Table of Contents

WRAPP

1. WHAT IS IT ?!

A generic swiss knife wanna be bash / perl centric application wrapper for quicky packaging and deploying your tools, create new tools out of your existing ones , generating code for additonal functions, search and replace in both file paths and contents ... and all the rest not mentioned actions in the sfw/bash/wrapp/tests/all-wrapp-tests.lst file ...

2. INSTALLATION AND CONFIGURATION

The wrapp instances and clones have been running on *Nix boxes with bash > 3.0 … , but some actions require 4.0 and older …
Cygwin has been tested also …

2.1. Prerequisites

The must have binaries are:
bash, perl, zip

The nice to have are:
tmux, vim ,ctags

The examples are for Ubuntu - use you OS package manager …

```
apt-get autoclean
apt-get install --only-upgrade bash

sudo apt-get install -y perl

# optionally
sudo apt-get install -y excuberant-ctags
sudo apt-get install -y 7z

apt-get upgrade
```

2.2. Fetch the source

Fetch the source from git hub as follows:

```
# create your product dir:
mkdir -p /opt/csitea/wrapp
cd /opt/csitea/wrapp/
```

```
# fetch the source
git clone git@github.com:YordanGeorgiev/wrapp.git
```

2.3. Build the first wrapp instance

When doing for first time do exactly as shown bellow, otherwise no joy ...
Each wrapp instance has it's own version, environmnt type and owner. For now just follow the
instruction - after half an hour you will be hacking this …

```
# build your product version dir - a kind of "this instance of the thingy dir"
mv -v /opt/csitea/wrapp/wrapp /opt/csitea/wrapp/wrapp.1.1.5.dev.ysg
```

2.4. Create you local conf file

The default conf file provides only limited functionality (this is by design) , thus copy and configure
the configuration file for your host

```
# go to the product version dir
cd /opt/csitea/wrapp/wrapp /opt/csitea/wrapp/wrapp.1.1.5.dev.ysg

mv -v sfw/bash/wrapp/wrapp.set-your-host.conf sfw/bash/wrapp/wrapp.`hostname -s`.conf
```

2.5. Start hacking

Start hacking … or wait check at least the test call running all the functions of the tool …

```
# opionally if you are in the vim camp open the "project relative files list file"
vim meta/.dev.wrapp

# Ctrl + Z ,
bash sfw/bash/wrapp/test-wrapp.sh

# now clone your own instance
bash sfw/bash/wrapp/wrapp.sh -a to-app=my-tool
```