|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SITE  SHANGHAI | | | | Software specifications | | |
| Originator  Cui Dahui | | | | **SOFTWARE Technical note** | | |
| How to work in windows phone 8909 | | | | | | |
| **Domain: Architecture**  **Rubric:** technical document | | | | | | |
| **CONTENTS:** | | | | | | |
| **KEY WORDS :** Windows phone | | | | | | |
| **DISTRIBUTION LIST** | |  | | | | |
|  | |  | | | | |
| *MPD/DTD/SWD* | | SHEN JIAN;CUI DAHUI;ZHANG XUDONG;xubing;hubo | | | | |
| *MPD/DTD/SAD* | |  | | | | |
| *MPD/PPD* | |  | | | | |
| *MPD/VAL* | |  | | | | |
| *MPD/MPSD* | |  | | | | |
| *MPD/design* | |  | | | | |
| *MPD/Quality* | |  | | | | |
|  | |  | | | | |
|  | |  | | | | |
| \* = mandatory reader | |  | | | | |
|  | AUTHOR | | APPROVALS | | | QUALITY | |
|  |  | | LEVEL 1 | | LEVEL 2 |  | |
| NAME | Cui Dahui | | SHEN JIAN | | ZHANGXUDONG |  | |
| FUNCTION | Team leader | |  | |  |  | |
| DATE |  | |  | |  |  | |
| SIGNATURE |  | |  | |  |  | |

HISTORY

|  |  |
| --- | --- |
| 2015-08-12 | Create draft |
| 2015-08-20 | Hu Bo |
| 2015-09-04 | Hu Bo Update AK to 10240 and Baseline to 1074 |
| 2015-09-11 | Hu Bo   1. Update environment variable for QWPCT build spkg. 2. Update Compile code method |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Table**

[1. Windows Phone architecture overview 5](#_Toc427409412)

[2. Wp Env setup prerequisites(PC) 6](#_Toc427409413)

[3. How to get source code from git 9](#_Toc427409414)

[4. How to complier your code 10](#_Toc427409415)

[5. Package and subsystem 14](#_Toc427409416)

[6. Create FFU image use IcD tool 15](#_Toc427409417)

[7. Image (BIN) type 17](#_Toc427409418)

[8. Download image 19](#_Toc427409419)

[9. How to change phone to MMOS mode 20](#_Toc427409420)

[10. Flash a phone image to a device 20](#_Toc427409421)

[11. Manufacturing 21](#_Toc427409422)

[12. Debugging Windows Phone 23](#_Toc427409423)

**Images ：**There are various kinds

**FFU images**

Sector-based format used by Windows Phone

**WIM images**

File-based format used by Windows

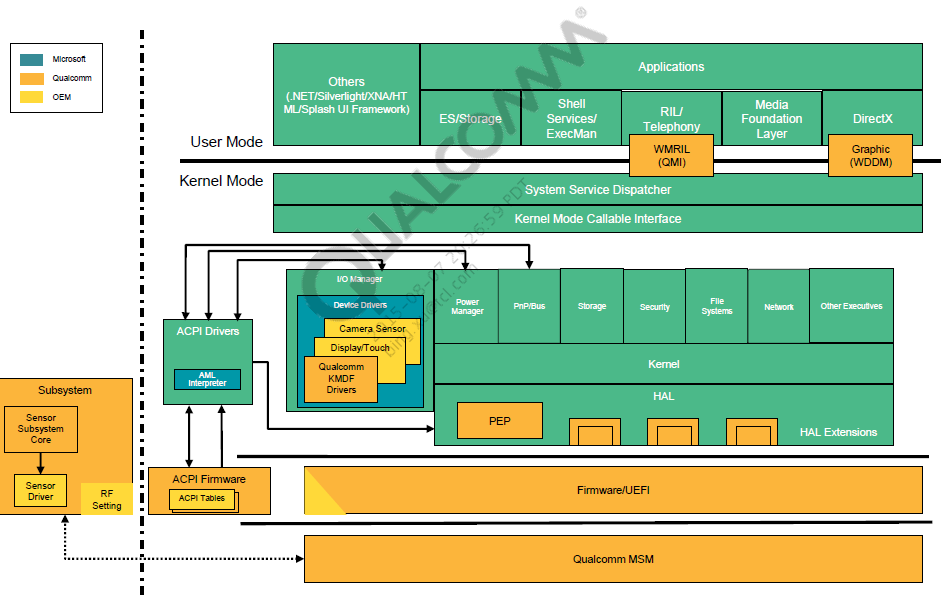
Can be made bootable so Windows Phone also uses them for

