

SDM850 Windows 10 Desktop BSP Setup and Build

Index:

1. Build machine and host OS (Windows 10 PRO) setup.....	2
2. BSP source building.....	4
2-1 non-HLOS.....	4
2-2 HLOS (WP).....	5
3. Create WinPE manually.....	7
4. Develop Win10 desktop new driver.....	8
5. Build WD Image with adding drivers.....	10
6. Flash Windows Desktop Builds	11

1. Build machine and host OS (Windows 10 PRO) setup

1. Set system locale to English

Settings -> Time & Language -> Region -> Additional date, time & regional settings

-> Region -> Administrative -> Language for non-unicode programs -> Change system locale... -> Current system locale:
Select "English (United States)"

2. Install .NET Framework 3.5

A. Execute "Win10 Settings" and follow bellows to install.

Settings ->[Apps & Features] ->[Programs and Features] ->[Turn Windows features on or off] ->[.NET Framework 3.5]
->check [**Windows Communication Foundation Non-HTTP Activation**]

B. You can install .NET Framework 3.5 from Windows ISO CD

```
# dism /online /enable-feature /featurename:netfx3 /all /limitaccess /source:E:\source\sxs
```

Ps: /source:[CD Drive]:\source\sxs , change [CD Drive] to yours

3. Change PowerShell policy for ACPI building

In Windows "Search" -> Input "Local Computer Policy"

->[Computer Configuration] ->[Administrative Templates] ->[Windows Components] ->[Windows PowerShell] ->[Turn on Script Execution] ->check [Enabled] ->[Option: Execution Policy] ->select [Allow all scripts]

4. Install EWDK

Download build version(match with BSP release note) of EWDK packages from M\$.

<https://docs.microsoft.com/zh-tw/windows-hardware/drivers/download-the-wdk>

Unzip it to your special path for example D:\.

5. Unzip Apps.zip

Unzip it to your special path for example D:\.

5. Modify environment parameter in setenv.bat in your BSP_ROOT path. (Used for nonHLOS building)

```
set APPS_PATH=D:\Apps

set PYTHON_PATH=%APPS_PATH%\Python27\2.7.6
set PYTHONPATH=%APPS_PATH%\Python27\2.7.6
set PYTHONBIN=%PYTHON_PATH%
set python_win=%APPS_PATH%\Python27\2.7.6

set ARMTOOLS=LLVM
set LLVM_VERSION=3.9.6
set CLANG_VERSION=3.9.6

set ARMLMD_LICENSE_FILE=8888@172.20.77.250
set ARM_COMPILER_TOOLCHAIN=%ARMTOOLS%\%LLVM_VERSION%
set ARM_COMPILER_PATH=%APPS_PATH%\%ARM_COMPILER_TOOLCHAIN%\bin
set ARMHOME=%APPS_PATH%\%ARM_COMPILER_TOOLCHAIN%
set ARMINC=%APPS_PATH%\%ARM_COMPILER_TOOLCHAIN%\include
set ARMLIB=%APPS_PATH%\%ARM_COMPILER_TOOLCHAIN%\lib
set ARMBIN=%APPS_PATH%\%ARM_COMPILER_TOOLCHAIN%\bin
set ARMPATH=%APPS_PATH%\%ARM_COMPILER_TOOLCHAIN%\bin
set ARMINCLUDE=%APPS_PATH%\%ARM_COMPILER_TOOLCHAIN%\include

set SD_LLVM_ROOT=%APPS_PATH%\%ARM_COMPILER_TOOLCHAIN%
set SD_LLVM_LIB=%APPS_PATH%\%ARM_COMPILER_TOOLCHAIN%\armv7m-none-eabi\lib
```

```
set SD_LLVM_LIBC=%APPS_PATH%\%ARM_COMPILER_TOOLCHAIN%\armv7m-none-eabi\libc
set SD_LLVM_RTLIB=%APPS_PATH%\%ARM_COMPILER_TOOLCHAIN%\lib\clang\%CLANG_VERSION%\lib\baremetal

set HEXAGON_ROOT=%APPS_PATH%\Qualcomm\HEXAGON_Tools
set HEXAGON_RTOS_RELEASE=8.1.04
set HEXAGON_Q6VERSION=v65
set HEXAGON_IMAGE_ENTRY=0x8E000000

set MAKE_PATH=%APPS_PATH%\%ARM_COMPILER_TOOLCHAIN%\bin
set GNUPATH=%APPS_PATH%\cygwin\bin
set CRMPERL=%APPS_PATH%\Perl64\5.20.2\bin
set PERLPATH=%APPS_PATH%\Perl64\5.20.2\bin

set PATH=.;%PYTHON_PATH%;%ARMBIN%;%GNUPATH%;%PERLPATH%;%PYTHONBIN%;%HEXAGON_ROOT%;%PATH%

set WORKING_PATH=\BSP\NP850
cd %WORKING_PATH%
```

