System Functional Test on Windows 10 Mobile (UWP app version)

Oct 2015

System Functional Test on Windows 10 Mobile (UWP app version)

Microsoft Corporation Technical Documentation License Agreement (Standard)
READ THIS! THIS IS A LEGAL AGREEMENT BETWEEN MICROSOFT CORPORATION ("MICROSOFT") AND THE RECIPIENT OF THESE MATERIALS, WHETHER AN INDIVIDUAL OR AN ENTITY ("YOU"). IF YOU HAVE ACCESSED THIS AGREEMENT IN THE PROCESS OF DOWNLOADING MATERIALS ("MATERIALS") FROM A MICROSOFT WEB SITE, BY CLICKING "I ACCEPT", DOWNLOADING, USING OR PROVIDING FEEDBACK ON THE MATERIALS, YOU AGREE TO THESE TERMS. IF THIS AGREEMENT IS ATTACHED TO MATERIALS, BY ACCESSING, USING OR PROVIDING FEEDBACK ON THE ATTACHED MATERIALS, YOU AGREE TO THESE TERMS.

1. For good and valuable consideration, the receipt and sufficiency of which are acknowledged, You and Microsoft agree as follows:

(a) If You are an authorized representative of the corporation or other entity designated below ("**Company**"), and such Company has executed a Microsoft Corporation Non-Disclosure Agreement that is not limited to a specific subject matter or event ("**Microsoft NDA**"), You represent that You have authority to act on behalf of Company and agree that the Confidential Information, as defined in the Microsoft NDA, is subject to the terms and conditions of the Microsoft NDA and that Company will treat the Confidential Information accordingly;

(b) If You are an individual, and have executed a Microsoft NDA, You agree that the Confidential Information, as defined in the Microsoft NDA, is subject to the terms and conditions of the Microsoft NDA and that You will treat the Confidential Information accordingly; or

(c)If a Microsoft NDA has not been executed, You (if You are an individual), or Company (if You are an authorized representative of Company), as applicable, agrees: (a) to refrain from disclosing or distributing the Confidential Information to any third party for five (5) years from the date of disclosure of the Confidential Information by Microsoft to Company/You; (b) to refrain from reproducing or summarizing the Confidential Information; and (c) to take reasonable security precautions, at least as great as the precautions it takes to protect its own confidential information, but no less than reasonable care, to keep confidential the Confidential Information. You/Company, however, may disclose Confidential Information in accordance with a judicial or other governmental order, provided You/Company either (i) gives Microsoft reasonable notice prior to such disclosure and to allow Microsoft a reasonable opportunity to seek a protective order or equivalent, or (ii) obtains written assurance from the applicable judicial or governmental entity that it will afford the Confidential Information the highest level of protection afforded under applicable law or regulation. Confidential Information shall not include any information, however designated, that: (i) is or subsequently becomes publicly available without Your/Company's breach of any obligation owed to Microsoft; (ii) became known to You/Company prior to Microsoft's disclosure of such information to You/Company pursuant to the terms of this Agreement; (iii) became known to You/Company from a source other than Microsoft other than by the breach of an obligation of confidentiality owed to Microsoft; or (iv) is independently developed by You/Company. For purposes of this paragraph, "Confidential Information" means nonpublic information that Microsoft designates as being confidential or which, under the circumstances surrounding disclosure ought to be treated as confidential by Recipient. "Confidential Information" includes, without limitation, information in tangible or intangible form relating to and/or including released or unreleased Microsoft software or hardware products, the marketing or promotion of any Microsoft product, Microsoft's business policies or practices, and information received from others that Microsoft is obligated to treat as confidential.

2. You may review these Materials only (a) as a reference to assist You in planning and designing Your product, service or technology ("Product") to interface with a Microsoft Product as described in these Materials; and (b) to provide feedback on these Materials to Microsoft. All other rights are retained by Microsoft; this agreement does not give You rights under any Microsoft patents. You may not (i) duplicate any part of these Materials, (ii) remove this agreement or any notices from these Materials, or (iii) give any part of these Materials, or assign or otherwise provide Your rights under this agreement, to anyone else.

3. These Materials may contain preliminary information or inaccuracies, and may not correctly represent any associated Microsoft Product as commercially released. All Materials are provided entirely "AS IS." To the extent permitted by law, MICROSOFT MAKES NO WARRANTY OF ANY KIND, DISCLAIMS ALL EXPRESS, IMPLIED AND STATUTORY WARRANTIES, AND ASSUMES NO LIABILITY TO YOU FOR ANY DAMAGES OF ANY TYPE IN CONNECTION WITH THESE MATERIALS OR ANY INTELLECTUAL PROPERTY IN THEM.

4. If You are an entity and (a) merge into another entity or (b) a controlling ownership interest in You changes, Your right to use these Materials automatically terminates and You must destroy them.

5. You have no obligation to give Microsoft any suggestions, comments or other feedback ("Feedback") relating to these Materials. However, any Feedback you voluntarily provide may be used in Microsoft Products and related specifications or other documentation (collectively, "Microsoft Offerings") which in turn may be relied upon by other third parties to develop their own Products. Accordingly, if You do give Microsoft Feedback on any version of these Materials or the Microsoft Offerings to which they apply, You agree: (a) Microsoft may freely use, reproduce, license, distribute, and otherwise commercialize Your Feedback in any Microsoft Offering; (b) You also grant third parties, without charge, only those patent rights necessary to enable other Products to use or interface with any specific parts of a Microsoft Product that incorporate Your Feedback; and (c) You will not give Microsoft any Feedback (i) that You have reason to believe is subject to any patent, copyright or other intellectual property claim or right of any third party; or (ii) subject to license terms which seek to require any Microsoft Offering incorporating or derived from such Feedback, or other Microsoft intellectual property, to be licensed to or otherwise shared with any third party.

6. Microsoft has no obligation to maintain confidentiality of any Microsoft Offering, but otherwise the confidentiality of Your Feedback, including Your identity as the source of such Feedback, is governed by Your NDA.

7. This agreement is governed by the laws of the State of Washington. Any dispute involving it must be brought in the federal or state superior courts located in King County, Washington, and You waive any defenses allowing the dispute to be litigated elsewhere. If there is litigation, the losing party must pay the other party's reasonable attorneys' fees, costs and other expenses. If any part of this agreement is unenforceable, it will be considered modified to the extent necessary to make it enforceable, and the remainder shall continue in effect. This agreement is the entire agreement between You and Microsoft concerning these Materials; it may be changed only by a written document signed by both You and Microsoft.

MICROSOFT LAW AND CORPORATE AFFAIRS
LCA

Contents

Change history

Version 1.0.6

Description
Document created (app for mobile beta version).

Version 1.0.8

Description
Update to conform the changes made in SFT mobile v.1.0.8.0 release (this is still a beta version for mobile). 1. Provide spkg to preload appx. 2. Change appx publisher name and self-signing key to "Windows Phone OEM App Test Cert 2013 (TEST ONLY)". 3. Enable SFT app for URI association, protocol = "oem-tool-sft". An example mobile use case is, a dialer app may use this URI to launch SFT app. 4. Add IMEI, Cellular technology info, Mobile data class info in SIM test page. 5. Add auto-pass configuration for SIM test by setup SIM status filter. 6. Add WLAN MAC, BT MAC, IMEI, Cellular technology info, Mobile data class info in system information page. 7. Add phone dialer test item. 8. Enable FM Radio test for SOC 8909/8994/8992 (for windows phone devices which are upgraded from WP8.1 might be failed to run this test). 9. Fix bug for system information page to show OS Version (<major>.<minor>.<QFE>.<build>). 10. Fix bug for Earpiece test item to ensure audio captured from handset mic and ensure audio output to handset speaker.

Overview

System Functional Test UWP app (abbreviated as SFT) is a sample application Microsoft developed to support partners' Windows 10 system manufacturing process. The goal of SFT is to demonstrate an integrated application to run on Windows 10 OS. Partners can use SFT as a reference for their manufacturing functional test tool.

In this document, we will provide details about the contents in the released package, the supported test items and test configurations.

This document describes the 1.0.8 version of the SFT. Before version 1.0, SFT support only desktop OS. Since version 1.0.6, SFT supports mobile OS as well, version 1.0.8.0 is still the beta version for mobile.

This document describes SFT functions for mobile only. All the functions of SFT and how SFT will be released will be subject to change in the future.

Contents

SFT consists of 3 components – application, service and configuration. When you get the zip file that contains system functional test binaries for mobile, please extract the zip file, and find the installation resources for each component as described below:

- **SystemFunctionalTest_1.0.8.0_ARM_Test** (folder) – The application package folder contains SystemFunctionalTest App package (appx) and its dependency packages.
- **OEMName.DiagTool.SystemFunctionalTest.spkg** – The spkg to preload SFT application at image build time.
- **OEMName.DiagTool.SystemTestService.spkg** – A service program used by the SystemFunctionalTest app to support hardware access functions.
    - Microsoft.Temp.VC140Runtime.spkg - This spkg contains VC runtime dlls that will be referenced by service, and has to be installed once per test device. This spkg might not be needed in later release.
- **OEMName.DiagTool.SFTConfig.spkg** – A package which contains only one configuration file (SFTConfig.xml) for the application to use.
- **PreloadPackages.xml** – an example feature manifest file to use in image build to preload the above spkg files.
- **SFTConfig.xml** – a configuration sample to define the test items.
- **SFTConfig.oem.pkg.xml** – a package project XML file for OEMName.DiagTool.SFTConfig.spkg.
- **Makecfgpkg.cmd** – a command line script that use SFTConfig.oem.pkg.xml and SFTConfig.xml to generate OEMName.DiagTool.SFTConfig.spkg.

System Functional Test on Windows 10 Mobile (UWP app version)

These SFT binaries are built with Visual Studio 2015 RTM version and Windows 10 SDK build 10240. The application (.appx) were signed with OEM test certificate (“*Windows Phone OEM App Test Cert 2013 (TEST ONLY)*”) which is provided in WDK installation.

These sample packages and executables were signed with OEM test certificates (“*Windows Phone OEM Test Cert 2013*”).

For more technical details and references which are not elaborated in this document, refer to *Windows 10 partner documentation*.

System Functional Test (SFT)

SFT is a UWP (Universal Windows Platform) app supporting functional tests while running full Windows OS. Some test items require lower level driver accessing to validate hardware functions. These kinds of hardware functions are restricted and prohibited from being accessed at UWP app level based on security policy. A native service program will be hereby presented to handle these hardware access and information queries. SFT as an application with service is not feasible to publish to store, it should be preloaded into windows image at build time or be side-loaded manually.

Function overview

- Multiple language support:
    - en-US (English)
    - zh-TW (Traditional Chinese)
    - zh-CN (Simplified Chinese)

- Basic tool functions:
    - **Auto Test**: When auto test is selected, this app will automatically start the functional tests one by one automatically until any stop condition is met.
    - **Manual Test**: There will be a page that lists all test items. Test engineer may select to run the test items one by one manually.
    - **Clear Test Result**: This function can clear the test result saved in app private space. The result will be kept until one the following conditions is met:
        - User chooses to clear test result
        - App re-install
        - System reset
    - **System Information**: Use to query system information of the device, such as OS version.
    - **Reset Phone**: Use to perform factory reset and restore the system.
    - **Shutdown Phone**: Use to shut down the phone.

System Functional Test on Windows 10 Mobile (UWP app version)

- Supported test items (25 items in total for windows 10 mobile device test):
  - Human Interface: Touch / Display / Brightness / Keypad Backlight / Keypad (Hardware Buttons, Not supported in beta)
  - Connectivity: Wi-Fi / Bluetooth
  - Card Slots: SD / SIM
  - Camera: Rear Camera / Front Camera / Flashlight
  - Radio: FM Radio (supported only for some SOC models)
  - Sensors: Compass / Accelerometer / Gyro meter / Proximity / Light / GPS
  - Audio Input / Output: Speaker+Mic / Headset / Earpiece + Handset mic
  - Misc.: Vibrator / Battery / Dialer

- Configurable test phase and test item:

  Test Item: A test item is a single test case.
  Test Phase: A test phase is a set of test items.

  In configuration file, multiple test items could be grouped in a test phase with partner defined phase name which will be shown in phase selection page during app startup. If no test phase is defined in configuration file, app will launch a default predefined phase; in this case, all test items will be included in this default phase.

  SFT has an internal default configuration. SFT also supports external configuration for partners to customize functional tests. The internal configuration will be used only if an external configuration file does not exist.

  Connect the phone to host PC (via MTP), and find the external configuration file by using file explorer:
  External configuration location: "This PC > Windows Phone > Phone > Pictures"
  External configuration file name: SFTConfig.xml

- URI association enabled:
  SFT application handles URI protocol named "**oem-tool-sft**". OEM applications could launch SFT by using this URI.
- Log:
  There will be a log generated after running test items, this log will record the start time and end time for test item that has been tested, as well as the test result.
  This log file will be saved at user public storage folder "Pictures". Each test phase will have its own log file, and log records will be appended to that file each time you run the test of that test phase.

Connect the phone to host PC (via MTP), and find the log file by using file explorer:
Location: "This PC > Windows Phone > Phone > Pictures"
Log file name: SystemFunctionalTest_<*PhaseName*>TestState.dat
(for example: `SystemFunctionalTest_AllTestState.dat`)

How to utilize SFT

This section describes how partners can utilize SFT by setup a Windows 10 mobile device and run SFT for functional tests in full OS environment.

There are 3 components of SFT – application, service, configuration file – that required to be installed to the device. These 3 parts could be installed manually during mobile OS runtime. Another option is to preload these components to mobile image during image build time, and we are not going to elaborate this option in this document, please refer to partner document for the details about how to preload spkg and appx to image.

