



Unisoc Confidential For hiar

中国电信移动终端自注册客制化指导手册

文档版本
发布日期

V1.2
2020-08-14

版权所有 © 紫光展锐科技有限公司。保留一切权利。

本文件所含数据和信息都属于紫光展锐所有的机密信息，紫光展锐保留所有相关权利。本文件仅为信息参考之目的提供，不包含任何明示或默示的知识产权许可，也不表示有任何明示或默示的保证，包括但不限于满足任何特殊目的、不侵权或性能。当您接受这份文件时，即表示您同意本文件中内容和信息属于紫光展锐机密信息，且同意在未获得紫光展锐书面同意前，不使用或复制本文件的整体或部分，也不向任何其他方披露本文件内容。紫光展锐有权在未经事先通知的情况下，在任何时候对本文件做任何修改。紫光展锐对本文件所含数据和信息不做任何保证，在任何情况下，紫光展锐均不负任何与本文件相关的直接或间接的、任何伤害或损失。

请参照交付物中说明文档对紫光展锐交付物进行使用，任何人对紫光展锐交付物的修改、定制化或违反说明文档的指引对紫光展锐交付物进行使用造成的任何损失由其自行承担。紫光展锐交付物中的性能指标、测试结果和参数等，均为在紫光展锐内部研发和测试系统中获得的，仅供参考，若任何人需要对交付物进行商用或量产，需要结合自身的软硬件测试环境进行全面的测试和调试。非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

Unisoc Confidential For hiar

紫光展锐科技有限公司



前 言

概述

本文介绍紫光展锐 Android 手机平台对电信自注册的支持情况，以及如何对紫光展锐平台电信移动终端自注册功能进行客制化，为客户电信入库送测做参考指导。详细介绍了电信移动终端数据自注册、CDMA 短信自注册和 IMS 短信自注册相关功能。

读者对象


适用于与电信移动终端自注册相关的研发和测试人员。

缩略语

缩略语	英文全名	中文解释
DM	Device Management	终端管理
CDMA	Code Division Multiple Access	码分多址
GSM	Global System for Mobile Communications	全球移动通信系统
WCDMA	Wideband Code Division Multiple Access	宽带码分多址增强型数据速率 GSM 演进技术
LTE	Long Term Evolution	长期演进技术
WIFI	Wireless Fidelity	基于 IEEE 802.11b 标准的无线局域网
IMS	IP Multimedia Subsystem	因特网协议（IP）多媒体子系统
ICCID	Integrate Circuit Card Identity	集成电路卡识别码
MEID	Mobile Equipment Identifier	移动设备识别码

符号约定

在本文中可能出现下列标志，它所代表的含义如下。

符号	说明
 说明	用于突出重要/关键信息、补充信息和小窍门等。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害。

变更信息

文档版本	发布日期	修改说明
V1.0	2019-03-19	第一次正式发布。
V1.1	2019-08-08	<ul style="list-style-type: none"> 修改 Android 4.4 平台适配，去掉可选字段 增加支持 Android 9.0 平台 增加支持短信自注册
V1.2	2020-08-14	<ul style="list-style-type: none"> 增加 5G 终端平台支持及说明 增加支持 Android 10.0 平台 增加支持 Android 11.0 平台 文档名由《中国电信移动终端自注册客户指导手册》改为《中国电信移动终端自注册客制化指导手册》

关键字

数据自注册、CDMA 短信自注册、IMS 短信自注册、电信入库。

目 录

1 概览	1
2 电信移动终端自注册	2
2.1 电信移动终端数据自注册介绍	2
2.1.1 自注册流程	2
2.1.2 平台支持情况	5
2.1.3 测试说明	6
2.2 电信移动终端 CDMA 短信自注册介绍	9
2.2.1 自注册流程	9
2.2.2 平台支持情况	10
2.2.3 测试说明	10
2.3 电信移动终端 IMS 短信自注册介绍	13
2.3.1 自注册流程	13
2.3.2 平台支持情况	14
2.3.3 测试说明	14
3 电信移动终端自注册各平台适配	17
3.1 电信移动终端 Mocom5 平台自注册模块适配	17
3.1.1 代码路径	17
3.1.2 DM 版本号	17
3.1.3 厂商名称、手机型号	17
3.1.4 软件版本号、系统版本号	17
3.1.5 上报字段配置	17
3.2 电信 LTE 移动终端 Android 9.0 平台自注册模块适配	18
3.2.1 代码路径	18
3.2.2 DM 版本号	19
3.2.3 厂商名称、手机型号	19
3.2.4 软件版本号、系统版本号	19
3.2.5 CDMA 短信自注册需要合入 CP MEID 写 CSIM 卡的修改	19
3.3 电信 5G 移动终端 Android 9.0 平台送测说明	19
3.3.1 代码路径	19
3.3.2 DM 版本号	20
3.3.3 厂商名称、手机型号	20
3.3.4 软件版本号、系统版本号	20
3.3.5 上报字段配置	20
3.3.6 测试说明	21
3.4 电信 LTE 移动终端 Android 10.0 平台自注册模块适配	22

3.4.1 代码路径.....	22
3.4.2 DM 版本号.....	22
3.4.3 厂商名称、手机型号.....	22
3.4.4 软件版本号、系统版本号	22
3.4.5 CDMA 设备关闭开机存储 MEID 功能	22
3.4.6 电信自注册功能开关.....	22
3.5 电信 LTE 移动终端 Android 11.0 平台自注册模块适配	23
3.5.1 代码路径.....	23
3.5.2 DM 版本号.....	23
3.5.3 厂商名称、手机型号.....	23
3.5.4 软件版本号、系统版本号	23
3.5.5 CDMA 设备关闭开机存储 MEID 功能	23
3.5.6 电信自注册功能开关.....	23
4 电信移动终端自注册 FAQ.....	24
5 参考文献.....	27

Unisoc Confidential For hiar

图目录

图 2-1 电信移动终端数据自注册流程	2
图 2-2 BASE64 在线工具解码.....	8
图 2-3 电信服务器 log 信息	9
图 2-4 电信服务器 log 字段解读	9

Unisoc Confidential For hiar

表目录

表 2-1 终端自注册信息内容.....	3
表 2-2 电信自注册服务器返回结果.....	5
表 2-3 平台数据自注册的支持情况.....	5
表 2-4 平台 CDMA 短信自注册的支持情况	10
表 2-5 平台 IMS 短信自注册的支持情况.....	14

Unisoc Confidential For hiar

1 概览

电信自注册是电信入库的基本需求，也是电信入库测试非常重要的一环。由于以下事项：

- 自注册模块需要与平台其他多个模块、电信 DM 服务器进行交互。
- 测试用例多。
- 涉及到多种场景，如 WIFI 网络、数据域网络、不同运营商 SIM 卡、不同卡商、SIM 热插拔、开关机、恢复出厂设置、OTA 升级等。
- 电信测试每次送测都需要进行预检，预检过程对自注册功能关注度很高。

在自注册送测之前需做好充分的准备和自测试：

- 送测前要确保 SIM 热插拔功能稳定可靠（如果支持的话），需要做压力测试。
- 送测前要对开关机进行压力测试，解决定屏、死机、不驻网、不识卡、不驻留 4G 等问题。
- 电信测试报告出来后，对于失败的 case 可以申请复测，或者申请抓 log。
- 对于自测试始终无法复现的问题建议申请电信复测，并说明情况。

Unisoc Confidential For hiar

2 电信移动终端自注册

2.1 电信移动终端数据自注册介绍

电信数据自注册功能，是指将终端相关信息通过数据域上报给电信自注册服务器的过程。请参考电信官方发布的《中国电信移动终端需求白皮书-数据自注册功能分册（2018.V2）20190111174357531.pdf》（若电信官方发布新文档，则以最新文档为准）。该规范已经于 2019 年 1 月 15 日正式生效，2019 年 1 月 15 日之后送测的终端版本，都需要满足此规范。

