10 [References], page 20.
[4] See item [**Parabola**] in Chapter 10 [References], page 20.
[5] See item [**systemd**] in Chapter 10 [References], page 20.

4.2 Debian

TODO

5 Components download

This section only applies to Cplint on SWISH and Cplint on SWISH binary.

Before running the daemons some components must be downloaded. The first thing to do is to start Docker[1] manually. See Chapter 6 [Daemons management], page 8, for more information.

5.1 Rserve sandbox and Rserve sandbox binary

To download the Docker image file as well as all dependencies for R[2], or to load the docker image, run the following command:

```
$ sudo -u rsd rserve-sandbox-docker -i
```

5.2 Cplint on SWISH and Cplint on SWISH binary

Before reading on, start the Rserve sandbox service, to avoid errors on some prolog tests executed by the Cplint R library.

To download all the prolog packages necessary in order to run Cplint on SWISH correcly:

```
$ sudo -u swish swish-cplint -i
```

[1] See item **[Docker]** in Chapter 10 [References], page 20.

[2] See item **[R]** in Chapter 10 [References], page 20.

6 Daemons management

Commands to manage the daemons depend on the init system in use.

6.1 Systemd

6.1.1 Cplint on SWISH

Docker is a dependency which must be started manually during the setup.

```
# systemctl start docker
```

The following command will run `swish-cplint` as well as `rserve-sandbox-docker` as its dependency:

```
# systemctl start swish-cplint
```

To start `swish-cplint` and `rserve-sandbox-docker` at boot:

```
# systemctl enable swish-cplint
```

You can stop both services with:

```
# systemctl stop swish-cplint
# systemctl stop rserve-sandbox-docker
```

> **Note:** stopping `swish-cplint` does not imply that `rserve-sandbox-docker` will be stopped.

To check the status of both daemons:

```
# systemctl status swish-cplint
# systemctl status rserve-sandbox-docker
```

6.1.2 Cplint on SWISH binary

For simplicity, services names do not change, so the same instructions of Cplint on SWISH are also applicable here.

6.1.3 SWISH

Running and managing SWISH alone is very similar to the previous method: instead of using `swish-cplint` as part of the commands, you must use `swish`. You don't need to worry about Rserve or Docker.

7 Accessing the server

To access the web interface you need a JavaScript[1] enabled browser and you have to connect to port 3050. For example: `http://localhost:3050` or `http://127.0.0.1:3050`

[1] See item **[JavaScript]** in Chapter 10 [References], page 20.

8 Development

8.1 Repository structure

```
.
|-- common
|   |-- rserve-sandbox-docker
|   |-- rserve-sandbox-docker-bin
|   |-- swish
|   |-- swish-cplint
|   `-- swish-cplint-bin -> swish-cplint/
`-- distributions
    |-- archLinux-based
    |   |-- dest
    |   |-- packages
    |   |   |-- rserve-sandbox-docker
    |   |   |-- rserve-sandbox-docker-bin
    |   |   |-- swish
    |   |   |-- swish-cplint
    |   |   `-- swish-cplint-bin
    |   `-- test
    `-- debian-based
```

The `distributions` directory contains all the files useful to build packages for a specific distribution. Makefiles are used to achieve this.

The `common` directory contains all the files which are not distribution specific.

8.1.1 Distribution-specific files

See Chapter 8 [Building the packages], page 10,

8.1.2 Common files

```
common/
|-- rserve-sandbox-docker
|   `-- run.sh
|-- rserve-sandbox-docker-bin
|   `-- run.sh -> ../rserve-sandbox-docker/run.sh
|-- shared_functions.sh
|-- swish
|   |-- run.pl
|   `-- run.sh
|-- swish-cplint
|   |-- install_web_iface_deps.pl
|   |-- run.pl
|   `-- run.sh
`-- swish-cplint-bin -> swish-cplint/
```

The `run.sh` files are helpers to start and stop the daemons. These helpers should work on any distribution. You can edit the variables at the top of each file accordingly.

