

[Personal](#)[Open source](#)[Business](#)[Explore](#)[Pricing](#)[Blog](#)[Support](#)[This repository](#)[Sign in](#)[Sign up](#)[littlekernel](#) / [lk](#)[Watch](#)

144

[★ Star](#)

690

[Fork](#)

172

[Code](#)[Issues](#) 13[Pull requests](#) 9[Projects](#) 0[Wiki](#)[Pulse](#)[Graphs](#)

# Building Outside The LK Tree

Brian Swetland edited this page on 20 Jun 2015 · 4 revisions

The build system provides a mechanism to overlay additional projects, targets, platforms, apps, libs, etc, from sibling directory hierarchies. This allows you to more easily build and maintain projects that are not part of the main lk tree.

For example, say you have LK checked out in an `lk` directory, and next to that you have a `mystuff` directory (a checkout of your own source tree). It can contain your own projects, libraries, and apps, like so:

```
mystuff/project/toaster-oven.mk
mystuff/target/toaster-oven/{rules.mk, init.c, ...}
mystuff/app/toaster-ui/{rules.mk, main.c, ...}
mystuff/lib/cool-lcd/{rules.mk, lcddriver.c, ...}
mystuff/lib/pid-controller/...
```

If you copy `lk/makefile` to `mystuff/makefile` and create `lk_inc.mk` like so:

```
LOCAL_DIR := mystuff
```

## ▼ Pages 7

[Home](#)[Building Outside The LK Tree](#)[Introduction](#)[Kernel APIs And Primitives](#)[LK Code Style](#)[The Build System](#)[Things You Should Know](#)

## Clone this wiki locally

<https://github.com/littlekernel/lk/wiki/Building-Outside-The-LK-Tree>

```
LKMAKEROOT := ..
LKROOT := lk
LKINC := $(LOCAL_DIR)
DEFAULT_PROJECT ?= toaster-oven
BUILDRROOT ?= $(LOCAL_DIR)
```

You can then `cd` to `mystuff` and run `make` to build your project. You don't need to modify anything in the main `lk` tree, so it's easy to keep that up to date — no need to fork it, merge or rebase your changes, etc.

## Details

`LKMAKEROOT` is the relative path from the overlay tree you're building in to the directory that's above the `lk` directory and any overlay tree directories (you can have multiple of these if you like). This variable is the **only** one that can have `..` path elements in it. Such elements used anywhere else will cause the build system to misbehave.

`LKROOT` is the path relative to `LKMAKEROOT` to `LK` itself

`LKINC` is the set of paths relative to `LKMAKEROOT` to overlay directories

`DEFAULT_PROJECT` does what it says — sets the project to build if another is not specified when `make` is invoked.

`BUILDRROOT` is relative to `LKMAKEROOT` and is the directory where `build-$(PROJECT)` will live, containing the build results.

You can also specify your `TOOLCHAIN_PREFIX` here to point at your chosen cross-compiler for this overlay.

## Alternate Approach

If you'd rather run `make` from the top directory (that contains `lk` and `mystuff`), you can copy

lk/makefile and set up lk\_inc.mk there instead. Just change the value of LKMAKER00T and BUILDROOT both to . and you're all set.

