一个Android.mk file用来向编译系统描述你的源代码。具体来说：该文件是GNU Makefile的一小部分，会被编译系统解析一次或多次。你可以在每一个Android.mk file中定义一个或多个模块，你也可以在几个模块中使用同一个源代码文件。编译系统为你处理许多细节问题。

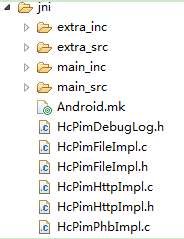
一个工程中的源文件不计数，其按类型、功能、模块分别放在若干个目录中，makefile定义了一系列的规则来指定，哪些文件需要先编译，哪些文件需要后编译，哪些文件需要重新编译，甚至于进行更复杂的功能操作，因为 makefile就像一个Shell脚本一样，其中也可以执行操作系统的命令。

makefile带来的好处就是——“自动化编译”，一旦写好，只需要一个make命令，整个工程完全自动编译，极大的提高了软件开发的效率。make是一个命令工具，是一个解释makefile中指令的命令工具，一般来说，大多数的IDE都有这个命令，比如：Delphi的make，Visual C++的nmake，Linux下GNU的make。可见，makefile都成为了一种在工程方面的编译方法。

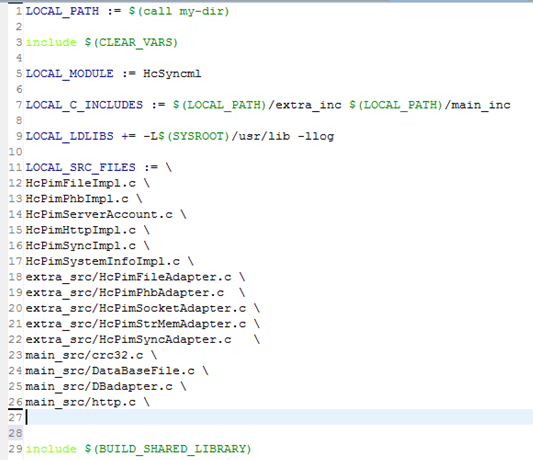
Make工具最主要也是最基本的功能就是通过makefile文件来描述源程序之间的相互关系并自动维护编译工作。而makefile 文件需要按照某种语法进行编写，文件中需要说明如何编译各个源文件并连接生成可执行文件，并要求定义源文件之间的依赖关系。makefile 文件是许多编译器--包括 Windows NT 下的编译器--维护编译信息的常用方法，只是在集成开发环境中，用户通过友好的界面修改 makefile 文件而已。

下面直接看一个实例：

Jni 文件树 如下



Android.mk文件 如下：



、LOCAL\_PATH := $(call my-dir)

一个Android.mk file首先必须定义好LOCAL\_PATH变量。它用于在开发树中查找源文件。在这个例子中，宏函数’my-dir’, 由编译系统提供，用于返回当前路径（即包含Android.mk file文件的目录）。

2、include $( CLEAR\_VARS)

CLEAR\_VARS 由编译系统提供，指定让GNU MAKEFILE为你清除许多LOCAL\_XXX变量（例如 LOCAL\_MODULE, LOCAL\_SRC\_FILES, LOCAL\_STATIC\_LIBRARIES, 等等...),除LOCAL\_PATH 。这是必要的，因为所有的编译控制文件都在同一个GNU MAKE执行环境中，所有的变量都是全局的。

3、LOCAL\_MODULE := HcSyncml

LOCAL\_MODULE变量必须定义，以标识你在Android.mk文件中描述的每个模块。名称必须是唯一的，而且不包 含任何空格。注意编译系统会自动产生合适的前缀和后缀，换句话说，一个被命名为'HcSyncml'的共享库模块，将会生成'libHcSyncml.so'文件。

4、LOCAL\_C\_INCLUDES := $(LOCAL\_PATH)/extra\_inc$(LOCAL\_PATH)/main\_inc

LOCAL\_C\_INCLUDES 中加入所需要包含的头文件路径

5、LOCAL\_SRC\_FILES

LOCAL\_SRC\_FILES中加入源文件路径(需要编译的文件),多个文件用 ‘\’ 隔开

6、LOCAL\_LDLIBS += -L$(SYSROOT)/usr/lib –llog

表示允许打印Log