The `run.pl` files are a modified version of the original files with the same name. These have been created in order to lauch SWISH as a background program. Without the changes contained in those files, you couldn't run SWISH in the background.

Another important file is `shared_functions.sh` which contains all common functions for the `run.sh` files. It must be appended to every `run.sh`, within a Makefile for example with the following:

```
$ cat shared_functions.sh >> {rserve-sandbox*,swish,swish-cplint}/run.sh
```

Cplint on SWISH contains a post installation script called `install_web_iface_deps.pl`. This script, called by `run.sh`, simply installs all the necessary Prolog dependencies automatically.

8.2 Guidelines to create packages

Makefiles are used to build the package in a new directory by copying all necessary files (also from the `common` directory). Each Makefile is distribution-based specific.

You will read general information about the packages, all their install and remove actions and how to build your version of the modified packages.

Installation and removal actions must be done sequentially.

8.2.1 Rserve sandbox

8.2.1.1 General information

- Data directory

  `/usr/share/rserve-sandbox-docker`
- Dependencies
  - R
  - Docker

8.2.1.2 Install actions

- Pre
  - None
- During
  - Make a symbolic link to be able to call `rserve-sandbox-docker` from `/usr/bin`

    ```
    $ ln -s /usr/share/rserve-sandbox-docker/run.sh \
    /usr/bin/rserve-sandbox-docker
    ```
- Post
  - Add `rserve` user and group.

    ```
    # getent group rserve &>/dev/null || groupadd -g 32749 -r rserve >/dev/null
    # getent passwd rserve &>/dev/null || useradd -u 32749 -m -d /home/rserve \
    -s /bin/false -r -g rserve rserve >/dev/null
    # chmod 750 /home/rserve
    ```
  - Add `rsd` user and group.

    ```
    # getent group rsd &>/dev/null || groupadd -r rsd >/dev/null
    ```

```
# getent passwd rsd &>/dev/null || useradd \
-s /bin/false -r -g rsd rsd >/dev/null
```

- Add the new user to the `docker` group

  ```
  # gpasswd -a rsd docker >/dev/null
  ```

- Change ownership of the package data directory

  ```
  # chown -R rsd:rsd /usr/share/rserve-sandbox-docker
  ```

  Note: the `rsd` user and group was created in order to avoid privilege escalation, since any user which belongs to the `docker` group is equivalent to `root`[1].

8.2.1.3 Remove actions

- Pre
  - None
- During
  - None
- Post
  - Tell the user that `/home/rserve`, `rsd` user and group, `rserve` user and group and all the Docker files can be removed (this depends on the package remove policies of the chosen distro).

8.2.2 Rserve sandbox binary

- Same as Rserve sandbox.

8.2.3 Cplint on SWISH

8.2.3.1 General information

- Data directory

  `/usr/share/swish-cplint`
- Dependencies
  - SWI Prolog (developement version)
  - Git[2]
  - Graphviz[3]
  - libXinerama[4]
  - libXpm[5]
  - Rserve sandbox
  - Bower (make dependency)[6]

[1] See item [**Docker root privileges**] in Chapter 10 [References], page 20.
[2] See item [**Git**] in Chapter 10 [References], page 20.
[3] See item [**Graphviz**] in Chapter 10 [References], page 20.
[4] See item [**libXinerama**] in Chapter 10 [References], page 20.
[5] See item [**libXpm**] in Chapter 10 [References], page 20.
[6] See item [**Bower**] in Chapter 10 [References], page 20.