2.1.1 自注册流程

主要流程图，请参见图 2-1。

图2-1 电信移动终端数据自注册流程



主要流程如下：

步骤 1 获取移动终端相关信息，上报必选字段。参数如表 2-1。

表2-1 终端自注册信息内容

参数名称	说明	最大字节数	数据类型	支持程度	备注
REGVER	自注册格式版本号。表示当前数据域注册协议的版本。初始版本号为1.0。	6个字节	字符串	必选	版本号的升级格式为1.0、2.0、3.0.....当前版本为8.0。
MEID	终端的串号	14个字节	字符串	必选	必须与终端机身的MEID、包装盒的MEID以及终端查询到的MEID保持一致。
MODEL	机型字段。组成：“厂商名称代码-手机型号”。	20个字节	字符串	必选	对应短信自注册机型字段，取值跟短信自注册型号保持一致。
SWVER	软件版本号	60个字节	字符串	必选	与申请表保持一致。
SIM1CDMAIMSI	卡槽1，CDMA的IMSI	15个字节	字符串	必选	不插卡时填充值为空”，不能为0。
UETYPE	终端类型：1~8 <ul style="list-style-type: none"> • 1 代表 phone • 2 代表 pad • 3 代表 DataCard • 4 代表 CPE • 5 预留 • 6 预留 • 7 代表物联网通信模块 • 8 代表物联网终端 	2个字节	字符串	必选	无
SIM1ICCID	卡槽1，ICCID	20个字节	字符串	必选	不插卡时填充值为空”，不能为0。
SIM1LTEIMSI	卡槽1，LTE IMSI	15个字节	字符串	必选	不插卡时填充值为空”，不能为0。
SIM2CDMAIMSI	卡槽2，CDMA IMSI	15个字节	字符串	必选	<ul style="list-style-type: none"> • 单卡终端填充值为空” • 双卡终端不插卡时填充值为空”，不能为0。
SIM2ICCID	卡槽2，ICCID	20个字节	字符串	必选	<ul style="list-style-type: none"> • 单卡终端填充值为空” • 双卡终端不插卡时填充值为空”，不能为0。
SIM2LTEIMSI	卡槽2，LTE IMSI	15个字节	字符串	必选	<ul style="list-style-type: none"> • 单卡终端填充值为空” • 双卡终端不插卡时填充值为空”，不能为0。

参数名称	说明	最大字节数	数据类型	支持程度	备注
MACID	终端Mac地址	48个字节	字符串	必选	WLAN网卡地址，与终端显示一致。
OSVER	操作系统版本号，IOS写运营商版本。该字段应为“操作系统名称+版本号”组合。	32个字节	字符串	必选	举例：例如Android的5.0版本，该字段应为“android5.0”。
IMEI1	卡槽1，IMEI号	15个字节	字符串	必选	无
IMEI2	卡槽2，IMEI号	15个字节	字符串	必选	单卡终端填充值为空值“”
SIM1CELLID	卡槽1的CELLID	20个字节	字符串	必选	<ul style="list-style-type: none"> 应该上报当前工作模式的CELLID，上报次序为LTE>WCDMA>GSM。 不插卡时填充值为空“”，不能为0。
SIM2CELLID	卡槽2的CELLID	20个字节	字符串	必选	<ul style="list-style-type: none"> 应该上报当前工作模式的CELLID，上报次序为LTE>WCDMA>GSM。 不插卡时填充值为空“”，不能为0。 异网卡可以上报为空“”。
DATASIM	目前使用移动数据的卡槽	2个字节	字符串	必选	<ul style="list-style-type: none"> 卡槽一或单卡终端为“1” 卡槽二为“2”

步骤 2 以 Json 格式封装。

Json 格式编码时使用一级平铺方式，不能使用嵌套、数组或者其他方式。

步骤 3 以 BASE64 方式编码。

终端上报注册信息时，应对符合要求的自注册信息数据进行加密处理，加密方式为在网络侧可传输的数据密文 A=Base64 编码方式（明文）。

步骤 4 通过 Http post 方法发送到电信自注册服务器。

- 消息类型：HTTP POST
- 自注册手机终端服务器：http://zzhc.vnet.cn/
- 内容类型：Content-Type=application/encrypted-json
- 内容说明：符合模块自注册功能上报信息加密方法要求上报信息的密文

步骤 5 如果电信自注册服务器校验成功，则向终端返回 **Success**。终端自注册的信息会同时保存到电信自注册服务器上。具体返回结果请参见表 2-2。

表2-2 电信自注册服务器返回结果

resultCode（最大 3 个字节）	resultDesc（最大 50 个字节）	中文解释
0	Success	成功
1	Decode error	解码错误
2	Check error	校验错误
其他保留待扩展	无	无

----结束

2.1.2 平台支持情况

紫光展锐手机平台数据自注册的支持情况，请参见表 2-3。

表2-3 平台数据自注册的支持情况

系统版本	软件版本	终端类型	工程支持	是否支持 CDMA	默认开启
Mocor5	MOCOR5_SFPHONE_17F_RLS1_W18.45.5 及后续版本	移动终端	CTCC 工程	芯片不支持 CDMA	是
Android6.0	MOCORDROID6.0_TRUNK_16B_RLS2_W17.15.5+PATCH	泛智能终端	CTCC 工程	芯片不支持 CDMA	是
Android8.1	MOCORDROID8.1_TRUNK_18A_RLS2_W19.26.5 及后续版本	泛智能终端	CTCC 工程	芯片不支持 CDMA	是
Android9.0	MOCORDROID9.0_Trunk_SHARKL5_19A_W19.22.5 及后续版本	移动终端	CTCC 工程	芯片支持 CDMA	是
Android9.0	MOCORDROID9.0_Trunk_19B_ROC1_ORCA_W19.51.5（5G）及后续版本	移动终端	CTCC 工程	芯片不支持 CDMA	是
Android10.0	MOCORDROID10_Trunk_19C_W20.07.4 及后续版本	移动终端	CTCC 工程	以芯片 CDMA 支持能力为准	是
Android11.0	MOCORDROIDR_Trunk_W20.24.5 及后续版本	移动终端	CTCC 工程	以芯片 CDMA 支持能力为准	是

2.1.3 测试说明

电信移动终端数据自注册的适配工作完成后，即可开展相关的测试工作。请参考电信官方发布的《中国电信移动终端测试方法全集-数据自注册功能分册 2018.V220190111100843520.pdf》。

电信测试会完全按照这些测试用例来测试，但测试顺序可能会调整。例如电信可能会把终端版本升级放在最后测试。

同时，因为电信数据自注册没有提供对应的适配服务器供研发和测试验证，研发和测试只能通过终端 **Android log** 和 **Cap** 包来确认。对于不支持 **CDMA** 的电信 **4G** 终端，电信自注册平台目前无法查询到自注册结果，需要申请电信服务器部门协助查询。

2.1.3.1 测试 log 抓取

对于 SC9820E Mocor5 平台，步骤如下：

步骤 1 输入 ***##83781#*##**，打开工程模式，在 **slog** 中选上 **Cap** 包。

步骤 2 根据电信数据自注册触发条件：第一次开机、更换 **SIM** 卡、主副卡切换、**30** 天重发，触发数据自注册，抓取完整的 **Android slog**（**/data/slog/**）。

步骤 3 在 **Android slog** 中的 **main log** 文件中搜索关键字 **DmService**，可查看电信数据自注册模块所有 **log** 信息。

- **Cap** 包也在 **slog** 目录下。
- 如果不插入 **SD** 卡，**Android log** 路径在 **/data/slog** 目录下。当存储空间满时，新的 **log** 会把老的 **log** 覆盖。
- 在插入 **SD** 卡的情况下，完整 **log** 包含两部分：**/data/slog**（**Android log** 开机部分）和 **SD** 卡根目录的 **slog**。

步骤 4 在网络 **TCP Cap** 包里搜索服务器网址 **zzhc.vnet.cn**，查看终端跟电信服务器交互的网络数据。

----结束

对于 **UMS312 Android9.0** 平台，说明如下：

- **Android log** 和网络 **Cap** 包默认都会抓取。
- 不插入 **SD** 卡，**Android log** 路径在 **/storage/emulated/0/ylog** 目录下。
- 在插入 **SD** 卡的情况下，**Android log** 在 **SD** 卡根目录下的 **ylog** 目录下。

2.1.3.2 测试 log 分析

以 **SC9820E Mocor5** 平台为例，分析电信数据自注册 **log**，方式如下：

- 在 **Android log** 中的 **main log** 中搜索关键字 **DmService**，查看当前获得的终端信息。

```
03-20 15:13:28.170 1246 1260 D DmService: REGVER = 8.0
03-20 15:13:28.170 1246 1260 D DmService: MEID = 8689880400000001
03-20 15:13:28.180 1246 1260 D DmService: MODEL = HS-EG981
03-20 15:13:28.180 1246 1260 D DmService: SWVER = Mocor 1.0
03-20 15:13:28.180 1246 1260 D DmService: SIM1CDMAIMSI =
03-20 15:13:28.180 1246 1260 D DmService: UETYPE = 1
03-20 15:13:28.180 1246 1260 D DmService: SIM1ICCID = 89860312805320133131
03-20 15:13:28.180 1246 1260 D DmService: SIM1LTEIMSI = 460111234567890
```