Update OS image, MMOS image

**MBN images**

Qualcomm image format for their firmware images

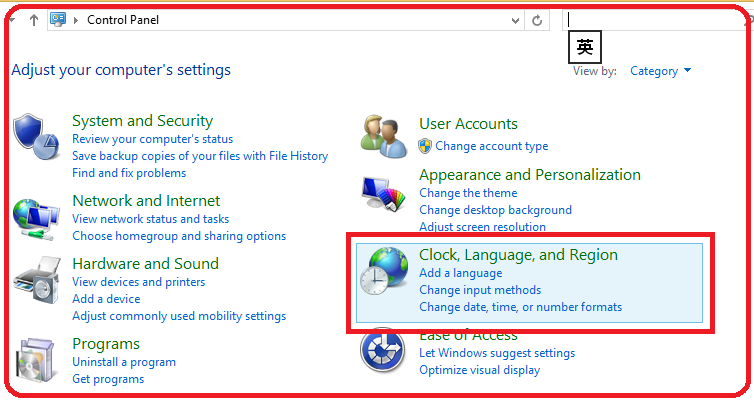
# Windows Phone architecture overview

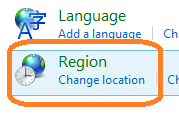


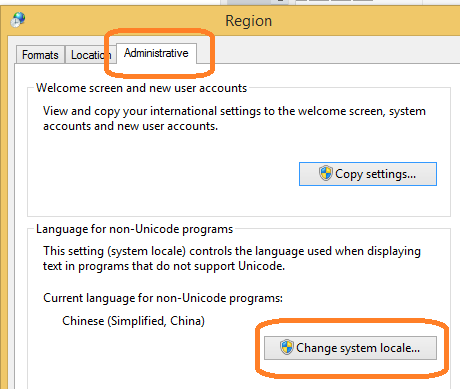
# Wp Env setup prerequisites(PC)

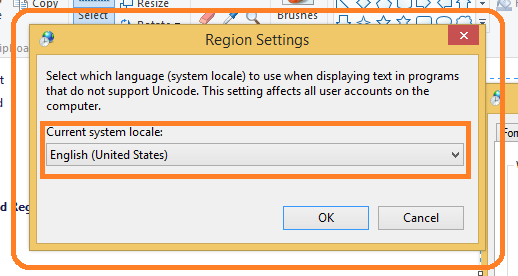
**Set system language:**

Control panel –>Clock, Language, and Region->Region->Administrative->Change system locale->English (United States)









The following software are operating system requirements

[Visual Studio 2013](mk:@MSITStore:E:\WinPhone-环境\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/7b1cd6ec-5aba-41e9-8f90-488f8cb9f02d.htm#vs)

[Install the AK10](mk:@MSITStore:E:\WinPhone-环境\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/7b1cd6ec-5aba-41e9-8f90-488f8cb9f02d.htm#sdk)

[Install OEM test certificates](mk:@MSITStore:E:\WinPhone-环境\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/7b1cd6ec-5aba-41e9-8f90-488f8cb9f02d.htm#testcerts)

[Install Windows symbols](mk:@MSITStore:E:\WinPhone-环境\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/7b1cd6ec-5aba-41e9-8f90-488f8cb9f02d.htm#symbols)

[Install other tools](mk:@MSITStore:E:\WinPhone-环境\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/7b1cd6ec-5aba-41e9-8f90-488f8cb9f02d.htm#tools)

Install qualcomm tool

[Set up environment variables](mk:@MSITStore:E:\WinPhone-环境\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/7b1cd6ec-5aba-41e9-8f90-488f8cb9f02d.htm#envvar)

[**Uninstalling earlier versions of the tools and development kits**](mk:@MSITStore:E:\WinPhone-环境\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/7b1cd6ec-5aba-41e9-8f90-488f8cb9f02d.htm#uninstalling) **, specially win8.1 ENV**

**Steps of installing software for compile:**

1. Install vs2013-pro （[\\172.24.220.210\share\jrd-tool\microsoft-win10\vs2013-pro](file:///\\172.24.220.210\share\jrd-tool\microsoft-win10\vs2013-pro)）

**You should run “vs\_professional.exe” as Administrator** and not change the setup directory and any default options.

1. Install all Kits such as adk, hlk etc.

([\\172.24.220.210\share\jrd-tool\microsoft-win10\10240\Kits](file:///\\172.24.220.210\share\jrd-tool\microsoft-win10\10240\Kits))

**You should run all setup files as Administrator** and not change the setup directory.

1. Install ADK
2. Install powerAddOn
3. Install Standalonesdk
4. Install WDK
5. Install WDKVisualStudioDev12 for vs patch
6. Install HLK
7. HLK\StrongNameSigning

The following steps only need to be performed ONCE per OS installation. All actions need to be performed from an account with administrator privileges on the controller, and any clients with a RTM’d OS installed.

* Install the test-certificate provided in the StrongNameSigning folder (testroot and testroot-SHA2”).

