# 32. DPDK 根目录 Makefile 理解

DPDK提供了一个根目录级别的Makefile，包含配置，构建，清理，测试，安装等目的。 这些操作将在下面的部分中进行解释。

## 32.1. 配置 Targets

配置 target 需要使用 T=mytarget 指定target的名称，这个操作不能省略。 可用的target列表位于 $(RTE\_SDK)/config 中(移除defconfig \_ 前缀)。

配置target还支持使用 O=mybuilddir 来指定输出目录的名称。 这是一个可选配置，默认的输出目录是build。

Config

这将创建一个构建目录，并从模板中生成一个配置。 同时会在构建目录下生成一个 Makefile 文件。

例如：

make config O=mybuild T=x86\_64-native-linuxapp-gcc

## 32.2. 构建 Targets

构建 targets 支持输出目录名称可选规则，使用 O=mybuilddir。 默认的输出目录是build。

all, build or just make

在前面由make config创建的目录上构建DPDK。

例如：

make O=mybuild

clean

清除所有由 make build 生成的目标文件。

例如：

make clean O=mybuild

%\_sub

只构建某个目录，而不管对其他目录的依赖性。

例如：

make lib/librte\_eal\_sub O=mybuild

%\_clean

清除对子目录的构建操作结果。

例如：

make lib/librte\_eal\_clean O=mybuild

## 32.3. 安装 Targets

* Install  
  可用的 targets 列表位于 $(RTE\_SDK)/config (移除 defconfig\_ 前缀).  
  可以使用 GNU 标准的变量：[http://gnu.org/prep/standards/html\_node/Directory-Variables.html](https://link.jianshu.com?t=http://gnu.org/prep/standards/html_node/Directory-Variables.html" \t "_blank) 和[http://gnu.org/prep/standards/html\_node/DESTDIR.html](https://link.jianshu.com?t=http://gnu.org/prep/standards/html_node/DESTDIR.html" \t "_blank)  
  例如：

make install DESTDIR=myinstall prefix=/usr

## 32.4. 测试 Targets

test

对使用 O=mybuilddir 指定的构建目录启动自动测试。 这是可选的，默认的输出目录是build。

例如：

make test O=mybuild

## 32.5. 文档 Targets

doc

生成文档（API和指南）。

doc-api-html

在html中生成Doxygen API文档。

doc-guides-html

在html中生成指南文档。

doc-guides-pdf

用pdf生成指南文档。

## 32.6. 其他 Targets

help

显示快速帮助。

## 32.7. 其他有用的命令行变量

以下变量可以在命令行中指定：

V=

启用详细构建（显示完整的编译命令行和一些中间命令）。

D=

启用依赖关系调试。 这提供了一些关于为什么构建目标的有用信息。

EXTRA\_CFLAGS=, EXTRA\_LDFLAGS=, EXTRA\_LDLIBS=, EXTRA\_ASFLAGS=, EXTRA\_CPPFLAGS=

附加特定的编译，链接或汇编标志。

CROSS=

指定一个交叉工具链头部，该头部将作为所有gcc/binutils应用程序的前缀。这只适用于使用gcc。

## 32.8. 在需要构建的目录中执行Make

上面描述的所有目标都是从SDK根目录 $(RTE\_SDK) 调用的。 也可以在build目录中运行相同的Makefile target。 例如，下面的命令：

cd $(RTE\_SDK)

make config O=mybuild T=x86\_64-native-linuxapp-gcc

make O=mybuild

相当于：

cd $(RTE\_SDK)

make config O=mybuild T=x86\_64-native-linuxapp-gcccd mybuild

# no need to specify O= now

make

## 32.9. 编译为调试 Target

要编译包含调试信息和优化级别设置为0的DPDK和示例应用程序，应在编译之前设置EXTRA\_CFLAGS环境变量，如下所示：

export EXTRA\_CFLAGS='-O0 -g'