The following installation instructions assume the contents of installation are copied to development PC at this folder:

```
C:\Tools\SFT_mobile_1.0.8.0
```

Under that folder, the installation contents could be found in following subfolder and files:

- SFT UWP App Installation Content Folder:

  `C:\Tools\SFT_mobile_1.0.8.0\SystemFunctionalTest_1.0.8.0_ARM_Test`

- SFT UWP App SPKG:

  `C:\Tools\SFT_mobile_1.0.8.0\OEMName.DiagTool.SystemFunctionalTest.spkg`

- SFT Service SPKG:

  `C:\Tools\SFT_mobile_1.0.8.0\OEMName.DiagTool.SystemTestService.spkg`

  `C:\Tools\SFT_mobile_1.0.8.0\Microsoft.Temp.VC140Runtime.spkg`

- SFT Configuration SPKG:

  `C:\Tools\SFT_mobile_1.0.8.0\OEMName.DiagTool.SFTConfig.spkg`

- SFT Configuration File and SPKG generation script:

  `C:\Tools\SFT_mobile_1.0.8.0\SFTConfig.xml`

  `C:\Tools\SFT_mobile_1.0.8.0\SFTConfig.oem.pkg.xml`

  `C:\Tools\SFT_mobile_1.0.8.0\makecfgpkg.cmd`

Prerequisites

- **A test device**: The windows 10 mobile device, usually phone device, or small screen tablet.
- **A development machine (Windows PC)**: This windows PC will work as a host machine to deploy SFT to test device. This PC should be Windows 8.1 or Windows 10, and installed with **Windows 10 SDK and Tools**, so to have those tools required.

  Partners may refer to Windows 10 partner documentation (some skills could also be found in Windows Phone 8.1 GDR1 partner document) for details on how to setup a development machine.

  The following tools will be required:

  - Windows 10 SDK tools: pkggen.exe, iutool.exe, WinAppDeployCmd.exe, imggen.cmd.
  - Phone debugging tools are required to install and setup properly: TShell and IPOverUSB or Virtual Ethernet. For detail setup information, please refer to the following topics:
    - IP Over USB:
      https://sysdev.microsoft.com/en-us/Hardware/oem/docs/Phone_Testing/IP_over_USB
    - Virtual Ethernet:
      Please find "Running Virtual Ethernet" section in this page.
      https://sysdev.microsoft.com/en-us/Hardware/oem/docs/Phone_Testing/Installing_and_configuring_TShell
    - TShell: Installing and configuring TShell (Please skip the step of "Configuring the phone for KDNet over USB"):
      https://sysdev.microsoft.com/en-us/Hardware/oem/docs/Phone_Testing/Installing_and_configuring_TShell
  - OEM test certifications are required to install on the machine (refer to Windows 10 partner document, chapter - "Set up the signing environment").

Install SFT

A. Prepare test device

A test device must be installed with Windows 10 mobile test image or production image. The required OS version should be:

- **TH2 build 10546 (QFE 13013)** or later versions.

The latest released OS version is highly recommended.

SFT must be installed and run under full OS mode (not supported in MMOS or manufacturing mode).

SFT could be preload to image at image build time.
For SFT to install at OS runtime, before installation, the test device must be configured properly as below:

- **Turn on developer mode to enable evaluation of SFT:**

1. On your device that you want to enable, go to **Settings**. Choose **Update & security**. Then choose **For developers**.
2. Select **Developer Mode** to enable developer mode.

Note: For detail information about developer mode, please refer to this web link - Enable your device for development: https://msdn.microsoft.com/en-us/library/windows/apps/dn706236.aspx

B. Install SFT at OS runtime

1. Install SFT service and configuration:
   a. Connect test phone device to this PC via USB. Make sure that phone device has MTP enabled, and "kernel-mode debugging" disabled.
   b. Open command prompt.
   c. Setup environment variables.

```
> set WPDKCONTENTROOT=C:\Program Files (x86)\Windows Kits\10
> set PATH=%PATH%;%WPDKCONTENTROOT%\Tools\bin\i386
```

[Note] According to your Windows architecture type, "Program Files (x86)" in this example could be "Program Files" for your Windows.

   d. Change folder to SFT service installation content folder:

```
> cd C:\Tools\SFT_mobile_1.0.8.0
```

   e. Run iutool to update service spkg to test device:

```
> iutool -p OEMName.DiagTool.SystemTestService.spkg;
Microsoft.Temp.VC140Runtime.spkg;OEMName.DiagTool.SFTConfig.spkg
```

[Note] For details about iutool, please refer to Windows 10 partner document chapter [IUTool.exe: Update packages on a phone].

2. Install SFT UWP App by using **winappdploycmd** tool:

   Winappdeploycmd will use TCP/IP to connect host PC and target phone device, hence both machine must be assigned with valid IP within the same network without the need of gateway. Based on this requirement, you may choose to use USB tethering with IP Over USB or Virtual Ethernet running on host PC and target phone device; or you may setup a Wifi environment for both machine to connect over Wifi (two machines must connect to the same Wifi access point).

   In this example, we will demonstrate the usage of USB tethering and use IP Over USB (the connected phone device will always map to IP 127.0.0.1 on host machine).

   a. Connect test phone device to this PC via USB. Make sure that phone device has MTP enabled, and "kernel-mode debugging" disabled. The host PC must have IPOverUSB setup properly.
   b. Open **admin-mode** command prompt.
   c. Setup environment variables.

```
> set WPDKCONTENTROOT=C:\Program Files (x86)\Windows Kits\10
> set
WIN10_TOOL=%WPDKCONTENTROOT%\bin\x86;%WPDKCONTENTROOT%\Tools\bin\
i386
> set PATH=%PATH%;%WIN10_TOOL%
```

   [Note] According to your Windows architecture type, "Program Files (x86)" in this example could be "Program Files" for your Windows.

   d. Change folder to SFT service installation content folder:

```
> cd C:\Tools\SFT_mobile_1.0.8.0
```

   e. Check device connectivity, and get the IP of your test device:

```
> winappdeploycmd.exe devices
```

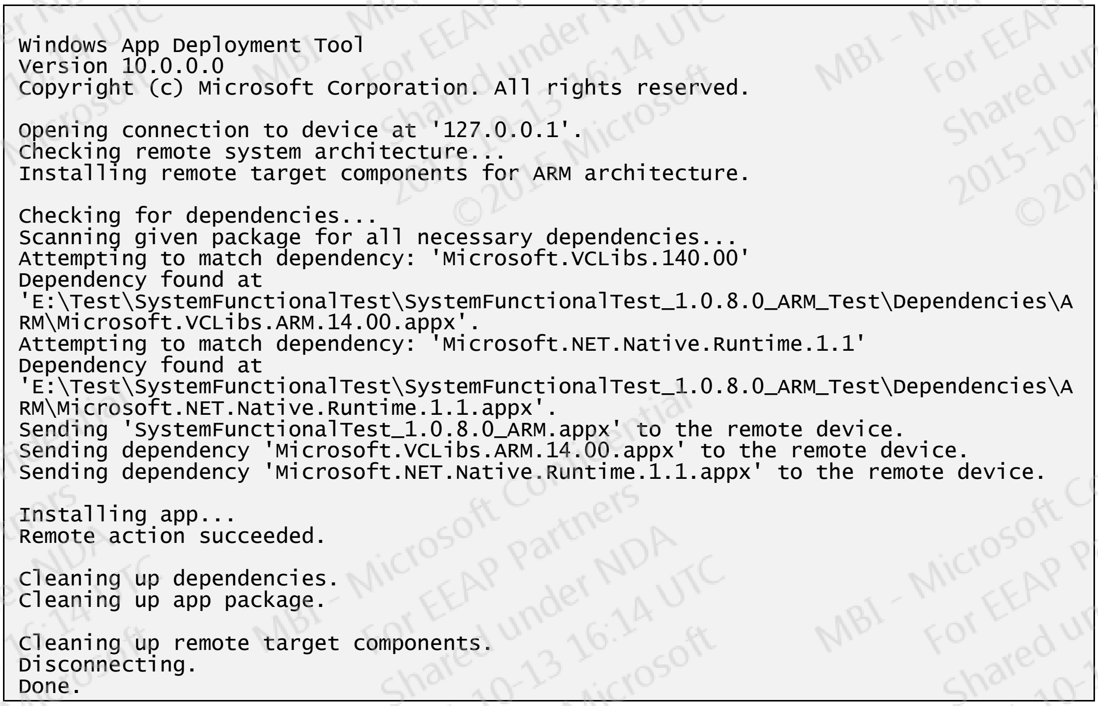
```
Windows App Deployment Tool
Version 10.0.0.0
Copyright (c) Microsoft Corporation. All rights reserved.

Discovering devices...
IP Address      GUID                                    Model/Name
127.0.0.1       00000000-0000-xxxx-0000-000000000000    QRD 8x26
Done.
```

   [Troubleshooting] If device is connected using Wifi, this command (winappdeploycmd devices) might be failed to detect such devices. For such case, please ignore this step, or try running **step f.** and then **step h.** with or without –pin option (winappdeploycmd list –ip xxxx [-pin xxx]) to check connectivity.

f. On phone device, go to **Settings** > **Update & Security** > **For developers**, Set "**Device Discovery**" to "**On**", and press "**Pair**" to get a **PIN** number (read this pin number on phone screen, for example *xX1234*, and use this pin number in WinAppDeployCmd.exe command line argument).

g. Run WinAppDeployCmd.exe to side-load app to test device. Get back to host PC command prompt, side-load appx to the connected phone device with PIN code to have this connection authorized to run app installation.

```
> winappdeploycmd.exe install -file
SystemFunctionalTest_1.0.8.0_ARM_Test\SystemFunctionalTest_1.0.8.
0_ARM.appx -ip 127.0.0.1 -pin xX1234
```



```
Windows App Deployment Tool
Version 10.0.0.0
Copyright (c) Microsoft Corporation. All rights reserved.

Opening connection to device at '127.0.0.1'.
Checking remote system architecture...
Installing remote target components for ARM architecture.

Checking for dependencies...
Scanning given package for all necessary dependencies...
Attempting to match dependency: 'Microsoft.VCLibs.140.00'
Dependency found at
'E:\Test\SystemFunctionalTest\SystemFunctionalTest_1.0.8.0_ARM_Test\Dependencies\A
RM\Microsoft.VCLibs.ARM.14.00.appx'.
Attempting to match dependency: 'Microsoft.NET.Native.Runtime.1.1'
Dependency found at
'E:\Test\SystemFunctionalTest\SystemFunctionalTest_1.0.8.0_ARM_Test\Dependencies\A
RM\Microsoft.NET.Native.Runtime.1.1.appx'.
Sending 'SystemFunctionalTest_1.0.8.0_ARM.appx' to the remote device.
Sending dependency 'Microsoft.VCLibs.ARM.14.00.appx' to the remote device.
Sending dependency 'Microsoft.NET.Native.Runtime.1.1.appx' to the remote device.

Installing app...
Remote action succeeded.

Cleaning up dependencies.
Cleaning up app package.

Cleaning up remote target components.
Disconnecting.
Done.
```

[Note] Under the same connection condition, there's no need to use "-pin" option in later WinAppDeployCmd.exe commands. Pairing the test device and the PC is one time work per connection or per device.

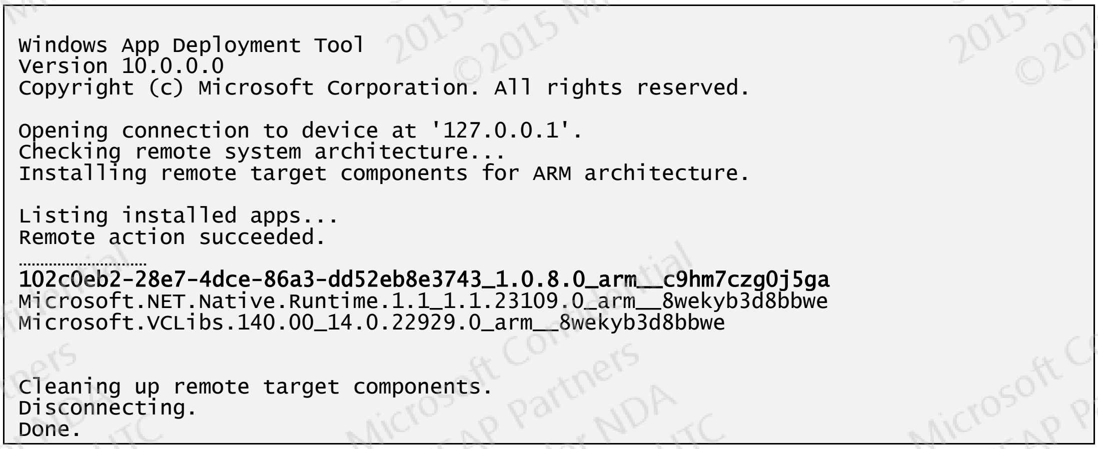
[Troubleshooting.1] There also exists situations that using "-pin" will cause error, please try the command with and without "-pin" option to see which way works.