Ps: Replace blue to yours

6. Run administrator command shell and set environment before all compilation building.

```
# cd D:\EWDK
# LaunchBuildEnv.cmd
# cd [to your working path]
# setenv.bat
```

2. BSP source building

2-1 non-HLOS

1. Bootloader

- (1) Modify "boot_images\QcomPkg\toolchainconfig.json"
Find "C:\\apps\\LLVM\\3.9.1\" and replace with yours.

- (2) build and clean

```
# cd boot_images\QcomPkg\  
# python buildex.py --variant WP -r RELEASE -t SDM845Pkg,QcomToolsPkg  
# python buildex.py --variant WP -r RELEASE -t SDM845Pkg,QcomToolsPkg --build_flags=cleanall
```

2. TZ

- (1) Rename "build_config_deploy.xml" to "build_config_windows.xml"

- (2) Modify "trustzone_images\build\ms\build_config_windows.xml"
Find all "method" property and change value as "declare".
Fins all value of "value" and set to match your environment.

- (3) Build and clean

```
# cd trustzone_images\build\ms  
# python build_all.py CHIPSET=sdm845 -b TZ.XF.5.0.1 --cfg build_config_windows.xml --recompile  
# python build_all.py CHIPSET=sdm845 -b TZ.XF.5.0.1 --cfg build_config_windows.xml --clean
```

3. AOP

- (1) Modify "aop_proc\build\build_845.bat" to match your environment

- (2) build and clean

```
# cd aop_proc\build  
# build_845.bat  
# build_845.bat -c
```

4. aDSP

- (1) cd to "adsp_proc\build"

- (2) build and clean

```
# python build.py -c sdm845 -o all -f aDSP  
# python build.py -c sdm845 -o clean -f aDSP
```

5. cDSP

- (1) cd to "cdsp_proc\build"

- (2) build and clean

```
# python build.py -c sdm845 -o all -f CDSP  
# python build.py -c sdm845 -o clean -f CDSP
```

6. MPSS(modem)

- (1) cd to "modem_proc\build\ms"

- (2) build and clean

```
# python build_variant.py sdm845.gen.prod bparams=-k  
# python build_variant.py sdm845.gen.prod --clean
```

7. SLPI

(1) Download "nanopb-0.3.6-windows-x86.zip" from "http://jpa.kapsi.fi/nanopb/download/"

(2) copy "nanopb-0.3.6-windows-x86.zip" to "slpi_proc\ssc\tools\"

(3) unzip nanopb-0.3.6-windows-x86.zip and set nanopb dependency

```
# cd slpi_proc
# python ssc\build\config_nanopb_dependency.py -f nanopb-0.3.6-windows-x86
```

(4) Modify "slpi_proc\build\build.py"

Search "os.environ['LLVMTOOLS'] = 'C:\\Apps\\LLVM\\3.9.3\\tools\\bin'" and set to match your LLVM path
Search "os.environ['LLVMHOME'] = 'C:\\Apps\\LLVM\\3.9.3'" and set to match your LLVM path