8.2.3.2 Install actions

- Pre
  - Compile the web dependencies
    ```
    $ bower --allow-root install
    $ make src
    ```
  - Copy `run.pl`, `run.sh` and `install_web_iface_deps.pl` in SWISH's root directory.
- During
  - Make a symlink to be able to call `swish-cplint` from `/usr/bin`
    ```
    $ ln -s /usr/share/swish-cplint/run.sh /usr/bin/swish-cplint
    ```
- Post
  - Add `swish` user and group
    ```
    # getent group swish &>/dev/null || groupadd -r swish >/dev/null
    # getent passwd swish &>/dev/null || useradd -m -d /home/swish \
    -r -g swish swish >/dev/null
    ```
  - Add `swish` user to the previously created `rserve` group.
    ```
    # gpasswd -a swish rserve >/dev/null
    ```
  - Change ownership of the package data directory
    ```
    # chown -R swish:swish /usr/share/swish-cplint
    ```

8.2.3.3 Remove actions

- Pre
  - None
- During
  - None
- Post
  - Tell the user that `/home/swish` and `swish` user and group can be removed (this depends on the package remove policies of the chosen distro).

8.2.4 Cplint on SWISH binary

8.2.4.1 General information

- Data directory
  `/usr/share/swish`
- Dependencies
  - SWI Prolog (developement version)
  - Git
  - Graphviz
  - libXinerama
  - libXpm
  - Rserve sandbox binary

  - curl (make dependency)[7]
  - UnZip (make dependency)[8]

8.2.4.2 Install actions

- Pre
  - Download and unzip the web dependencies

    ```
    $ curl -o swish-bower-components.zip \
    http://www.swi-prolog.org/download/swish/swish-bower-components.zip
    $ unzip swish-bower-components.zip
    $ rm -rf swish-bower-components.zip
    ```

  - Copy `run.pl`, `run.sh` and `install_web_iface_deps.pl` in SWISH's root directory.
- During
  - Same as Cplint on SWISH
- Post
  - Same as Cplint on SWISH

8.2.4.3 Remove actions

- Same as Cplint on SWISH

8.2.5 SWISH

8.2.5.1 General information

- Data directory

  `/usr/share/swish`
- Dependencies
  - SWI Prolog (developement version)
  - libXinerama
  - libXpm
  - Bower (make dependency)

8.2.5.2 Install actions

- Pre
  - Compile the web dependencies

    ```
    $ bower --allow-root install
    $ make src
    ```

  - Copy `run.pl` and `run.sh` and in SWISH's root directory.
- During
  - Make a symlink to be able to call `swish` from `/usr/bin`

    ```
    $ ln -s /usr/share/swish/run.sh /usr/bin/swish
    ```

[7] See item **[curl]** in Chapter 10 [References], page 20.

[8] See item **[UnZip]** in Chapter 10 [References], page 20.

- Post
  - Add `swish` user and group

    ```
    # getent group swish &>/dev/null || groupadd -r swish >/dev/null
    # getent passwd swish &>/dev/null || useradd -m -d /home/swish \
    -r -g swish swish >/dev/null
    ```
  - Change ownership of the package data directory

    ```
    # chown -R swish:swish /usr/share/swish
    ```

8.2.5.3 Remove actions

- Pre
  - None
- During
  - None
- Post
  - Tell the user that `/home/swish` and `swish` user and group can be removed (this depends on the package remove policies of the chosen distro).

8.3 Building the packages

8.3.1 Arch Linux

```
archLinux-based/
|-- Makefile
|-- packages
|   |-- rserve-sandbox-docker
|   |   |-- .install
|   |   |-- PKGBUILD
|   |   `-- rserve-sandbox-docker.service
|   |-- rserve-sandbox-docker-bin
|   |   |-- .install -> ../rserve-sandbox-docker/.install
|   |   |-- PKGBUILD
|   |   `-- rserve-sandbox-docker.service -> ../rserve-sandbox-docker/rserve-sandbox-docker
|   |-- swish
|   |   |-- .install
|   |   |-- PKGBUILD
|   |   `-- swish.service
|   |-- swish-cplint
|   |   |-- .install
|   |   |-- PKGBUILD
|   |   `-- swish-cplint.service
|   `-- swish-cplint-bin
|       |-- .install -> ../swish-cplint/.install
|       |-- PKGBUILD
|       `-- swish-cplint.service -> ../swish-cplint/swish-cplint.service
`-- test
```

```
    `-- swish_installer_full_test
```

Once you've made changes you can run `$ make` then change directory into one of the new `dest/*.aur` generated directories and finally run `$ makepkg -sri` to install the package.