To install,

a. Right click on the certificate.

b. Select “Install Certificate”

c. Select “Next”

d. Accept default for certificate store, and select “Next”

e. Select “Finish”

* Disable Strong-name signing. This is done by installing the provided registry keys located in the \Kits\HLK\StrongNameSigning\ folder:

StrongNameBypass.reg

WOW64StrongNameBypass.reg

To install,

a. From the controller machine right-click on the registry key.

b. Select “Merge”

c. Select “Run”

d. Select “Yes”

1. COPY MobileOS ALL SUBFOLDER to C:\Program Files (x86)\Windows Kits\10

（[\\172.24.220.210\share\jrd-tool\microsoft-win10\10240\MobileOS](file:///\\172.24.220.210\share\jrd-tool\microsoft-win10\10240\MobileOS)）

1. Arm complie tool RVCT5

Copy setup zip file (DS500-PA-00002-r5p0-05rel0.zip) to your own PC and unzip it to install:

([\\172.24.220.210\share\jrd-tool\arm](file:///\\172.24.220.210\share\jrd-tool\arm))

Do not change any default setup options.

1. Thirdparty tools：

Git, MiniGW, perl, python, Sygwin

**You should run all setup files as Administrator.**

([\\172.24.220.210\share\jrd-tool\thirdparty](file:///\\172.24.220.210\share\jrd-tool\thirdparty))

Python:

When you install python, should follow the following sequence:

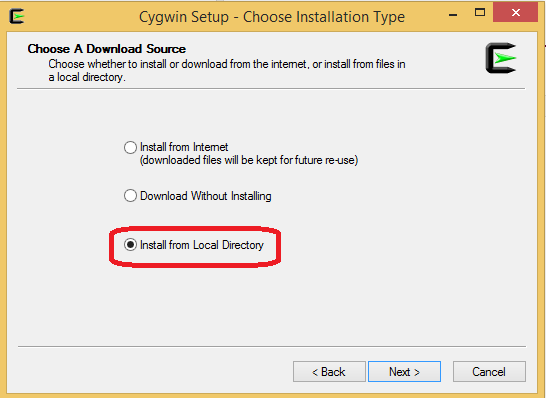
a. python-2.7.6.amd64

b. pywin32-219.win-amd64-py2.7

c. python-2.6.6.amd64

Sygwin:

Please select “Install from Local Directory”



1. HEXAGON\_Tools dev by qualcomm and will update according the BSP version

([\\172.24.220.210\share\jrd-tool\HEXAGON\_Tools\6.4.05](file:///\\172.24.220.210\share\jrd-tool\HEXAGON_Tools\6.4.05))

Copy the directory “6.4.05” to the path “C:\Qualcomm\HEXAGON\_Tools”.

If the directory does not exist. Please create a new folder “Qualcomm” and sub folder named “HEXAGON\_Tools”

1. Install OEM test certificates by running the script “C:\Program Files (x86)\Windows Kits\10\Tools\bin\i386\ InstallOEMCerts.cmd” (this is a one-time step).
2. Add a new environment firstly

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

Attention, the value of environment should be end with “\”

1. Run “InstallOEMCerts.cmd” as Administrator
2. Replace the following files:
3. Replace “C:\Program Files (x86)\Windows Kits\10\build\ WindowsDriver.Default.props” with “[\\172.24.220.210\share\WindowsDriver.Default.props](file:///\\172.24.220.210\share\WindowsDriver.Default.props)”.

1. Replace “C:\Program Files (x86)\Microsoft Visual Studio 12.0\Common7\Tools\ vsvars32.bat” with “\\172.24.220.210\share\ vsvars32.bat”
2. Replace “C:\Program Files (x86)\Microsoft Visual Studio 12.0\Common7\Tools\ VCVarsQueryRegistry.bat” with “\\172.24.220.210\share\ VCVarsQueryRegistry.bat”

9． Merge the register file about SDK 10“\\172.24.220.210\share\ SDK.reg”

for the error of can’t find /warning.h.

**Steps of setting environment variables:**

Set the environment variables as the following:



Attention:

* Some variables should be set to your own information please. Such as %HOMEPATH%, %COMPUTERNAME% etc.
* Please set “C:\Program Files (x86)\Windows Kits\10\Tools\bin\i386;” to the front of the environment variable “path”.

When you use QWPCT to build spkg, we should ensure signtool.exe in the directory “C:\Program Files (x86)\Windows Kits\10\Tools\bin\i386” will be used instead of in the other directories.Otherwise signtool.exe will crash and lead to build spkg fail.