- 查看移动终端信息的 json 格式化封装。

- 查看移动终端通过 http post 发送的 BASE64 编码数据。

- 终端收到的服务器响应。

- 查看终端 ap 侧的 Cap 包。

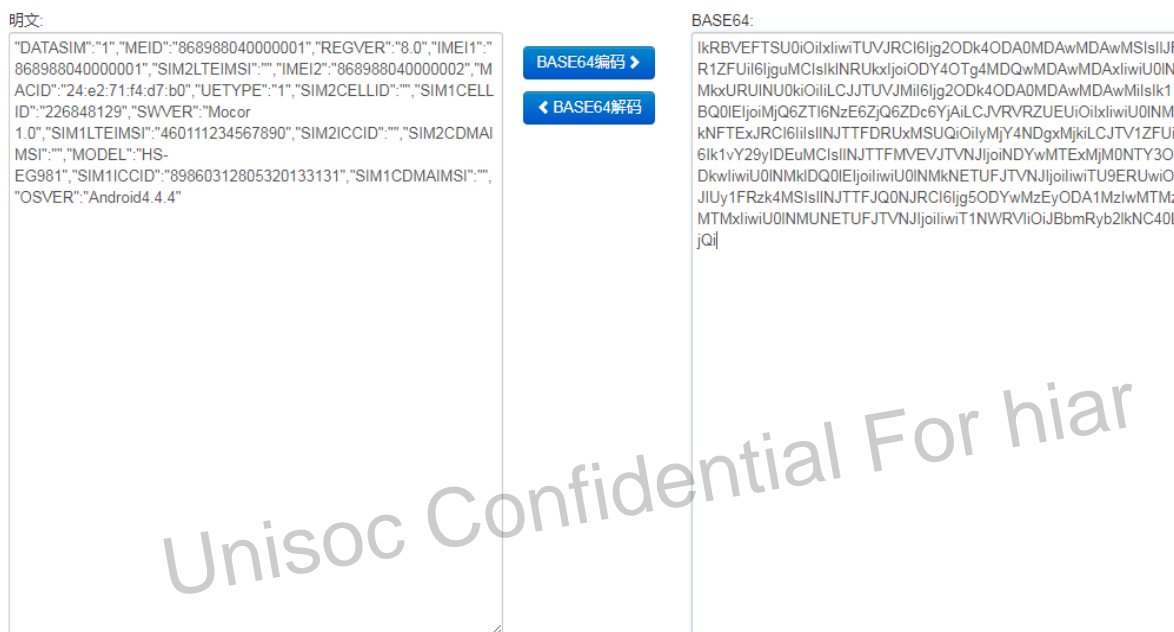
文档版本 V1.2 (2020-08-14)

```
JiJoiIiwiT1NWRViiOiJBbmRyb2lkNC40LjQi //BASE64 明文编码, 可以使用在线工具解码
http://tool.oschina.net/encrypt?type=3

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/plain; charset=ISO-8859-1
Content-Length: 41
Date: Wed, 20 Mar 2019 07:13:29 GMT
{"resultCode": "0", "resultDesc": "Success"}
```

- 网络报文中 **BASE64** 明文编码部分。可以使用在线工具解码。

图2-2 BASE64 在线工具解码



- 对于不支持 **CDMA** 的电信 **4G** 终端, 电信数据自注册平台可能查询不到上报信息。需要电信测试人员通过邮件, 向电信服务器部门申请获取电信服务器 **log** 信息。以 2018 年 V1 版电信需求送测为例, **DM** 数据自注册版本为 7.0, 电信服务器 **log** 信息如图 2-3 所示。

步骤 2 终端以普通短信方式（TeleserviceID 是 4098）向服务器（10659401）发送 CDMA 短信自注册。

- 短信以 8bit 编码
- 短信头部是 4 个字节的 CDMA 短信自注册协议头
- 短信内容

```
<a1>
<b1>Device Model</b1>
<b2>MEID</b2>
<b3>CDMAIMSI</b3>
<b4>Software Version</b4>
</a1>CRC
```

步骤 3 服务器向终端反馈“注册成功确认”短消息（TeleserviceID 是 65005，短信内容（0x01,0x04））。CDMA 短信自注册成功。

---结束

2.2.2 平台支持情况

紫光展锐手机平台 CDMA 短信自注册的支持情况，请参见表 2-4。

表2-4 平台 CDMA 短信自注册的支持情况

系统版本	软件版本	工程支持	是否支持 CDMA	默认支持
Android9.0	MOCORDROID9.0_Trunk_SHARKL5_19A_W19.22.5 及后续版本	CTCC 工程	芯片支持 CDMA	是
Android10.0	MOCORDROID10_Trunk_19C_W20.07.4 及后续版本	CTCC 工程	以芯片 CDMA 支持能力为准	是
Android11.0	MOCORDROIDR_Trunk_W20.24.5 及后续版本	CTCC 工程	以芯片 CDMA 支持能力为准	是

2.2.3 测试说明

电信移动终端数据自注册的适配工作完成后，即可开展相关的测试工作。请参考电信官方发布的《中国电信移动终端测试方法全集-短信自注册功能分册 2018.V220190111100854593.pdf》（若电信官方发布新文档，则以最新文档为准）。

2.2.3.1 测试 log 抓取

对于 UMS312 Android9.0 平台，说明如下：

- Android log 和网络 Cap 包默认都会抓取。

- 不插入 SD 卡，Android log 路径在/storage/emulated/0/ylog 目录下。
- 在插入 SD 卡的情况下，Android log 在 SD 卡根目录下的 ylog 目录下。

