# 任务目标

通过dialer code 启动app。针对本次具体任务，即为通过##2886#启动SFT APP(用于手机各项软硬件的检测)

此任务的一个block点在于，dialer code只支持Windows Phone Silverlight 8.0 app，而SFT APP是UWP app。所以只能通过dialer code启动一个Silverlight 8.0 app(DialerAppSimple),然后在此app中启动SFT APP。

# 任务思路

1. 将SFT APP预制到FFU里面
2. 将dialerAPPSimple预制到FFU里面
3. Dialer Code 关联dialerAPPSimple
4. dialerAPPSimple启动SFT APP

# 文件夹内容介绍

**SFT\_mobile\_v1.0.8.0 ：** SFT生成好的spkg，还有预制需要的xml，可以直接用

SystemFunctionalTest\_SampleCode-v1.0.8.0.zip：SFT源码，需要安装，并且有说明文档

**SFT UWP mobile v 1.0.8.pdf ：** SFT项目的说明文档，有如何push SFT APP到手机，如何将SFT编译进FFU，如何引用SFT的URI等。

**SFT UWP mobile v 1.0.8\_CHS.pdf：** 与上同，部分已经翻译成中文。

**Example.DialerAppSimple.zip ：** 一个Silverlight 8.0 APP的源码，用来启动SFT APP

**OEM\_App\_SelfSign\_Extras.zip ：** 打包成spkg需要用到的文件

**ReferenceDocument ：** 过程中用到的文件，修改的文件

# 完成步骤

## 绑定SFT URI生成可执行文件

在DialerAppSimple中绑定SFT的URI：将代码里的oem-factory-tool 换成SFT的URI(oem-tool-sft)

用VS生成.xap文件。(VS2015,或是VS2013 update4)

## sign可执行文件

参考文档：Presubmission testing for preloaded applications

准备文件：

解压缩后：

步骤：

1. 先sign下环境（以后此步骤可省，详情可参考Set up the signing environment）

执行"%WPDKCONTENTROOT%\tools\bin\i386\InstallOEMCerts.cmd"

可通过Certmgr.msc，查看是否生成Personal->certificates->oem hal test cert…

1. 将DialerAppSimple\_Debug\_ARM.xap改为DialerAppSimple\_Debug\_ARM.zip,然后解压缩
2. Sign解压后的DialerAppSimple.dll

管理员权限打开控制台：

set WPDKCONTENTROOT=C:\Program Files (x86)\Windows Kits\10

set PATH=%PATH%;%WPDKCONTENTROOT%\Tools\bin\i386

set SIGN\_WITH\_TIMESTAMP=1

Sign.cmd –oem –app DialerAppSimple.dll

1. 把WMAppPRHeader.xml拷贝到解压目录下，然后一起打包并改名成DialerAppSimple\_Debug\_ARM.xap
2. sign下打包好的xap文件

Sign.cmd –oem –app DialerAppSimple\_Debug\_ARM.xap

## 生成spkg包

参考文档：Creating packages

参考文件：ReferenceDocument\DialerApp\

备注：此为发布test版本的步骤，此过程用到的license为一个通用的test license，如果要发布正式版本，需要将xap文件上传至微软指定地址得到唯一对应的license。

生成spkg包需要3个文件：

1. xap文件，上一个步骤产生，sign之后的
2. provxml文件，这个是自己创建的，第一个guid是productId，可在.xap解压出来的WMAppManifest.xml中找到，后面两个guid任意制定

参考Create a preloaded application->Preload of a .XAP package

1. license.xml,拷贝前面提到的TestApp\_License.xml

三个文件准备好后，创建一个APPname.pkg.xml文件，在里面指定上面三个文件的路径。

可以参考ReferenceDocument\DialerApp\ DialerAppSimple.pkg.xml

用pkggen生成spkg包，可参考ReferenceDocument\DialerApp\ ReferenceDocument\DialerApp\ makecfgpkg.cmd，详细用法请参考PkgGen.exe command-line options

## 编译到ffu

1. 在 \WP\prebuilt\8909\FMFiles\QCFM.xml文件中为dialSImple和SFT制定feature ID。加在oem标签下。
2. WP\prebuilt\8909\OEMInputTemplates\oeminput\_TestOS.xml中加入刚添加的feature ID。

需要在执行下QWPCT打包生成output目录下的oeminput\_TestOS.xml

可以参考ReferenceDocument目录下的对应文件，添加的feature ID为 DIAGNOSTIC\_APP\_PRELOAD和SFT\_APP\_PRELOAD，也可以新建类似于QCFM的xml文件，然后import到testos.xml文件里

## 制定dialer code

dialer code必须以##开头

在WP\prebuilt\8909\OEMCustomization\目录下新建SystemFunctionalTest.xml文件，然后import到QCOEMCustomization.xml文件中，当然也可以直接在后一个文件中修改，参考ReferenceDocument\SystemFunctionalTest.xml。

其中，PartnerAppTaskUri的value中的guid为第二个步骤中解压xap中生成的WMAppManifest.xml文件中的ProductID，后面的default为TaskName。

PartnerAppProvisioningFilePath的value为第三个步骤中DialerAppSimple.pkg.xml制定的provxml的DestinationDir。

PartnerImmediateDialString即为制定的dialer code。

文件QCOEMCustomization.xml也叫作Customization answer file

参考文档：Dialer codes to launch diagnostic applications. step 5 ~step 9

Customization answer file

## 生成FFU

用现在的命令即可

# 隐藏APP icon

在应用列表中隐藏DialerAPP和SFT可以通过下面的方法实现：

 隐藏DialerAPP如附件DialerAppSimple.pkg.xml文件中<!--hide DialerApp from APP list-->以下的代码；

 隐藏SFT如附件SFTConfig.oem.pkg.xml文件中<!--hide SFT from APP list-->以下代码

（ReferenceDocument目录下也有）



# 参考资料

参考文档：Dialer codes to launch diagnostic applications

参考目录：\WP\prebuilt\8909\AddOnPackages

文中的参考文档均来自：WP\_Blue\_GDR\_Documentation.chm

2015年11月17日 V1.0 myang T:67219