**Steps of installing software for debug or download:**

* 1. Install USB driver:

([\\172.24.220.210\share\jrd-tool\Android\_USB\_Driver\_Q\_7.0.0.0\_2014\_10\_16\_All\_in\_One\_noinstall](file:///\\172.24.220.210\share\jrd-tool\Android_USB_Driver_Q_7.0.0.0_2014_10_16_All_in_One_noinstall))

* 1. Install Symbols to C:\Symbols

([\\172.24.220.210\share\jrd-tool\microsoft-win10\10240\Symbols](file:///\\172.24.220.210\share\jrd-tool\microsoft-win10\10240\Symbols))

* 1. Qualcomm tools for download and debug

（[\\172.24.220.210\share\jrd-tool\qualcomm](file:///\\172.24.220.210\share\jrd-tool\qualcomm)）

a. QPST.2.7.421-72-V1400-157

b. qxdm-72-V1189-173

c. qcat

d. QDART-72-VB981-72\_00024

e. EnableQCComp(debug tool)

* + - * install drive of qualcomm usb QUD.WIN.1.1 Installer-10030.1
      * Connect with WP and Put device in mass storage mode
      * run attached script: N is the drive letter for MainOS partiton.

([\\172.24.220.210\share\jrd-tool\qualcomm\EnableQCComp](file:///\\172.24.220.210\share\jrd-tool\qualcomm\EnableQCComp))

enableQCComp.bat N:\ SerialCompositeConfig

# How to get source code from git

You can get code from our server

1. Create your local folder in your computer just as D:\WP
2. Entry the folder cd /d D:\WP, right click the directly and click “Git Bash”
3. Input the following cmd to download code

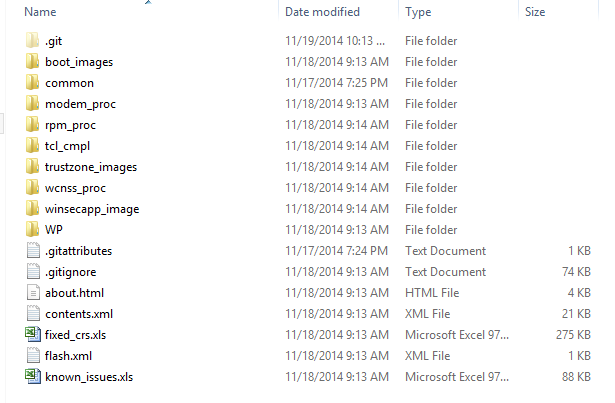
git clone [git@172.24.61.94:winphone/amss\_8909.git](mailto:git@172.24.61.94:winphone/amss_8909.git)

1. after download finish ,your need to checkout code branch:

cd /d D:\WP\amss\_8909

git checkout amss-pop3-5.5-dev

1. make sure you can see the source code as follow



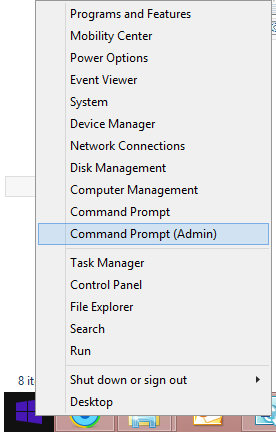
Some useful git commnd

Git add /git commit /git push /git pull /git clean –df/gitk …………….

# How to complier your code

1. Run command as Administrator firstly.

Right click on the bottom left corner and click “Command Prompt (Admin)”.



1. Please call wp\_build\_set\_env.bat to set environment variables before you build.



Attention:

Please set “%SOURCECODE\_ROOT\_DIR%” to your own code path!

1. Complier subsystem and boot

You also can use \int\_tools\make\make.bat to compile subsystem and boot without input the compile command.

Please refer \int\_tools\make\readme.txt to use make.bat.

make –b uefi/make –b modem/make –b bootloader/make -b tz/make -b rpm

or you can use

make –b all

to build all

And you can user

Make –c all

to clean all before you build.

Tip: If use make.bat to compile code, you need not to call wp\_build\_set\_env.bat, as it has been called in make.bat before compile.

Compile command:

echo \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

echo \*\*\*\*\*\*\*\*\*\*\*\*\*Build UEFI begin\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

echo \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

cd \WP\src\uefi\rel\9.2\edk2\QcomPkg\Msm8909Pkg

b.bat -cfg uefiplatWP.cfg RELEASE

echo \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

echo \*\*\*\*\*\*\*\*\*\*\*\*\*Build MODEM begin\*\*\*\*\*\*\*\*\*\*\*\*\*\*

echo \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

cd \modem\_proc\build\ms

build.cmd 8909.gen.prod -k

echo \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

echo \*\*\*\*\*\*\*\*\*\*\*\*\*Build BOOTLOADER begin\*\*\*\*\*\*\*\*\*\*\*\*\*\*

echo \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

cd \boot\_images\build\ms

build.cmd --prod TARGET\_FAMILY=8909

echo \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

echo \*\*\*\*\*\*\*\*\*\*\*\*\*Build TRUSTZONE begin\*\*\*\*\*\*\*\*\*\*\*\*\*\*

echo \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

cd \trustzone\_images\build\ms

build.cmd CHIPSET=msm8909 tz sampleapp tzbsp\_no\_xpu playready widevine keymaster commonlib