(5) build and clean

```
# cd build\
# python build.py -c sdm845 -o all -f SNS_DISABLE_ISLAND,SNS_REDUCED_SDC_PRAM
# python build.py -c sdm845 -o clean
```

2-2 HLOS (WP)

1. Set environment parameters for WP building.

```
set BUILD_ROOT=%CD%
set EWDK_ROOT=D:\EWDK

set QCPlatform=850
set platform=arm64
set CPU_TYPE=arm64
set device=850

set WDK_CURRENT_KIT_VERSION=10
set _WDKCOMP_ROOT_PATH=%EWDK_ROOT%
set WDKCONTENTROOT=%EWDK_ROOT%\Program Files\Windows Kits\%WDK_CURRENT_KIT_VERSION%

set _QCDK_ROOT_PATH=%BUILD_ROOT%\QCDK
set QCExtPath=%BUILD_ROOT%\build\Qualcomm
set ExtensionTasksPath=%QCExtPath%
set SECTOOLS_PATH=%QCExtPath%\sectools\850
```

You can add these parameters into setenv.bat in section 1.

Ps: **_WDKCOMP_ROOT_PATH** should be same as your EWDK path. Don't modify qcpackbuild.cmd and qcbuildwp63.cmd in "WP\src\tools\rel\10.4\"

2. copy ACPI compiler (This step can skip)

copy "D:\EWDK\Program Files\Windows Kits\10\Tools\arm64\ACPIVerify\asl.exe" to "WP\src\acpi\rel\10.4\bin\"

3. Modify "WP\src\acpi\acpi.wp63.proj"

Add "<Target Name="Clean"> </Target>" after "<Target Name="Build"> </Target>"

4. Modify BSP\WP\src\tools\rel\10.4\qcbuildwp63.cmd\qcbuildwp63.cmd

Find all "ATTRIB +R" and replace to "ATTRIB -R"

Sdm850 Windows 10 Desktop Development Environment and building

5. Delete or Rename "WP\src\tools\rel\10.4\CrmParallelization.py" to skip this python script be runned.

6. Compilation

run administrator command shell

```
# cd EWDK\ (Skip this if you have done)
# EWDK\LaunchBuildEnv.cmd (Skip this if you have done)
# cd \ (Skip this if you have done)
# setenv.bat (Skip this if you have done)
# cd BSP\WP\src
# .\tools\rel\10.4\qcpackbuild.cmd -p 850 -platform ARM64 (build for all)
# .\tools\rel\10.4\qcpackbuild.cmd -p 850 -platform ARM64 -componentsinput %1 (build component)
```

Ps: The first time building must build for all.

7. If ACPI compile fail at first time. Please rebuild it again with below command.

```
# .\tools\rel\10.4\qcpackbuild.cmd -p 850 -platform ARM64 -componentsinput acpi
```

3. Create WinPE manually

Recommend use steps in Section 5 to build all images include WinPE.

The steps in this section are referred and improved from MS document. Bad and stupid.