[Troubleshooting.2] If error happens and complains about "dependency", this usually happens when running winappdeploycmd command on Win 8.1 host PC, for this situation, you must add "-d" option in the command line to explicitly specify dependency files, and make sure these dependency files exist (please find these dependency files at "*SystemFunctionalTest_1.0.8.0_ARM_Test\Dependencies\ARM*"):

```
> winappdeploycmd.exe install -file
SystemFunctionalTest_1.0.8.0_ARM_Test\SystemFunctionalTest_1.0
.8.0_ARM.appx -d
SystemFunctionalTest_1.0.8.0_ARM_Test\Dependencies\ARM\
Microsoft.NET.Native.Runtime.1.1.appx
SystemFunctionalTest_1.0.8.0_ARM_Test\Dependencies\ARM\
Microsoft.VCLibs.ARM.14.00.appx -ip 127.0.0.1 … (other
options)
```

h. Use WinAppDeployCmd.exe to double check if app is installed. This command line could also be used to check the connectivity with target device (instead of using command line "winappdeploycmd devices").

```
> winappdeploycmd.exe list -ip 127.0.0.1
```



```
Windows App Deployment Tool
Version 10.0.0.0
Copyright (c) Microsoft Corporation. All rights reserved.

Opening connection to device at '127.0.0.1'.
Checking remote system architecture...
Installing remote target components for ARM architecture.

Listing installed apps...
Remote action succeeded.
........................
102c0eb2-28e7-4dce-86a3-dd52eb8e3743_1.0.8.0_arm__c9hm7czg0j5ga
Microsoft.NET.Native.Runtime.1.1_1.1.23109.0_arm__8wekyb3d8bbwe
Microsoft.VCLibs.140.00_14.0.22929.0_arm__8wekyb3d8bbwe


Cleaning up remote target components.
Disconnecting.
Done.
```

[Note] In the displayed installed app list, "102c0eb2-28e7-4dce-86a3-dd52eb8e3743_1.0.8.0_arm__c9hm7czg0j5ga" is the package full name of SystemFunctionalTest app. This package full name could be used to uninstall this app by using WinAppDeployCmd.exe.

📝 **Note**: If WinAppDeployCmd is failed with error message:
"0x80070002 - The system cannot find the file specified. (Exception from HRESULT: 0x80070002)"
Please first confirm that the file does exist. If problem still remains, go try **step f.** to resolve the error. Please refer to the following link for more details about "Install Universal Windows Apps with the WinAppDeployCmd tool":
https://msdn.microsoft.com/en-us/library/mt203806(v=vs.140).aspx

C. Preload SFT at image build time

To illustrate how to preload SFT to image at image build time, we'll use "imggen.cmd" tool to build image.

SFT spkg files are test-signed, the target image type is restricted to test image or production image.

In this example, we will show you how to build a test image that has SFT application preloaded.

1. Mobile OS version requirement: The same OS version requirement as described in "A. Prepare test device". In this example, we assume you install the mobile OS prebuilt packages at:
   ```
   C:\Program Files (x86)\Windows Kits\10\MSPackages
   ```

2. Locate SFT spkg files:
   Create a subfolder "AddOnPackages" in BSP root folder (where BSP.config.xml locates). For example, BSP root is:
   ```
   C:\BSP\SOC_XXX\WP\prebuilt
   ```

   Copy "SFT_mobile_v1.0.8.0" folder to "AddOnPackages" folder, so the contents could be found at:
   ```
   C:\BSP\SOC_XXX\WP\prebuilt\AddOnPackages\SFT_mobile_v1.0.8.0
   ```

3. Create a working folder for image build, for example:
   ```
   C:\OEM\Img.WP10\TestImage
   ```

4. Modify OEMInput file:
   a. Test image OEMInput file template could be found at:
      ```
      C:\BSP\SOC_XXX\WP\prebuilt\OEMInputTemplates\output\
      oeminput_testos.xml
      ```

   b. Create an MyOEMInput file for test image at:
      ```
       C:\OEM\Img.WP10\TestImage\MyOEMInputFile.xml
      ```

   c. Modify MyOEMInputFile.xml to include SFT "PreloadPackages.xml", so the spkg files defined in "PreloadPackages.xml" will be built-in to image:

      ```
      <AdditionalFMs>
      . . .
      <AdditionalFM>%QCPackageDir%\AddOnPackages\SFT_mobile_v1.0.8.0
      \PreloadPackages.xml</AdditionalFM>

      </AdditionalFMs>
      ```

5. Generate image:

Open an admin-mode command prompt:

a. Setup environment variables and path:

```
> set WPDKCONTENTROOT=C:\Program Files (x86)\Windows Kits\10
> set
WIN10_TOOL=%WPDKCONTENTROOT%\bin\x86;%WPDKCONTENTROOT%\Tools\b
in\i386
> set PATH=%PATH%;%WIN10_TOOL%
```

b. “**QCPackageDir**” macro will be referenced in “MyOEMInputfile.xml” and SFT “PreloadPackages.xml”, so this environment variable should be set properly (set this value to the path where “BSP.config.xml” file is located):

```
> set QCPackageDir=C:\BSP\SOC_XXX\WP\prebuilt
```

c. Run “imggen.cmd” to build image:

```
> cd C:\OEM\Img.WP10\TestImage
> mkdir output
> imggen “.\output\flash.ffu“ “MyOEMInputFile.xml"
"%WPDKCONTENTROOT%\MSPackages"

> dir output\flash.ffu
```

d. After **step c.** is successfully done, sign the image if it’s required (please refer to partner document for detail information about how to sign the image).
Make sure the target phone device is boot up to download mode and is connected to host PC via USB.
Use ffutool.exe to flash this image to target phone device to test:

```
> cd C:\OEM\Img.WP10\TestImage\output
> dir flash.ffu
> ffutool -list
> ffutool -flash flash.ffu
```

Launch SFT by using URI association

For OEM application to launch SFT by using URI association, here’s the code example:

```
var isSuccess = await Windows.System.Launcher.LaunchUriAsync(new Uri(“oem-
tool-sft:”));
```

Run the test

1. Power on Windows mobile device.
2. If the test requires audio file for audio tests, copy the audio test file to device folder "Music". And do rename the file name to match to the names defined in configuration file. For this example, as configured in SFTConfig.xml, you may prepare one .wav audio file and copy to "This PC > Windows Phone > Phone > Music > StereoChannelTest.wav".
3. Go to start menu, select "All Apps" to show all apps list.
4. Find the "SystemFunctionalTest" app listed down in "S" category (figure. 1). Select "SystemFunctionalTest" to start the app. Follow the GUI guidance to run the test.

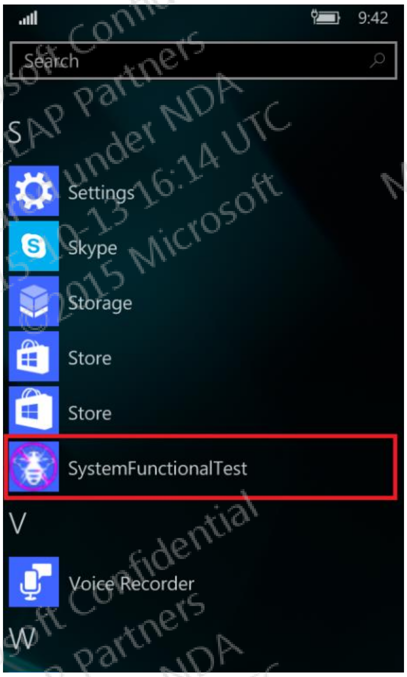


Figure. 1

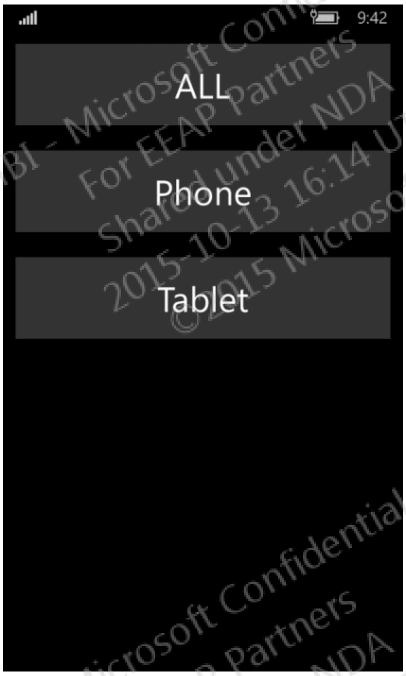


Figure. 2

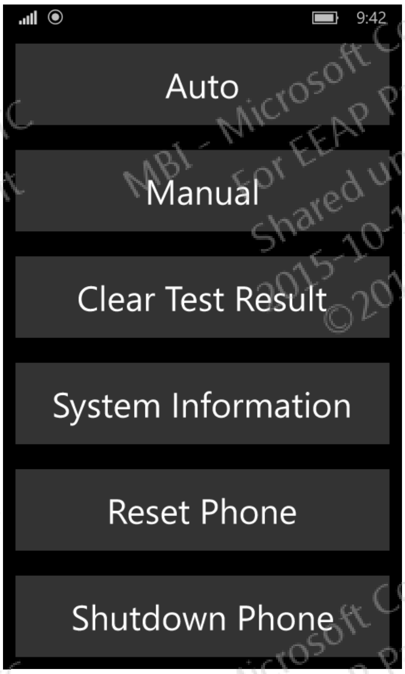


Figure. 3

5. GUI / **Phase Selection Page**: (figure. 2)
   This page shows the available test phases. Select a preferred test phase to enter.

   Note that if SFTConfig.xml contains only one test phase, then this page/step will be skipped and goes directly to step 6 (Tool Function Page).

   - Select a phase to enter step 6 (Tool Function), or,
   - Click on "Close" icon on app title bar to exit the app.

6. GUI / Tool Function Page: (figure. 3)
   This page shows a predefined test phase specified in SFTConfig.xml (or default built-in test phase if SFTConfig.xml is not found).

Default available tool functions: Auto / Manual / Clear Test Result / System Information / Shutdown Phone / Reset Phone

- Choose "Auto" will go to step 8 (Auto Test Sequence).
- Choose "Manual will go to step 7 (Test Item List).
- Choose "Clear Test Result" will clear all test results after user's confirmation.
- Choose "System Information" will enter system Information page.
- Choose "Shutdown Phone" will shutdown the phone after user's confirmation.
- Choose "Reset Phone" will start factory reset process after user's confirmation.
- Click on "Back" button will exit the app.

7. GUI / **Test Item List Page (for Manual Test)**: (figure. 4)

This page lists a set of test items configured in SFTConfig.xml for a specific test phase (depending on the number of test items, user may need to scroll up/down to access all test items). Select one of the test item to test.

- Select test item will go to step 9 (Test Item).
- Or select "Back to Main Page" at the bottom of this page to get back to step 6 (Tool Function).

Each test item will be finished once test engineer feedbacks the test result as "fail" or "pass", and then get back to this page. The placeholder of the test item will change color to indicate the status of the test item.

Test item status: Transparent for "Not tested" / Green for "Pass" / Red for "Fail".

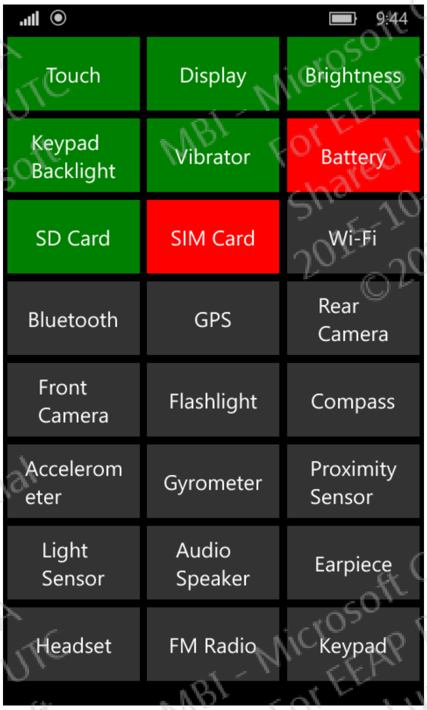


Figure. 4

Available test items (Total 25 items): Touch / Display / Brightness / Keypad Backlight / Vibrator / Battery / SD / SIM / Wi-Fi / Bluetooth / GPS / Rear Camera / Front Camera / Flashlight / Compass / Accelerometer / Gyro meter / Proximity / Light / Speaker / Earpiece / Headset / FM Radio / Keypad / Dialer

8. GUI / **Auto Test Sequence (for Auto Test):**

The auto test sequence will proceed from item to item automatically, without back and forth to item list page. It starts from first test item all the way to the last item. The auto test sequence will be stopped only when all test items are finished.

Once auto test sequence is stopped, it automatically goes to step 7 (Test Item List). Pressing "Close" button on app title bar will exit the program.

9. GUI / **Test Item:** (figure. 5)

For some test items, the program will prompt Pass/Fail/Retry selections to test engineer and wait until the engineer determines the result.

For test items the program can determine the result directly, these test items will show the test result once the tests are finished.

Test Engineer Feedback Option: Pass / Fail / Retry

For "Auto" test: If test engineer chooses "Retry", then it restarts current item. If test result is "Pass", it proceeds to next test item. If result is "Fail", it goes to step 7 (Test Item List).

For "Manual" Test: If test engineer feedbacks "Retry", then it restarts current item. If test engineer feedbacks "Pass" or "Fail", it goes to step 7 (Test Item List).