echo \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

echo \*\*\*\*\*\*\*\*\*\*\*\*\*Build RPM begin\*\*\*\*\*\*\*\*\*\*\*\*\*\*

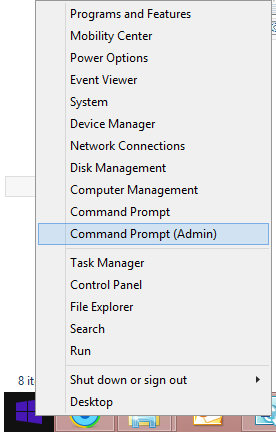
echo \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

cd \rpm\_proc\build

build\_8909.bat

1. Complier ACPI and all driver :
2. Run command as Administrator firstly.

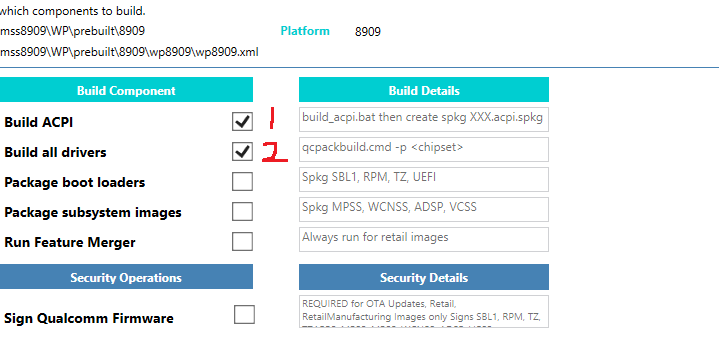
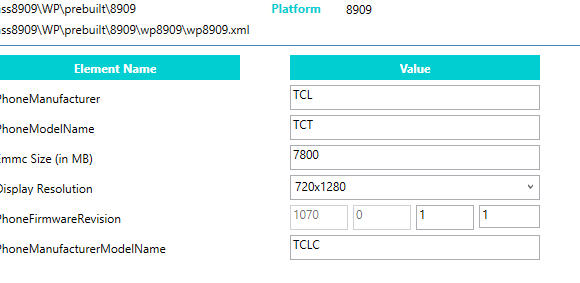
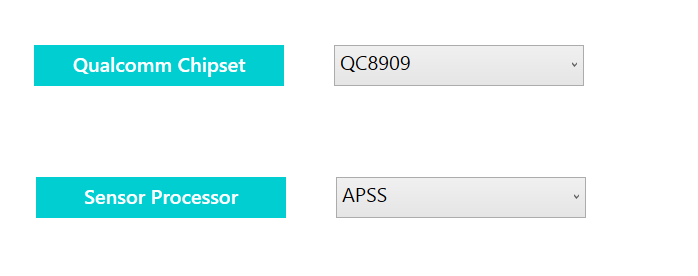
Right click on the bottom left corner and click “Command Prompt (Admin)”.



1. Entry \amss\_8909\WP\prebuilt\8909\app\Qwpct
2. Attention, please call wp\_build\_set\_env.bat to set environment variables before you build.

Please get wp\_build\_set\_env.bat from 4.b (Page 15)

1. Execute the Qwpct.exe
2. select BSP config file amss\_8909\WP\prebuilt\BSP.config.xml

Build ACPI with command line: msbuild acpi.wp63.vcxproj /p:platform=arm /p:targetversion=windows10 /p:configuration=Release

# Package and subsystem

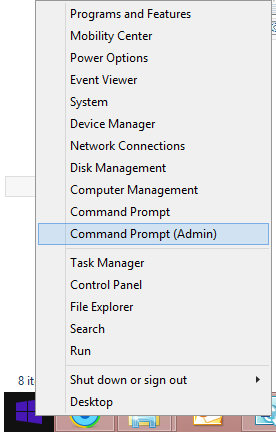
After complier done , maybe you need package mbn/bin/package files to pkg use QWPCT in your project amss\_8909\WP\prebuilt\app\Qwpct, right clip icon

Attention, please call wp\_build\_set\_env.bat to set environment variables before you build.

**Package boot loaders and package susbsystem drivers:**

1. Run command as Administrator firstly.

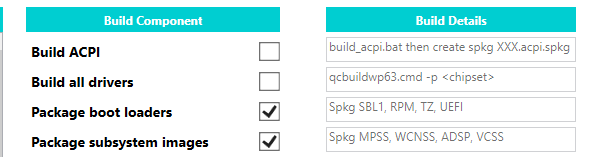
Right click on the bottom left corner and click “Command Prompt (Admin)”.