1. Create folder "WP\prebuilt\850\ISOGEN\emmcld_method"
2. Copy folder "WP\prebuilt\850\ISOGEN_QCReference\BOOTCHAIN_MTP" to "WP\prebuilt\850\ISOGEN\emmcld_method\"
3. Copy file "WP\prebuilt\850\firmware\850\850_MTP\Fat12.bin" to "WP\prebuilt\850\ISOGEN\emmcld_method\BOOTCHAIN_MTP\"
4. Copy files in "WP\prebuilt\850\DesktopScripts\" to "WP\prebuilt\850\ISOGEN\emmcld_method\Thumbdrive"
5. Copy folder "WP\src\build\Qualcomm\DISM" to "WP\prebuilt\850\ISOGEN\emmcld_method\Thumbdrive"
6. Copy folder "WP\src\build\Qualcomm\BCDBoot" to "WP\prebuilt\850\ISOGEN\emmcld_method\Thumbdrive"
7. Copy folders "boot", "efi", "sources" and "support" from WD_OS iso image to "WP\prebuilt\850\ISOGEN\emmcld_method\Thumbdrive"
8. Set environment parameters before WinPE building

```
set DISM_PATH = %BUILD_ROOT%\build\Qualcomm\DISM
set BCDBOOT_PATH = %BUILD_ROOT%\build\Qualcomm\BCDBoot

set ThumbPath = %BUILD_ROOT%\..\prebuilt\850\ISOGEN\Thumbdrive
set BCDFile = %ThumbPath%\efi\microsoft\boot\bcd
set BOOTWIM_PATH = %ThumbPath%\sources
set INSTALL_WIM = %ThumbPath
set Boot = %ThumbPath%\Boot
set Support = %ThumbPath%\support
```

9. Start building

```
# %ThumbPath%\WinPEDriverInstall.cmd -Source %BUILD_ROOT%\..\prebuilt\850\ -Destination %ThumbPath%\ -Option
'DriverCopy'
# %ThumbPath%\WinPEDriverInstall.cmd -Source %BUILD_ROOT%\..\prebuilt\850\ -Destination %ThumbPath%\ -Option
'WINPE'
```

10. Copy all files and folders in "WP\prebuilt\850\ISOGEN\emmcld_method\Thumbdrive\" to USB disk.
USB disk must be formatted as FAT32.
Disk size must larger then 4GB.
11. Install WinPE.
Plug USB disk into 1st USB port on device.
Open PUTTY or other console terminal tools.
Boot device to BDS menu. (Press UP key in putty while booting)
Select " 8 Boot USB First "

4. Develop Win10 desktop new driver

1. Setup windows driver compile environment.

- (1) Download Visual Studio 2017 and WDK installer.

<https://docs.microsoft.com/zh-tw/windows-hardware/drivers/download-the-wdk>

Download Visual Studio Professional 2017

<https://developer.microsoft.com/zh-tw/windows/downloads>

or

<https://www.visualstudio.com/thank-you-downloading-visual-studio/?sku=Professional&rel=15>

Download WDK for Windows 10, version 1809

<https://go.microsoft.com/fwlink/?linkid=2026156>

- (2) Install Visual Studio 2017 professional.

In Visual Studio Installer, check installed components in Individual components TAB in Modifying window:

[C++/CLI support](#)

[Windows 10 SDK \(10.0.16299.0\) for UWP: C++](#)

[Windows 10 SDK \(10.0.16299.0\) for Desktop C++ \[x86 and x64\]](#)

[Windows 10 SDK \(10.0.16299.0\) for Desktop C++ \[ARM and ARM64\]](#)

[Visual C++ compilers and libraries for ARM64](#)

[C++ Universal Windows Platform tools for ARM64](#)

[Visual C++ compilers and libraries for ARM](#)