2.2.3.2 测试 log 分析

UMS312 Android9.0 平台，在 ylog 的 main.log 中搜索关键字：DmService。

正常 CDMA 自注册流程如下：

```
03-20 22:19:13.673 2024 2024 D DmService: onReceive, action is android.intent.action.SERVICE STATE
03-20 22:19:13.673 2024 2024 D DmService: onReceive, Service state changed.....
03-20 22:19:13.673 2024 2024 D DmService: handleServiceStateChanged, ant: 4
03-20 22:19:13.673 2024 2024 D DmService: handleServiceStateChanged, state: 0
03-20 22:19:13.673 2024 2024 D DmService: handleServiceStateChanged, CDMA registered.
-----驻 CDMA 网络
03-20 22:19:13.676 2058 2058 D DmServiceSms: isNeedCdmaSelfReg, Begin, slot = 0
03-20 22:19:13.688 2058 2058 I DmService: isMasterSlotCtccSimCard, imsi = 460110136600638
03-20 22:19:13.690 2058 2058 D DmServiceSms: isNeedCdmaSelfReg, ue cdma imsi1 = 460036181429919
03-20 22:19:13.698 2058 2058 D DmServiceSms: isNeedCdmaSelfReg, ue cdma imsi2 = 460037801287791
03-20 22:19:13.724 2058 2058 I DmService: isMasterSlotCtccSimCard, imsi = 460016241514946
03-20 22:19:13.753 2058 2058 D DmServiceSms: isNeedCdmaSelfReg, cdma imsi1 = 460036181429919
03-20 22:19:13.753 2058 2058 D DmServiceSms: isNeedCdmaSelfReg, cdma imsi2 = 460016241514946
03-20 22:19:13.753 2058 2094 D DmServiceSms: SelfRegisterHandler:generate data
03-20 22:19:13.754 2058 2094 D DmService: getModel, modelInfo = ZXS-ums312_1h10_nati
03-20 22:19:13.754 2058 2094 D DmService: getSwVer, swVer = ums312_1h10_ctcc_nosec-userdebug 9 PPR1.
03-20 22:19:13.767 2058 2094 D DmService: getMeid, meid = A00000203EDE34
03-20 22:19:13.772 2058 2094 D DmService: XmlHelper, createXMLData, xmlData = <a1><b1>ZXS-ums312_1h1</b1><b2>A00000203EDE34</b2><b3>460036181429919</b3><b4>ums312_1h10_ctcc_nosec-userdebug 9 PPR1</b4></a1>----- CDMA 自注册短信内容
03-20 22:19:13.772 2058 2094 D DmServiceSms: generateCdmaRegData, xml data length = 127
03-20 22:19:13.772 2058 2094 D DmServiceSms: generateData, msgData: __□ <a1><b1>ZXS-ums312_1h1</b1><b2>A00000203EDE34</b2><b3>460036181429919</b3><b4>ums312_1h10_ctcc_nosec-userdebug 9 PPR1</b4></a1>
03-20 22:19:13.772 2058 2094 D DmServiceSms: generateData, msgData length = 131
03-20 22:19:13.774 2058 2094 D DmServiceSms: sendSmsRegDataMessage ====data : ----发送 CDMA 自注册短信
03-20 22:19:13.774 2058 2094 D DmService: dumpDataInHex,
=====Begin=====
03-20 22:19:13.774 2058 2094 D DmService: dumpDataInHex, dataLen = 139
03-20 22:19:13.774 2058 2094 D DmService: dumpDataInHex:
02037f003c61313e3c62313e5a58532d756d733331325f3168313c2f62313e3c
03-20 22:19:13.774 2058 2094 D DmService: dumpDataInHex:
62323e4130303030303032303345444533343c2f62323e3c62333e343630303336
03-20 22:19:13.774 2058 2094 D DmService: dumpDataInHex:
3138313432393931393c2f62333e3c62343e756d733331325f316831305f6374
03-20 22:19:13.774 2058 2094 D DmService: dumpDataInHex:
63635f6e6f7365632d757365726465627567203920505052313c2f62343e3c2f
03-20 22:19:13.774 2058 2094 D DmService: dumpDataInHex: 61313e3934623265366662
03-20 22:19:13.774 2058 2094 D DmService: dumpDataInHex,
=====End=====
03-20 22:19:13.791 2058 2094 D DmService: getSubId, subId = 1
03-20 22:19:13.791 2058 2094 D DmServiceSms: sendSmsRegDataMessage: slot = 0
03-20 22:19:13.792 2058 2094 D DmServiceSms: sendSmsRegDataMessage: subId = 1
03-20 22:19:13.803 2058 2094 D DmServiceSms: sendSmsRegDataMessage: messageUri = 2019-03-20 22:19:13
```

```

03-20 22:19:13.804 2058 2094 D DmServiceSms: sendSmsRegDataMessage: mPendingMessageMap = {2019-03-20
22:19:13=SendResult:Pending=1,HighestFailureLevel=0}
03-20 22:19:13.804 2058 2094 D DmServiceSms: sendSmsRegDataMessage: port = 0
03-20 22:19:15.960 2024 2024 D DmService: onReceive, action is
com.sprd.dm.DmSmsService.SendStatus.MESSAGE_SENT
03-20 22:19:15.960 2024 2024 D DmService: onReceive, resultCode = -1
03-20 22:19:15.960 2024 2024 D DmService: onReceive, requestId = 2019-03-20 22:19:13
03-20 22:19:15.960 2024 2024 D DmService: onReceive, EXTRA_ERROR_CODE: 0
03-20 22:19:15.960 2024 2024 D DmService: onReceive, EXTRA_PART_ID: 0
03-20 22:19:15.960 2024 2024 D DmService: onReceive, EXTRA_ERROR_CODE: 1
03-20 22:19:15.961 2024 2024 D DmService: printIntent, intent = Intent
{ act=com.sprd.dm.DmSmsService.Message.Send pkg=com.sprd.dm.mbsselfreg (has extras) }
03-20 22:19:15.961 2024 2024 D DmService: printIntent, key = partId, value = 0
03-20 22:19:15.961 2024 2024 D DmService: printIntent, key = resultCode, value = -1
03-20 22:19:15.961 2024 2024 D DmService: printIntent, key = Uri, value = 2019-03-20 22:19:13
03-20 22:19:15.961 2024 2024 D DmService: printIntent, key = subId, value = 1
03-20 22:19:15.961 2024 2024 D DmService: printIntent, key = errorCode, value = 0
03-20 22:19:15.967 2058 2058 D DmServiceSms: onStart, action = com.sprd.dm.DmSmsService.Message.Send
03-20 22:19:15.967 2058 2058 D DmServiceSms: handleSendStatus, requestId = 2019-03-20 22:19:13
03-20 22:19:15.967 2058 2058 D DmServiceSms: handleSendStatus, resultCode = -1
03-20 22:19:15.967 2058 2058 D DmServiceSms: handleSendStatus, EXTRA_ERROR_CODE: 0
03-20 22:19:15.967 2058 2058 D DmServiceSms: handleSendStatus, EXTRA_PART_ID: 0
03-20 22:19:15.967 2058 2058 D DmServiceSms: handleSendStatus, EXTRA_ERROR_CODE: 1
03-20 22:19:15.967 2058 2058 D DmServiceSms: setResult: requestId = 2019-03-20 22:19:13
03-20 22:19:15.967 2058 2058 D DmServiceSms: setResult: mPendingMessageMap = {2019-03-20
22:19:13=SendResult:Pending=1,HighestFailureLevel=0}
03-20 22:19:15.968 2058 2094 D DmServiceSms: sendSmsRegDataMessage: waiting for message send
result...
03-20 22:19:15.968 2058 2094 D DmServiceSms: sendSmsRegDataMessage: sending completed. dest=10659401
message= <al><b1>ZXS-
ums312 1h1</b1><b2>A00000203EDE34</b2><b3>460036181429919</b3><b4>ums312 1h10 ctcc nsec-userdebug 9
PPR1</b4></al>94b2e6fb result=SendResult:Pending=0,HighestFailureLevel=0
03-20 22:19:15.968 2058 2094 D DmServiceSms: sendSmsRegDataMessage: send selfregister sms success.
-----CDMA 自注册短信发送成功
03-20 22:19:15.969 2058 2094 D DmServiceSms: saveCdmaSelfRegInfo, cmda slot id = 0
03-20 22:19:15.972 2058 2094 D DmServiceSms: putUeImsi imsi sim1:460036181429919
03-20 22:19:15.978 2058 2094 I DmService: isMasterSlotCtccSimCard, imsi = 460016241514946
03-20 22:19:15.991 2058 2094 D DmServiceSms: putUeImsi imsi sim2:460016241514946
03-20 22:19:16.001 2058 2094 D DmService: getMeid, meid = A00000203EDE34
03-20 22:19:16.001 2058 2094 I DmService: setMeidToSim: meid = A00000203EDE34
03-20 22:19:24.828 2024 2024 D DmService: onReceive, action is android.intent.action.SERVICE_STATE
03-20 22:19:24.828 2024 2024 D DmService: onReceive, Service state changed.....
03-20 22:19:24.829 2024 2024 D DmService: handleServiceStateChanged, ant: 2
03-20 22:19:24.829 2024 2024 D DmService: handleServiceStateChanged, state: 0
03-20 22:19:26.615 2024 2024 D DmService: onReceive, action is android.intent.action.SERVICE_STATE
03-20 22:19:26.615 2024 2024 D DmService: onReceive, Service state changed.....
03-20 22:19:26.616 2024 2024 D DmService: handleServiceStateChanged, ant: 2
03-20 22:19:26.616 2024 2024 D DmService: handleServiceStateChanged, state: 0
03-20 22:20:30.433 2058 2071 D DmServiceSms: onReceiveTextSms: format = 3gpp2 -----收到服务器返回的短信
03-20 22:20:30.433 2058 2071 D DmServiceSms: onReceiveTextSms: destPort = -1
03-20 22:20:30.433 2058 2071 D DmServiceSms: onReceiveTextSms: subId = 1
03-20 22:20:30.433 2058 2071 D DmServiceSms: onReceiveTextSms: pdu = android.service.carrier.MessagePdu@d0e9e1
03-20 22:20:30.433 2058 2071 D DmServiceSms: onFilterSms, onReceiveResult: result = true
03-20 22:20:30.433 2058 2071 D DmServiceSms: onReceiveTextSms: format = 3gpp2
03-20 22:20:30.433 2058 2071 D DmServiceSms: DmUtil.dumpDataInHex ====>pduList[0]:

```



```

03-20 22:20:30.433 2058 2071 D DmService: dumpDataInHex,
=====Begin=====
03-20 22:20:30.433 2058 2071 D DmService: dumpDataInHex, dataLen = 61
03-20 22:20:30.433 2058 2071 D DmService: dumpDataInHex:
000000000000fded000000000000000008313036353934303100000000000000
03-20 22:20:30.433 2058 2071 D DmService: dumpDataInHex:
000000190003100000001040010082003061903202216270801000a0100
03-20 22:20:30.433 2058 2071 D DmService: dumpDataInHex,
=====End=====
03-20 22:20:30.435 2058 2071 D DmServiceSms: onReceiveTextSms: fromAddress = 10659401 -----服务器地址 10659401
03-20 22:20:30.435 2058 2071 D DmServiceSms: onReceiveTextSms: userData:
03-20 22:20:30.435 2058 2071 D DmService: dumpDataInHex,
=====Begin=====
03-20 22:20:30.435 2058 2071 D DmService: dumpDataInHex, dataLen = 2
03-20 22:20:30.435 2058 2071 D DmService: dumpDataInHex: 0104 -----服务器回复短信内容 0x01, 0x04 表示 CDMA 短信自注册成功
03-20 22:20:30.436 2058 2071 D DmService: dumpDataInHex,
=====End=====
03-20 22:20:30.436 2058 2071 D DmServiceSms: onReceiveTextSms: teleServiceID = 65005 -----服务器短信的 teleServiceID 为 65005
03-20 22:20:30.436 2058 2071 D DmServiceSms: onReceiveTextSms: fullMsgBody =
03-20 22:20:30.438 2058 2071 D DmServiceSms: onFilterSms: slotId = 0
03-20 22:20:30.439 2058 2071 D DmServiceSms: checkCdmaResponse dumpDataInHex responseData====>
03-20 22:20:30.439 2058 2071 D DmService: dumpDataInHex,
=====Begin=====
03-20 22:20:30.439 2058 2071 D DmService: dumpDataInHex, dataLen = 2
03-20 22:20:30.439 2058 2071 D DmService: dumpDataInHex: 0104
03-20 22:20:30.439 2058 2071 D DmService: dumpDataInHex,
=====End=====
03-20 22:20:30.443 2058 2071 D DmServiceSms: Cdma Sms self register success!

```

2.3 电信移动终端 IMS 短信自注册介绍

电信 IMS 短信自注册功能，主要是指根据电信的需求白皮书，将终端相关信息通过 IMS 短信上报给电信自注册服务器的过程。请参考电信官方发布的《3-中国电信移动终端需求白皮书-短信自注册功能分册（2019.V1_20190314）.docx》（若电信官方发布新文档，则以最新文档为准）。该规范中 IMS 短信自注册功能已经于 2019 年 9 月 1 日正式生效，2019 年 9 月 1 日之后送测的终端版本，都需要满足此规范中对于 IMS 短信自注册的要求。

2.3.1 自注册流程

主要流程如下：

步骤 1 当终端登录到 IMS 网络时，若满足以下条件会触发自注册。

终端记录的当前卡槽的 LTE IMSI 和当前插入 SIM 卡的 LTE IMSI 不一致。

步骤 2 终端以普通短信方式（TeleserviceID 是 4098）向服务器（10659401）发送 IMS 自注册短信。

- 短信以 8bit 编码。
- 短信头部是 4 个字节的 IMS 短信自注册协议头。

– 短信内容：

```
<a1>
<b1>Device Model
<b2>IMEI1 Number (以 ASCII 码方式显示。例如：为字符串“862031030150330”)
<b3>IMEI2 Number (以 ASCII 码方式显示。例如：为字符串“862031030150330”)
<b4>IMSI Number (15 位十进制数字，例如：为字符串“460111234567890”)
<b5>Software Version (不超过 40 个字节)
</a1>
```

步骤 3 服务器向终端反馈“注册成功确认”短消息（TeleserviceID 是 65005，短信内容（0x03,0x04））。IMS 短信自注册成功。

----结束

2.3.2 平台支持情况

紫光展锐手机平台 IMS 短信自注册的支持情况，请参见表 2-5。

表2-5 平台 IMS 短信自注册的支持情况

Android 版本	软件版本	工程支持	是否支持 CDMA	默认开启
Mocor5	MOCOR5_SFPHONE_17F_RLS1_W19.31.1 及后续版本	CTCC 工程	芯片不支持 CDMA	是
Android9.0	MOCORDROID9.0_Trunk_SHARKL5_19A_W19.26.5 及后续版本 MOCORDROID9.0_Trunk_19B_ROC1_ORCA_W19.51.5	CTCC 工程	以芯片 CDMA 支持能力为准	是
Android10.0	MOCORDROID10_Trunk_19C_W20.07.4 及后续版本	CTCC 工程	以芯片 CDMA 支持能力为准	是
Android11.0	MOCORDROIDR_Trunk_W20.24.5 及后续版本	CTCC 工程	以芯片 CDMA 支持能力为准	是

2.3.3 测试说明

电信移动终端数据自注册的适配工作完成后，即可开展相关的测试工作。请参考电信官方发布的《中国电信移动终端测试方法全集-短信自注册功能分册 2018.V220190111100854593.pdf》（若电信官方发布新文档，则以最新文档为准）。

2.3.3.1 测试 log 抓取

对于 UMS312 Android9.0 平台，说明如下：

- Android log 和网络 Cap 包默认都会抓取。
- 不插入 SD 卡，Android log 路径在/storage/emulated/0/ylog 目录下。
- 在插入 SD 卡的情况下，Android log 在 SD 卡根目录下的 ylog 目录下。

2.3.3.2 测试 log 分析

UMS312 Android9.0 平台，在 ylog 的 main.log 中搜索关键字：DmService。

正常 CDMA 自注册流程如下：

```

M00AC0D 06-18 09:35:23.493 1979 1979 D DmServiceSms: onStart, action =
com.sprd.dm.DmSmsService.ImsReg ----终端收到 IMS 驻网广播
M00AC5E 06-18 09:35:23.558 1979 1979 D DmServiceSms: isNeedImsSelfReg, ims1 = 460110130472497 ---
-电信卡的 LTE IMSI
M00AC5F 06-18 09:35:23.559 1979 1979 D DmServiceSms: isNeedImsSelfReg, ueimsi sim1 =
0000000000000000
M00AC6C 06-18 09:35:23.569 1979 2034 D DmService: getImei, imei = 867400020316612 ----IMEI1
M00AC78 06-18 09:35:23.574 1979 2034 D DmService: getImei, imei = 867400020316620 ----IMEI2
M00ACC0 06-18 09:35:23.603 1979 2034 D DmServiceSms: generateImsRegData getModel, devMod = HS-
ums312_1h10_nativ -----Device Model
M00ACC1 06-18 09:35:23.603 1979 2034 D DmServiceSms: generateImsRegData getSwVer, swVer =
ums312_1h10_ctcc_nosec-userdebug 9 PPR1. -----软件版本号
DmServiceSms: generateImsRegData, message: <a1><b1>HS-
ums312_1h10_nativ<b2>867400020316612<b3>867400020316620<b4>460110130472497<b5>ums312_1h10_ctcc_nosec-
userdebug 9 PPR1.</a1>
M00ACC5 06-18 09:35:23.603 1979 2034 D DmServiceSms: DmUtil.dumpDataInHex ====>msgData:
M00ACC6 06-18 09:35:23.604 1979 2034 D DmService: dumpDataInHex,
=====Begin=====
M00ACC7 06-18 09:35:23.604 1979 2034 D DmService: dumpDataInHex, dataLen = 138
03033f003c61313e3c62313e48532d756d733331325f316831305f6e61746976
3c62323e3836373430303032303331363631323c62333e383637343030303230
3331363632303c62343e3436303131303133303437323439373c62353e756d73
3331325f316831305f637463635f6e6f7365632d757365726465627567203920
505052312e3c2f61313e -----IMS 自注册短信 16 进制内容，包含 4 个字节的 IMS 短信自注册协议头
=====End=====
M00B21D 06-18 09:35:24.475 1979 2034 D DmServiceSms: sendSmsRegDataMessage: sending completed.
dest=10659401 message=◆◆◆◆<a1><b1>HS-
ums312_1h10_nativ<b2>867400020316612<b3>867400020316620<b4>460110130472497<b5>ums312_1h10_ctcc_nosec-
userdebug 9 PPR1.</a1> result=SendResult:Pending=0,HighestFailureLevel=0
M00B21E 06-18 09:35:24.475 1979 2034 D DmServiceSms: sendSmsRegDataMessage: send selfregister sms
success. -----IMS 自注册短信发送成功
06-18 10:20:39.598 1979 2016 D DmServiceSms: onReceiveTextSms: format = 3gpp
06-18 10:20:39.598 1979 2016 D DmServiceSms: DmUtil.dumpDataInHex ====>pduList[0]:
06-18 10:20:39.599 1979 2016 D DmService: dumpDataInHex,
=====Begin=====
06-18 10:20:39.599 1979 2016 D DmService: dumpDataInHex, dataLen = 29
06-18 10:20:39.599 1979 2016 D DmService: dumpDataInHex:
099164000349145802f02408a101564910000491608101020123020304 ----收到电信服务器回复短信内容
06-18 10:20:39.599 1979 2016 D DmService: dumpDataInHex,
=====End=====
06-18 10:20:39.614 1979 2016 D DmServiceSms: onReceiveTextSms: fromAddress = 10659401 ----电信服务器地
址
06-18 10:20:39.615 1979 2016 D DmServiceSms: onReceiveTextSms: userData:
06-18 10:20:39.615 1979 2016 D DmService: dumpDataInHex,
=====Begin=====
  
