## 什么是ATFWD？

简单的说，那就是一个进程，进行数据中转的进程：数据从modem下传上来先通过venderril，再从venderril下传到framework（或者别的什么地方）下进行处理，ATFWD便是venderril下的一个中转站。其中与modem的通讯方式采用QMI，与framework采用bingerserver方式通讯

## At Command大致处理流程



## 关键类和方法

**/vendor/qcom/proprietary/commonsys/telephony-apps/ATFwd/src/com/qualcomm/atfwd/**

1. AtFwdAutoboot.java:

接收开机广播，先启动服务AtFwdService，然后启动服务AtCmdFwdService

1. AtCmdFwd.java：

initAtCmdHandlers方法：将各类at命令的处理类（例如AtCkpdCmdHandler）添加进HashMap变量**mCmdHandlers**中，进行统一处理.  
processAtCmd实际的处理类，当被调用时调用对应的xxxAtCmdHandler的handleCommand处理，有返回消息.

1. 接口AtCmdHandler，主要有两个方法**getCommandName**（作为mCmdHandlers的键值）、**handleCommand**（实际的处理类）
2. 关于AT命令的处理类**xxxAtCmdHandler**，实现了接口AtCmdHandler，不同处理类处理不同at命令
3. At命令基本类AtCmd，存储一些At命令关键信息，一条At命令请求生成一个AtCmd对象。处理At命令失败时会调用**getAtCmdErrStr**，不同的xxxAtCmdHandler会返回不同的error

**vendor/qcom/proprietary/telephony-apps/ATFWD-daemon**

1、Android.mk，将ATFWD-daemon编译到（vendor/bin）下面做实时监听从modem下传来的AT命令

2、sendcmd.cpp  
    **sendit** 获取服务AtCmdFwdService  
    **sendAtCmd** 调用AtCmdFwdService的processAtCmd，猜想handleResp处理返回值  
    initWithMapSize 初始化binder服务，以此实现AT命令传到实际处理的地方，在atfwd\_daemon的mian函数调用

3、atfwd\_daemon.c

ATFWD-daemon进程的入口是main()函数，最后函数会循环在while(1)中，当modem下有传来需要处理的命令的时候，newRequest会置为1

**atCmdFwdReqType**变量存储所有At命令

tryInit();实现两块块的初始化，首先是QMI的初始化，其次是进行QMI连接的初始化，这两块主要是实现能接受到从modem下传上来的At命令。

## 如何扩展添加一个AP Site的AT命令

扩展AP侧的AT命令task4359782  
  
**1、在atfwd\_daemon.c添加AT Command的注册**  
    { //AT command fwd type  
        1, // Number of commands  
        {  
            { QMI\_ATCOP\_AT\_CMD\_NOT\_ABORTABLE, "+CDVOL"},  
        }  
    },

**2、需要实现一个完整的AtCdvolCmdHandler.java**  
public class AtCdvolCmdHandler extends AtCmdBaseHandler implements   
AtCmdHandler

**3、在AtCmdFwdService.java将处理类添加进map**  
            try {  
                cmd = new AtCdvolCmdHandler(c);  
mCmdHandlers.put(cmd.getCommandName().toUpperCase(), cmd);  
            } catch (AtCmdHandlerInstantiationException e) {  
                Log.e(LOG\_TAG, "Unable to instantiate command   
AtCdvolCmd", e);  
            }  
  
**4、在Modem侧添加自定义的AT Command的注册**  
/amss\_2017\_la3.0/MPSS.AT.3.1/modem\_proc/datamodem/interface/atcop/src/dsatclient\_ex.c  
  LOCAL byte   
allowed\_list[][MAX\_CMD\_SIZE]={"+RIMRATPREF","+RIMDATASERVICE","+CLVL","+CKPD","+CMUT",  
"+CTSA","+CBKLT","+CFUN","+CDIS","+CRSL","+CMAR",  
"+CSO","+CSS","+CBC","$QCPWRDN","+RIMTETHER","+RIMWLANAPCFG",  
"+CGMM","+CGMR","+CRST","+CSHM","+CTACT","+CCALD","+CCLGS",  
"+CDUR","+CWAP","+CDCONT","+CMMS","+CBCAST","+CBLTH",  
                                           "+RIMCMGS","+CTBCPS",  
                                           /\*[FEATURE]-Add-BEGIN by   
TCTNB.caixia.chen,03/01/2017,Task 4289970 \*/  
"+CEMAIL","+CBCAST","+CIMACT","+CSYNC","+CALRM","+CRMIND","+CTMRV",  
"+CVRCD","+CSMCT","+CLOCD","+CWLNT","+CNPAD","+CSCNT","+CTASK","+CMSG",  
                                           /\*[FEATURE]-Add-END by   
TCTNB.caixia.chen \*/  
                                           "+CDVOL","+CDDUR",""};

**5、修改数组大小**  
avengers/codeaurora\_msm8953/MSM8953/amss\_8953\_la2.0 /   
MPSS.TA.2.3/modem\_proc/datamodem/interface/atcop/inc/dsatclient.h  
#define MAX\_CMD\_LIST             (50)  /\* total no of commands that can be