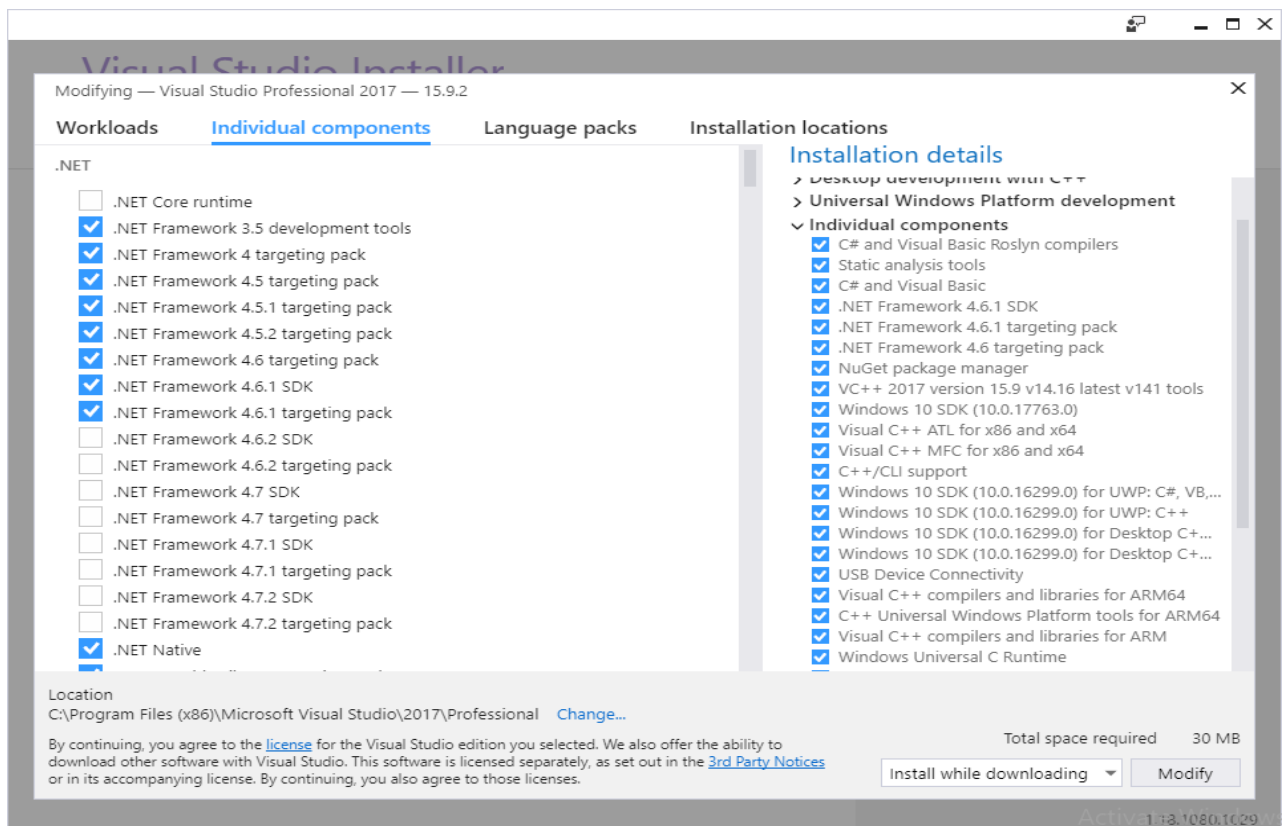
[Visual Studio SDK](#)

[Visual C++ ATL for ARM](#)

[Visual C++ ATL for ARM64](#)

[Visual C++ MFC for ARM](#)

[Visual C++ MFC for ARM64](#)



(3) Install WDK for Win10 version 1809

2. Create new KMDF driver in VisualStudio

(1) [File]->[New]->[Project...]

(2) Installed -> Visual C++ -> Windows Drivers -> Kernel Mode Driver (KMDF)

Ps: If WDK not install [Visual C++ -> Windows Drivers] will not present.

3. Copy your driver project folder to BSP\WP\src\

4. Build new adding driver

```
# .\tools\rel\10.4\qcpackbuild.cmd -p 850 -platform ARM64 -componentsinput NEW_DRIVER_NAME
```

5. Driver signing

Test Signing procedure is automatic process in BSP while build driver modules.

(1) WDK offer a test certification for driver test sign.

It will be copied as YourDriverName.cer in your driver folder while driver building.

(2) BSP use **INF2CAT** to check .inf content structure and create YourDriverName.cat.

ex:

```
Inf2CatItems=  
D:\BSP\NP850\WP\src\KMDF-Hello\ARM64\Release\KMDF-Hello\KMDF-Hello.inf  
D:\BSP\NP850\WP\src\KMDF-Hello\ARM64\Release\KMDF-Hello\KMDF-Hello.sys  
  
D:\EWDK\Program Files\Windows Kits\10\bin\10.0.17763.0\x86\inf2cat.exe /os:Server10_arm64  
/driver:D:\BSP\NP850\WP\src\KMDF-Hello\ARM64\Release\KMDF-Hello\
```

Then KMDF-Hello.cat will be produced.