```

```
06-18 10:20:39.615 1979 2016 D DmService: dumpDataInHex, dataLen = 2
06-18 10:20:39.615 1979 2016 D DmService: dumpDataInHex: 0304 ----解析后电信服务器回复 IMS 短信自注册内容
=====End=====
06-18 10:20:39.619 1979 2016 D DmServiceSms: checkImsResponse register success
06-18 10:20:39.619 1979 2016 D DmServiceSms: saveImsSelfRegInfo, slot = 0
06-18 10:20:39.635 1979 2016 D DmServiceSms: Save sim1 LTE imsi:460110130472497 ---- IMS 短信自注册相关
信息保存到终端
06-18 10:20:39.635 1979 2016 D DmServiceSms: Ims sms self register success! ----IMS 短信自注册流程成功
06-18 10:20:39.637 1979 1979 D DmServiceSms: DmSmsService onDestroy
```

Unisoc Confidential For hiar

3 电信移动终端自注册各平台适配

3.1 电信移动终端 Mocor5 平台自注册模块适配

3.1.1 代码路径

Mocor5 平台电信 DM 自注册模块代码路径：

vendor/sprd/operator/ctcc/packages/app/loTSelfRegister

3.1.2 DM 版本号

数据自注册上报的 DM 版本号，需在如下文件中配置对应 register_version 字段：

vendor/sprd/operator/ctcc/packages/app/loTSelfRegister/res/values/config.xml

```
<string name="register_version" translatable="false">8.0</string>
```

版本号 8.0 取自电信白皮书《中国电信移动终端需求白皮书-数据自注册功能分册（2018.V2）20190111174357531.pdf》。若后续电信白皮书有升级，请以最新白皮书说明为准。

3.1.3 厂商名称、手机型号

数据自注册上报的厂商名称，需在如下文件中配置对应 manu_id 字段：

vendor/sprd/operator/ctcc/packages/app/loTSelfRegister/res/values/config.xml

```
<string name="manu_id" translatable="false">厂商代码（客户根据项目厂商配置）</string>
```

手机型号通过 Build.MODEL 属性配置，即在对应项目的 mk 文件中配置：

```
PRODUCT_MODEL := 手机型号（客户根据项目型号配置）
```

3.1.4 软件版本号、系统版本号

数据自注册上报的软件版本号通过 ro.version.software 属性配置，以 SC9820E Mocor5 平台为例，在 device/sprd/sharkle/sp9820e_2h10/system.prop 中配置：

```
ro.version.software=软件版本号（需要客户根据项目的软件版本配置）
```

SC9820E Mocor5 平台系统版本号在如下文件中通过 DEF_FIRMWARE_VERSION 配置：

vendor/sprd/operator/ctcc/packages/app/loTSelfRegister/src/com/sprd/dm/selfreg/DmMessage.java

```
private static final String DEF_FIRMWARE_VERSION = "Mocor 1.0（默认）"
```

3.1.5 上报字段配置

数据自注册去掉电信 V2 版本可选字段，保留必选字段。单卡终端必选字段如下：

vendor/sprd/operator/ctcc/packages/app/loTSelfRegister/res/values/arrays.xml

```
<string-array name="singlesim self register message string" translatable="false" >
    <item>REGVER</item>
    <item>UETYPE</item>
    <item>MEID</item>
    <item>MODEL</item>
    <item>SWVER</item>
    <item>OSVER</item>
    <item>MACID</item>
    <item>IMEI1</item>
    <item>IMEI2</item>
    <item>DATASIM</item>
    <item>SIM1ICCID</item>
    <item>SIM1LTEIMSI</item>
    <item>SIM1CDMAIMSI</item>
    <item>SIM1CELLID</item>
    <item>SIM2CDMAIMSI</item>
    <item>SIM2ICCID</item>
    <item>SIM2LTEIMSI</item>
    <item>SIM2CELLID</item>
</string-array>
```

双卡终端必选字段如下:

```
<string-array name="dualsim self register message string" translatable="false" >
    <item>REGVER</item>
    <item>UETYPE</item>
    <item>MEID</item>
    <item>MODEL</item>
    <item>SWVER</item>
    <item>OSVER</item>
    <item>MACID</item>
    <item>IMEI1</item>
    <item>IMEI2</item>
    <item>DATASIM</item>
    <item>SIM1ICCID</item>
    <item>SIM1LTEIMSI</item>
    <item>SIM1CDMAIMSI</item>
    <item>SIM1CELLID</item>
    <item>SIM2CDMAIMSI</item>
    <item>SIM2ICCID</item>
    <item>SIM2LTEIMSI</item>
    <item>SIM2CELLID</item>
</string-array>
```

3.2 电信 LTE 移动终端 Android 9.0 平台自注册模块适配

3.2.1 代码路径

Android 9.0 平台电信 DM 自注册模块代码路径:

vendor/sprd/feature_configs/carriers/ctcc/apps/loTSelfRegister

3.2.2 DM 版本号

数据自注册上报的 DM 版本号，需在如下文件中配置对应 `register_version` 字段：

`vendor/sprd/feature_configs/carriers/ctcc/apps/IoTSelfRegister/res/values/config.xml`

```
<string name="register_version" translatable="false">8.0</string>
```

版本号 8.0 取自电信白皮书《中国电信移动终端需求白皮书-数据自注册功能分册（2018.V2）20190111174357531.pdf》。若后续电信白皮书有升级，请以最新白皮书说明为准。

3.2.3 厂商名称、手机型号

数据自注册上报的厂商名称，需在如下文件中配置对应 `manu_id` 字段：

`vendor/sprd/feature_configs/carriers/ctcc/apps/IoTSelfRegister/res/values/config.xml`

```
<string name="manu_id" translatable="false">厂商代码（客户根据项目厂商配置）</string>
```

手机型号通过 `Build.MODEL` 属性配置，即在对应项目的 `mk` 文件中配置：

```
PRODUCT_MODEL := 手机型号（客户根据项目型号配置）
```

3.2.4 软件版本号、系统版本号

数据自注册上报的软件版本号，需通过 `ro.version.software` 属性配置，可以在 `system.prop` 中配置：

```
ro.version.software=软件版本号（需要客户根据项目的软件版本配置）
```

9.0 平台电信自注册系统版本号通过属性 `ro.build.version.release` 加前缀 “android” 配置，默认为 “android9”，一般情况下不需要修改。

3.2.5 CDMA 短信自注册需要合入 CP MEID 写 CSIM 卡的修改

同一张 SIM 卡，在测试机中 CDMA 短信自注册成功，取出并放入配合终端（辅助测试机）成功完成 CDMA 短信自注册后，再次将该 SIM 卡放入测试机对应卡槽，不能发起自注册。

电信 CDMA 短信自注册 MEID 写 CSIM 卡流程跟 3gpp2 协议冲突，无法通过泰尔实验室 CDMA 认证。因此当前 CP 的这项修改未合入量产分支。

如何获取修改：请提 CQ 申请，提供专门的 Modem 版本。

合入建议：如果不做泰尔实验室认证，则直接合入 CP 修改就可以。如果需要过泰尔实验室认证，建议先过泰尔实验室认证，再合入 CP 修改过电信广院 CDMA 短信自注册认证。

3.3 电信 5G 移动终端 Android 9.0 平台送测说明

以版本 MOCORDROID9.0_Trunk_19B_ROC1_ORCA_W19.51.5 为例说明 5G 终端的数据自注册情况。

3.3.1 代码路径

Android 9.0 平台电信 DM 自注册模块代码路径：

`vendor/sprd/feature_configs/carriers/ctcc/apps/IoTSelfRegister`

3.3.2 DM 版本号

数据自注册上报的 DM 版本号，需在如下文件中配置对应 `register_version` 字段：

`vendor/sprd/feature_configs/carriers/ctcc/apps/IoTSelfRegister/res/values/config.xml`

```
<string name="register_version" translatable="false">8.0</string>
```