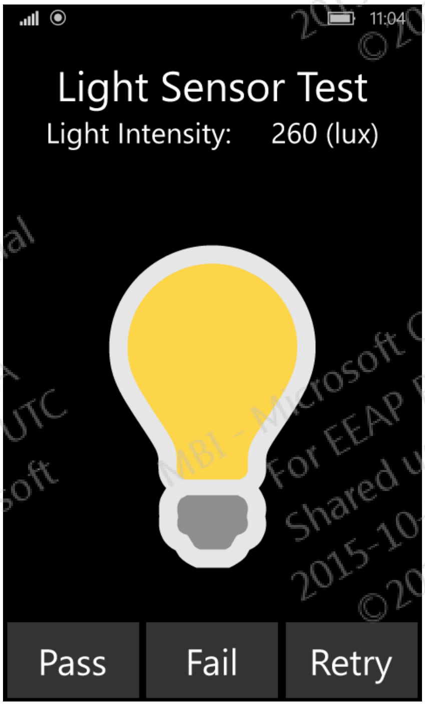


Figure. 5

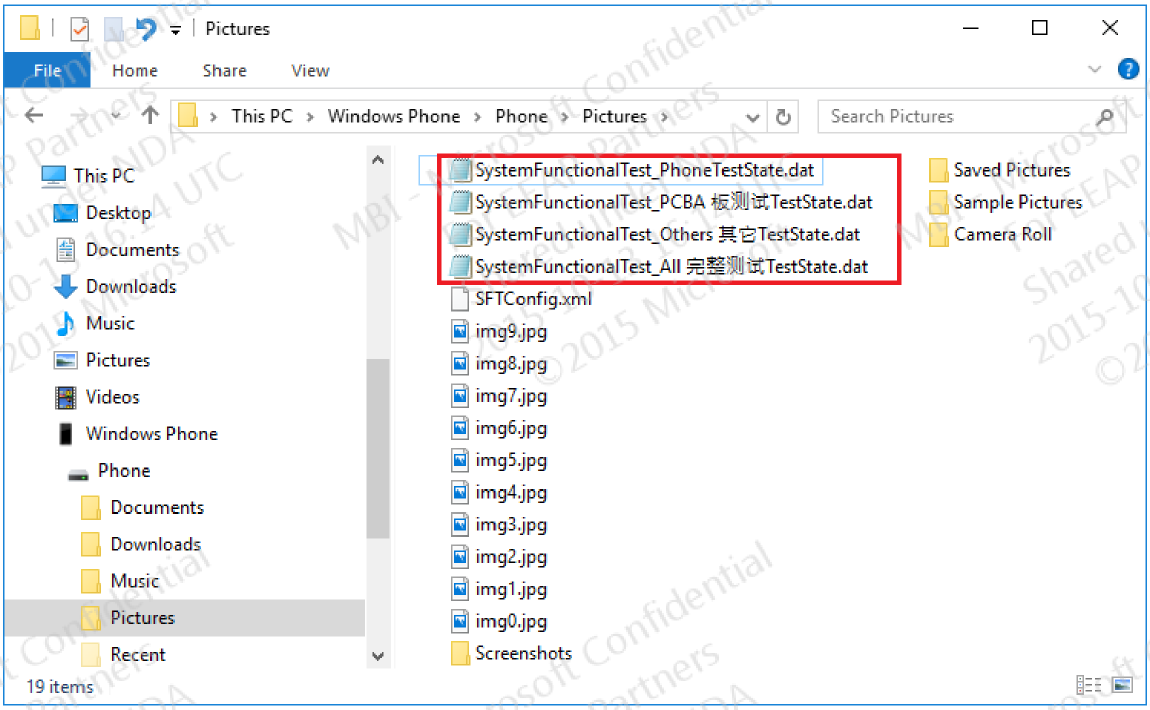
Note: For more function / test item details, refer to chapter "SFT Supported Test Items".

Log

Each time the MMI test is done by using SFT, a log file will be generated:

1. Ensure device OS system is up and running, Connect the phone device to host PC. On the host PC, wait until phone device MTP connection is established. Open file explorer and find "This PC" > "Windows Phone" > "Phone". Click to open the "Pictures" folder on "Phone" storage. ((figure. 6)
2. Find logs in file name started with SystemFunctionalTest_<*PhaseName*>TestState.dat

Figure. 6



SystemFunctionalTest_PhoneTestState.dat
SystemFunctionalTest_PCBA 板测试TestState.dat
SystemFunctionalTest_Others 其它TestState.dat
SystemFunctionalTest_All 完整测试TestState.dat
SFTConfig.xml
img9.jpg
img8.jpg
img7.jpg
img6.jpg
img5.jpg
img4.jpg
img3.jpg
img2.jpg
img1.jpg
img0.jpg
Screenshots
Saved Pictures
Sample Pictures
Camera Roll

SFT configuration

Partners could configure SFT by using an external configuration file. The details of the configuration will be described below.

Configuration file

SFT app will prioritize the use of external configuration file, it checks if an external configuration file exists at user public storage folder “Pictures”. While phone is connected to PC, this file could be found at “This PC > Windows Phone > Phone > Pictures > SFTConfig.xml”. If there is no external configuration file found, SFT app will use the default configuration inside the app’s private space. The following explains what customization can be done through SFTConfig.xml file:

There is a sample configuration file provided with this example. If you follow the instructions to install the example files, you can find the sample configuration file here: `C:\Tools\SFT_mobile_1.0.8.0\SFTConfig.xml`

- To specify user preferred UI language:

  ```
  <SystemFunctionalTest>
  <Language Name=”predefined language name” />
  </SystemFunctionalTest>
  ```

  - One or none <Language> could be set within <SystemFunctionalTest>
  - The value of attribute “Name” will be used to override SFT app language preference. SFT app will select the language by the following rules: If no preferred language is configured, app will select last configured language. If app has currently no preferred language set, app will select OS system preferred language. If the selected language is not supported by SFT app, SFT app will use default language - “en-US”.
  - Valid values for “Name” are: `en-US / zh-TW / zh-CN`

- To setup logging preference:

  ```
  <SystemFunctionalTest>
  <Log Enable=”True or False” />
  </SystemFunctionalTest>
  ```

  - One or none <Log> could be set within <SystemFunctionalTest>
  - The value of attribute “Enable” will be used to enable (“True”)/ disable (“False”) logging. Default (if this tag is not set or invalid) is enabled.
  - Valid values for “Enable” are: `True / False`

- To add / modify test phases:

  ```
  <SystemFunctionalTest>
  <Phase Name=”partner defined phase name”></Phase>
  </SystemFunctionalTest>
  ```

  - One or multiple <Phase> could be set within <SystemFunctionalTest>
  - The value of attribute “Name” of <Phase> is defined by partner. This value will be used in “Phase Selection Page” in app.

- To add / modify tool functions:

```
<Phase>
<MainMenu>
<MenuItem Name=”predefined function name” />
</MainMenu>
</Phase>
```

  - Only one <MainMenu> is allowed within <Phase>
  - One or multiple <MenuItem> could be set within <MainMenu>
  - The value of attribute “Name” of <MenuItem> should be one from the predefined functions:

    `Auto / Manual / ClearResult / SystemInformation / ShutdownPhone / ResetPhone`

- To add / modify test items:

```
<Phase>
<TestMenu>
<MenuItem Name=”predefined item name” />
</TestMenu>
</Phase>
```

  - Only one <TestMenu> is allowed within <Phase>
  - One or multiple <MenuItem> could be set within <TestMenu>
  - SFT will follow the order of the items defined by <MenuItem> to list the test items in “Test Item List Page” in app. This order will also reflects the test sequence in “Auto Test”.
  - The value of attribute “Name” of <MenuItem> should be one from the predefined test items:

    `Touch / Display / Brightness / Backlight / Vibrator / Battery / SD / WiFi / Bluetooth / SIM / RearCamera / FrontCamera / Flashlight / Compass / Accelerometer / Gyrometer / Proximity / Light / Speaker / Earpiece / Headset / FM Radio / Keypad / GPS / Dialer`

- To configure function item - “SystemInformation”:

```
<MainMenu>
<MenuItem Name=”SystemInformation”>
<Property>predefined property name</Property>
</MenuItem>
</MainMenu>
```

  - One or multiple <Property> could be set within <MenuItem Name=”SystemInformation”>.
  - The value for the <Property> should be one from the predefined properties:

```
Friendly Name / Device Name / Manufacture / Model Name / Operator
Name / OS Version / Chip SOC Version / Firmware Version / Hardware
Version / Radio Software Version / Radio Hardware Version / Screen
Resolution / App Version / Cellular Class / Mobile Data Class / IMEI
/ WLAN MAC / BT MAC
```

- To configure test item - "Wifi":

```
<TestMenu>
<MenuItem Name="Wifi">
<AvailableName>Wifi SSID to match for autopass</AvailableName>
<ConnectionName>Wifi SSID to try connection for
autopass</ConnectionName>
<Threshold SignalBar="Signal Strength Indicator Value" />
</MenuItem>
</TestMenu>
```

  - <AvailableName>, <ConnectionName> and <Threshold> defines 3 kinds of auto-pass conditions. <Threshold> is effective only when <AvailableName> is set. <AvailableName> and <ConnectionName> could be used solely or together.
  - None or multiple (at most 10) <AvailableName> could be set within <MenuItem Name="Wifi">.
  - The value for the <AvailableName> should be the Wifi SSID to be matched to trigger pass / fail automatically without the manual feedback from test engineer (aka. auto-pass).
  - The value for the <ConnectionName> should be the Wifi SSID to try connection with. If connection could be setup successfully (will be disconnected right after), it will contribute a pass condition, otherwise a failure condition, for auto-pass.
  - SFT will match the configured SSID string with the scanned Wifi SSID list by using case sensitive comparison. If any of the defined SSID is not found in scanned list, SFT will fail this test item directly.
  - None or one <Threshold> could be set within <MenuItem Name="Wifi">.
  - The attribute "SignalBar" in <Threshold> is the signal strength indicator for each Wifi SSID scanned. Usually the signal bar value is among 0 to 5, from low to high. This "SignalBar" attribute defines the minimal signal strength that a Wifi network should meet. Only when all the listed Wifi networks (defined by <AvailableName>) are found and each has signal strength value equal to or larger than this attribute value, then this will be determined as a positive "pass" condition.
  - If there is no auto-pass criteria defined, manual feedback from test engineer is required.

- To configure test item - "Bluetooth":

```
<TestMenu>
<MenuItem Name="Bluetooth">
<AvailableName>Bluetooth device name to match for
autopass</AvailableName>
<Threshold AllowedMaxSearchTime="Search time in seconds"/>
</MenuItem>
</TestMenu>
```

  - <AvailableName> and <Threshold> defines two kinds of auto-pass conditions. Any of these conditions could be used solely or together.
  - None or multiple (at most 10) <AvailableName> could be set within <MenuItem Name="Bluetooth">.
  - The value for the <AvailableName> should be the Bluetooth device name to be matched to trigger pass / fail automatically without the manual feedback from test engineer (aka. auto-pass).
  - SFT will match the configured Bluetooth device name string with the scanned Bluetooth device name list by using case sensitive comparison. If any of the defined Bluetooth device name is not found in scanned list, SFT will fail this test item directly.
  - The attribute "AllowedMaxSearchTime" in <Threshold> defines a search time window for a BT scan. There's no guarantee that longer the search time the more BT devices might be searched. The value for this attribute is the time in seconds from 1 to 10. The time defined is just for the BT scan process to use as a timeout reference, the actual scan time could be shorter or longer (usually longer). This value should be set only if that value found to best fit for your test need.
  - If there is no auto-pass criteria defined, manual feedback from test engineer is required.

- To configure test item - "SD":

```
<TestMenu>
<MenuItem Name="SD">
<Threshold Count="Number of SD storage found to autopass" />
</MenuItem>
</TestMenu>
```

  - None or one <Threshold> could be set within <MenuItem Name="SD">.
  - <Threshold> defines the minimum number of SD storage found to trigger pass / fail automatically without the manual feedback from test engineer (aka. auto-pass).
  - The value for <Threshold> should be a positive integer value. Please notice that, this version of SFT can detect at most 1 SD storage, hence, only "1" is the meaning threshold count value to set.
  - If a threshold count is defined, SFT will auto-pass this test item while the number of SD storage found is greater than or equal to this threshold count, otherwise, SFT will fail this test item.
  - If there is no auto-pass criteria defined, manual feedback from test engineer is required.

- To configure test item - "Speaker", "Headset", "Earpiece":

```
<TestMenu>
<MenuItem Name="Speaker">
<Property JackAutoDetection="True or False" />
<Property AudioOutSource="True or False" />
<Property EnsureAudioOut="Yes or No or Auto" />
```

```
</MenuItem>
</TestMenu>
```

- None or multiple <Property> could be set within <MenuItem Name="Speaker"> (or Name="Headset", Name="Earpiece").
- None ore multiple attributes could be set in <Property>. Valid attributes are: `JackAutoDetection | AudioOutSource | EnsureAudioOut`
- JackAutoDetection: valid values are: `True | False`.

  [Note] This attribute is for desktop only, and is not applicable for mobile.
- AudioOutSource: valid value is the audio file name for speaker / headphone audio output test. This attribute is meaningful only if "EnsureAudioOut" is set to "Yes" or "Auto". The audio file must be placed under "Music" library folder.
- EnsureAudioOut: valid values are: `Yes | No | Auto.`
  - If attribute is set to "Yes", the audio test will require test engineer to press "Play" button to a clip of audio source (set in attribute "AudioOutSource") to ensure the audio output end-point working properly before proceeding to audio capture / replay test.
  - If this attribute is set to "Auto", audio test will try to search for audio file as an audio output source, if audio file is not found, the test will start from audio capture directly.
  - If this attribute is set to "No", audio test will start from audio capture without checking the existence of audio file defined in "AudioOutSource".

- To configure test item - "Keypad":

```
<TestMenu>
<MenuItem Name="Keypad">
<Button>predefined hardware button name</Button>
<Property RepeatCount="Required Key Repeat Count" />
</MenuItem>
</TestMenu>
```

- One or multiple <Button> could be set within <MenuItem Name="Keypad">.
- The value for the <Button> should be one from the predefined hardware button names:
  `Back / Start / Search / VolUp / VolDown / Power / Camera / HeadsetButton`
- One or none <Property RepeatCount="*Count*"> could be set within <MenuItem Name="Keypad">.
- "RepeatCount" attribute in <Property> defines a key press repeated count required for key press test. A hardware button must be pressed continuously as many times as the "RepeatCount" defined. The valid repeat count value is an integer value from 1 to 3, the default value is 1.

