

markd's make rules

Mark Diekhans
markd@cbse.ucsc.edu

Sep 20, 2005

Features

- centralized variable and rule definitions
 - minimal makefiles
 - easy to make changes to the build process
 - local override of defaults
 - limited automatic configuration (not environment dependencies)
- one tree can
 - build for multiple architectures
 - build with different compiler options (-O3, -g, -pg)without a `make clean`
- header and library dependencies
- stable build
 - running `make` a second time will not rebuild files
- errors are propagated
 - `make -k` works!

Source Tree

Makefile

build/defs.mk

 /rules.mk

 /defRules.mk

src/Makefile

defs.local.mk

Source Tree

Makefile

build/defs.mk

 /rules.mk

 /defRules.mk

src/Makefile

defs.local.mk

obj/i686/opt/...

 /dbg/...

bin/i686/opt/...

 /dbg/...

lib/i686/opt/...

 /dbg/...

Example makefiles

- build a single program

```
ROOT = ../../..
```

```
PROGS = fred
```

```
include ${ROOT}/defRules.mk
```

Example makefiles

- build a single program

```
ROOT = ../../..  
PROGS = fred  
include ${ROOT}/defRules.mk
```

- build a single program with multiple object files

```
ROOT = ../../..  
OBJS = foo.o baz.o  
PROGS = fred2  
include ${ROOT}/defRules.mk
```