版本号 **8.0** 取自电信白皮书《中国电信 5G 全网通终端需求白皮书（试行）-数据自注册功能分册.doc》。若后续电信白皮书有升级，请以最新白皮书说明为准。

3.3.3 厂商名称、手机型号

数据自注册上报的厂商名称，需在如下文件中配置对应 `manu_id` 字段：

`vendor/sprd/feature_configs/carriers/ctcc/apps/IoTSelfRegister/res/values/config.xml`

```
<string name="manu_id" translatable="false">厂商代码（客户根据项目厂商配置）</string>
```

手机型号通过 `Build.MODEL` 属性配置，即在对应项目的 `mk` 文件中配置：

```
PRODUCT_MODEL := 手机型号（客户根据项目型号配置）
```

3.3.4 软件版本号、系统版本号

数据自注册上报的软件版本号，需通过 `ro.version.software` 属性配置，可以在 `system.prop` 中配置：

```
ro.version.software=软件版本号（需要客户根据项目的软件版本配置）
```

9.0 平台电信自注册系统版本号通过属性 `"ro.build.version.release"` 加前缀 `"android"` 配置。默认为 `"android9"`，一般情况下不需要修改。

3.3.5 上报字段配置

5G 终端数据自注册上报字段的配置：

```
<string-array name="dualsim self register message string" translatable="false" >
    <item>REGVER</item>
    <item>UETYPE</item>
    <item>MEID</item>
    <item>MODEL</item>
    <item>SWVER</item>
    <item>OSVER</item>
    <item>MACID</item>
    <item>IMEI1</item>
    <item>IMEI2</item>
    <item>DATASIM</item>
    <item>SIM1ICCID</item>
    <item>SIM1LTEIMSI</item>
    <item>SIM1CDMAIMSI</item>
    <item>SIM1CELLID</item>
    <item>SIM2CDMAIMSI</item>
    <item>SIM2ICCID</item>
    <item>SIM2LTEIMSI</item>
    <item>SIM2CELLID</item>
    <item>NETWORKTYPE</item> //该字段为 5G 终端新增字段
</string-array>
```

3.3.6 测试说明

支持 5G 的终端需要同时支持 5G 自注册和 LTE 自注册。两者的主要区别为网络类型的差异。所以在进行 5G 自注册时，需要保证 5G 网络稳定。否则即使 LTE 类型上报成功，仍然被认为对应 case Fail。

以 UDS710_UDX710 平台为例，分析 5G 终端电信数据自注册 log，方式如下：

- 在 Android log 中的 main log 中搜索关键字 DmService，查看当前获得的终端信息。

```
DmService: REGVER = 8.0
DmService: UETYPE = 5 // 5g 终端 type 类型
DmService: getSlotCount, count = 2
DmService: getImei, imei = 866806040001334
DmService: MEID = 866806040001334//不支持 CDMA 终端默认为 IMEI1
DmService: getModel, modelInfo = HS-HNR550T
DmService: MODEL = HS-HNR550T
DmService: getSwVer, swVer = L1707.6.01.01.00
DmService: SWVER = L1707.6.01.01.00
DmService: OSVER = android9
DmService: getWifiMac, macAddress = 02:00:00:00:00:00
DmService: ReadWifiMacFromFile: strFilePath = /mnt/vendor/wifimac.txt
DmService: MACID = //以实际获取值为准
DmService: getSlotCount, count = 2
DmService: getImei, imei = 866806040001334
DmService: IMEI1 = 866806040001334
DmService: getSlotCount, count = 2
DmService: getImei, imei = 866806040001342
DmService: IMEI2 = 866806040001342
DmService: DATASIM = 1
DmService: SIM1ICCID = 89861119140211203055
DmService: getSimLteImsi, lteImsi = 460110120300115
DmService: SIM1LTEIMSI = 460110120300115
DmService: SIM1CDMAIMSI = //不支持 CDMA 终端为空
DmService: getSimCellId, slotId = 0
DmService: getSimCellId, simcellid lte = 95434753
DmService: SIM1CELLID = 95434753
DmService: SIM2CDMAIMSI = //不支持 CDMA 终端为空
DmService: SIM2ICCID = 898600810917F7045227
DmService: getSimLteImsi, lteImsi = 460007753138460
DmService: SIM2LTEIMSI = 460007753138460
DmService: getSimCellId, slotId = 1
DmService: getSimCellId, simcellid lte = 13668177
DmService: SIM2CELLID = 13668177
DmService: getNetworkType, type = 5
DmService: NETWORKTYPE = 5 //如果网络类型为 5g，则该值为 5，类型为 LTE，则该值为 4.
```

- 终端收到服务器响应。

```
URL=http://zzhc.vnet.cn
Content-Length: 588
Content-Type: application/encrypted-json
Content-Type: text/plain; charset=ISO-8859-1 (assuming UTF-8)
finally response is{"resultCode":"0","resultDesc":"Success"}----终端收到服务器网络的响应
```

3.4 电信 LTE 移动终端 Android 10.0 平台自注册模块适配

3.4.1 代码路径

Android 10.0 平台电信 DM 自注册模块代码路径：

vendor/sprd/carriers/ctcc/packages/apps/SelfRegister/

3.4.2 DM 版本号

数据自注册上报的 DM 版本号，需在如下文件中配置对应 register_version 字段：

vendor/sprd/carriers/ctcc/packages/apps/SelfRegister/res/values/config.xml

```
<string name="register_version" translatable="false">8.0</string>
```

版本号 8.0 取自电信白皮书《中国电信移动终端需求白皮书-数据自注册功能分册（2018.V2）20190111174357531.pdf》。若后续电信白皮书有升级，请以最新白皮书说明为准。

3.4.3 厂商名称、手机型号

数据自注册上报的厂商名称，需在如下文件中配置对应 manu_id 字段：

vendor/sprd/carriers/ctcc/packages/apps/SelfRegister/res/values/config.xml

```
<string name="manu_id" translatable="false">厂商代码（客户根据项目厂商配置）</string>
```

手机型号通过 Build.MODEL 属性配置，即在对应项目的 mk 文件中配置：

```
PRODUCT_MODEL := 手机型号（客户根据项目型号配置）
```

3.4.4 软件版本号、系统版本号

数据自注册上报的软件版本号，需通过 ro.version.software 属性配置，可以在 product.prop 中配置：

```
ro.version.software=软件版本号（需要客户根据项目的软件版本配置）
```

10.0 平台电信自注册系统版本号通过属性"ro.build.version.release"加前缀“android”配置，默认为“android10”，一般情况下不需要修改。

3.4.5 CDMA 设备关闭开机存储 MEID 功能

电信 CDMA 短信自注册 MEID 写 CSIM 卡流程跟 3gpp2 协议冲突，因此支持 CDMA 的终端在送测之前，需要先进入工程模式（unisoc 默认工程模式暗码：*##*83781#*##*），在“TELEPHONY”TAB 页中将“Store Meid At Sim Init”开关关闭并恢复出厂设置。

3.4.6 电信自注册功能开关

电信工程下（工程 mk 配置 PRODUCT_REVISION := ctcc），通过工程模式可开启或关闭电信自注册功能：

进入工程模式（unisoc 默认工程模式暗码：*##*83781#*##*），在“TELEPHONY”TAB 页中通过“电信自注册功能开关”项设置电信自注册功能开启或关闭。

3.5 电信 LTE 移动终端 Android 11.0 平台自注册模块适配

3.5.1 代码路径

Android 11.0 平台电信 DM 自注册模块代码路径：

vendor/sprd/carriers/ctcc/packages/apps/SelfRegister/

3.5.2 DM 版本号

数据自注册上报的 DM 版本号，需在如下文件中配置对应 register_version 字段：

vendor/sprd/carriers/ctcc/packages/apps/SelfRegister/res/values/config.xml

```
<string name="register_version" translatable="false">8.0</string>
```

版本号 8.0 取自电信白皮书《中国电信移动终端需求白皮书-数据自注册功能分册（2018.V2）20190111174357531.pdf》。若后续电信白皮书有升级，请以最新白皮书说明为准。