You also have the possibility to run the test script `$ ./test/swish_installer_full_test`. Here is the help page:

```
Usage: swish_installer_full_test [OPTION]
Full automated install test for the Arch Linux packages in swish-installer

Options are grouped in couples: one excludes the other

Options:
    -b, --binary                use binary packages where available
    -c, --compile               compile packages where available

    -d, --dev                   use development branches where available
    -m, --master                use default branches where available

    -e, --experimental          use swish-installer's dev branch
    -s, --stable                use swish-installer's master branch

    -h, --help                  print this help


Default: --compile --master --stable
Dependencies: git, wget.

Exit status:
 0  if OK,
 1  if an error occurred.

swish_installer_full_test online help:
<https://frnmst.github.io/swish-installer/swish-installer.html#Help-pages>
Full documentation at:
<https://frnmst.github.io/swish-installer/swish-installer.html>

Copyright    2017  Franco Masotti.  License GPLv3+: GNU GPL version 3 or
later <http://gnu.org/licenses/gpl.html>.
This  is  free  software: you are free to change and redistribute it.  There
is NO WARRANTY, to the extent  permitted by law.
```

8.4 Help pages

8.4.1 Rserve sandbox

```
rserve-sandbox-docker [OPTION]
Docker spec for running Rserve in a sandbox
```

```
Only a single option is permitted.
    -h      print this help
    -i      install image and dependencies
    -k      kill the container
    -r      remove the docker image
    -s      start the container

Exit status:
 0  if OK,
 1  some error occurred.

Full documentation at: <https://github.com/frnmst/rserve-sandbox>
```

8.4.2 Rserve sandbox binary

```
rserve-sandbox-docker [OPTION]
Docker spec for running Rserve in a sandbox

Only a single option is permitted.
    -h      print this help
    -i      load image
    -k      kill the container
    -r      remove the docker image
    -s      start the container

Exit status:
 0  if OK,
 1  some error occurred.

Full documentation at: <https://gitlab.com/frnmst/rserve-sandbox-bin>
```

8.4.3 Cplint on SWISH

```
swish-cplint [OPTION]
SWI-Prolog for SHaring: a SWI-Prolog web IDE integrated with the cplint suite

Only a single option is permitted.
    -h      print this help
    -i      install dependencies
    -k      kill swish-cplint
    -s      start swish-cplint

Exit status:
 0  if OK,
 1  some error occurred.

Full documentation at: <https://github.com/friguzzi/swish>
and at: <https://github.com/friguzzi/cplint>
```

8.4.4 SWISH

```
swish [OPTION]
SWI-Prolog for SHaring: a SWI-Prolog web IDE

Only a single option is permitted.
    -h      print this help
    -k      kill swish
    -s      start swish

Exit status:
 0  if OK,
 1  some error occurred.

Full documentation at: <https://github.com/SWI-Prolog/swish>
```

8.5 Compiling this documentation

The source of this documentation is under the `doc` directory.

To be able to compile it, you have to install several tex packages (for example: `texlive-most` and `texi2html` if you are using Arch Linux) that contain the following binaries:

```
makeinfo
texi2dvi
docbook2html
docbook2pdf
docbook2txt
texi2html
perl
```

After running `make`, a directory named `manual` is created and you can access the files by opening `index.html` with a browser.

9 Thanks

I want to thank the SWI Prolog, Arch Linux and Systemd communities as well the authors of the free software used here, which made the creation of these packages possible.

I also want to thank Fabrizio Riguzzi which tested the packages and gave me advices about them.

10 References

Some quotations reported here are taken directly from the respective web sites.

