



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/>

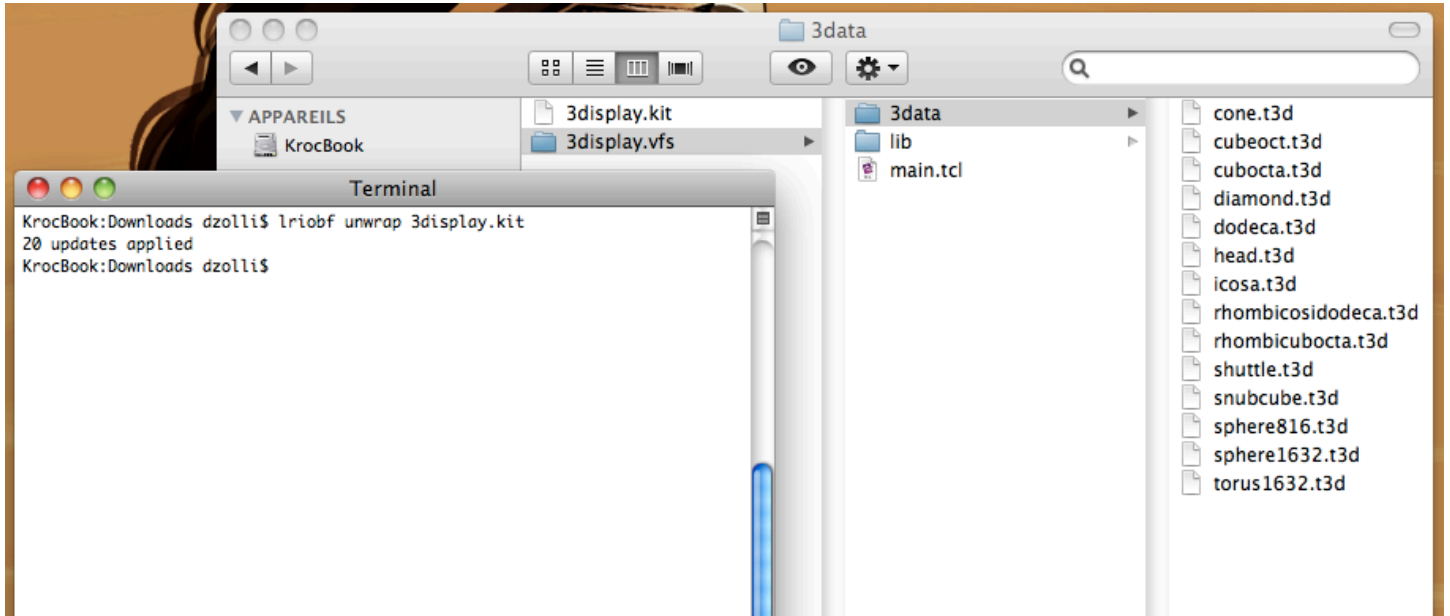
We will work with the first one on the list : [3display](#).

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 :



All data from starkit's **virtual file system** are extracted to `3display.vfs` directory. The only required script is `main.tcl` at root which 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 {rZ4BinAi4FGEzKAdsoll18e8PH2SsfgTDKrD4bP40SurZ4BinAGcLFL7CAHgzx/afG2jm0N74+Vu6V6nd8e
+XcXW5Sg9nlpSfpI4VTX6b4eQK+Aee8VAASFk9DdF31caGTtXafMBpQUyEacIu3z5Sw9AYfBTY40aqy6Kh5Fx6Anhc0ADtmHmA/
CkbRoDqw29zwdYxVG02JrHhcw7WCEfudWCGqbsEszIcW57iSvTbSNBV4vWqE4h1Y8tbRnfzjZMU6WENogctDyila/
mSCEZEBF32wpLmYDRyu4eCSm3tu+jKc0sYpjKP1QedpbYRFYEs7iZFGFFP8j8cDhhkA7351LAJa9VdxYK1Ph5sG21D95FCMbyt
+sIRFuaohc/XoeiQyi4FGEzKAdsoll18e8PH2SsfgTDKD4bP40SUGcLFL7CAHgzxMatL3nrxWX5miRT/
WQQUt9gIbwK65wiYoLn1GTf0a902rXBGsaaNBVD8ViKcXkYI7Iuhz9Bx0kDNBEPJ4wkE9svy0y1/
RsbeS1IWTGz13rUmpXfQ1aZe8qqsQdrZ4BinAhzpGj4BuDBR3dStV6VhtyW07UME2hjkehNySKG0+AQ4Efe7zjFn+It6S0cm8/
edDU9VD7+G6rNYtx0784YwoGmPV4Gj4BuDBR3dStV6VhtyW07UMEu0e79oBv9buv8+0/Uer/c6GRvBqQI35P+oEo6f
+xv21wRrJWrTLgy0TWPzED761XBudWCGqbsEszIcW5JXYHUTrdCXiI7yMgPvpGMfI1u7lbIvgvWm8p04/
IMkqQacBoyhC4BiRda8z61rZ4BinAQxm04SP3XlvzKTELHLn3KYhiyn/W3twcVZUd6+Gj4BuDBR3dStV6VhtyW07UME ...
```

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
```