3.5.3 厂商名称、手机型号

数据自注册上报的厂商名称，需在如下文件中配置对应 manu_id 字段：

vendor/sprd/carriers/ctcc/packages/apps/SelfRegister/res/values/config.xml

```
<string name="manu_id" translatable="false">厂商代码（客户根据项目厂商配置）</string>
```

手机型号通过 Build.MODEL 属性配置，即在对应项目的 mk 文件中配置：

```
PRODUCT_MODEL := 手机型号（客户根据项目型号配置）
```

3.5.4 软件版本号、系统版本号

数据自注册上报的软件版本号，需通过 ro.version.software 属性配置，可以在工程 mk 中配置：

```
PRODUCT_PROPERTY_OVERRIDES += \
    ro.version.software=软件版本号（需要客户根据项目的软件版本配置）
```

11.0 平台电信自注册系统版本号通过属性"ro.build.version.release"加前缀“android”配置，默认为“android11”，一般情况下不需要修改。

3.5.5 CDMA 设备关闭开机存储 MEID 功能

电信 CDMA 短信自注册 MEID 写 CSIM 卡流程跟 3gpp2 协议冲突，因此支持 CDMA 的终端在送测之前，需要先进入工程模式（unisoc 默认工程模式暗码：*##83781#*），在“TELEPHONY”TAB 页中将“Store Meid At Sim Init”开关关闭并恢复出厂设置。

3.5.6 电信自注册功能开关

电信工程下（工程 mk 配置 PRODUCT_REVISION := ctcc），通过工程模式可开启或关闭电信自注册功能：

进入工程模式（unisoc 默认工程模式暗码：*##83781#*），在“TELEPHONY”TAB 页中通过“电信自注册功能开关”项设置电信自注册功能开启或关闭。

4

电信移动终端自注册 FAQ

Q1: 电信正式送测出现测试 case fail 的情况，如何处理？

A1: 分为 2 种情况：

- 本地可以复现的问题，请自测试抓取 log 分析原因。
- 本地不能复现的问题，只能申请电信抓 log 或者复测。

电信入库测试过程中只提供结果，不提供 log。在一轮测试完成之后，有一次申请复测或者抓 log 的机会：

- 如果问题自测试实在无法复现，且认为终端没有问题，可以申请电信复测。
- 如果确认终端大概率有问题，则可以申请电信抓 log 分析问题。

Q2: 在哪些情况下电信正式送测的终端在电信测试平台上查不到数据自注册信息？

A2: 可能有如下几种情况：

- 电信 4G（不支持 CDMA）终端，由于自注册上报的信息中关键字段 CDMAIMSI 为空，所以在电信的测试平台上无法查到终端上报的数据域自注册信息。这种情况需要请电信测试人员帮忙发邮件给电信服务器部门查询。
- 电信 4G（支持 CDMA）终端，如果上报的 CDMAIMSI 字段错误，在电信的测试平台上也查不到终端上报的数据域自注册信息。
- 终端由于网络原因，发送数据域自注册失败，在电信的测试平台上也查不到终端上报的数据域自注册信息。遇到这种情况，可以在网络好的时候申请复测。
- 在使用其他运营商的卡做 WIFI 热点时，终端上报自注册概率性遇到服务器返回 403 错误。建议电信使用宽带路由 WIFI 测试，尽量不要使用移动数据承载的 WIFI 热点。
- 服务器返回 503 错误。这种情况下，电信服务器不可用。可以使用其他平台对比机做对比测试。遇到这种情况，只能申请其他时段复测。
- 电信测试规范《中国电信移动终端测试方法全集-数据自注册功能分册 2018.V220190111100843520.pdf》中 7.1.4 TC-MF_DATAREG-20104 [必选] 数据自注册功能要求测试（切换主副卡、双卡终端）（测试集选择码：111011）：

Step 12: 在不关机的情况下，拔出卡槽一的电信卡且打开终端 WIFI 数据连接，在 WIFI 数据连接开启的情况下，检查自注册平台上是否有该终端的自注册信息（主卡为中国移动 LTE 卡 B）；

这条 case，由于上报信息中只有移动 SIM 卡的信息，没有 CDMAIMSI 字段，所以在电信的测试平台上无法查到终端上报的数据自注册信息。需要请电信测试人员帮忙发邮件给电信服务器部门查询。

Q3: 电信白皮书中提到的“终端处于国际漫游状态”具体有哪些情况？

A3: 只考虑电信卡是否处于国际漫游状态。

Q4: SC9820E Mocor5 平台电信数据自注册上报 MEID 字段为什么是 15 位，而不是 14 位？

A4: SC9820E Mocor5 平台支持电信 4G 但不支持 CDMA，所以没有 MEID（14 字节），电信要求 MEID 以 15 个字节的 IMEI1 内容上报电信自注册服务器。

其中 CI 为需要查询的 cellid，LAC/TAC 可以从 Android log 中找到。

电信 4G:

```
R007B18 05-21 14:23:19.004 962 1086 D RadioInteractor: [3689]< REQUEST GET CELL INFO LIST
[CellInfoLte:{mRegistered=YES mTimeStampType=oem ril mTimeStamp=36521464704ns
mCellConnectionStatus=2147483647 CellIdentityLte:{ mCi=226848129 mPci=385 mTac=30040 mEarfcn=100
CellSignalStrengthLte: ss=50 rsrp=-90 rsrq=-24 rssnr=2147483647 cqi=2147483647 ta=2147483647}}] [SUB1]
```

联通卡:

```
R00B514 05-21 14:50:51.233 957 1344 D NetworkRegistrationManager:
onGetNetworkRegistrationStateComplete result 0 state NetworkRegistrationState{transportType=1
domain=CS regState=HOME accessNetworkTechnology=LTE reasonForDenial=0 emergencyEnabled=false
supportedServices=[I@c7177f cellIdentity=CellIdentityLte:{ mCi=100776195 mPci=-1 mTac=9477 mEarfcn=-1
mBandwidth=2147483647 mMcc=null mMnc=null mAlphaLong=null mAlphaShort=null}
voiceSpecificStates=VoiceSpecificRegistrationStates
```

Q5: 电信入库测试之前自注册模块需要检查哪些信息？

A5: 主要检查如下几个方面：

- 终端基本功能正常，驻网状态正常，开关机正常，不会出现 Modem Assert 异常。
- *#0000#界面、设置界面、自注册上报的相关信息（MEID、软件版本号、硬件版本号、操作系统版本号、CELLID 等信息）要保持一致。
- *#0000#界面检查：
 - MEID
 - IMEI1、IMEI2
 - CELLID
 - LTEIMSI
 - 软件版本号
 - 硬件版本号
 - 终端型号
 - 操作系统版本号
- 设置界面检查：
 - 软件版本号
 - 硬件版本号
 - 终端型号
 - 操作系统版本号
 - WIFI MAC
- 自注册模块 log 检查：
 - MEID

- IMEI1、IMEI2
- CELLID
- LTEIMSI、CDMAIMSI
- WIFI MAC
- 软件版本号
- 终端型号
- 操作系统版本号

Q6: 如何关闭 Mocor5 平台 IMS 短信自注册功能?

A6: 如下文件中配置 dm_ims_sms 字段为 false:

vendor/sprd/operator/ctcc/packages/app/loTSelfRegister/res/values/config.xml

```
<!-- ims sms switch ,default on,if false,switch off.VALEU[true,false] -->
<string name="dm_ims_sms" translatable="false">>false</string>
```

Q7: 如何关闭 UMS312 Android 9.0 平台 IMS 短信自注册功能?

A7: 如下文件中配置 dm_ims_sms 字段为 false:

vendor/sprd/feature_configs/carriers/ctcc/apps/loTSelfRegister/res/values/config.xml

```
<!-- ims sms switch ,default on,if false,switch off.VALEU[true,false] -->
<string name="dm_ims_sms" translatable="false">>false</string>
```

Unisoc Confidential For hiar

5

参考文献

以下参考文献均为电信正式发布的文档，可以通过电信获取。

- 《4-中国电信移动终端需求白皮书-数据自注册功能分册（2018.V2）20190111174357531.pdf》
- 《中国电信移动终端测试方法全集-数据自注册功能分册 2018.V220190111100843520.pdf》
- 《3-中国电信移动终端需求白皮书-短信自注册功能分册（2019.V1_20190314）.docx》
- 《中国电信移动终端测试方法全集-短信自注册功能分册 2018.V220190111100854593.pdf》

Unisoc Confidential For hiar