

简述

- 结束WEA流程;跟进PLMN流程结束;跟进SPN流程中。

PLMN和SPN

- PLMN(Public Land Mobile Network):当前SIM所驻留的网络
- SPN(Service Provider Name) : 当前发行SIM卡的运营商的名称

PLMN流程

1. TeleService 系统应用加载Telephony业务模型的过程中，同步完成 ServiceStateTracker 对象的创建，同时注册网络变化相关的Registration。

- ServiceStateTracker 位于
frameworks/opt/telephony/src/java/com/android/internal/telephony/ServiceStateTracker.java

```
■ //ServiceStateTracker类体
protected final TransportManager mTransportManager;
protected final SparseArray<NetworkRegistrationManager>
mRegStateManagers = new SparseArray<>();
mTransportManager = mPhone.getTransportManager();
for (int transportType :
mTransportManager.getAvailableTransports()) {
    mRegStateManagers.append(transportType, new
NetworkRegistrationManager(
    transportType, phone));

    mRegStateManagers.get(transportType).registerForNetworkRegistration
InfoChanged(
        this, EVENT_NETWORK_STATE_CHANGED, null);
}
```

2. 当网络发生变化时ServiceStateTracker响应EVENT_NETWORK_STATE_CHANGED消息

```
◦ public void handleMessage(Message msg) {
    AsyncResult ar;
    int[] ints;
    Message message;

    if (VDBG) log("received event " + msg.what);
    switch (msg.what) {
        case EVENT_NETWORK_STATE_CHANGED:
            pollStateInternal(true);
            break;
    }
}

protected void pollStateInternal(boolean modemTriggered) {
    // MTK-END
```

```

mPollingContext = new int[1];
mPollingContext[0] = 0;

log("pollState: modemTriggered=" + modemTriggered);

switch (mCi.getRadioState()) {
    case TelephonyManager.RADIO_POWER_UNAVAILABLE:
        mNewSS.setStateOutOfService();
        setSignalStrengthDefaultValues();
        mLastNitzData = null;
        mNitzState.handleNetworkUnavailable();
        pollStateDone();
        break;

    case TelephonyManager.RADIO_POWER_OFF:
        mNewSS.setStateOff();
        setSignalStrengthDefaultValues();
        mLastNitzData = null;
        mNitzState.handleNetworkUnavailable();
        // don't poll when device is shutting down or the poll was
not modemTriggered
        // (they sent us new radio data) and current network is not
IWLAN
        if (mDeviceShuttingDown ||
            (!modemTriggered &&
ServiceState.RIL_RADIO_TECHNOLOGY_IWLAN
            != mSS.getRilDataRadioTechnology())) {
            pollStateDone();
            break;
        }

    default:
        // Issue all poll-related commands at once then count down
the responses, which
        // are allowed to arrive out-of-order
        mPollingContext[0]++;
        //获取PLMN信息
        mCi.getOperator(obtainMessage(EVENT_POLL_STATE_OPERATOR,
mPollingContext));
        break;
}
}

```

◦ modem侧获取PLMN优先级

1. Eons(Enhanced Operator Name String), 也就是从SIM的EF_OPL和EF_PNN分区来读取Plmn Name
 - EF:Elementary File
2. CPHS ONS(Common PCN Handset Specification Operator Name String), 该字符串也是保存在SIM文件系统中
3. NITZ Operator Name
4. 配置文件读取
5. MCCMNC数字作为Plmn Name

3. 接收到RIL的返回数据时，ServiceStateTracker响应EVENT_POLL_STATE_OPERATOR消息

```
o public void handleMessage(Message msg) {
    AsyncResult ar;
    int[] ints;
    Message message;

    if (VDBG) log("received event " + msg.what);
    switch (msg.what) {
        case EVENT_POLL_STATE_OPERATOR:
            ar = (AsyncResult) msg.obj;
            handlePollStateResult(msg.what, ar);
            break;
    }
}

protected void handlePollStateResult(int what, AsyncResult ar) {
    // Ignore stale requests from last poll
    if (ar.userObj != mPollingContext) return;

    if (ar.exception != null) {
        ...
    } else try {
        handlePollStateResultMessage(what, ar);
    }
}

protected void handlePollStateResultMessage(int what, AsyncResult ar) {
    int ints[];
    switch (what) {
        case EVENT_POLL_STATE_OPERATOR: {
            if (mPhone.isPhoneTypeGsm()) {
                String opNames[] = (String[]) ar.result;

                if (opNames != null && opNames.length >= 3) {
                    //protected ServiceState mNewSS;
                    mNewSS.setOperatorAlphaLongRaw(opNames[0]);
                    mNewSS.setOperatorAlphaShortRaw(opNames[1]);
                    // FIXME: Giving brandOverride higher precedence, is
                    this desired?

                    String brandOverride = getOperatorBrandOverride();
                    mCdnr.updateEfForBrandOverride(brandOverride);
                    if (brandOverride != null) {
                        log("EVENT_POLL_STATE_OPERATOR: use
brandOverride=" + brandOverride);
                        mNewSS.setOperatorName(brandOverride,
brandOverride, opNames[2]);
                    } else {
                        mNewSS.setOperatorName(opNames[0], opNames[1],
opNames[2]);
                    }
                }
            }
            break;
        }
    }
}
```

```
protected String getOperatorBrandOverride() {
    // MTK-END
    UiccCard card = mPhone.getUiccCard();
    if (card == null) return null;
    UiccProfile profile = card.getUiccProfile();
    if (profile == null) return null;
    return profile.getOperatorBrandOverride();
}
```

4. ServiceState的setOperatorName()

- ```
public void setOperatorName(String longName, String shortName, String
numeric) {
 mOperatorAlphaLong = longName;
 mOperatorAlphaShort = shortName;
 mOperatorNumeric = numeric;
}
```

#### 5. ServiceState提供了三个对应的getXXX()方法获取PLMN

- ```
//Get current registered operator name in long alphanumeric format.
//In GSM/UMTS, long format can be up to 16 characters long.
//In CDMA, returns the ERI text, if set. Otherwise, returns the ONS.
public String getOperatorAlphaLong() {
    return mOperatorAlphaLong;
}

//Get current registered operator numeric id.
//In GSM/UMTS, numeric format is 3 digit country code plus 2 or 3 digit
//network code.
public String getOperatorNumeric() {
    return mOperatorNumeric;
}
```

SPN流程

从SIM卡读取

从配置文件读取