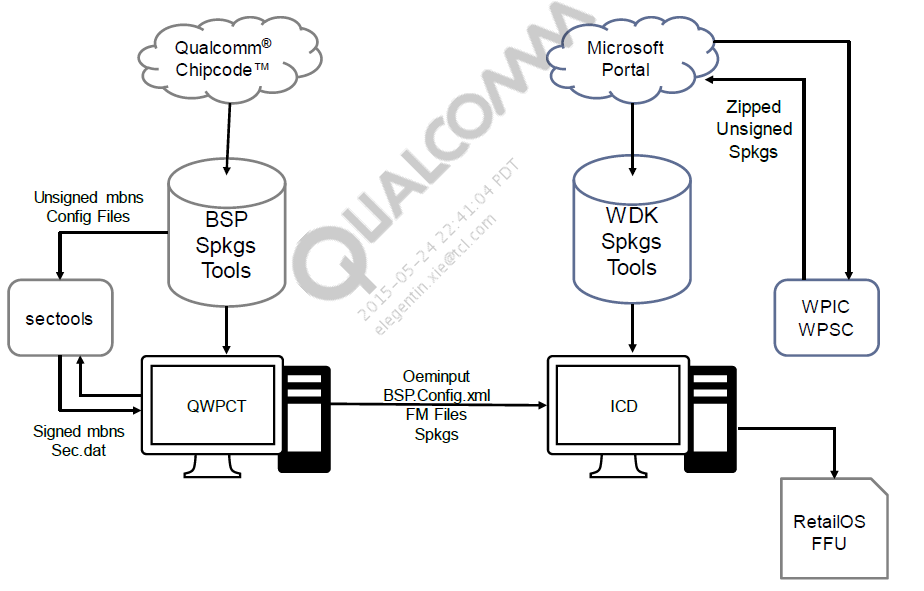
1. Entry \amss\_8909\WP\prebuilt\8909\app\Qwpct
2. Attention, please call wp\_build\_set\_env.bat to set environment variables before you build.

Please get wp\_build\_set\_env.bat from 4.b (Page 15)

1. Execute the Qwpct.exe
2. select BSP config file amss\_8909\WP\prebuilt\BSP.config.xml

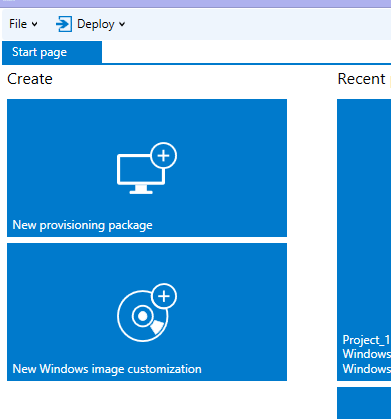


# Create FFU image use IcD tool



You can find IcD tool in "C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Imaging and Configuration Designer\x86\ICD.exe"

Please run it use administrator



\\172.24.220.210\share\3. WP10 Customization v2 and Build Image.pdf

# Image (BIN) type

|  |  |  |
| --- | --- | --- |
| Image type | Description | OEMInput samples |
| Retail | Retail images are the images that are flashed to final retail phones. Retail images must use Microsoft-signed packages that are returned to OEMs after submitting production images to Microsoft by using the OEM submission tool. For more info, see [Submit binaries to be retail signed](mk:@MSITStore:E:\WinPhone-环境\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/66bff648-a159-446e-a72b-0377ebe92f57.htm).  Retail images include the following:   * Production version of core Windows components included in Windows Phone. * Production Windows Phone OS components | RetailOEMInput.xml |
| Production | Production images are similar to final retail images, but they have test signing enabled to run OEM-signed components as well as production-signed components, and they may contain test-related packages as well as production packages. Production images can be used for engineering work as well as mobile operator trials and other certification processes. Production images are submitted to Microsoft by using the OEM submission tool to be production signed by Microsoft before generating the final retail image. For more info, see [Submit binaries to be retail signed](mk:@MSITStore:E:\WinPhone-环境\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/66bff648-a159-446e-a72b-0377ebe92f57.htm).  Production images include the following:  Production version of core Windows components included in Windows Phone.  Production Windows Phone OS components.  Test signing enabled. | ProductionOEMInput.xml |
| Test | Test images can be run in offsite test labs to test the functionality of the OS and drivers on a device. Test images include the following:  Test version of core Windows components included in Windows Phone.  Production Windows Phone OS components.  Test signing enabled.  Test applications, drivers, and other components to use for testing the OS in different conditions.   |  | | --- | | To generate an image that includes OS tools such as ipconfig.exe, kill.exe, ping.exe, minshutdown.exe, reg.exe, tracelog.exe, sc.exe, and tlist.exe, build a test image. | | TestOEMInput.xml |
| Health | Health images to be run in offsite test labs to test the power and performance capabilities of the device. Health images are similar to production images, with the addition of components for running tests related to power and performance. | HealthOEMInput.xml |
| Manufacturing | Manufacturing images to be used in the manufacturing environment. For more info, see [MMOS image definition](mk:@MSITStore:E:\WinPhone-环境\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/3bc8ed74-81a4-43d0-8ab6-5a321c16570e.htm).. | MfgOEMInput.xml |
| Customer care | Customer care images include MMOS for retail customer care scenarios. For more info, see [MMOS image definition](mk:@MSITStore:E:\WinPhone-环境\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/3bc8ed74-81a4-43d0-8ab6-5a321c16570e.htm). | CustomerCareOEMInput.xml |