- **[Cplint]** "A suite of programs for reasoning with probabilistic logic programs". See `https://github.com/friguzzi/cplint`

- **[SWISH]** A web browser interface for SWI Prolog to share code. See `https://github.com/SWI-Prolog/swish` for the original version and `https://github.com/friguzzi/swish` for the version made by Fabrizio Riguzzi that uses Cplint.

- **[Rserve]** A docker image that enables to use the R and Rserve environment in a secure way. See `https://github.com/JanWielemaker/rserve-sandbox` for the original version made by Jan Wielemaker and `https://github.com/frnmst/rserve-sandbox/tree/distro-package` version by Franco Masotti which is used here. A client to access Rserve from Prolog is also necessary. See `https://github.com/JanWielemaker/rserve_client`

- **[R]** "R is an integrated suite of software facilities for data manipulation, calculation and graphical display". See `https://cran.r-project.org/doc/manuals/r-release/R-intro.html`
- **[Swipl]** "SWI-Prolog offers a comprehensive free Prolog environment. Since its start in 1987, SWI-Prolog development has been driven by the needs of real world applications. SWI-Prolog is widely used in research and education as well as commercial applications". See `http://www.swi-prolog.org/` and `https://github.com/SWI-Prolog/swipl-devel` which is the developement version used here.

- **[Yaourt]** "A pacman wrapper with extended features and AUR support". To install Yaourt follow the instructions reported on `https://archlinux.fr/yaourt-en`.

- **[AUR]** "The Arch User Repository (AUR) is a community-driven repository for Arch users. It contains package descriptions (PKGBUILDs) that allow you to compile a package from source with makepkg and then install it via pacman. The AUR was created to organize and share new packages from the community and to help expedite popular packages' inclusion into the community repository". See `https://aur.archlinux.org/` for the AUR homepage and `https://wiki.archlinux.org/index.php/Arch_User_Repository` for a complete explanation.

- **[Parabola]** "A fully free, simple, and lightweight operating system". See `https://parabola.nu`.

- **[systemd]** "systemd is a suite of basic building blocks for a Linux system. It provides a system and service manager that runs as PID 1 and starts the rest of the system". See `https://www.freedesktop.org/wiki/Software/systemd/`.

- [**Docker**] "Docker containers wrap a piece of software in a complete filesystem that contains everything needed to run: code, runtime, system tools, system libraries anything that can be installed on a server. This guarantees that the software will always run the same, regardless of its environment". See `https://www.docker.com/`.

- [**Packages on the AUR**] Here follows a list to the AUR packages:
  - SWI Prolog (developement version) `https://aur.archlinux.org/packages/swi-prolog-devel`
  - Rserve sandbox `https://aur.archlinux.org/packages/rserve-sandbox-docker/`
  - Cplint on SWISH `https://aur.archlinux.org/packages/swish-cplint/`
  - SWISH `https://aur.archlinux.org/packages/swish/`

- [**JavaScript**] "JavaScript is a high-level, dynamic, untyped, and interpreted programming language". See `https://en.wikipedia.org/wiki/JavaScript`

- [**Docker root privileges**] See `https://docs.docker.com/engine/security/security/#/docker-daemon-attack-surface`

- [**Git**] "Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency". See `https://git-scm.com/`

- [**Graphviz**] "Graphviz is open source graph visualization software". See `http://graphviz.org/`

- [**libXinerama**] "libXinerama - API for Xinerama extension to X11 Protocol". See `https://cgit.freedesktop.org/xorg/lib/libXinerama`

- [**libXpm**] "libXpm - X Pixmap (XPM) image file format library". See `https://cgit.freedesktop.org/xorg/lib/libXpm`

- [**Bower**] "A package manager for the web". See `https://bower.io/`

- [**curl**] "An URL retrieval utility and library". See `https://curl.haxx.se`

- [**UnZip**] "For extracting and viewing files in .zip archives". See `http://www.info-zip.org/UnZip.html`