A better railing

Maxim Kharchenko, Cloudozer LLP

11/11/2014

1 Overview

The lifecycle of a LING image include three stages: build, boot, and run. The output of the build stage is a Xen-bootable image. The boot process culminates in a running LING kernel. The user application starts at the last – run – stage.

All three stages take a multitude of configurable parameters. The purpose of the rewrite of the *railing* utility is to minimise the number of configuration files and simplify the entire lifecycle.

2 Build options

The railing utility needs to following information to produce an image:

- The name of the output image;
- A list of Erlang libraries to import from OTP;
- A list of the project BEAM files to convert and import;
- A list of auxilliary files to import;
- Whether to use a debug version of LING.

All the above are set using command line parameters that follow the `image` selector:

```
railing image [options]
```

The railing utility derives the name of the project from the name of the directory the project resides in. By default, the name of the image is `<project-name>.img`.

To obtain a list of BEAM files to import, railing scans the project directory for any subdirectory those path ends with `/ebin`. All `*.beam` and `*.app` files in such subdirectories are imported and embedded into the image. There is a command-line option to exclude certain subdirectory from the import list.

A list of railing options for the `image` selector:

Option	Meaning
`-n name` or `--name name`	set image name (default: projname.img)
`-l lib` or `--library lib`	import lib from Erlang OTP
`-x path` or `--exclude path`	do not import directories that start with path
`-k type` or `--kernel type`	Use debug or normal (default) LING kernel