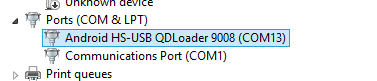
# Download image

Prepare and use EMMC download

1. 一： Android software: to winphone software
2. First program boot loaders Please check you phone if under mode 9008 download mode

Try booting up your phone and holding vol + and vol- key during boot(just holding vol + after first download); if it does not display the following screen, follow details in the firmware download section to recover the phone

Open your device manager ,pls make sure you can see



1. Copy prog\_emmc\_firehose\_8909\_lite.mbn、Flash.FFU to your emmcdl tools directory just as follow path
2. Move to \amss\_8909\WP\prebuilt\8909\app\emmcdl
3. emmcdl.exe -p COM3(need change) -f prog\_emmc\_firehose\_8909\_lite.mbn -ffu Flash.FFU
4. emmcdl.exe -p 1(need change ) -e 0 100000 in mode 9006 to entry 9008 emmc download mode
5. it will auto power on after download ready
6. ： in winphone mode
7. power off phone
8. long press power key then will entry mass store mode
9. plugin USB
10. \amss\_8909\WP\prebuilt\8909\app\emmcdl
11. emmcdl.exe -p 1 -e 0 100000 (pls sure yours disk num ,it dangerous)
12. success then plugout USB
13. Plugin USB phone will entry 9008 mode
14. Flash your FFU image to phone

# How to change phone to MMOS mode

Pls sure u in the FOLLOW mode :

Long press VOL+ then plugin USB



Open a commond prompt (admin)

ffutool -setBootMode 1 factory

normal mode

ffutool -setBootMode

# Flash a phone image to a device

* [Use the flashing tools provided by Microsoft](mk:@MSITStore:E:\WinPhone-环境\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/1e461311-9928-4380-b428-49b39811e385.htm)

Microsoft provides a tool set for flashing images to devices. This tool set includes ffutool.exe, Can custom

Initial device-side & Host-side

|  |
| --- |
| **Important note.1** |
| The flashing tool set provided by Microsoft should not be used in a manufacturing environment. For flashing images to phones during manufacturing, OEMs must build their own flashing tools by using the information provided in [Developing custom OEM flashing tools](mk:@MSITStore:E:\WinPhone-环境\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/f46c6203-8fd0-4299-9786-a4b4713c3f94.htm). |
| **Important note.2** |
| The device-side UEFI flashing application from Microsoft is automatically included in all phone images. This application must be included in all retail phones.  The OEM may need to develop a custom flashing tool to address the life cycle needs of the phone. Other flashing options have limitations that the OEM should understand before deciding to use them.  The following table summarizes the flashing tool options.   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Scenario** | **Microsoft FFU Engineering Tool** | **OEM Custom FFU Tool** | **SoC Provided Manufacturing Flashing tool** | **Gang Programmer** | | Engineering and Development | Yes | Yes | Yes | N/A | | Manufacturing | No | Yes | Yes | Yes | | Service Center | Yes | Yes | No | N/A | |

# Manufacturing

|  |
| --- |
| **Note** |
| Images prepared for manufacturing must meet the requirements described in [Retail OS requirements checklist](mk:@MSITStore:E:\WinPhone-env-doc\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/995c0e0c-34d5-4d3e-814b-5ec498772b03.htm). |

Simple factory flow



**Board tests/SMT** – Image is flashed via gang programmer.

**Final assembly, Boot** – Marry board with plastic; the first time the device is booted on the manufacturing floor.

**Manual tests** – Line worker runs device tests such as sound, vibration, camera, keyboard, and so on.

**RF/Call testing** – Automated testing in which the device is tethered to enable power and the recording of test data.

**Final provisioning** – Automated process where IMEI data is written, customizations are loaded, and labeling is completed.

**Final QA/Packaging** – Final manual verification of the device, then packaging.

**Random sample testing** – A specified number of devices are removed from packaging and tested. If failures reach a certain threshold, the entire line may be recalled.

guidance that works for your manufacturing processes and business

Manufacturing process option 1: boot from WIM MMOS image

The advantage of using this approach for servicing is that you will not need to reserve space on the retail OS for code that is only used in servicing. Minimizing the space that is consumed by the OS is an important consideration in low cost devices.

|  |
| --- |
| **Important note** |
| Only MMOS test images are currently supported. Retail signing is not currently supported. |

see [Working with WIM MMOS images](mk:@MSITStore:E:\WinPhone-env-doc\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/e09e6e71-d83d-4858-806c-8ae754a1dd5f.htm).