- To configure test item - "Display":

```
<TestMenu>
<MenuItem Name=”Display”>
<ColorTest Color=”RBG Color Code”>Color Name</ColorTest>
</MenuItem>
</TestMenu>
```

- None or multiple <ColorTest> could be set within <MenuItem Name=”Display”>. Each <ColorTest> will define a test step in display test.
- “Color” is an attribute for <ColorTest> to define the color to be displayed in fullscreen. The value of “Color” attribute should be a RGB color code in this format: `#RRGGBB`
- The “Color Name” value for <ColorTest> is not used in current version, any value specified will be ignored.

- To configure test item - “Touch”:

```
<TestMenu>
<MenuItem Name=”Touch”>
<SubItem Name=”MultiPoints”><Property
MultiPointCount=”Number”/></SubItem>
< SubItem Name=”MultiDraw”><Property
MultiDrawCount=”Number”/></SubItem>
</MenuItem>
</TestMenu>
```

  - None or multiple <SubItem> could be set within <MenuItem>
  - Each <SubItem> must have “Name” attribute set to one of the following value : `MultiPoints | MultiDraw`
  - <SubItem Name=”MultiPoints”> defines a touch test sub item to run multi-point test. <Property> with attribute “MultiPointCount” must be set to describe this sub item. Valid value for “MultiPointCount” should be integer that represents how many multi-point contacts on touch screen is expected for this sub item. The range of the integer is usually the maximum number of multi-touch-points supported by the test device.
  - <SubItem Name=”MultiDraw”> defines a touch test sub item to run multi-draw test. <Property> with attribute “MultiDrawCount” must be set to describe this sub item. Valid value for “MultiDrawCount” should be integer that represents how many multi-points is expected to draw on touch screen for this sub item. The range of the integer is among 1 to 3.

- To configure test item - “GPS”:

```
<TestMenu>
<MenuItem Name=”GPS”>
<Threshold LocationMatch=”Any ” AllowedMaxSearchTime=”Search time in
seconds”/>
</MenuItem>
</TestMenu>
```

  - None or one < Threshold > could be set within <MenuItem Name=”GPS”>. <Threshold> defines the condition for auto-pass.
  - The attribute “LocationMatch” will define the location to match for auto-pass. Only one value is allowed for current version – “Any”.

  - The attribute "AllowedMaxSearchTime" defines a search timeout (in seconds) to wait for GPS position fix. The search time measured since SFT starts till this GPS test is requested. If the measured search time has not yet reached the timeout value, GPS test will wait till the timeout is reached or any auto-pass / auto-fail condition is met. This value should be set to a reasonable GPS position-fix time, for example, it might take 2 minutes or more for GPS to fix position since cold-boot.
  - It's recommended to have GPS test to run at the end of test sequence, so the total item test time will cover the time required for a GPS position-fix.

- To configure test item - "Battery":

```
<TestMenu>
<MenuItem Name=”Battery”>
<Threshold MinCapacity=”Minimal Remaining Capacity In Percentage” />
</MenuItem>
</TestMenu>
```

  - One or none <Threshold> could be set within <MenuItem Name="Battery">.
  - "MinCapacity" attribute of <Threshold> defines the minimal remaining capacity of the battery for the auto-pass criteria. Only if the battery remaining capacity (in percentage) is equal to or larger than the value defined by "MinCapacity", it will be determined as a positive pass condition. Valid value for this attribute is an integer from 0 to 100. Default value is 0.

- To configure test item - "FrontCamera", "RearCamera":

```
<TestMenu>
<MenuItem Name=”FrontCamera”>
<Property Width=”Photo Width in px” Height=”Photo Height in px” />
</MenuItem>
</TestMenu>
```

  - One or none <Property> could be set within <MenuItem Name="FrontCamera"> (or "RearCamera").
  - "Width" and "Height" attribute of <Property> defines the photo width and height in pixels. Usually there's no need to configure photo resolution, it should be configured only if default photo size cannot meet the test need.

- To configure test item - "FMRadio":

```
<TestMenu>
<MenuItem Name=”FMRadio”>
<RadioSearch RegionCode=”region code”/>
</MenuItem>
</TestMenu>
```

  - One or none <RadioSearch> could be set within <MenuItem Name="FMRadio">.
  - The value of attribute "RegionCode" will be used to set radio search region: North America ("1")/ World ("2") / Japan ("3"). Default region (if this tag is not set or invalid) is World ("2").

  - Valid values for "RegionCode" are: `1 / 2 / 3`

```
<TestMenu>
<MenuItem Name="FMRadio">
<Channel Frequency="Frequency value in MHz"/>
</MenuItem>
</TestMenu>
```

  - Multiple, one or none <Channel> could be set within <MenuItem Name="FMRadio">.
  - The value of attribute "Frequency" is defined by partner. The value will be used to set one channel frequency for quick radio channel search. Default channel search (if no valid channel is defined) is the whole FM radio frequency band of the specified region.

- To configure test item - "SIM":

```
<TestMenu>
<MenuItem Name="SIM">
<Property ShowInfo="True or False"/>
</MenuItem>
</TestMenu>
```

  - One or none <Property> could be set within <MenuItem Name="SIM">.
  - The value of attribute "ShowInfo" will be used to show (set to "True") / hide (set to "False") extra SIM information (IMEI, Cellular technology info, Mobile data class info), default is "False" (if no valid attribute value is set).
  - Valid values for "ShowInfo" are: `True / False`

```
<TestMenu>
<MenuItem Name="SIM">
<Threshold SIM1="State Filter" SIM2="State Filter" />
</MenuItem>
</TestMenu>
```

  - One or none <Threshold> could be set within <MenuItem Name="SIM">.
  - For single SIM phone device, only "SIM1" attribute should be set in <Threshold> to indicate only one SIM slot; For dual SIM phone device, both "SIM1" and "SIM2" attributes should be set to indicates two SIM slots for this device. The value of "SIM1" or "SIM2" attribute will be used to setup SIM state filter for auto-pass.
  - If only "SIM1" is set and two SIM slots are found, then this test will be failed; if both "SIM1" "SIM2" are set and only one SIM slot is found, then this test will be failed.
  - SIM states will be classified to 3 stages: 1: NoCard (SIM card not inserted); 2: NotReady (SIM card detected, but the card is invalid or disabled or locked, or is currently unable to register to cellular network); 3: Ready (SIM card is detected and can successfully register to cellular network). Default value is "NoCard" (if no valid value is set).
  - If the state filter is set at stage 1 (NoCard), SIM slot in later stages (NotReady, Ready) will pass the test. If the state filter is set at stage 3 (Ready), SIM slot in earlier stage (NoCard, NotReady) will fail the test.

- Valid values for "SIM1", "SIM2": `NoCard / NotReady / Ready`

- To configure test item - "Dialer":

```
<TestMenu>
<MenuItem Name="Dialer">
<DialerNumber>Destination Phone Number</DialerNumber>
<DialerName>Display Name of DialerNumber</DialerName>
</MenuItem>
</TestMenu>
```

- One or none <DialerNumber> could be set within <MenuItem Name="Dialer">.
- <DialerName> should be paired with <DialerNumber>. One or none <DailerName> could be paired with <DialerNumber>.
- Value of <DialerNumber> will be used to setup initial destination phone number in Dialer test, test engineer may use this phone number to make phone call directly, or use dial pad to input a new destination phone number.
- Value of <DialerName> will be used as display name of the phone number set in <DialerNumber>, and this display name will be shown in Dialer test page while the phone call is in progress.

Snapshot of the sample configuration file

In this sample configuration file, the test phase "All 完整测试" demonstrates all available tests on tablet with configurable variants while the test phase "PCBA 板测试" demonstrates a test configuration and sequence that simplifies the test sequence with least human interaction.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Example Mobile Configuration File for SystemFunctionalTest UWP App Version : 1.0.8.0 -->
<SystemFunctionalTest>

    <!-- default is phone language dependent -->
    <Language Name="zh-CN" />

    <!-- default is true -->
    <Log Enable="True" />

    <Phase Name="All 完整测试">
        <MainMenu >
            <MenuItem Name="Auto"/>
            <MenuItem Name="Manual"/>
            <MenuItem Name="ClearResult"/>

            <MenuItem Name="SystemInformation">
                <Property>Friendly Name</Property>
                <Property>Device Name</Property>
                <Property>Manufacture</Property>
                <Property>Model Name</Property>
                <Property>Operator Name</Property>
                <Property>OS Version</Property>
                <Property>Chip SOC Version</Property>
                <Property>Firmware Version</Property>
                <Property>Hardware Version</Property>
                <Property>Radio Software Version</Property>
                <Property>Radio Hardware Version</Property>
                <Property>Screen Resolution</Property>
                <Property>App Version</Property>
                <!-- additional service info (for phone only) -->
                <Property>Cellular Class</Property>
                <Property>Mobile Data Class</Property>
                <Property>IMEI</Property>
                <Property>WLAN MAC</Property>
                <Property>BT MAC</Property>
            </MenuItem>

            <MenuItem Name="ResetPhone"/>
            <MenuItem Name="ShutdownPhone"/>
        </MainMenu>

        <TestMenu>
            <MenuItem Name="Touch">
                <!-- Optional additional tests : MultiPoints, MultiDraw -->
                <SubItem Name="MultiPoints">
                    <!-- maximum number is valid from 1 to 10, depending on actual multi-touch
capability -->
                    <Property MultiPointCount="2" />
                </SubItem>
                <SubItem Name="MultiDraw" >
                    <!-- maximum number is valid from 1 to 3, depending on actual multi-touch
capability -->
                    <Property MultiDrawCount="2" />
                </SubItem>
            </MenuItem>
            <MenuItem Name="Display">
                <!-- user defined colors, maximum 10 addtional color for colortest
                 if no valid colortest is defined, basic colors will be used for test
                 basic colors are : white/black/blue/red/green -->
                <ColorTest Color="#FFFFFF" >White</ColorTest>
                <ColorTest Color="#8F8F8F" >Gray</ColorTest>
                <ColorTest Color="#000000" >Black</ColorTest>
                <ColorTest Color="#FF0000" >Red</ColorTest>
                <ColorTest Color="#00FF00" >Green</ColorTest>
                <ColorTest Color="#0000FF" >Blue</ColorTest>
                <ColorTest Color="#FFFF00" >Yellow</ColorTest>
                <ColorTest Color="#00FFFF" >Cyan</ColorTest>
                <ColorTest Color="#FF00FF" >Magenta</ColorTest>
            </MenuItem>
            <MenuItem Name="Brightness"/>
            <MenuItem Name="Backlight"/>
            <MenuItem Name="Vibrator"/>
            <MenuItem Name="Battery"/>

            <MenuItem Name="SD"/>

            <MenuItem Name="SIM">
              <Property ShowInfo="True" />
            </MenuItem>

            <MenuItem Name="WiFi"/>
            <MenuItem Name="Bluetooth"/>
            <MenuItem Name="GPS"/>

            <MenuItem Name="RearCamera"/>
            <MenuItem Name="FrontCamera"/>
            <MenuItem Name="Flashlight"/>

            <MenuItem Name="Compass"/>
            <MenuItem Name="Accelerometer"/>
            <MenuItem Name="Gyrometer"/>
            <MenuItem Name="Proximity"/>
            <MenuItem Name="Light"/>

            <MenuItem Name="Speaker">
                <!-- configure to play a audio clip before recording to verify audio output is OK. -->
                <Property EnsureAudioOut="Auto" AudioOutSource="StereoChannelTest.wav" />
            </MenuItem>
            <MenuItem Name="Earpiece"/>
            <MenuItem Name="Headset"/>
            <MenuItem Name="FMRadio">
                <RadioSearch RegionCode="2"/>
                <!-- 1:NorthAmerica, 2:World(default), 3:Japan -->
                <Channel Frequency="91.7" />
                <Channel Frequency="98.8" />
                <Channel Frequency="98.5" />
                <Channel Frequency="100.7" />
            </MenuItem>

            <MenuItem Name="Keypad">
                <Button>Back</Button>
                <Button>Start</Button>
                <Button>Search</Button>
                <Button>VolUp</Button>
                <Button>VolDown</Button>
                <Button>Camera</Button>
                <Button>Power</Button>
                <Button>HeadsetButton</Button>
            </MenuItem>

            <MenuItem Name="Dialer">
                <!-- Set dialer number and name.  For mobile platform only. -->
                <DialerNumber>0123456789</DialerNumber>
                <DialerName>Friend</DialerName>
            </MenuItem>

        </TestMenu>
    </Phase>

    <Phase Name="PCBA 板测试">
        <MainMenu>
            <MenuItem Name="SystemInformation">
                <Property>Device Name</Property>
                <Property>Model Name</Property>
                <Property>Firmware Version</Property>
                <Property>Hardware Version</Property>
                <Property>Chip SOC Version</Property>
                <Property>Radio Software Version</Property>
                <Property>Radio Hardware Version</Property>
                <Property>Screen Resolution</Property>
                <Property>App Version</Property>
            </MenuItem>

            <MenuItem Name="Auto"/>
            <MenuItem Name="Manual"/>
            <MenuItem Name="ClearResult"/>
        </MainMenu>

        <TestMenu>
            <MenuItem Name="SD">
                <!-- AutoPass Criteria: Setup a detected (concurrent) SD storage count to trigger
pass/fail automatically -->
                <Threshold Count="1" />
            </MenuItem>

            <MenuItem Name="SIM">
              <!-- AutoPass Criteria: Setup a entry state of SIM to trigger pass/fail automatically.
                   State of SIM : NoCard (default) / NotReady (card inserted but not ready) / Ready
              -->
              <!-- For single SIM -->
              <Threshold SIM1="NotReady" />
              <!-- For dual SIM -->
              <!--
              <Threshold SIM1="NotReady" SIM2="NoCard" />
              -->
            </MenuItem>/>