(3) BSP use signtool.exe to sign YourDriverName.cat and YourDriverName.sys with YourDriverName.cer

It's not important for driver sign during driver module building. The formal driver signing will be done while build WDOS image. And formal certification for release build should be put in "WP\prebuild\850". In original Qualcomm BSP the test certification is **inf2cat.cer** used for default building.

6. Set device WIN10 to Testing Mode for test your driver.

Execuate below commands in administrator command shell.

```
# Bcdedit.exe - set TESTSIGNING ON  
# Bcdedit.exe /debug on
```

7. Install your driver to device as desktop windows driver installation procedure.

5. Build WD Image with adding drivers

1. Build parameters are same as section [2-2 HLOS(WP)].
2. Extract WDOS ISO image 17763.17.181009-1247.rs5_release_svc_refresh_CLIENTENTERPRISE_OEM_A64FRE_en-us.iso

Create new directory **DesktopOS** in "WP\prebuild\
Extract WDOS .iso to "WP\prebuild\DesktopOS\"

3. Copy folder "WP\src\build\Qualcomm\DISM" to "WP\prebuild\DesktopOS\"
4. Copy folder "WP\src\build\Qualcomm\BCDBoot" to "WP\prebuild\DesktopOS\"
5. Remove some .proj files to avoid building.

```
createwdtargetsfile_19H1.proj  
isogen_thumbdrive_19H1.proj
```

Or you can rename them to .proj_

6. Modify "WP\prebuild\850\isogen_thumbdrive.proj" files

Modify all "~~\$(DesktopOS)\Install.wim~~" to "~~\$(DesktopOS)\sources\Install.wim~~"

Search ~~<WD_Target Condition="\$(WD_Target)'=='>MTP;QRD;CLS</WD_Target>~~

Edit as ~~<WD_Target Condition="\$(WD_Target)'=='>MTP</WD_Target>~~

7. Modify "WP\prebuild\850\isogen.proj" files

Search ~~<WD_Target Condition="\$(WD_Target)'=='>MTP;CLS</WD_Target>~~

Edit as ~~<WD_Target Condition="\$(WD_Target)'=='>MTP</WD_Target>~~

8. Copy your driver files(.ing .sys .cat .pdb...) to "WP\prebuilt\850\driver\common\"
9. Modify "WP\prebuilt\850\DesktopScripts\drivers.txt to add your driver name.
10. Delete folder "WP\prebuild\850\ARM64\" and "WP\prebuild\850\ISOGEN\" if you have before image building everytime.
11. Start building

```
# cd WP\prebuild\  
# postbld.bat
```

Be patient with long time building without any message output.

12. Windows Desktop Image and WinPE will be produced in "WP\prebuild\850\ISOGEN\".

6. Flash Windows Desktop Builds

1. Switch the device to EDL mode and connect to host machine.
2. Unzip “WP\prebuild\850\ISOGEN\DesktopImage\MP\installer.zip” into a folder, for example “wdinstaller”.
3. Open administrator command prompt and change directory (cd) to “wdinstaller” folder.
4. Run the following commands if the storage type of the target device is UFS. <com port> is emergency download mode port.

```
# emmcdl.exe -p <com port> -f prog_firehose_ddr.elf -Memoryname UFS -x rawprogram1.xml -x rawprogram4.xml -x  
rawprogram3.xml -x rawprogram2.xml -x rawprogram5.xml -SetActivePartition 1  
# emmcdl.exe -p <com port> -f prog_firehose_ddr.elf -x WDFlash.xml -memoryname UFS
```

5. Reboot the device.