LRIOBF

Quick user guide

A) Installing LRIOBF :

LRIOBF is available on La Rochelle Innovation website : http://www.lr-i.com/LRIOBF.html. Once you have downloaded and uncompressed LRIOBF.zip archive, you will find three command line binaries : one for Windows, one for Mac OS X and one for Linux. This guide assumes you will move the binary for your working platform in a directory listed in your ::env(PATH) such as C:\Windows\system\ or /usr/bin/ (depending on the platform). If you don't do that, you should add your full path to all `lriobf` commands examples above.

B) Installing Tclkit runtimes :

If you want to try your starkits or if you want to build starpacks, you will also need to download Tclkit runtimes from the web. This runtimes could be found on two websites :

- http://www.equi4.com/tclkit/download.html
- http://code.google.com/p/tclkit/downloads/list

This guide assumes you have downloaded a runtime for Windows renamed **tclkit-win32.exe**, a runtime for Mac OS X renamed **tclkit-darwin** and a runtime for Linux renamed **tclkit-linux**.

C) Getting our first example :

As a first example, we will use a sample starkit from Tcl developer Xchange starkits repository :

- http://www.tcl.tk/starkits/

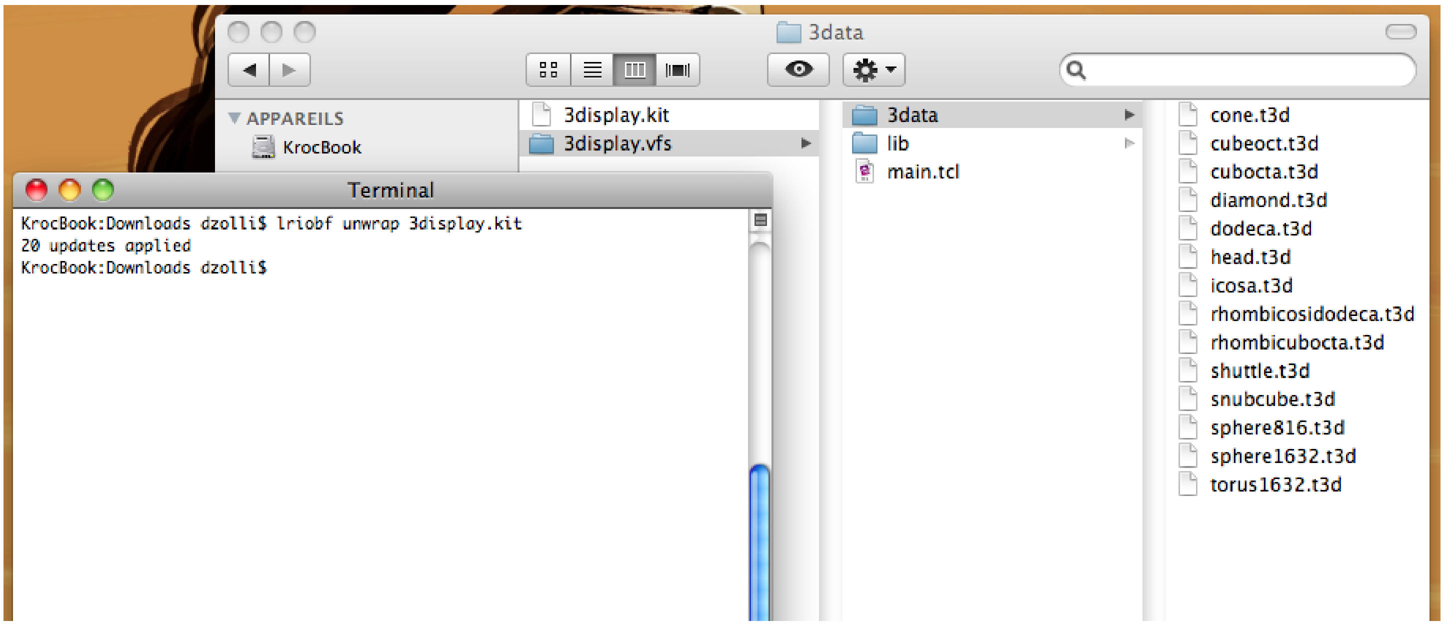
We will work with the first one on the list : 3display.

D) unwrap :

The first command we will see is `unwrap`. Open a command line terminal, move to the directory where you save 3display.kit and type this command:

```
lriobf unwrap 3display.kit
```

You should see something like that :



```
KrocBook:Downloads dzolli$ lriobf unwrap 3display.kit
20 updates applied
KrocBook:Downloads dzolli$
```

All data from starkit's virtual file system are extracted to 3display.vfs directory. The only required script is main.tcl at root wich is automatically sourced when a starkit is run. The two first lines of main.tcl should be :

```
package require starkit
starkit::startup
```

This performs a few standard startup tasks - it initialises the starkit::topdir variable to point to the top directory in the Starkit VFS (which can be used to relatively access files in the VFS). It also adds the Starkit lib directory to the Tcl auto_path variable, thus making available any packages stored in that directory.

If you want more details about starkit internals, you should read this presentation : Beyond Tclkit - starkits, starpacks and other stuff.

E) wrap and protect :

The wrapping mechanism turn a VFS directory to a starkit. With the `wrap` command, everything will be copied as is, while the `protect` command will crypt every Tcl scripts to something unreadable for humans like this:

```
if {[catch {::Lriobf::eval {}}]} {return}
```

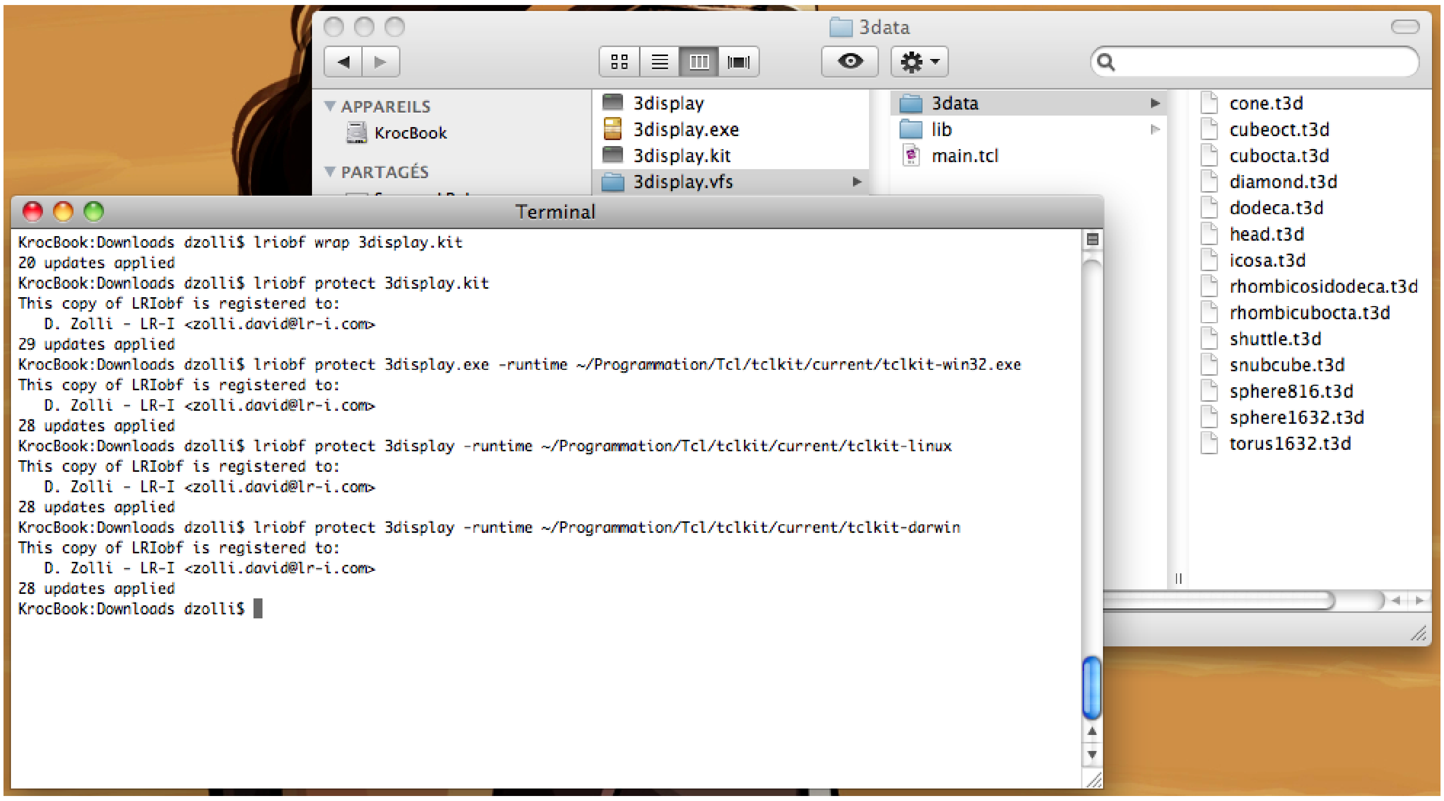
```
::Lriobf::eval {i4FGEzKAdsoll18e8PH2SsfgTDKD4bP40SUGcLFL7CAHzgx/afG2jm0N74+Vu6V6nd8e+XcXW5Sg9nlPsfp
x7EzI4VTX6b4eQK+Aee8VAASFK9DdF31caGTtXafMBpQUyEacIu3z5Sw9AYfBTY40aqy6Kh5Fx6Anhc0ADtmHmA/CkbRoDqw29z
wdYxVG02JrHhcw7WCEfudWCGqbsEszIcW57iSvTbSNBV4vWqE4h1Y8tbRnfzjZMU6WENogctDyila/mSCEZEBF32wpJKQRLYlp7
FJUjLmYDRyu4eCSm3tu+jKcOsYpjKP1QedpbYRFYES7iZFGfFfP8j8cDhhkA735lLAJa9VdxYK1Ph5sG21D95FCMbyt+sIRFuao
hc/XoeiQyMatL3nnKrxWX5miRT/WQQUtx9gIbwK65wiYoLn1GTf0a902rXBGSaaNBYD8ViKcXkYI7Iuhz9BxOkDNBEpJ4wkE9sv
y0y1/RsbeS1IWTGz13rUmpXfQhzp2hjkehNySKG0+AQ4Efe7zjFn+It6SOcm8/edDU9VD7+G6rNYtxO784YwoGmPV4u0e79oBv9
buv8+O/Uer/c6GRvBqQI35P+oEo6f+xv21wRrJWrTLgyOTwPzED761XBJXYHUTrdCXiI7yMgPvpGMfI1u7lbIvgvWm8pO4/IMkq
QacBoyhc4BiRda8z61rZ4BinAQxm04SP3XlvzKTElHLn3KYhiyn/W3twcVZUd6+Gj4BuDBR3dStV6VhtyWO7UME ...
```

Both wrap and `protect` commands `accept` a `-runtime` argument to output a starpack instead of a starkit. A starpack is a special version of a starkit that combines a starkit with a Tclkit runtime into a single file. Starpacks are standalone executables which run out of the box, making them even easier to distribute and use than starkits.

Here are some commands you can try:

```
lriobf wrap 3display.kit
lriobf protect 3display.exe
lriobf protect 3display.exe -runtime /pathto/tclkit-win32.exe
lriobf protect 3display -runtime /pathto/tclkit-darwin
lriobf protect 3display -runtime /pathto/tclkit-linux
```

As you can see, you can create starpacks for any platform supported by LRIOBF from your working platform.



```
KrocBook:Downloads dzolli$ lriobf wrap 3display.kit
20 updates applied
KrocBook:Downloads dzolli$ lriobf protect 3display.kit
This copy of LRIobf is registered to:
   D. Zolli - LR-I <zolli.david@lr-i.com>
29 updates applied
KrocBook:Downloads dzolli$ lriobf protect 3display.exe -runtime ~/Programmation/Tcl/tclkit/current/tclkit-win32.exe
This copy of LRIobf is registered to:
   D. Zolli - LR-I <zolli.david@lr-i.com>
28 updates applied
KrocBook:Downloads dzolli$ lriobf protect 3display -runtime ~/Programmation/Tcl/tclkit/current/tclkit-linux
This copy of LRIobf is registered to:
   D. Zolli - LR-I <zolli.david@lr-i.com>
28 updates applied
KrocBook:Downloads dzolli$ lriobf protect 3display -runtime ~/Programmation/Tcl/tclkit/current/tclkit-darwin
This copy of LRIobf is registered to:
   D. Zolli - LR-I <zolli.david@lr-i.com>
28 updates applied
KrocBook:Downloads dzolli$ 
```

F) help :

The `help` command will display help and available sub-commands on the standard output.

`lriobf help`

```
Specify one of the following commands:
      protect    Pack a file system directory area to a crypted starkit
      unwrap     Unpack a starkit into a new directory
      wrap       Pack a file system directory area to a starkit
For more information, type:      /usr/bin/lriobf help ?command?
```

`lriobf help protect`

```
Pack a file system directory area to a crypted starkit

             Usage: protect name ?options?
             -interp      name    Start something other than "tclkit" up
             -nocomp              Do not compress files added to starkit
             -runtime     file    Take Tclkit runtime prefix from file
             -verbose             Report actions taken
             -writable            Allow modifications (must be single writer)

             Expects a directory called "name.vfs", and creates a fresh
             starkit from it, called "name". If a Tclkit is specified as
             runtime prefix, then files will be merged with it.
```

`lriobf help wrap`

```
Pack a file system directory area to a starkit

             Usage: wrap name ?options?
             -interp      name    Start something other than "tclkit" up
             -nocomp              Do not compress files added to starkit
             -runtime     file    Take Tclkit runtime prefix from file
             -verbose             Report actions taken
             -vfs         dir     Use this directory as the vfs tree
             -writable            Allow modifications (must be single writer)

             Expects a directory called "name.vfs", and creates a fresh
             starkit from it, called "name". The -vfs option lets you
             use something other than "name.vfs". If a Tclkit is specified
             as runtime prefix, then files will be merged with it.
```

`lriobf help unwrap`

```
Unpack a starkit into a new directory

             Usage: unwrap name
             The name specified is the name of the starkit file.
             The results are placed in a directory "name.vfs", which must
             not yet exist.
```