            <MenuItem Name="Vibrator"/>
            <MenuItem Name="Brightness"/>
            <MenuItem Name="Backlight"/>
            <MenuItem Name="Keypad">
                <Button>Back</Button>
                <Button>Start</Button>
                <Button>Search</Button>
                <Button>VolUp</Button>
                <Button>VolDown</Button>
            </MenuItem>

            <MenuItem Name="Accelerometer"/>
            <MenuItem Name="Proximity"/>
            <MenuItem Name="Light"/>

            <MenuItem Name="RearCamera"/>

            <MenuItem Name="Speaker"/>
            <MenuItem Name="Earpiece"/>
            <MenuItem Name="Headset"/>
            <MenuItem Name="FMRadio"/>

            <MenuItem Name="WiFi">
                <!-- Predefined available network name list used for AutoPass (Max count = 10). -->
                <AvailableName>WifiTest_A</AvailableName>
                <AvailableName>WifiTest_B</AvailableName>
                <ConnectionName>WifiTest_A</ConnectionName>
                <Threshold SignalBar="4" />
            </MenuItem>
            <MenuItem Name="Bluetooth">
                <!-- Predefined available Bluetooth name list used for AutoPass (Max count = 10). -->
                <AvailableName>Windows Phone</AvailableName>
            </MenuItem>
            <MenuItem Name="GPS">
                <!-- Maximum allowed position fix time (since app start) set in seconds for AutoPass -
->

                <Threshold AllowedMaxSearchTime="180" LocationMatch="Any" />
            </MenuItem>
            <MenuItem Name="Dialer">
                <!-- Set dialer number and name.  For mobile platform only. -->
                <DialerNumber>0123456789</DialerNumber>
                <DialerName>Friend</DialerName>
            </MenuItem>
        </TestMenu>
    </Phase>

    <Phase Name="Others 其它">
        <MainMenu>
            <MenuItem Name="SystemInformation"/>
            <MenuItem Name="Manual"/>
            <MenuItem Name="ClearResult"/>
            <MenuItem Name="ResetPhone"/>
            <MenuItem Name="ShutdownPhone"/>
        </MainMenu>

        <TestMenu>
            <MenuItem Name="Battery"/>
        </TestMenu>
    </Phase>

</SystemFunctionalTest>
```

To update the configuration

The modified configuration file should be installed to device at user public storage “Pictures” folder: “C:\data\users\public\Pictures\SFTConfig.xml”.

A. To install configuration file at build time

To install this configuration file at build time, please follow the instructions to create a package to be included in image build. There's spkg file ready in this example (OEMName.DiagTool.SFTConfig.spkg), which partners may use directly if they find it feasible for their tests. You may skip this section if you decide to use the sample configuration file directly.

(Please notice that any file installed at device user data storage will be erased after phone reset)

1. Open command prompt, and change directory.

   ```
   > cd C:\Tools\SFT_mobile_v1.0.8.0
   ```

2. Modify the SFTConfig.xml.

   ```
   > notepad SFTConfig.xml
   ```

3. Backup original spkg file (OEMName.DiagTool.SFTConfig.spkg) before we generate new package for the modified configuration.

   ```
   > ren OEMName.DiagTool.SFTConfig.spkg OEMName.DiagTool.SFTConfig.spkg.bk
   ```

4. Use script – makecfgpkg.cmd to generate package. This script will utilize pkggen.exe tool provided by Windows 10 SDK or ADK. Find the generated package at: C:\Tools\SFT_mobile_v1.0.8.0\OEMName.DiagTool.SFTConfig.spkg

   ```
   > makecfgpkg
   > dir OEMName.DiagTool.SFTConfig.spkg
   ```

5. Rebuild the image to include the latest update from this spkg file.
6. Flash this image to device to verify if app adopts the new configuration settings.

B. To install configuration file at OS runtime

[Method.1] Use "iutool" to update the configuration file spkg to device.

1. Follow A. step 1 to step 3 to generate a spkg file that contains a modified configuration file. And Stay in the command prompt window which is used to generate the spkg.

   Note: If you plan to use "iutool" to update the SPKG package on a phone, please make sure the new SPKG package has higher version number. For example, if last spkg version on device is 10.0.0.0, then you have to build the updated spkg file with a greater version number, such as 10.0.0.1, to override previous version on device. If you use makecfgpkg.cmd to generate configuration package, please modify "pkggen" command line written in makecfgpkg.cmd to change the version number.

2. Connect the test device to host PC. Use "iutool" to update the generated spkg file to device:

```
>iutool -p OEMName.DiagTool.SFTConfig.spkg
```

3. Wait for the device to reboot and finish the updating.
4. Once the update is done, launch “SystemFunctionalTest” app to verify the new configuration settings.

[Method.2] To update configuration file on device flashed with test image, you could just use TShell to update files on device (command: putd / getd). Please refer to Windows 10 partner document.

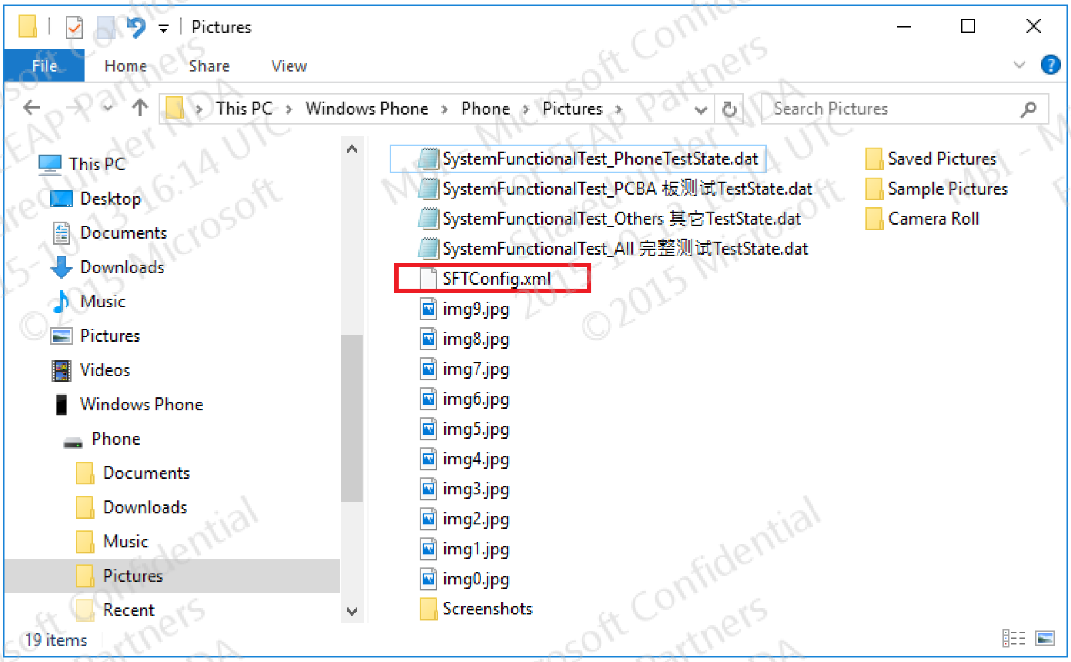
1. Modify SFTConfig.xml.
2. Connect the test device to host PC. Open TShell command prompt window. Use “putd” to push updated configuration file to device:

```
PS>cd C:\Tools\SFT_mobile_v1.0.8.0
PS>notepad SFTConfig.xml
PS>open-device 127.0.0.1
PS>putd .\SFTConfig.xml c:\data\users\public\Pictures\.
```

3. Get back to device, go to App menu and open “SystemFuncitonalTest” app, verify if the app adopts new configuration settings.

[Method.3] To update configuration file on device by using MTP.

1. Modify SFTConfig.xml
2. Connect the test device to host PC, open file explorer and browse to phone Pictures folder: “This PC > Windows Phone > Phone > Pictures”
3. Copy your source SFTConfig.xml and paste to this MTP “Pictures folder.

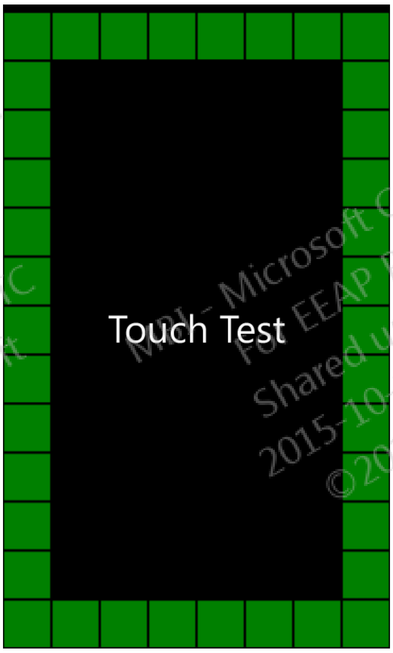


SFT supported test items

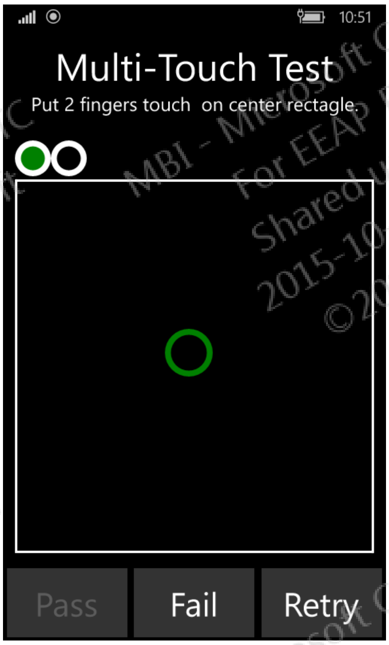
SFT provides 25 test items for windows 10 mobile device test. This section describes these test items and shows how to run these tests.

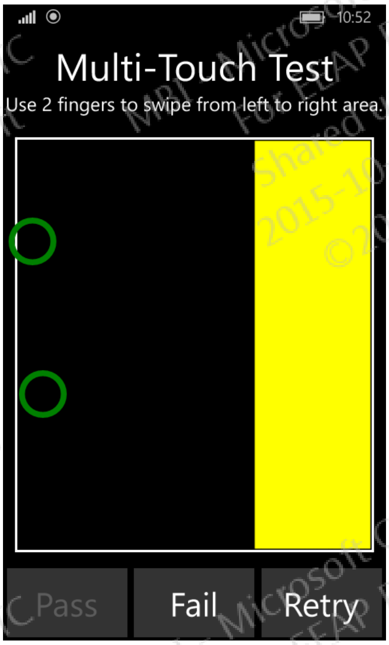
Touch Test

- Test engineer must use finger to tap (or swipe) on the green tiles on screen, when a finger touch is detected upon green tile, it will erase the touched tile.
    - You must erase all green tiles to pass the test.
    - When all green tiles are erased, the app will automatically show the result as "passed" and proceed to next step.
    - Test engineer may press physical "Back" button 3 times to fail this test.



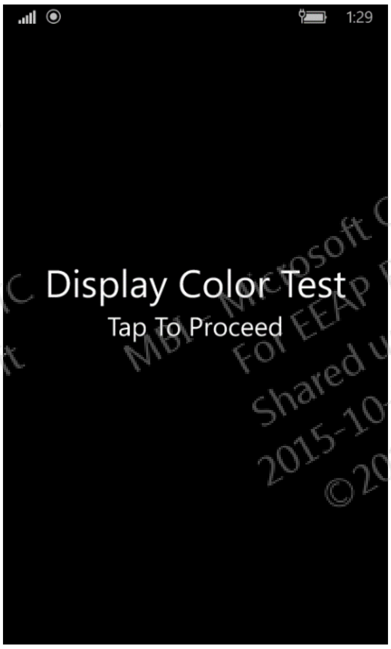
- Optional sub test item – Multi-Points:
    - Put the number of fingers required on center rectangle area.
    - When required touch contacts are detected, it will automatically proceed to next step.





- Optional sub test item – Multi-Draw:
  - Put the number of fingers required for this test inside the left yellow rectangle. Once the required touch contacts detected, the target rectangle (the yellow rectangle on right-hand side) will show, all the finger contacts must now move to hit the target rectangle at the right-hand side to pass this test.
- Test Engineer Feedback:
  - If user choose to abort the test, a feedback page will be shown for test engineer to feedback Fail / Retry.

Display Test

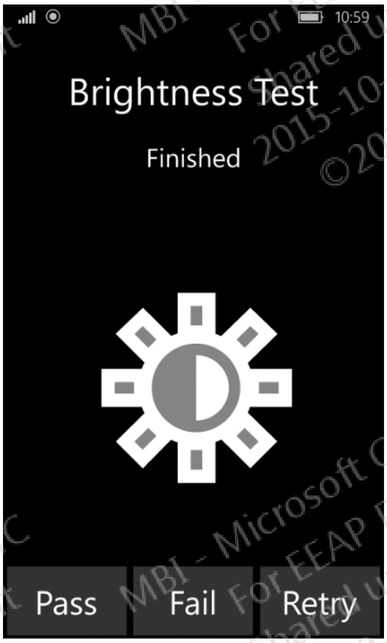


- Test engineer taps on screen to step through the color test. One color is displayed at a time in full screen:
  - Full screen color displayed in this sequence: White -> Black -> Blue -> Red -> Green. (Or the color sequence defined in configuration file)
  - Test engineer must step through all color display to enter the feedback page.
  - Test engineer may press physical "Back" button 3 times to fail this test.
- Test Engineer Feedback:
  - Pass/Fail/Retry

Brightness Test

- Test engineer observes if the display brightness changes.
    - Display brightness will be automatically changed from dim to brightest (set from 0% to 100% brightness) gradually within a few seconds, and end the test by resuming to the original display brightness.

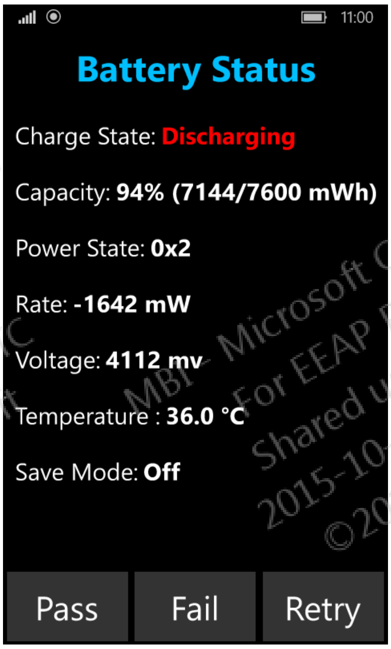
- Test Engineer Feedback
    - Pass/Fail/Retry



Battery Test

- Test engineer checks if the listed battery status is correct.
    - Test engineer may plug in/unplug power source to check if the battery status changes as expected.

- Test Engineer Feedback
    - Pass/Fail/Retry



Battery Status

Charge State: **Discharging**

Capacity: **94% (7144/7600 mWh)**

Power State: **0x2**

Rate: **-1642 mW**

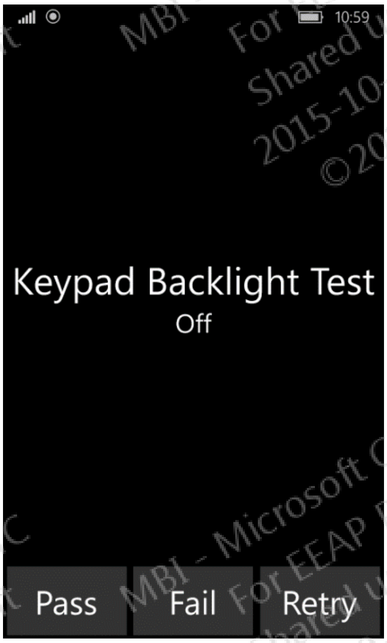
Voltage: **4112 mv**

Temperature : **36.0 °C**

Save Mode: **Off**

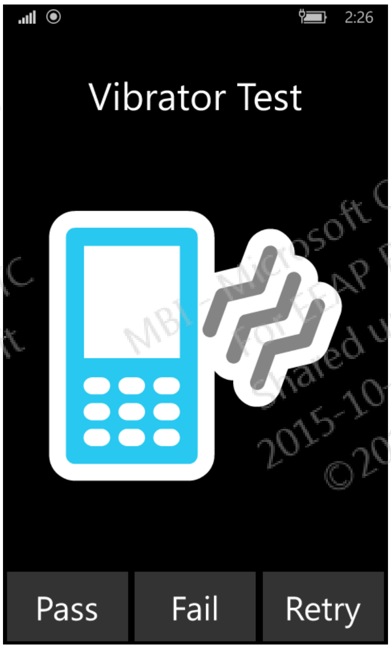
Keypad Backlight Test

- Test engineer observes backlight on/off of hardware navigation buttons.
  - Backlight LED of hardware navigation buttons (back/windows/search) will be automatically switched on/off for a few times.
- Test Engineer Feedback
  - Pass/Fail/Retry



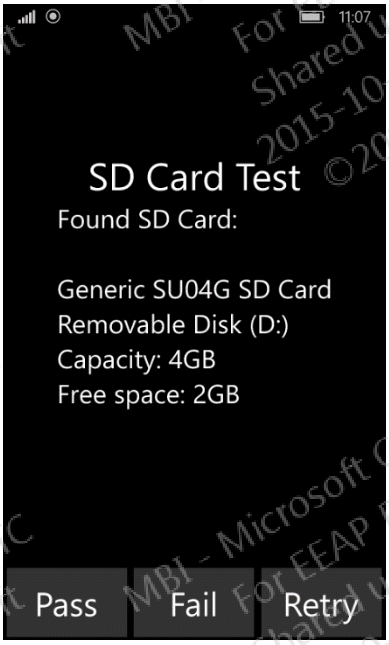
Vibrator Test

- Tester observes if the phone vibrator works.
  - The device vibrates until test engineer feedback.
- Test Engineer Feedback
  - Pass/Fail/Retry



SD Card Test

- Test engineer checks if SD Card is present.
  - SD card must be inserted to device before running this test, or press "Retry" to force detecting change of status.
- Test Engineer Feedback
  - Pass/Fail/Retry



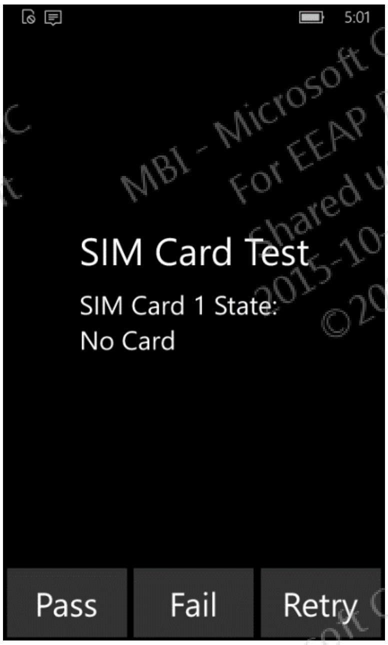
SD Card Test

Found SD Card:

Generic SU04G SD Card
Removable Disk (D:)
Capacity: 4GB
Free space: 2GB

SIM Card Test

- Test engineer checks if SIM Card is detected.
  - SIM card must be inserted into device before running this test.
- Test Engineer Feedback
  - Pass/Fail/Retry

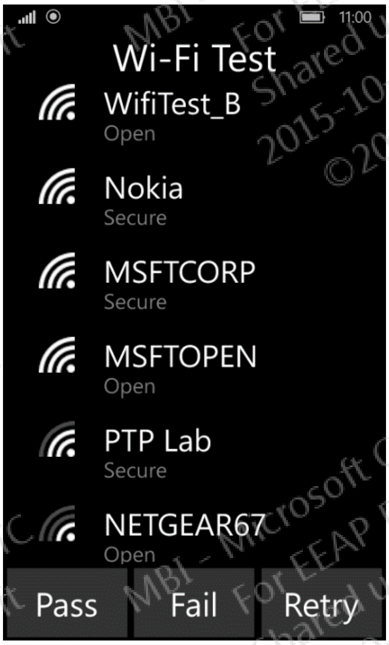


SIM Card Test

SIM Card 1 State:
No Card

Wi-Fi Test

- Checks if any Wi-Fi AP detected
  - Test engineer must wait to see at least one Wi-Fi AP is listed.
  - This test monitors changes of Wifi networks detected and update the information on screen accordingly.
- Test Engineer Feedback
  - Pass/Fail/Retry



Wi-Fi Test

WifiTest_B
Open

Nokia
Secure

MSFTCORP
Secure

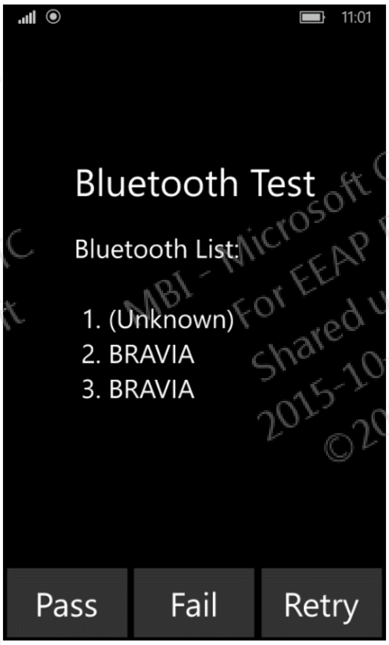
MSFTOPEN
Open

PTP Lab
Secure

NETGEAR67
Open

Bluetooth Test

- Checks if a Bluetooth device can be detected
  - Test engineer must wait to see if at least one Bluetooth device is detected or until the Bluetooth device search times out.
- Test Engineer Feedback
  - Pass/Fail/Retry

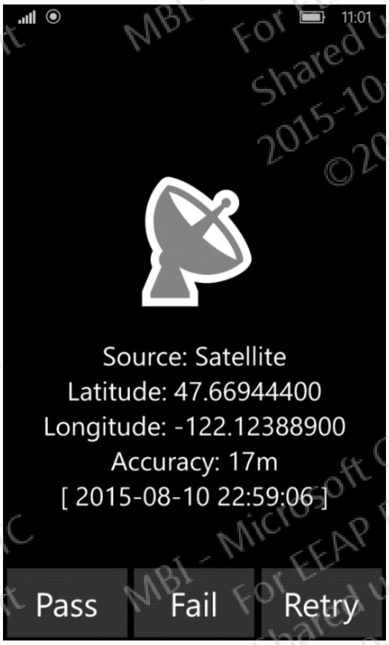


Bluetooth Test

Bluetooth List:

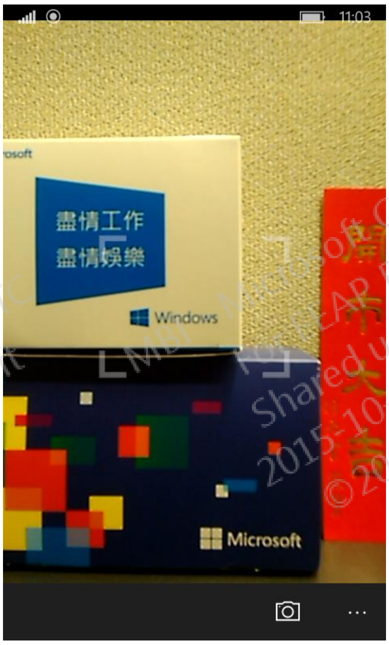
1. (Unknown)
2. BRAVIA
3. BRAVIA

GPS Test


Source: Satellite
Latitude: 47.66944400
Longitude: -122.12388900
Accuracy: 17m
[2015-08-10 22:59:06]

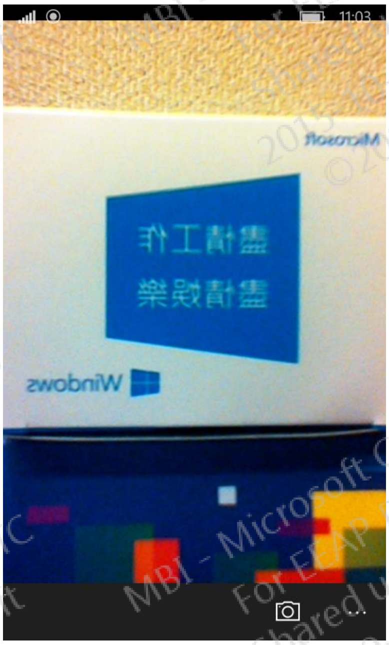
- Checks if a GPS satellite signal can be detected
  - While the position is fixed, the position data will be shown on screen.
  - Test engineer must wait to see if there is a GPS message shows up.
- Test Engineer Feedback
  - Pass/Fail/Retry

Rear Camera Test



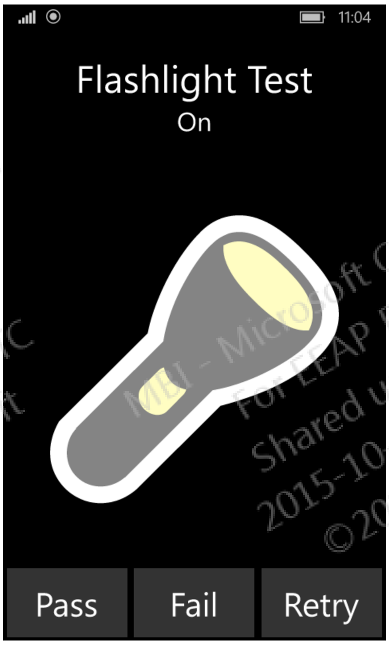
- Test engineer observes if camera preview/auto focus (if supported)/capture function are OK.
  - Default focus mode is continuous-focus, the focus frame will be found at the center of preview window.
  - If the camera supports touch to focus, the test engineer may touch the camera preview window, the focus frame will show where the test engineer touched the preview window, and at the same time trigger an auto-focus at that region.
  - Test engineer should press the shutter button to trigger picture shot to proceed to next page to review captured picture.
