统一编译规则：

每个库的makefile在对应目录的Trunk目录下，一般Trunk目录包含如下结构。

Include：放库所需的头文件。

Lib：生成的所有平台库文件。

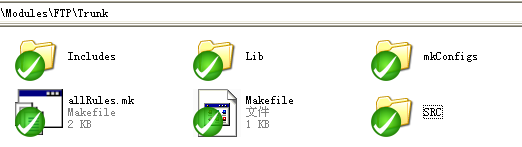
Src：库的代码。

mkConfigs：不同平台编译文件。现在mkconfigs的映射地址，

<http://10.6.5.2/svnpl/Renascence/mkConfigs/Trunk>

Makefile：make的主要文件。

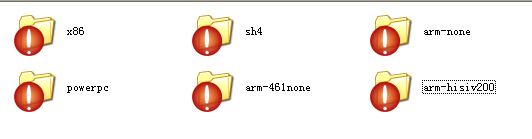
allRules.mk：大部分有，有些没有，库自己的通用配置，供makefile调用。



Lib下目录结构：

Lib目录下面文件夹以平台名命名，类似下图，组件输出库文件放入对应平台名下的文件夹中。

例如：arm-v5t生成库放入/Lib/arm-v5t 目录中。



Makefile通用命令限制：

1. 调用make clean；能够删除所有中间文件和目标文件。
2. 调用make；自动编译出所有mkconfigs中平台库，并放入Trunk/Lib目录对应平台名称下。

例如：v5t平台库放入：Trunk/Lib/arm-v5t目录中。

1. 生成单个平台的库使用方法：

make platform=平台名

或者：

make PLATFORM=平台名。

例如v5t平台：make platform=arm-v5t

需要支持大写和小写两个参数！

1. Mac平台由于不能和linux通用，所以配置文件单独在mkConfigs下建目录，这样既不影响原有配置编译，也可以统一到现在编译方式中去。Mac平台配置在mkConfigs/MacConfigs目录中。 编译方式和其他平台完全一样，只要名字前是mac，并且配置存在。64位服务器编译的配置文件保存在mkConfigs/m64Configs文件夹中，使用方式也一致！

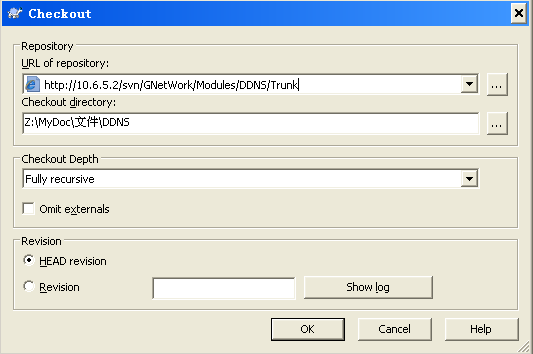
现在常用平台名和编译器对应关系：

|  |  |  |
| --- | --- | --- |
| 平台名： | 编译器 | 库路径 |
| sh4 | sh4-linux-uclibc- | ./Lib/sh4-linux |
| arm926 | arm-linux-uclibc- | ./Lib/arm926ejs-linux-uclibc |
| arm-none | arm-none-linux-gnueabi- | ./Lib/arm-none-linux-gnueabi |
| arm-v5t | arm\_v5t\_le- | ./Lib/arm-v5t-linux |
| arm-461none | arm-none-linux-gnueabi-4.6.1- | ./Lib/arm-461none-linux |
| arm-hisiv200 | arm-hisiv200-linux- | ./Lib/arm-hisiv200-linux |
| arm-hismall | arm-hismall-linux- | ./Lib/arm-hismall-linux |
| powerpc | powerpc-e300c3-linux-gnu- | ./Lib/powerpc-e300c3-linux |
| x86 |  | ./Lib/x86 |
| mips | mips-linux-gnu- | ./Lib/mips-linux |
| armv7 | armv7-linux- | ./Lib/armv7-linux- |
| arm482v7 | armv7-linux-4.8.2- | ./Lib/arm482v7-linux |
| arm-hisiv100nptl | arm-hisiv100nptl-linux- | ./Lib/arm-hisiv100nptl-linux |
| arm-hisiv100-linux | arm-hisiv100-linux- | ./Lib/arm-hisiv100-linux |
| mac-i386 | gcc(mac平台) | ./Lib/mac-i386 |
| mac-armv7 | llvm-gcc-4.2（平台） | ./Lib/mac-armv7 |
| m64x86 | (64位下的gcc) | ./Lib/m64x86 |
| m64-gnueabihf | arm-linux-gnueabihf-(32位) | ./Lib/m64-gnueabihf-linux |
| m32x86 | i386-centos5-linux-gnu-（32位下的gcc） | ./Lib/m32x86 |
|  |  |  |

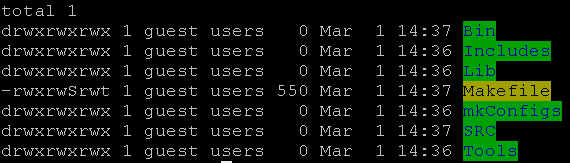
一、实例

以libddns.a libNetCon.a为例;

将需要编译的库的代码从SVN下check out，一般放到库名字对应的目录下DDNS目录下，并将该目录smbmnt到服务器（见相关文档）



2.登入服务器，cd 到DDNS目录下



3、运行make clean; make （这会生成4个平台的库）

如果要制定平台可执行 make clean; make platform=（arm-v5t，举例）

可支持自动化编译工具

4、生成的库在Trunk/Lib 目录下