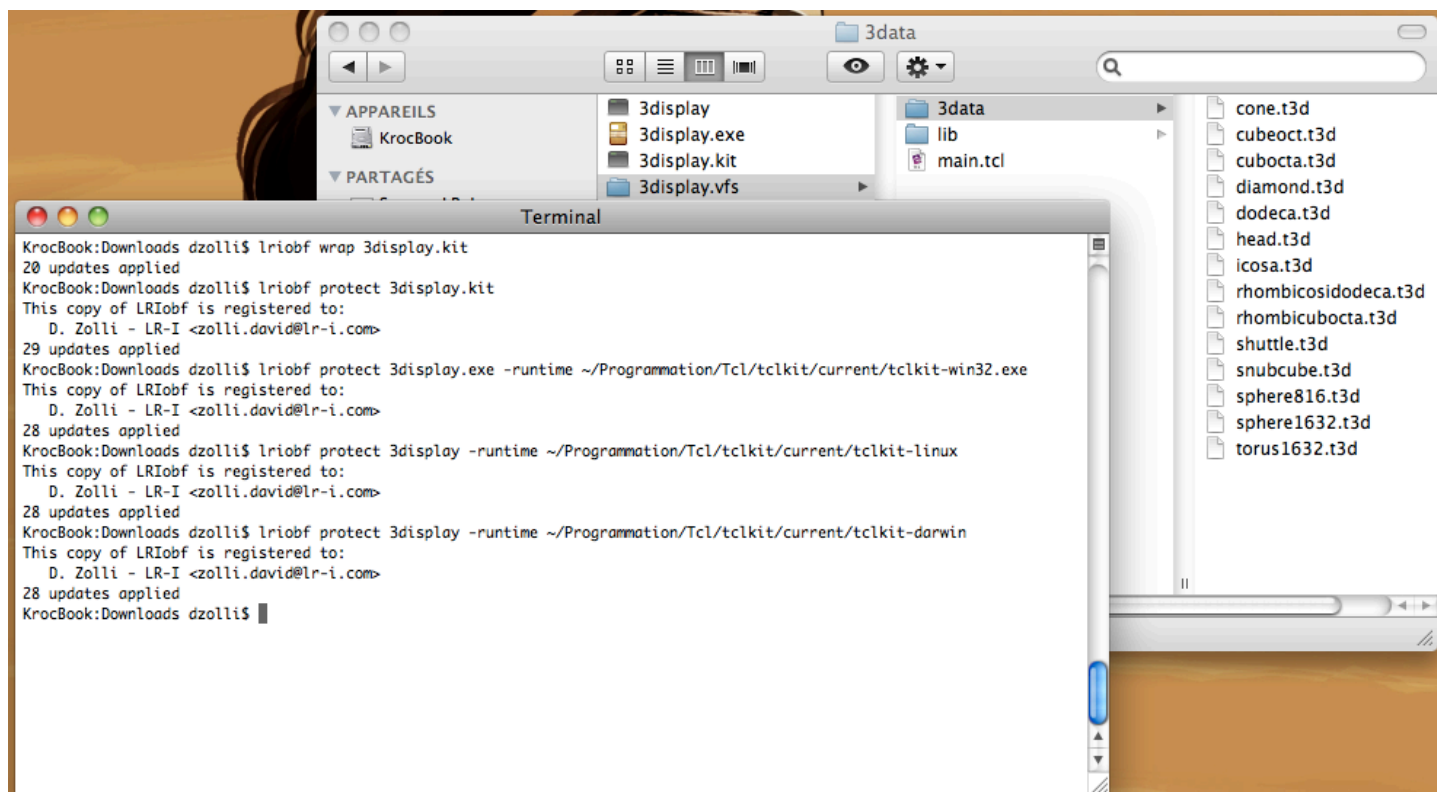
```
lriobf protect 3display.exe
```

```
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.



F) help :

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

lriobf help

Specify one of the following commands:

<code>protect</code>	Pack a file system directory area to a crypted starkit
<code>unwrap</code>	Unpack a starkit into a new directory
<code>wrap</code>	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?`

<code>-interp</code>	<code>name</code>	Start something other than "tclkit" up
<code>-nocomp</code>		Do not compress files added to starkit
<code>-runtime</code>	<code>file</code>	Take Tclkit runtime prefix from file
<code>-verbose</code>		Report actions taken
<code>-writable</code>		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?`

<code>-interp</code>	<code>name</code>	Start something other than "tclkit" up
<code>-nocomp</code>		Do not compress files added to starkit
<code>-runtime</code>	<code>file</code>	Take Tclkit runtime prefix from file
<code>-verbose</code>		Report actions taken
<code>-vfs</code>	<code>dir</code>	Use this directory as the vfs tree
<code>-writable</code>		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.