Manufacturing process option 2: reflash the phone

One tradeoff in this approach is that the manufacturing line must be designed to accommodate the reflashing time that occurs near the end of the manufacturing process.

see [Flashing tools](mk:@MSITStore:E:\WinPhone-env-doc\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/5eede945-b47b-40a3-9be1-ca38ab7343d2.htm).

# Debugging Windows Phone

1. The host computer and target phone are connected by a USB cable
2. Debugging Windows Phone requires tools that are included in the Windows Phone Kit
3. Debugging is supported only when the target phone is running a non-retail version of Windows
4. TOOLS NEED
   1. Debugging Tools for Windows
   2. TShell: This is a Windows PowerShell plug-in. With TShell, you can control the target phone by entering commands on the host computer. For more information, see [TShell](mk:@MSITStore:E:\WinPhone-env-doc\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/16fc2a98-582d-4da3-b5a4-28a92cbdb01b.htm).
   3. Virtual Ethernet (VirtEth.exe) This is an application that runs on the host computer. It provides Ethernet connectivity over USB. VirtEth.exe is part of the KDBG Connectivity package.
   4. IpOverUSB see [IP over USB](mk:@MSITStore:E:\WinPhone-env-doc\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/e361f9ff-7e47-4ba9-8891-fa0d1649ac3a.htm). feature is typically used for transferring files or testing programs via FTP or Telnet, and for debugging
5. Kernel mode and user mode debugging

In user-mode debugging, a debugger running on the host computer attaches to a single process that is running on the target phone. When you use the debugger to break in to the process, the threads in that process are stopped, but other threads keep running.

In kernel-mode debugging, a debugger attaches to the entire operating system running on the target phone. When you use the debugger to break in, all threads on the target phone are stopped.

For more information, see [User Mode and Kernel Mode](http://msdn.microsoft.com/library/windows/hardware/ff554836(v=vs.85).aspx).

Debugging Tools for Windows includes these debuggers:

* WinDbg – user mode or kernel mode
* CDB – user mode
* NTSD – user mode
* KD – kernel mode
* NTKD – kernel mode

[Set up Windows Phone debugging (kernel-mode)](mk:@MSITStore:E:\WinPhone-env-doc\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/c85486f0-9177-41df-a130-24077772d3ca.htm)

[Set up Window Phone debugging (user-mode)](mk:@MSITStore:E:\WinPhone-env-doc\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/6435b708-dbce-4809-9835-85b4e5449e10.htm)

1. Additional documentation for debugging Windows Phone

* [Debugger operation for Windows Phone](mk:@MSITStore:E:\WinPhone-env-doc\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/1e4b22fc-95ed-4ef5-b9ba-dd83911bec47.htm)
* [Symbol files for debugging Windows Phone](mk:@MSITStore:E:\WinPhone-env-doc\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/4f6ce8da-2760-443a-8295-bf0b199404a2.htm)
* [Windows Phone crash dump files](mk:@MSITStore:E:\WinPhone-env-doc\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/1696349c-eaa8-4782-a1de-b133e1e7d38b.htm)
* [Tools for Windows Phone debugging](mk:@MSITStore:E:\WinPhone-env-doc\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/d7a56942-576c-4e98-9312-3ee4118658d5.htm)
* [Debugging techniques for Windows Phone](mk:@MSITStore:E:\WinPhone-env-doc\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/c4d3f508-71b3-491d-869b-303eca3e48d5.htm)
* [Mass storage mode and boot configuration data](mk:@MSITStore:E:\WinPhone-env-doc\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/8b0400ac-5b62-4abd-ae18-ccc70693639a.htm)
* [Source code for Windows Phone debugging](mk:@MSITStore:E:\WinPhone-env-doc\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/043d2617-fe28-4e2f-bc92-c02d5070985f.htm)
* [Windows Phone debugging reference](mk:@MSITStore:E:\WinPhone-env-doc\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/bbb23eed-72f3-4c8a-a2fe-4472aad3fbcc.htm)
* [Debugger engine and extension APIs for Windows Phone](mk:@MSITStore:E:\WinPhone-env-doc\20140815-tang\WP_Blue_GDR_Documentation.chm::/html/50932c8d-d5e9-4786-abbd-39435356aced.htm)

PS:

**Link you can search keywords in WP\_Blue\_GDR\_Documentation.CHM**

YOU CAN FOLOW BELOW PATH TO GET PKG NEEDED BY WP10

[\\172.24.220.210\share\jrd-tool](file:///\\172.24.220.210\share\jrd-tool)

Please follow readme in above link to install, you add env variable first

Please get env\_var\_20150911.txt from page 12

And call wp\_build\_set\_env.bat before compile and build spkg

Please get wp\_build\_set\_env.bat from 4.b (Page 15)

According the chm file

[\\172.24.220.210\share\jrd-tool\WP\_Blue\_GDR\_Documentation.chm](file:///\\172.24.220.210\share\jrd-tool\WP_Blue_GDR_Documentation.chm)