  - After reviewing captured picture, test engineer should select the test result from the feedback section of the screen.
- Test Engineer Feedback
  - Pass/Fail/Retry

Front Camera Test



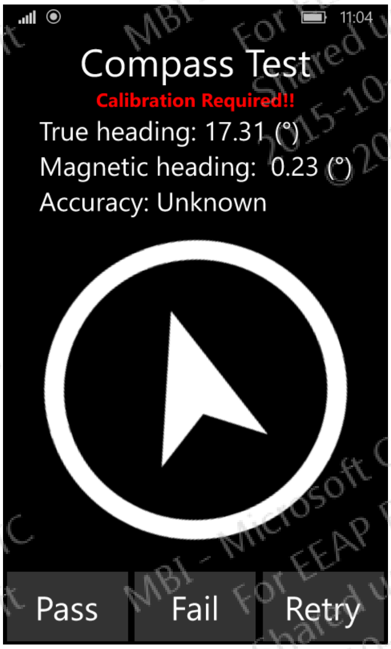
- Test engineer observes if camera preview/capture function are OK.
  - Test engineer should press the shutter button to trigger picture shot to proceed to next page to review captured picture.
  - After reviewing the captured picture, test engineer should select the test result.
- Test Engineer Feedback
  - Pass/Fail/Retry

Flash Light Test



- Test engineer observes if flashlight LED is lighting as expected.
  - Flashlight LED will be on and off periodically until the test engineer provides feedback.
- Test Engineer Feedback
  - Pass/Fail/Retry

Compass Test



Compass Test

Calibration Required!!

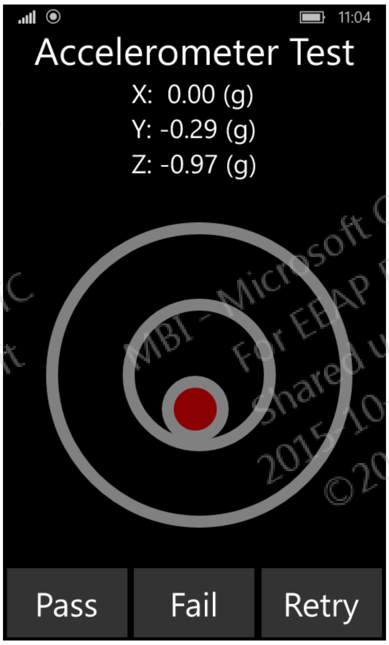
True heading: 17.31 (°)

Magnetic heading: 0.23 (°)

Accuracy: Unknown

- Test engineer observes if compass sensor returns values.
  - Test engineer should change the heading of the device to check if the value/status changes accordingly.
  - For some devices that require to do sensor calibration, there will be a warning shown on the screen to alert test engineer to do calibration.
- Test Engineer Feedback
  - Pass/Fail/Retry

Accelerometer Test



Accelerometer Test

X: 0.00 (g)

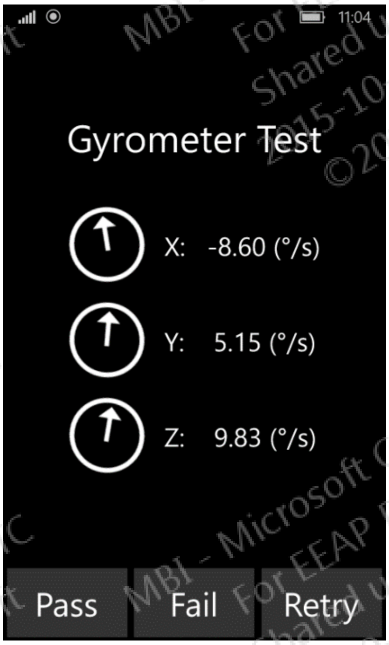
Y: -0.29 (g)

Z: -0.97 (g)

- Test engineer observes if the accelerometer sensor returns values.
  - Test engineer should move the device towards different orientation to check if the value changes accordingly.
- Test Engineer Feedback
  - Pass/Fail/Retry

Gyro meter Test

- Test engineer observes if the gyro meter sensor returns values.
  - Test engineer should rotate/spin/flip the device a little bit to see if value changes accordingly.
  - To avoid automatic display rotation, test engineer should change the system to "tablet mode", or set in action center to lock rotation to "landscape".
- Test Engineer Feedback
  - Pass/Fail/Retry



Gyrometer Test

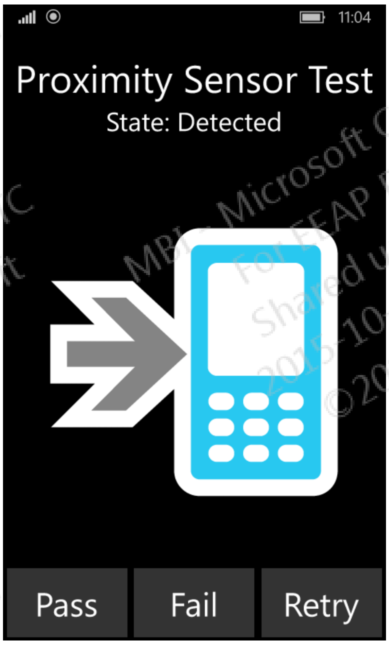
X: -8.60 (°/s)

Y: 5.15 (°/s)

Z: 9.83 (°/s)

Proximity Sensor Test

- Test engineer observes if the proximity sensor returns values.
  - Test engineer should touch where the proximity sensor is installed to see if the state value changes accordingly.
- Test Engineer Feedback
  - Pass/Fail/Retry

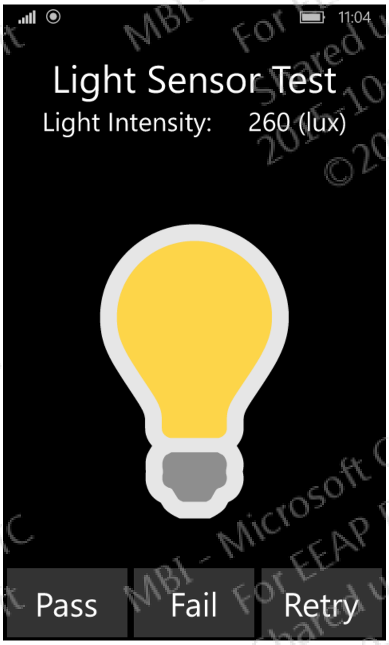


Proximity Sensor Test

State: Detected

Light Sensor Test

- Test engineer observes if the ambient light sensor returns values.
    - Test engineer should try to change the luminance condition at where ambient light sensor is installed to see if the value changes accordingly.
- Test Engineer Feedback
    - Pass/Fail/Retry

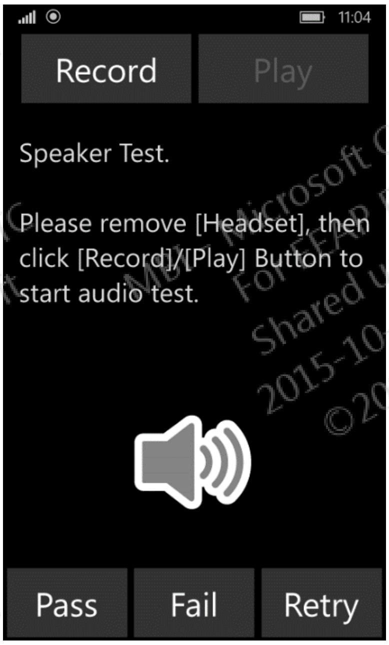


Light Sensor Test

Light Intensity: 260 (lux)

Speaker Test

- Test engineer observes if audio recording (mic) and audio playback (speaker) can work correctly.
    - Test engineer should generate some sounds for recording, and then check if the playback is OK.
    - Since this test is for the speaker and mic installed on the device, please make sure there is no headset connected
- Test Engineer Feedback
    - Pass/Fail/Retry

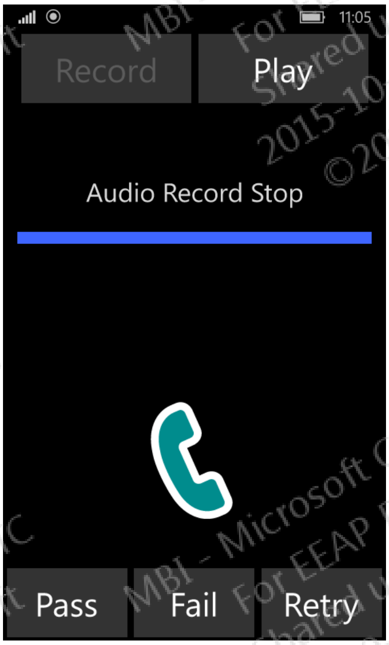


Speaker Test.

Please remove [Headset], then click [Record]/[Play] Button to start audio test.

Earpiece Test

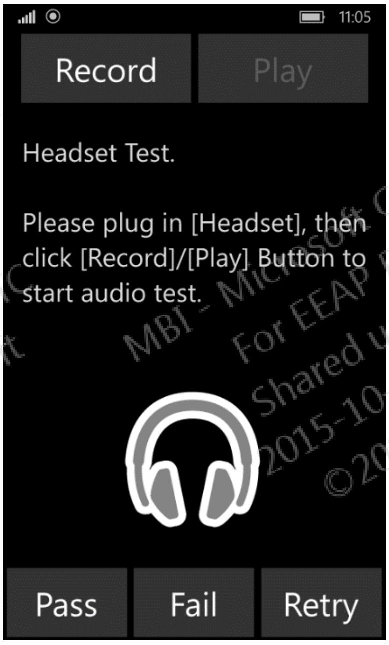
- Tester observes if audio can be heard from an earpiece.
- Tester Feedback
  - Pass/Fail/Retry



Audio Record Stop

Headset Test

- Test engineer observe if audio recording (headset mic) and audio playback (headset earphone) function OK.
  - Test engineer should generate some sound for recording, and then check if the playback is OK.
  - Headset is required for this test
- Test Engineer Feedback
  - Pass/Fail/Retry

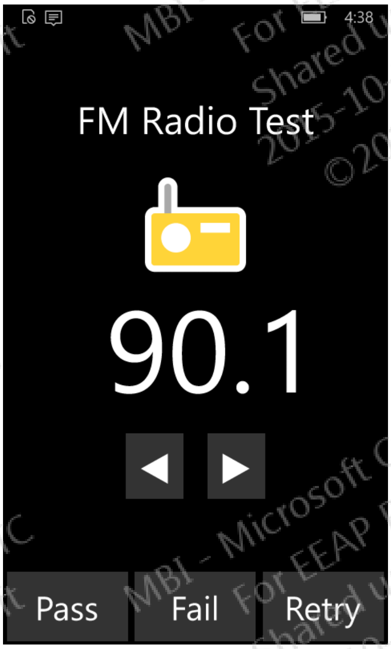


Headset Test.

Please plug in [Headset], then click [Record]/[Play] Button to start audio test.

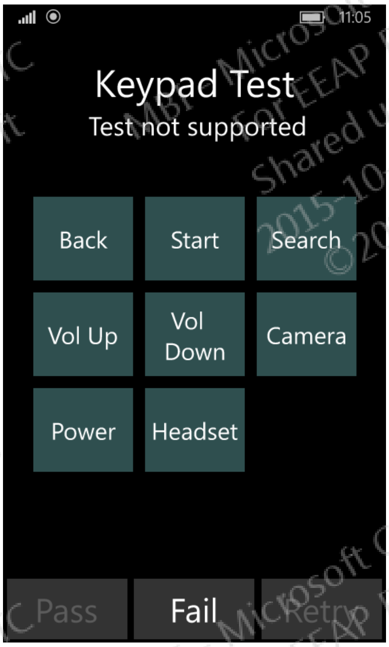
FM Radio Test

- Test engineer observes if the FM radio works as expected.
  - Tester should press the Next and Previous buttons to scan for an available FM channel. Before a channel is found, noise should be heard as an expected result.
  - A headset must be plugged in for this test.
  - If "FM Radio" application has been run, then this test will fail to initiate FM Radio receiver, device must be reboot to recover this error.
  - FM Radio test will fail on devices which image is built with BSP that's not natively support windows 10.
- Test engineer Feedback
  - Pass/Fail/Retry



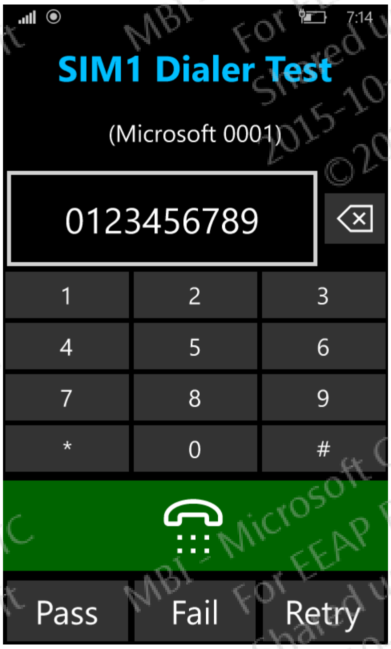
Keypad Test (Currently Not Support)

- Test engineer observes if the following hardware buttons are functioning correctly.
  - Test engineer should press the physical hardware buttons (according to those button icons on screen) one by one.
  - The button icon color changes to green if the corresponding hardware button is pressed.
  - When all buttons are pressed, the test will decide the result as "Pass".
  - Test engineer may press physical "Back" button 3 times to fail this test.
- Test engineer Feedback
  - Pass/Fail/Retry



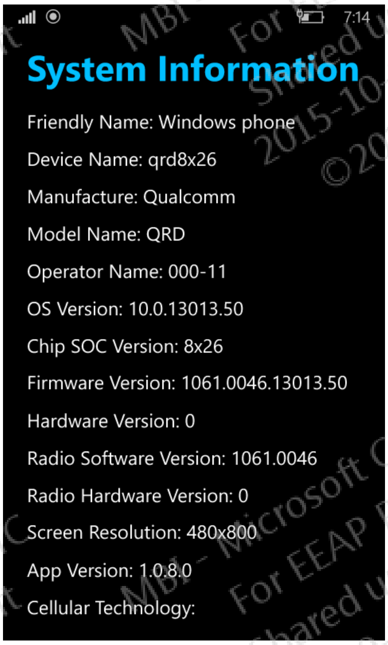
Dialer Test

- Test engineer observes if the phone call could be made as expected.
    - Test engineer could use default number and just press the green "Call" button to make the call and see if result is as expected.
    - Tester may choose not to use the default number, and dial a desired number to make the phone call.
    - Once the call is in progress, the green "Call" button will become "Hang up" button for test engineer to hang up the call.
    - If the phone device support dual SIM cards, after first SIM card phone call test is finished, the test will proceed to next SIM card phone call test page.
- Test engineer Feedback
    - Pass/Fail/Retry



System Information

- Select "System Information" from tool function page to show the system (or device) information page.
- "App Version" is the version number of this SFT App.



System Information

Friendly Name: Windows phone

Device Name: qrd8x26

Manufacture: Qualcomm

Model Name: QRD

Operator Name: 000-11

OS Version: 10.0.13013.50

Chip SOC Version: 8x26

Firmware Version: 1061.0046.13013.50

Hardware Version: 0

Radio Software Version: 1061.0046

Radio Hardware Version: 0

Screen Resolution: 480x800

App Version: 1.0.8.0

Cellular Technology: