Qualcomm Technologies, Inc.

Modem Software Configuration

80-N5576-96 Rev E

November 7, 2014

zhangnan@hipad.com

Confidential and Proprietary – Qualcomm Technologies, Inc.

© 2012-2014 Qualcomm Technologies, Inc. and/or its affiliated companies. All rights reserved.

NO PUBLIC DISCLOSURE PERMITTED: Please report postings of this document on public servers or websites to: DocCtrlAgent@qualcomm.com.

Not to be used, copied, reproduced, or modified in whole or in part, nor its contents revealed in any manner to others without the express written permission of Qualcomm Technologies, Inc.

Questions or comments: Questions or comments: Questions or comments: https://support.cdmatech.com

zhangnan@hipad.com

Restricted Distribution: Not to be distributed to anyone who is not an employee of either Qualcomm Technologies, Inc. or its affiliated companies without the express approval of Qualcomm Configuration Management.

Qualcomm is a trademark of Qualcomm Incorporated, registered in the United States and other countries. All Qualcomm Incorporated trademarks are used with permission. Other product and brand names may be trademarks or registered trademarks of their respective owners.

This technical data may be subject to U.S. and international export, re-export, or transfer ("export") laws. Diversion contrary to U.S. and international law is strictly prohibited.

Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

Revision history

Revision	Date	Description
A	Nov 2012	Initial release
B	Dec 2012	Updated
C	Jul 2013	Numerous changes were made to this document revision; it should be read in its entirety
D	Feb 2014	Numerous changes were made to this document revision; it should be read in its entirety
E	Nov 2014	Restructured and rewritten; this document should be read in its entirety

QUALCOMM®
zhangnan@hipad.com

Contents

QUALCOMM®
zhangtian@hipad.com

Figures

Tables

QUALCOMM® zhangnan@hinad.com

Confidential and Proprietary – Qualcomm Technologies, Inc.

1 Introduction

1.1 Purpose

This document provides information about a Modem Configuration (MCFG) framework. It provides instructions on building an MCFG software image (*mcfg_sw.mbn*) and loading it onto a device by:

- Using the macro-enabled workbook/spreadsheet (MCFG_SW_Items_List_Macro) to generate configurations
- Creating and using carrier configurations
- Loading modem binary (MBN) files onto Android

1.2 Conventions

Function declarations, function names, type declarations, attributes, and code samples appear in a different font, for example, `#include`.

Code variables appear in angle brackets, for example, `<number>`.

Button and key names appear in bold font, for example, click **Save** or press **Enter**.

File, folder, and path names appear in italics, for example, the *license.dll* files are in the *LINUX/android/vendor/qcom/proprietary/aost-lf* directory.

1.3 References

Ref.	Document	
Qualcomm Technologies, Inc.		
Q1	*Application Note: Software Glossary for Customers*	CL93-V3077-1
Q2	*Presentation: Secured MSM™ Code Signing Service*	80-V9807-1
Q3	*Presentation: Code Signing Management System Overview*	80-V3999-1

1.4 Technical assistance

For assistance or clarification on information in this document, submit a case to Qualcomm Technologies, Inc. (QTI) at https://support.cdmatech.com/.

If you do not have access to the CDMATech Support website, register for access or send email to support.cdmatech@qti.qualcomm.com.

QUALCOMM®
zhangna@tipad.com

1.5 Acronyms

Acronym	Definition
CSFB	Circuit Switched Fallback
DSDS	Dual Sim Dual Standby
EFS	Embedded File System
MBN	Modem configuration BiNary
MCFG	Modem Configuration
NV	Nonvolatile
QPST	Qualcomm® Product Support Tool

For more terms and abbreviations, see [Q1].

QUALCOMM® zhangnan@hipad.com

Confidential and Proprietary – Qualcomm Technologies, Inc.

2 Overview

This chapter provides an overview and reference information about the modem configuration framework, the macro-enabled workbook and worksheets used for creating custom modem configurations, default modem configurations, and descriptions of attribute fields and the XML schema of generated output.

For step-by-step procedures for generating, loading and activating configurations and images, and integrating the MBNs into Android, see all subsequent chapters.

2.1 MCFG build environment

MCFG MBN generation uses the same build environment and SCons framework as the modem image itself.

The following is required to build MCFG MBNs:

- A complete copy of the MPSS image build
- Hexagon™ (for MSM8916, version 6.2.09, 64-bit); the version needed is dependent on product line
- Python 2.7.5
- Perl 5.6 or above
- MS Office installation (if generating configurations from the MCFG Excel spreadsheet)

2.2 Introduction to MCFG framework

The goal is to have a single binary image paired with a configuration data/image that can support multiple software/hardware configurations.

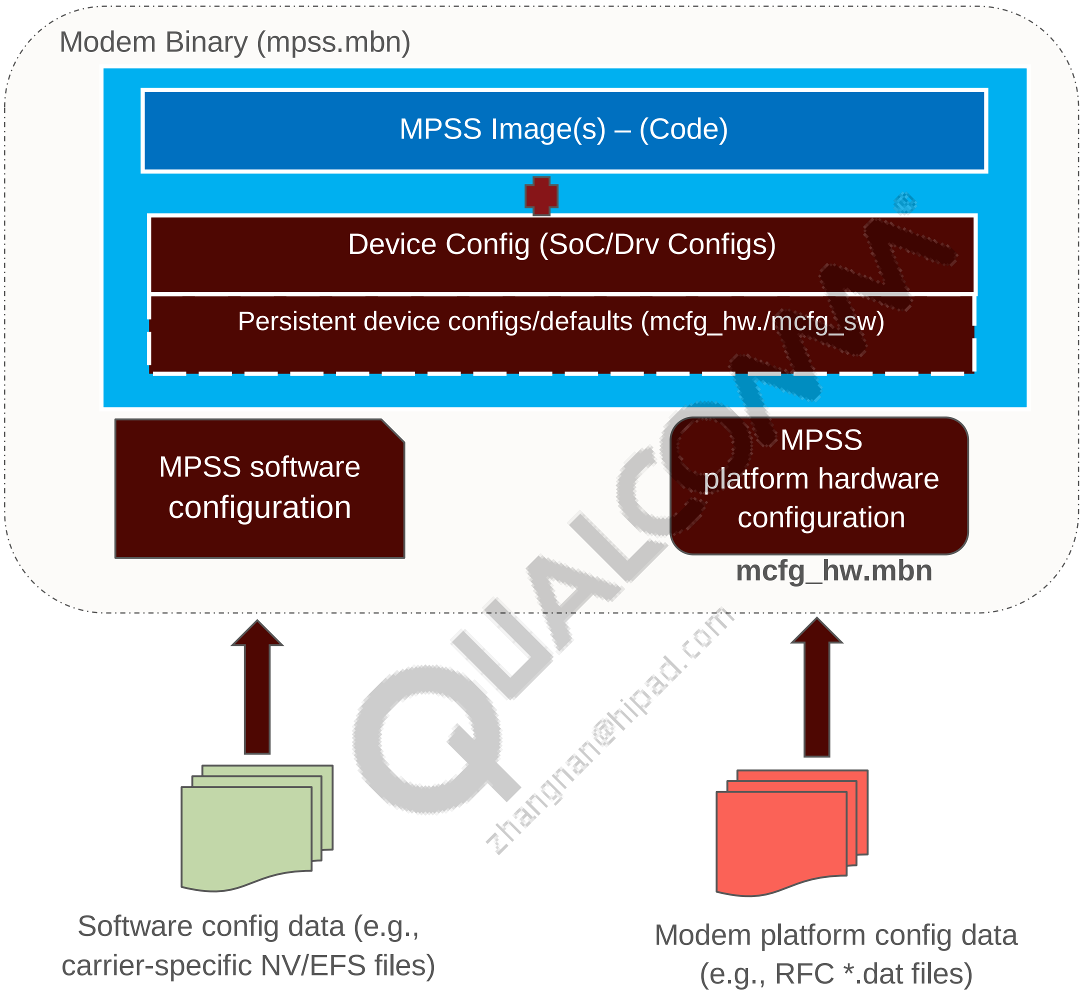
The modem software configuration allows licensees to configure the modem for various technologies, e.g., cdma2000, GSM-UMTS, LTE, software features, and carrier-specific customizations.

The mcfg_sw.mbn is comprised of NV/EFS settings specific to a carrier. The licensee groups the carrier-specific settings into an MBN that can be selected based on the carrier SIM (automatic) or manual methods.

There can be multiple carrier-specific settings saved in the device; however, only one carrier-specific setting is active at a given point in time, per subscription.

QUALCOMM®
zhangnan@sinacom

2.3 Binary MCFG feature concept


Modem Binary (mpss.mbn)
MPSS Image(s) – (Code)
Device Config (SoC/Drv Configs)
Persistent device configs/defaults (mcfg_hw./mcfg_sw)
MPSS software configuration
MPSS platform hardware configuration
mcfg_hw.mbn
Software config data (e.g., carrier-specific NV/EFS files)
Modem platform config data (e.g., RFC *.dat files)

QUALCOMM
zhangnan@hipad.com

Figure 2-1 Binary MCFG feature concept

2.4 MCFG framework

The MCFG framework involves:

- Generating the modem configuration
- Authenticating the modem configuration
- Processing the various modem configuration types (mcfg_hw – RFC, UIM configurations, mcfg_sw = NV/EFS, carrier-specific settings)
- Selecting/downloading the modem configurations using QPST or UICC-based (automatic) mechanisms

Confidential and Proprietary – Qualcomm Technologies, Inc.

The MCFG framework brings in the new modem configuration task that runs before any of the modem tasks are started. It configures the modem as per the selected configurations. If the UIM-based auto-selection feature is enabled, this selects the correct carrier software configuration based on the Issuer Identification Number (IIN) field of the ICCID in the UIM.

QPST is used to load the software/hardware configuration. This configuration data is version tagged to avoid processing the data again in case of a warm modem restart, etc. The process of activating a configuration involves a modem subsystem reboot. HLOS is not affected.

The modem hardware and software configurations are currently managed using a macro-enabled workbook/spreadsheet.

To leverage the MPSS image, OEMs typically modify the MCFG images. Qualcomm provides sample configurations for various operators for the reference platform. OEMs will modify/add new configurations according to their product requirements.

2.4.1 Security

The configuration data is authenticated before flashing into modem EFS, and every time it needs to be processed. Regarding the signing infrastructure, for the current implementation, the CSMS mechanism (available for main images) also applies for the MCFG images. See [Q2] and [Q3] for further information.

Software configuration supports the concept of multiple subscriptions via the use of a subscription bitmask. This mask determines for which subscription slots an NV/EFS setting will be active.

QUALCOMM®
zhangnan@hipad.com

Confidential and Proprietary – Qualcomm Technologies, Inc.

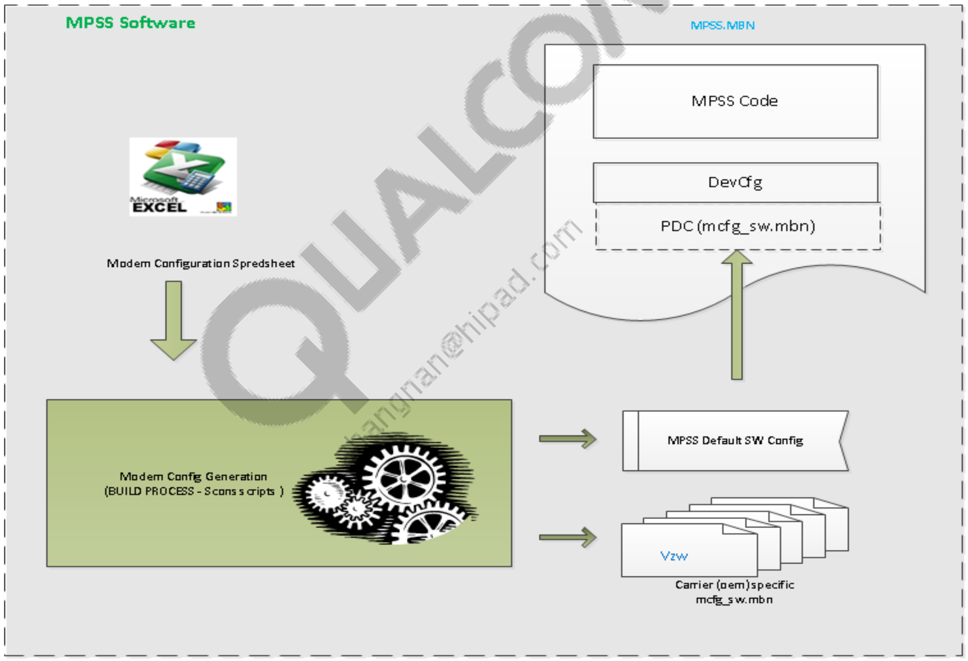
2.5 Generating MCFG software image using macro-enabled spreadsheet

This section is for information purposes. For step-by-step procedures, see Chapter 3.

The macro-enabled spreadsheet allows for creating custom configurations with the click of a button. The spreadsheet macros use build_mcfg.exe as a backend for this purpose.

Steps to generate a software MCFG image are:

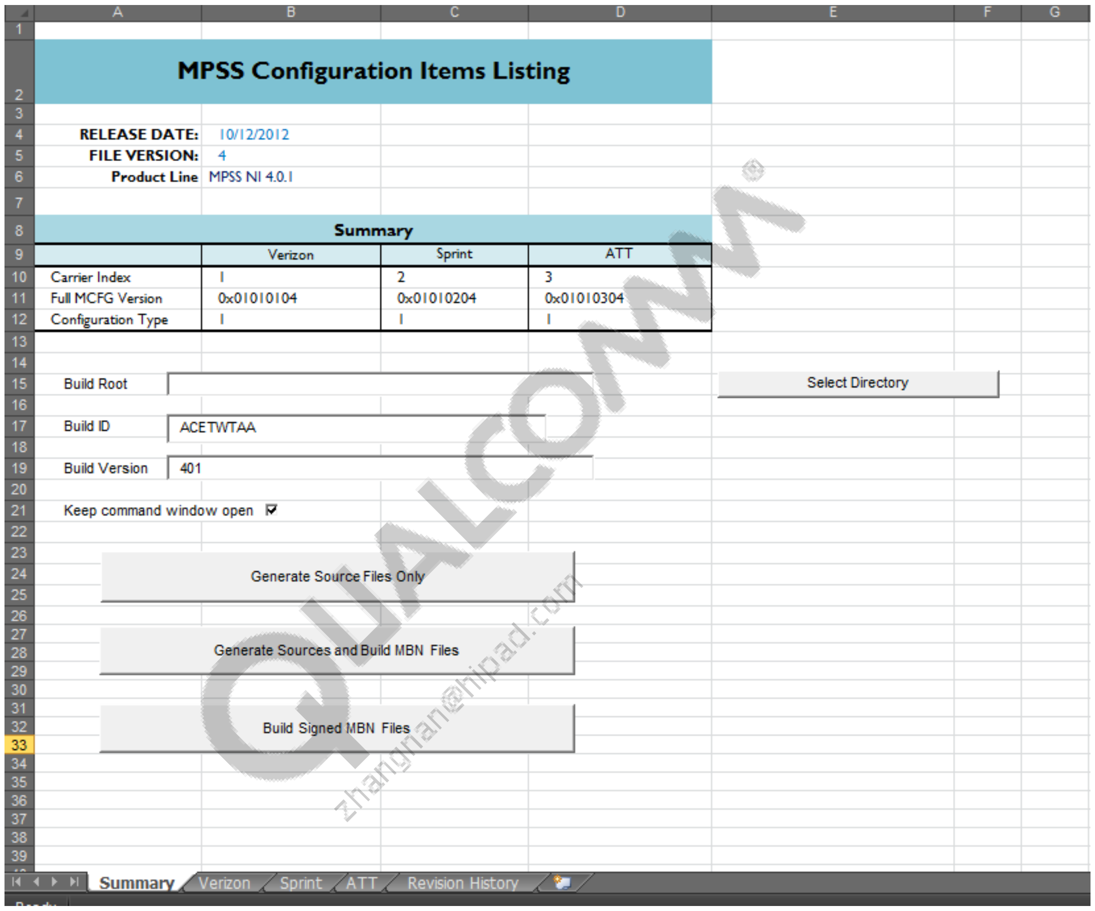
1. Locate and open a software MCFG workbook/spreadsheet.
2. Edit and save the spreadsheet.
3. Edit the Summary tab.
4. Generate source files.



QUALCOMM
zhangnan@hipad.com

2.5.1 Spreadsheet Summary tab

The workbook or spreadsheet should contain a worksheet named Summary along with a worksheet for each configuration to be generated. See Section 3.4 for detailed procedures.



MPSS Configuration Items Listing

RELEASE DATE: 10/12/2012
FILE VERSION: 4
Product Line MPSS NI 4.0.1

Summary			
	Verizon	Sprint	ATT
Carrier Index	1	2	3
Full MCFG Version	0x01010104	0x01010204	0x01010304
Configuration Type	1	1	1

Build Root
Build ID ACETWTAA
Build Version 401
Keep command window open ☑

QUALCOMM zhangnan@hipad.com

2.5.2 Summary worksheet format

The Summary worksheet in a configuration workbook (or spreadsheet) must be named Summary. This is the starting point for the script, where it determines which other worksheets to process.

The Summary worksheet must contain the word Summary somewhere in column A. Place the summary table directly below that word. An example of a summary table is shown here.

Summary			
	Verizon	**Sprint**	**ATT**
Carrier Index	1	2	3
Full MCFG Version	0x01010104	0x01010204	0x01010304
Configuration Type	1	1	1

- Each column header should match the name of the corresponding configuration worksheet in the workbook. In the example, since there are Verizon and Sprint columns, spreadsheets in the workbook named Verizon and Sprint would be expected.
- These are typical carrier names for software configurations, but they can be anything as long as they match the spreadsheet names.
- These names are used to differentiate the various output .xml files. A spreadsheet named Verizon would yield an output .xml file named <PREFIX>Verizon.xml, where <PREFIX> is whatever output prefix is specified in the command line arguments.

2.5.3 Configuration worksheet format

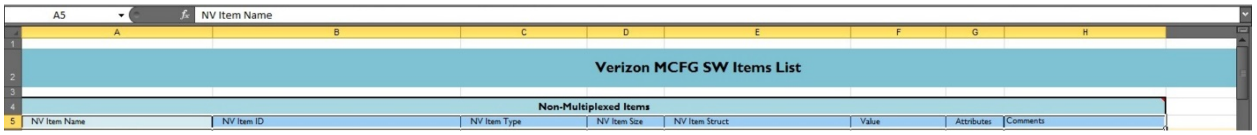
Configuration worksheets are divided into various MCFG items that can be one of several types:

- efs – Indicates an EFS file entry; this is more or less a direct copy of some file on the local machine to the EFS file system
- efs_dir – Directory of EFS files; all files in this directory and its subdirectories are read as if they had been individually listed in the spreadsheet with type "efs"
- efs_item – Like an EFS file entry, this type of item is data mapped to a specific path in the EFS file system; however, the data is written directly to the spreadsheet instead of being copied from a file on the local machine
- prl – Indicates a PRL file
- trl – Indicates the Trailer Record of the configuration
- OR – A numbered NV item mapped to the /nvm/num directory of the EFS file system

QUALCOMM®
zhangnan@hlipad.com

2.5.4 Numbered MCFG item entries

Numbered MCFG items are items with some ID number; they essentially reflect files created in the /nvm/num directory of the target device whose name is the ID number of that item. Data is manually input to the spreadsheet for these items.



A5 NV Item Name

	A	B	C	D	E	F	G	H
1								
2	Verizon MCFG SW Items List							
3								
4	Non-Multiplexed Items							
5	NV Item Name	NV Item ID	NV Item Type	NV Item Size	NV Item Struct	Value	Attributes	Comments

The exact format is:

- First row
  - Column A – Item description; this is not used by the generation script but is present for documentation
  - Column B – Item ID; this is some integer value
  - Column G – Attributes field; see Section 2.7 for details
- Row (2-N) – All rows following the first row for an item until the next items are values and appear as:
  - Column C – Value type
    - int – An integer value

    - int [ ] – An integer array of unspecified length; its size is dependent on how many elements are listed in the value column
    - int [n] – An integer value of length n
    - string – A character string
  - Column D – Value size
    - If type is int – The size in bytes of this integer value
    - If type is int [ ] or int [n] – The size in bytes of each element in the array
    - If type is string – A blank size means the length of the string should be the number of characters in the value + 1 for the NULL terminator
    - An integer means the string should be a fixed size; it will be padded with zeros if the value does not have this many characters
  - Column F – Value
    - If type is an int array, the value should be a list of integers or hex values delimited by commas
    - If type is an int value, the value should be an integer or hex value
    - If type is a string, the value can be any string
  - Prl item entries
    - Column G – Attributes hex value
    - Column C – prl
    - Column E – PRL file path on local file system
    - Column H – Local path to file relative to build root
- EFS file entries
  - Column B – EFS file path; this is the path to which the data should be copied
  - Column C – efs
  - Column G – Attributes as described previously
  - Column B – 257
- EFS Dir entries
  - Column B – EFS directory path; source efs files will be copied here
  - Column C – efs_dir
  - Column G – Attributes as described previously
  - Column H – Local directory path relative to build root
  - EFS item entries – Row 1
    - Column B – EFS file path; this is the path to which the data should be copied
    - Column C – efs_item
    - Column G – Attributes as described previously

QUALCOMM® zhangnan@hipad.com

- Trailer Record entries – Row 1
  - Column C – trl
  - Column G – 0x00

2.6 Default MCFG

The default MCFG allows for including the modem settings within the MPSS image. Typically, the NV/EFS settings persistent to device and common across the product line are expected to be included in the default configuration.

Default configurations are compiled into the modem image using the same XML schema as those generated by MCFG_SW_Items_List_macro.xlsm. This allows any carrier XML to be used as a substitute for the default XML, and subsequently compiled into the modem image as the default configuration.

Possible use cases of default configurations are:

- Automated generation of modem images with a default carrier
- Generating modem images containing updates to a default carrier
- Generating multiple modem images each with a different default carrier

NOTE: There can only be one default carrier configuration for any given image.

QUALCOMM® zhangnan@hipad.com

2.7 Description of attribute fields

Column G

Attribute field	Value	Description
#define U_ITEM_ATTRIB_CFG	0x01	/* C */ – This is a configuration bit. If this is set, the values for the MCFG item are taken from the spreadsheet.
#define U_ITEM_ATTRIB_MUXD	0x02	/* M */ – This is a multiplexed item. If this is set, the MCFG item becomes a symbolic link to the filename. *<MCFG_Item>_S<carrier_index>* – This attribute is used to support multiple subscriptions. Typical MCFG items that fall into this category include items that are set by the configuration and can also be modified via OTA or connection managers, e.g., PRL item.
#define U_ITEM_ATTRIB_WRITE_ONCE	0x04	/* W */ – When this is set, it implies that the MCFG item is written by the framework just once. Typical values include initial settings for the carriers prior to activation. Not many items fall into this category.
#define U_ITEM_ATTRIB_REST_FACT	0x08	/* R */ – This is always set to 1.
#define U_ITEM_ATTRIB_MULTISIM	0x10	/* S */ – This is for multi-SIM-related NVs.

Confidential and Proprietary – Qualcomm Technologies, Inc.

Attribute field	Value	Description
#define U_ITEM_ATTRIB_INDEXED	0x20	/* I */ – This is set to 1 for legacy, old-style indexed items.
#define U_ITEM_ATTRIB_DELETE	0x40	/* D */ – This is used if the set deletes the MCFG item from the device. Use caution and only if required.
#define U_ITEM_ATTRIB_UPDATE_ONLY	0x80	This bit allows for more optimization. If this is set, the values are read first and written only if different. This saves time in case of slow access flash devices like SPI-NOR where write times are significant.

2.8 XML schema of generated source files

Attribute field	Description
NvItemData	Used for normal NV items; "id" specifies the numerical item ID. "mcfgAttributes" is the 1-byte value used to specify attribute flags. The member list is a list of the various structs in the item. Each member has a "sizeOf" attribute and a "type" attribute. "sizeOf" is the length of the list of numbers inside the member tag. "type" is the size of each number in the list. "type" can be uint8, uint16, uint32, uint64, int8, int16, int32, or int64. int types are used when the data in the member tag is signed. uint is used otherwise.
NvEfsItemData	Used for EFS items whose data is listed explicitly in the XML file; "fullpathname" is the location on the target where the EFS file is to be placed. For information on the member list, see NvItemData.
NvEfsFile	Used for EFS items whose data is contained in a file on the build system; this data is to be read in and added to the final configuration when it is built. "targetPath" is the location on the target where the EFS file is to be placed. "buildPath" is the location on the build system relative to the root where the file resides.
NvEfsDir	This is similar to the NvEfsFile tag, but it specifies a whole directory whose contents should be copied to the specified location on the target.
NvPrlFile	This specifies the location of the prl file on the build system to be copied to the target.
NvTrlRecord	Contains data about the config's trailer record; the trailer record's purpose is to provide configuration metadata for internal processing by the MCFG framework. For information on the member list, see NvItemData.

Confidential and Proprietary – Qualcomm Technologies, Inc.

2.8.1 Schema of XMLs generated by macro-enabled spreadsheet

Examples listings of each XML element type

- NvItemData

```
<NvItemData id="256" mcfgAttributes="0x29">       <Member sizeOf="1"
type="uint8">0 </Member>       <Member sizeOf="1" type="uint8">1 </Member>
</NvItemData>
```

- NvEfsItemData

```
<NvEfsItemData mcfgAttributes="0x09"
fullpathname="/nv/item_files/data/3gpp2/ehrpd_to_hrpd_fallback">
<Member sizeOf="1" type="uint8">1 </Member>       <Member sizeOf="1"
type="uint8">1 </Member>       <Member sizeOf="1" type="uint8">0 </Member>
</NvEfsItemData>
```

- NvEfsFile

```
<NvEfsFile mcfgAttributes="0x09"
targetPath="/nv/item_files/ims/qp_ims_param_config"
buildPath="modem_proc\mcfg\mcfg_gen\scripts\data\efs_files\vzw\qp_ims_param
_
config/>
```

- NvEfsDir

```
<NvEfsDir mcfgAttributes="0x09" targetPath="/nv/item_files/ims/"
buildPath="modem_proc\mcfg\mcfg_gen\scripts\data\efs_files\vzw\/>
```

- NvPrlFile

```
<NvPrlFile mcfgAttributes="0x09“
buildPath="modem_proc\mcfg\mcfg_gen\scripts\data\efs_files\vzw\prlFile.txt/
>
```

- NvTrlRecord

```
<NvTrlRecord mcfgAttributes="0x00">       <Member sizeOf="7"
type="uint8">86 69 82 73 90 79 78 </Member>       <Member sizeOf="1"
type="uint8">4 </Member>       <Member sizeOf="1" type="uint16">6 </Member>
<Member sizeOf="1" type="uint8">0 </Member>       <Member sizeOf="1"
type="uint8">1 </Member>       <Member sizeOf="1" type="uint32">891480
</Member>   </NvTrlRecord>
```

- NvConfigurationData

```
<NvConfigurationData carrierIndex="1" version="0x02010108" type="1"/>
```

  Contains data related to configuration, including version, carrier index for muxed items, and type (hardware or software)

  - “carrierIndex” − Index used for muxed items
  - “version” − Configuration version
  - “type” − 1 for a software configuration, 0 for a hardware configuration

QUALCOMM® zhangnan@hmct.com

3 Macro-Enabled Spreadsheet Procedures

3.1 Locate and open the software MCFG workbook

1. Open the workbook/spreadsheet file from the path *$BUILD_ROOT\modem_proc\mcfg\mcfg_gen\generic*.
2. Allow editing and macros to run, if prompted by Excel.

3.2 Add and rename worksheets

The Summary table in the spreadsheet automatically updates the names of configurations when the **Refresh** button is clicked. This is important to do after renaming a worksheet or adding a new one.

Trust Center Settings option

1. In the worksheet, select **File**, **Options**, **Trust Center** (menu item), **Trust Center Settings…** (button).
2. In the Trust Center window, select **Macro Settings** from the menu.
3. Enable checkbox **Trust access to the VBA project object model**.
4. Select **OK**, then **OK** again.

Add a new worksheet

1. Select the configuration to be used as a base for your new configuration.
2. Right-click this tab name at the bottom and select **Move or Copy**.
3. Copy the worksheet to the end. Make it the last configuration before the Revision tab.
4. Go to the summary sheet and click **Refresh**. A new column and corresponding checkbox should be added to the end.

QUALCOMM
zhangjian@hipad.com

If the **Refresh** button is not present in your MCFG_SW spreadsheet (for older PLs), then add the new column manually:

- Select one of the existing configuration columns and select **Format Painter** from the Excel menu and paste it next to the last configuration column as shown:

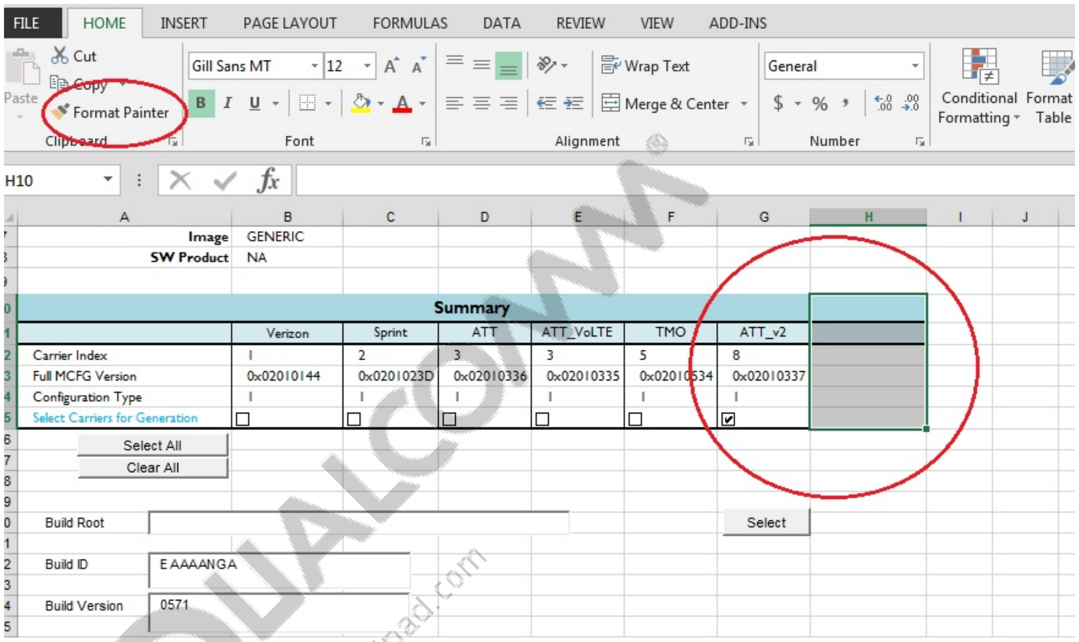


Image	GENERIC						
SW Product	NA						
Summary							
	Verizon	Sprint	ATT	ATT_VoLTE	TMO	ATT_v2	
Carrier Index	1	2	3	3	5	8	
Full MCFG Version	0x02010144	0x0201023D	0x02010336	0x02010335	0x02010534	0x02010337	
Configuration Type	1	1	1	1	1	1	
Select Carriers for Generation	☐	☐	☐	☐	☐	☑	

Build Root

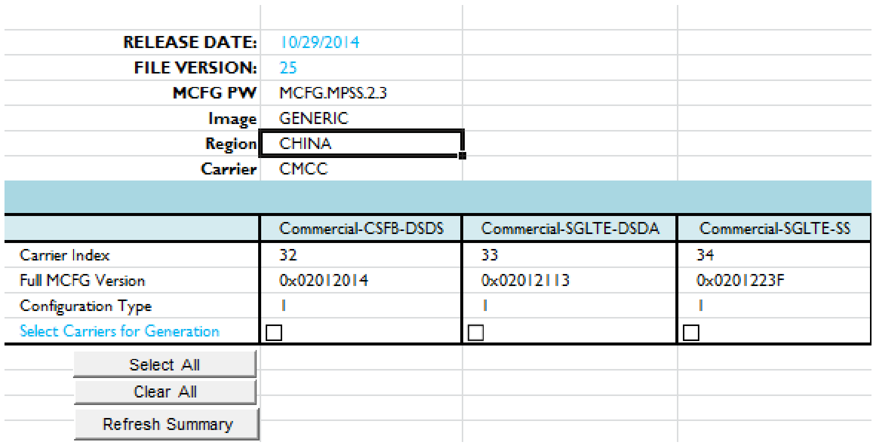
Build ID: E AAAANGA

Build Version: 0571

QUALCOMM zhangyan@hipad.com

5. Like the other entries, this listing will consist of the following (which will need to be entered manually before generation can take place):
   - Carrier index (decimal): This can be the same as previous entries.
   - MCFG software version: This is the same value listed in the trailer record of the corresponding worksheet.
   - Configuration type: 1 = software configuration, 0 = hardware configuration

   Image of table with **Refresh** button:



RELEASE DATE:	10/29/2014		
FILE VERSION:	25		
MCFG PW	MCFG.MPSS.2.3		
Image	GENERIC		
Region	CHINA		
Carrier	CMCC		
	Commercial-CSFB-DSDS	Commercial-SGLTE-DSDA	Commercial-SGLTE-SS
Carrier Index	32	33	34
Full MCFG Version	0x02012014	0x02012113	0x0201223F
Configuration Type	1	1	1
Select Carriers for Generation	☐	☐	☐

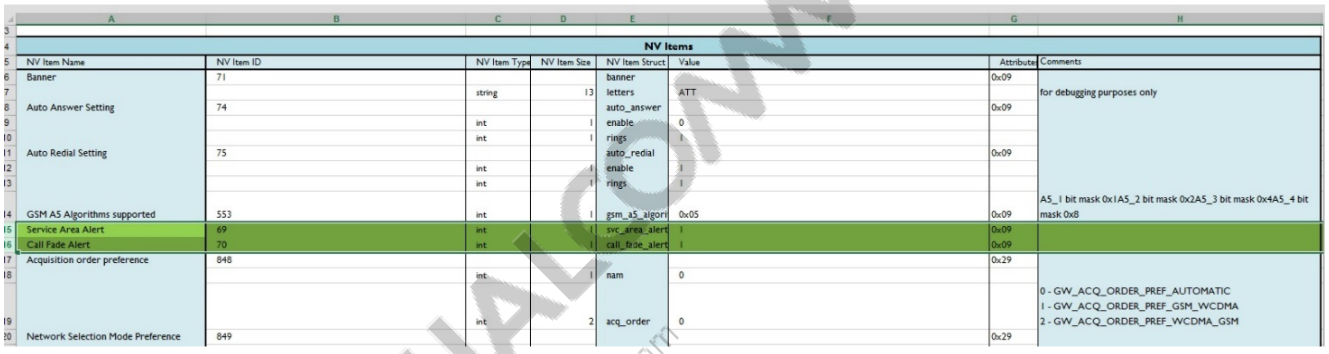
6. Edit the tab name for the configuration that needs to be updated.

7. Go to the summary sheet and click **Refresh**. The new name should appear in the Summary table.

Once the Summary table is up to date, you can begin generating source files and MBNs.

3.3 Edit the carrier-specific row in the worksheet

1. Add NV items by adding rows to the worksheet; for example, NV 69 and NV 70 are added as shown:



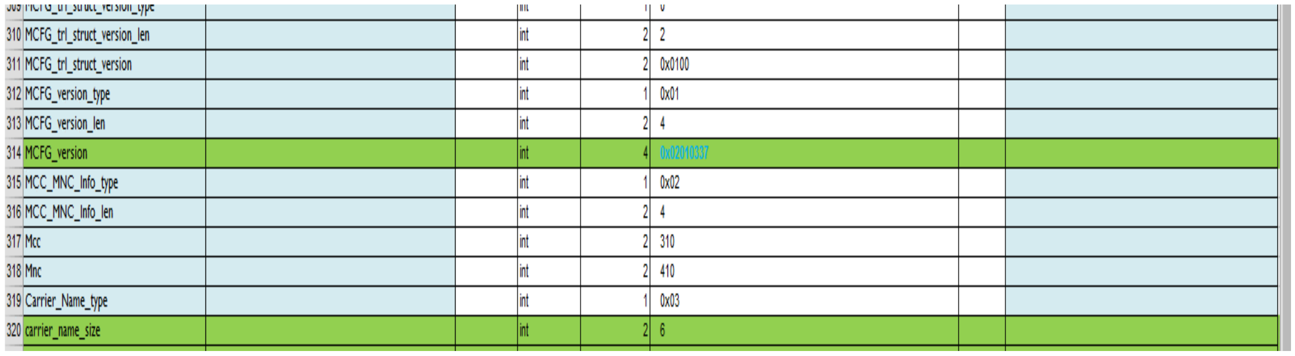
	A	B	C	D	E	F	G	H
3								
4	NV Items							
5	NV Item Name	NV Item ID	NV Item Type	NV Item Size	NV Item Struct	Value	Attributes	Comments
6	Banner	71			banner		0x09	
7			string	13	letters	ATT		for debugging purposes only
8	Auto Answer Setting	74			auto_answer		0x09	
9			int	1	enable	0		
10			int	1	rings	1		
11	Auto Redial Setting	75			auto_redial		0x09	
12			int	1	enable	1		
13			int	1	rings	1		
14	GSM A5 Algorithms supported	553	int	1	gsm_a5_algori	0x05	0x09	A5_1 bit mask 0x1A5_2 bit mask 0x2A5_3 bit mask 0x4A5_4 bit mask 0x8
15	Service Area Alert	69	int	1	svc_area_alert	1	0x09	
16	Call Fade Alert	70	int	1	call_fade_alert	1	0x09	
17	Acquisition order preference	848					0x29	
18			int	1	nam	0		
19			int	2	acq_order	0		0 - GW_ACQ_ORDER_PREF_AUTOMATIC 1 - GW_ACQ_ORDER_PREF_GSM_WCDMA 2 - GW_ACQ_ORDER_PREF_WCDMA_GSM
20	Network Selection Mode Preference	849					0x29	

2. Change the MCFG_version in the Trailer Record table.

NOTE: Any configurations generated by OEMs will need an MCFG software version different from what is released by Qualcomm. The second byte in the software version is reserved for this purpose. Example:

```
CMCC_CSFB_DSDS_Commercial from Qualcomm: 0x04012000
CMCC_CSFB_DSDS_Commercial update by OEM : 0x04022000
```

- Any set of changes to be included in an MBN will need a corresponding MCFG software version update as well.
- If carrier_name length has changed, the carrier_name_size and the Data Size column of the carrier name are automatically updated.



309	MCFG_trl_struct_version_type			int	1	0		
310	MCFG_trl_struct_version_len			int	2	2		
311	MCFG_trl_struct_version			int	2	0x0100		
312	MCFG_version_type			int	1	0x01		
313	MCFG_version_len			int	2	4		
314	MCFG_version			int	4	0x02010337		
315	MCC_MNC_Info_type			int	1	0x02		
316	MCC_MNC_Info_len			int	2	4		
317	Mcc			int	2	310		
318	Mnc			int	2	410		
319	Carrier_Name_type			int	1	0x03		
320	carrier_name_size			int	2	6		

Confidential and Proprietary – Qualcomm Technologies, Inc.

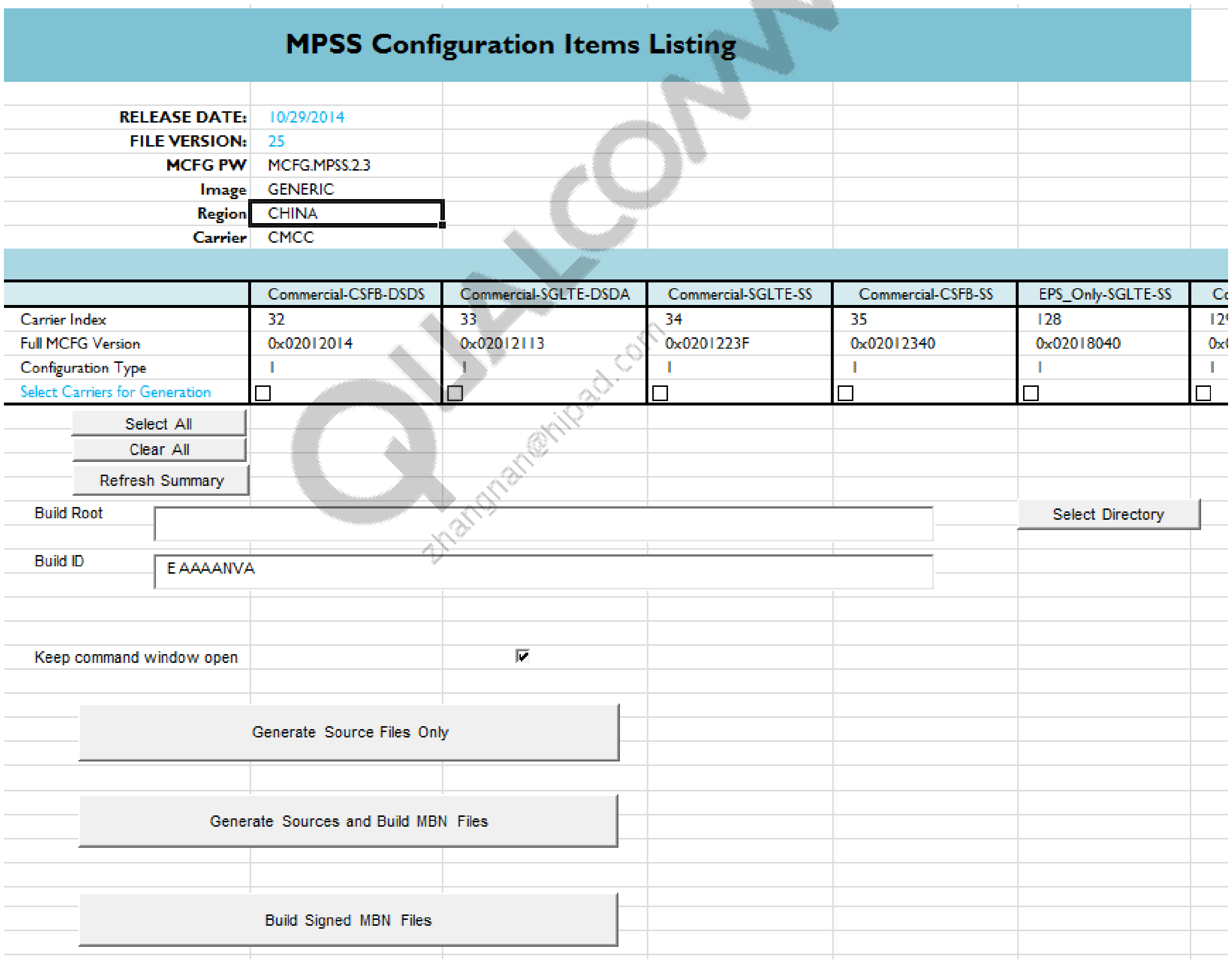
3.4 Edit the Summary tab and generate source files

1. Open the Summary tab and select configurations to generate.

   Each checkbox corresponds to the configuration name listed above it. Use the **Select All / Clear All** buttons to select or clear all checkboxes. Optionally, configurations can be manually selected/deselected one at a time by clicking the checkbox.

   Buttons to generate sources/MBNs will only affect configurations that have corresponding checkboxes checked.

2. Enter the build root, build ID, and build version.



MPSS Configuration Items Listing

RELEASE DATE:	10/29/2014
FILE VERSION:	25
MCFG PW	MCFG.MPSS.2.3
Image	GENERIC
Region	CHINA
Carrier	CMCC

	Commercial-CSFB-DSDS	Commercial-SGLTE-DSDA	Commercial-SGLTE-SS	Commercial-CSFB-SS	EPS_Only-SGLTE-SS	Co
Carrier Index	32	33	34	35	128	12
Full MCFG Version	0x02012014	0x02012113	0x0201223F	0x02012340	0x02018040	0x
Configuration Type	1	1	1	1	1	1
Select Carriers for Generation	☐	☐	☐	☐	☐	☐

Build Root

Build ID: EAAAANVA

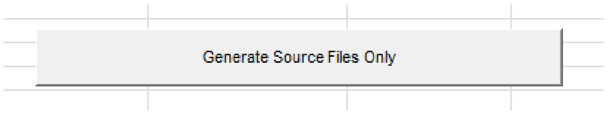
Keep command window open ✓

Generate Source Files Only
Generate Sources and Build MBN Files
Build Signed MBN Files

QUALCOMM zhangnan@hipad.com

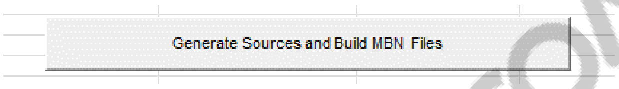
3.4.1 Generate Source Files Only

This option generates only the source files that are used in the configuration. Source files are located in the same directory as the spreadsheets/workbooks. They are used mainly for debugging purposes.



3.4.2 Generate Sources and Build MBN Files

This option generates the source files and creates unsigned .mbn files. The .mbn files are generated in the $BUILD_ROOT\modem_proc\mcfg\configs directory.



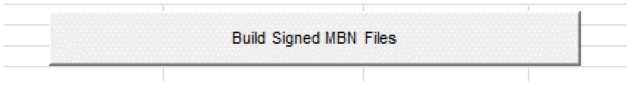
3.4.3 Build Signed MBN Files

This option is used in conjunction with (and after) the option **Generate Sources and Build MBN Files** for creating signed .mbn files.

For secboot-enabled build:

1. Select **Generate Sources and Build MBN Files** to generate the hash of unsigned .mbn files.

2. Use Code Signing Management System (CSMS) to generate a signature with Secure Boot Software Type as AMSS_HASH_TABLE(0002) and save the certificate. CSMS is available to Qualcomm OEMs as a security feature; see [Q2] and [Q3].
3. Select **Build Signed MBN Files** to make the final signed images.



- Ensure configuration updates are included by updating the version number in the trailer record.
- When a configuration update occurs, whether it be within the spreadsheet or outside (e.g., update to an EFS file), the version listed in the trailer record will also need to change within the spreadsheet (or XML if that is the primary generation source).

This updated version is what the MCFG framework evaluates when deciding if the MBN is an actual upgrade. If a configuration version is the same as one currently on the target, the file will not be loaded to the target.

Although there is no rule about version numbering, it is best that the configuration version be incremented by one for each set of changes.

QUALCOMM® zhangnan@itipad.com

3.5 MBN generation backend

The executable used to generate configurations is *build_mcfgs.exe*.

The file location is *<build_root>\modem_proc\mcfg\build*.

On Unix-based systems, the equivalent *build_mcfgs.pl* file can be used.

Buttons at the bottom of the spreadsheet all make calls to *build_mcfgs.exe*, but with different switch settings. The table shows what system call is made behind the scenes when a macro is triggered.

Spreadsheet button	Equivalent command
Generate Source Files Only	>build_mcfgs.exe -- build_id+TAAANAA –configs=mcfg_sw:all –force-regenerate –sources-only
Generate Sources and Build MBN Files	> build_mcfgs.exe –build_id=TAAANAA –configs=mcgf_sw:all –force-regenerate –force-rebuild

Available options to *build_mcfgs.exe* can be found using `build_mcfgs.exe --usage`.

QUALCOMM®
zhangnan@hipad.com

Confidential and Proprietary – Qualcomm Technologies, Inc.

4 Generating Carrier Configurations via Command Line

4.1 Generate a specific, single-carrier configuration

General form of the command used to build a single source or .mbn file

```
build_mcfgs.exe --build_id=<BUILD ID> --configs=<PLATFORM>:<CONFIG_NAME>
[--force-regenerate] [--force-rebuild]
```

Explanation of command arguments

- `<BUILD ID>` – Build variant ID of the image; listed in the summary spreadsheet of the macro-enabled Excel workbook
- `<PLATFORM>:<CONFIG_NAME>` – MBN platform and configuration name being generated
  - `PLATFORM` – mcfg_sw for carrier configurations, mcfg_hw for hardware configurations; these are the only two values currently accepted
  - `CONFIG_NAME` – Carrier name for software configurations, e.g., Verizon, or hardware configuration name, e.g., MTP8974_NA1
- `[--force-regenerate]` – Option to generate the source XML regardless if one is already present (optional, but recommended)
- `[--force-rebuild]` – Option to generate the *.mbn* file regardless if one is already present (optional, but recommended)

NOTE: If the force switches are not specified, *build_mcfgs.exe* will only generate the corresponding file if one does not already exist. Carrier names are case sensitive and must match the name listed in the spreadsheet exactly.

4.2 Generate a default configuration using carrier settings

If no carrier is specified in the image build command, then XML data from the default tab of *MCFG_SW_Items_List_Macro.xlsm* is used as the default configuration in the modem image, e.g., build.cmd 9x25.geni BUILD_ID=TAAAANAA BUILD_VER=0001_def_config.

- To specify an alternate (specific) carrier as the default configuration, add the MCFG_SW_TYPE option to the build command (see Section 4.1), e.g., build.cmd 9x25.geni BUILD_ID=TAAAANAA BUILD_VER=0001_scons_change MCFG_SW_TYPE=Verizon.

This example will compile an image with Verizon as the default carrier.

See Chapter 9 for any errors encountered.

QUALCOMM® zhongguanchipad.com

4.2.1 Process new default configurations

If the NVs related to processing default configurations are inactive, the behavior is to process both the default and regular, e.g., carrier, configurations any time either of the two is updated.

NOTE: NVs are altered *only* if the OEM wants to change the Qualcomm default behavior. Otherwise, it is recommended to *not* alter the NV.

Use the following NVs to explicitly control how and if configurations are processed:

- NV 71547 – Bitmap to enable/disable default modem configuration (Bit 0 – Hardware, Bit 1 – Software)
  - Enabled or Inactive – Default configuration type (hardware or software) will be processed once after an update has been loaded.
  - Disabled – Default configurations are not processed.
- NV 71548* – Bitmap to enable/disable force load of def cfg before activating new regular configuration (only applies when NV 71547 is enabled. General approach is to not set any NVs)
  - Enabled or Inactive – Default configuration is reactivated each time before a regular configuration of the same type is activated.
  - Disabled – If explicitly disabled, settings from the default configuration will not be reprocessed after a carrier update.

NOTE: Bit 0 – Hardware, Bit 1 – Software for each bitmap.

- NV 71549 – Bitmap to enable/disable force load of regular configuration after processing default configuration
  - Enabled or Inactive – If the default configuration has been updated, reprocess settings from the regular configuration of the same type.
  - Disabled – If explicitly disabled, settings from the currently active carrier configuration will not be reprocessed after a default configuration update.

4.3 Generate multiple images with default configurations

4.3.1 Procedure for multiple images with default carrier

1. Build single MPSS image with default carrier configuration.
2. Save firmware image to another location.
   - The original location is *<build_root>\modem_proc\build\ms\bin\ <build_id>\qdsp6sw.mbn*.

- If building images is for a different carrier (Verizon, ATT, etc.), repeat this process for each carrier.
- If building an image to update the same carrier, e.g., from Verizon to Verizon_2, see above.
  - Multiple image with default carrier updates

QUALCOMM® zhangnan@hipad.com

3. For carrier updates to take effect, e.g., Verizon to Verizon_2, the software version also needs to be updated.
   - The MCFG framework checks for a unique configuration ID and software version pair before it will load or activate a configuration.
   - If the default software version is the same as the current default software version, the update will not occur.

4.3.2 Update the software version listed in carrier XML

Increment the software version listed in the XML before building target firmware (version is listed in two places):

1. First location – Look for a line similar to:

```
<NvConfigurationData carrierIndex="1" version="0x0201010B" type="1"/>
```

2. Increment the version, e.g.,

```
<NvConfigurationData carrierIndex="1" version="0x0201010C" type="1"/>
```

3. Second location – Go to the trailer record and increment the sixth line down from the beginning of the record (here it must be in decimal):

```
<NvTrlRecord mcfgAttributes="0x00">
<Member sizeOf="1" type="uint32">33620235 </Member
```

- The above will change to:

```
<NvTrlRecord mcfgAttributes="0x00">
…
<Member sizeOf="1" type="uint32">33620236 </Member>
```

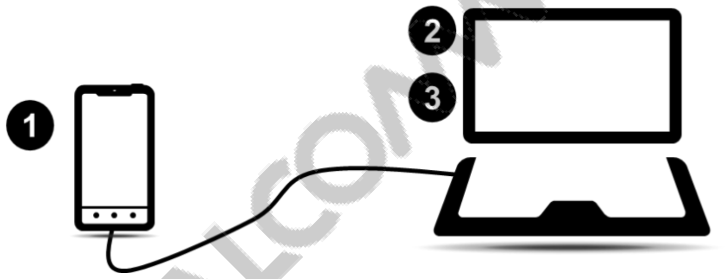
Note that the first edit is in hexadecimal and the second is in decimal.

4. Once this is done, rebuild the image.

QUALCOMM® zhanghuan@hipad.com

5 Loading, Activating, and Deactivating MBNs Using QPST

Setup and prerequisites



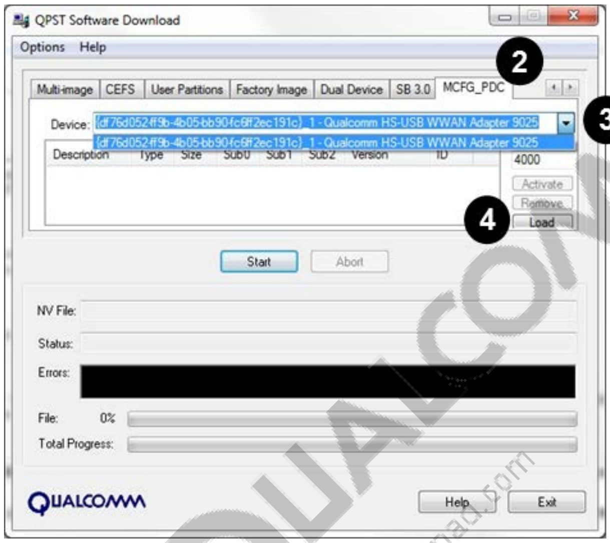
❶	UE is connected to a PC.
❷	PC is running QPST version 2.7.421 or later.
❸	Know your <MODEM_BUILD> path and be able to access it from the PC.

NOTE: If you disabled the RmNet port on the UE, ensure that it is enabled before performing these steps.

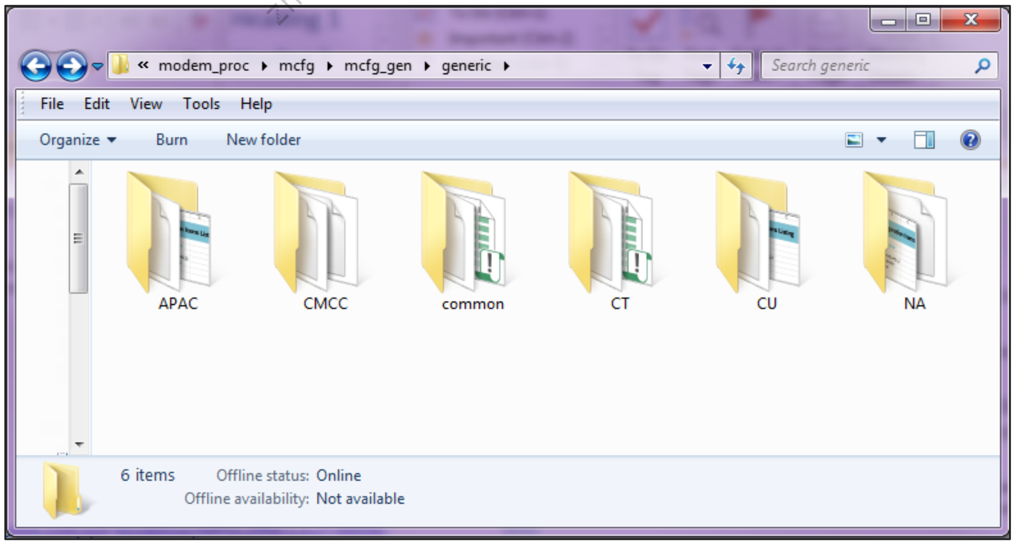
QUALCOMM®
zhangnan@chipad.com

5.1 Load and activate MBNs

1. Open the QPST Software Download module on the PC.
2. Select the MCFG-PDC tab on the far right of the application.
3. Click the drop-down arrow and select any RmNet port available from the device.
4. Select **Load**. A pop-up window appears.



5. Use Windows Explorer to navigate to *<MODEM_BUILD>\modem_proc\mcfg\configs\mcfg_sw\generic*. This path is known as the <swmbnpath>.


« modem_proc ▸ mcfg ▸ mcfg_gen ▸ generic ▸
Search generic
File Edit View Tools Help
Organize ▾ Burn New folder
APAC
CMCC
common
CT
CU
NA
6 items Offline status: Online
Offline availability: Not available

QUALCOMM zhanghuachipad.com

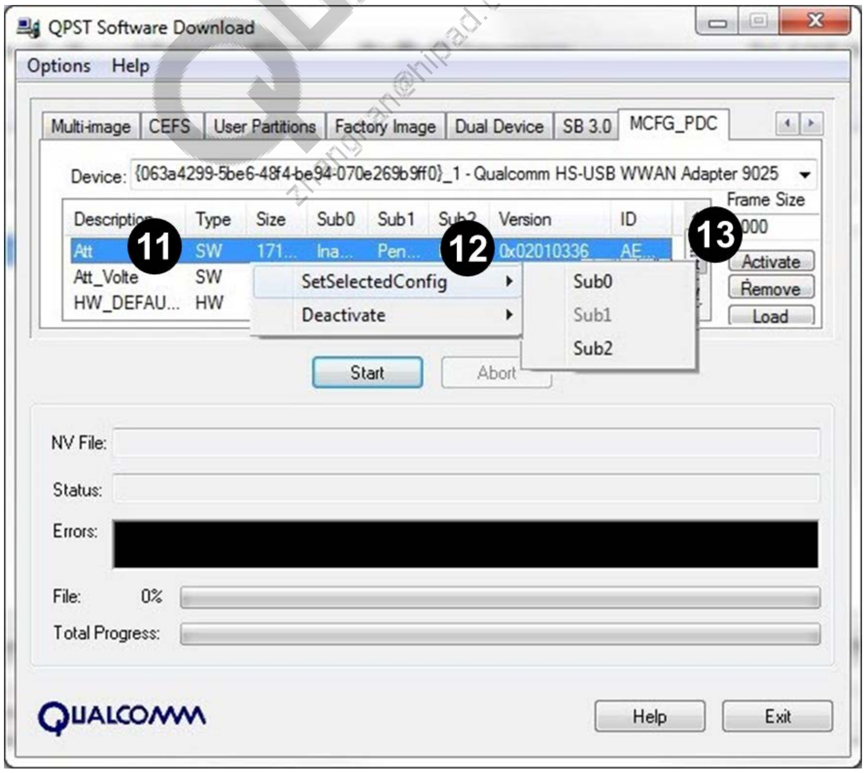
Confidential and Proprietary – Qualcomm Technologies, Inc.

There are directories in this path that organize the MBNs by geographic region or by carrier. Table 5-1 identifies the appropriate subdirectory for each carrier or region.

Table 5-1 MBN subdirectories by carrier and region

Carrier or region	Subdirectory containing the MBNs
Asia-Pacific – Carriers like Airtel, DCM (DOCOMO), KDDI, Reliance, and SBM (Softbank))	*<swmbnpath>\APAC*
Common	*<swmbnpath>\common*
China Mobile	*<swmbnpath>\CMCC*
China Telecom	*<swmbnpath>\CT*
China Unicom	*<swmbnpath>\CU*
North America – Carriers like Verizon Wireless, AT&T, Sprint, and T-Mobile)	*<swmbnpath>\NA*

6. In Explorer, open the directory for the appropriate carrier or region.
7. Copy the full path of the applicable directory then paste it in the QPST pop-up.
8. Click **Open**.
9. Double-click the *mcfg_sw.mbn* file. The file is now listed in QPST.
10. Repeat steps 5 through 8 until all of the MBNs applicable to the UE are loaded.



Device: {063a4299-5be6-48f4-be94-070e269b9ff0}_1 - Qualcomm HS-USB WWAN Adapter 9025

Description	Type	Size	Sub0	Sub1	Sub2	Version	ID
Att	SW	171...	Ina...	Pen...		0x02010336	AE...
Att_Volte	SW						
HW_DEFAU...	HW						

11. Highlight and right-click the MBN that is appropriate for the UE that you are configuring.

QUALCOMM zhangnan@hipad.com

12. Select **SetSelectedConfig** from the pop-up menu and then do one of the following:
    - If the UE is a single-SIM device, select Sub0.
    - If the UE is a dual-SIM device, select Sub0 then repeat steps 10 and 11 and select Sub1.

      The configuration state changes to Pending after the selection.
13. Click **Activate**. The device resets. In some cases, a crash dump occurs.
14. Turn off, then turn on the device. The selected configuration is now active on the UE.

5.2 Switch between MBNs

To switch between MBNs using QPST, do the following:

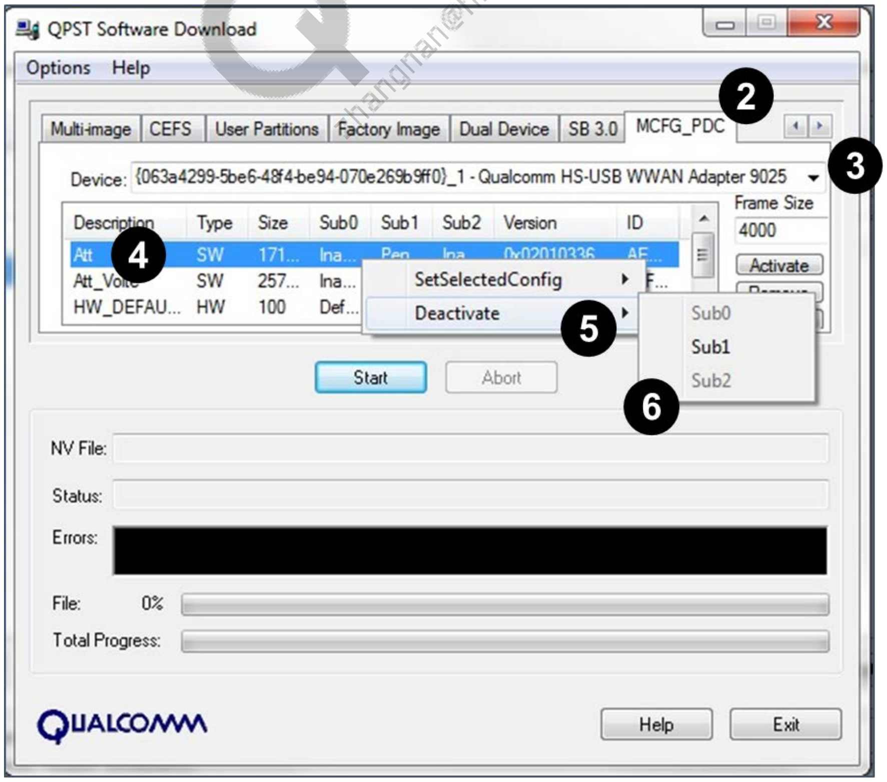
1. Deactivate the currently active MBN. See Deactivating an MBN for more information.
2. Load and activate the new MBN. See Section 5.1 for more information.

5.2.1 Deactivate an MBN

When you deactivate an MBN, the NV/EFS settings are rolled back to their prior state for the specified subscription. The inactive MBN remains on the UE and is available for reactivation.

To deactivate an MBN:

1. Open the QPST Software Download module on the PC.



Device: {063a4299-5be6-48f4-be94-070e269b9ff0}_1 - Qualcomm HS-USB WWAN Adapter 9025

Description	Type	Size	Sub0	Sub1	Sub2	Version	ID
Att	SW	171...	Ina...	Pen	Ina	0x02010336	AE...
Att_Vo...	SW	257...	Ina...				F...
HW_DEFAU...	HW	100	Def...				

2. Click the MCFG-PDC tab on the far right of the application. Click the drop-down arrow and select any RmNet port available from the UE.

QUALCOMM zhangnan@hipad.com

3. Highlight and right-click the MBN that you want to deactivate.
4. Select Deactivate. A list of subscriptions appears.
5. Do one of the following:
   - If the UE is a single-SIM device, select Sub0.
   - If the UE is a dual-SIM device, select Sub0, then repeat steps 4 and 5 and select Sub1.

   The status of the MBN changes to Inactive.

5.3 Delete an MBN from the UE

NOTE: Only MBNs in pending or inactive status can be deleted from the UE.

To delete an MBN from the Flash memory of the UE, do the following:

1. Open the QPST Software Download module on the PC.
2. Select the MCFG-PDC tab on the far right of the application.
3. Select an inactive/pending configuration and select **Remove**. This removes the MBN from the Flash memory of the UE.

5.4 Working with hardware/platform configurations

Unlike software/carrier configurations, a current hardware configuration must be deactivated before the next hardware configuration can be activated. The following is a use case for switching between the single-SIM and multi-SIM hardware configurations, while also using single-SIM and multi-SIM carrier configurations.

5.4.1 Switch between single-SIM and dual-SIM configurations while using both hardware and software MBNs

1. Load and activate the single-SIM hardware and software configurations. At this point, your target is ready for single-SIM testing.
2. To begin testing dual SIM, first deactivate the single-SIM hardware configuration.
3. Load and activate the dual-SIM hardware and software configurations. At this point, the target is ready for dual-SIM testing.
4. To switch from dual-SIM to single-SIM testing, perform these steps in reverse (and step 2 will consist of deactivating the dual SIM hardware configuration).

QUALCOMM®

6 Autoselection Mechanism

For software configurations, it is also possible to automatically select the right carrier configuration based on the UICC.

To enable the autoselect mechanism, first load the Commercial MBNs for CMCC, CU, and CT using QPST, but do not activate any subscription:

1. Load the MBN using the QPST Software Download tool, using the MCFG tab:
   a. Load CMCC MBN.
   b. Load CT MBN.
   c. Load CU MBN.

   See Chapter 5 for details.
2. In QXDM Professional™ (QXDM Pro), go to the NV Browser.
   a. Go to NV 71546.
   b. At the top of the NV Browser window, check **Multi SIM**.
   c. Select Subscription 0 from the drop-down menu.
   d. Write a value of 1 to NV 71546.
   e. Repeat steps b through d but for Subscription 1 you may also want to enable the autoselect mechanism for SIM 2.
   f. Reset the UE.

The automatic selection feature is based on the ICCID standardized by ITU.

The first 4 to 7 digits of the ICCID contain the IIN, consisting of:

- Major Industry Identifier (2 digits) – Always 89 for telecom
- Country Code (1 to 3 digits)
- Issuer Identifier Number (1 to 4 digits)
- Example – ICCID from Verizon 4G UICC: 89148000000014177464
  - 89, 1, 480 assigned to Verizon Wireless per ITU
  - ITU maintains lists of registered telecom IINs

The feature is disabled by default and can be turned on in specific product configurations, e.g., M2M, Windows Phone 8 dongles, etc.

QUALCOMM® zhangnan@hipad.com

7 Start-to-Finish Generation Process Example

This chapter is a sample session. General update procedures included in this chapter consist of the following basic steps:

1. Update source files via either the spreadsheet or XML.
2. Update the configuration version.
3. Save the source file.
4. Build the configuration(s).
5. Load and activate the configuration on the target to verify updates are present.

For more detailed information on any of these steps, refer to the respective chapter in this document.

zhongnan@hipad.com

7.1 Update source files

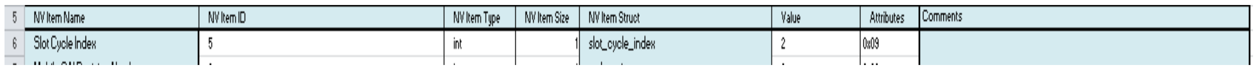
Make edits through either the spreadsheet or in the XMLs used to create the .c source files. Example updates to be made are:

Carrier	Updates
Verizon	▪ Add slot cycle index (NV 5). ▪ Remove Auto Answer (NV 74). ▪ Update NV 69744 with new structure members.
AT&T	▪ Set mode preference (NV 10) to GSM only for subscriptions 2 and 3. ▪ Add a new EFS file, *carrier_policy.xml*. ▪ Include updated mandatory APN list as EFS file.

7.1.1 Update source files via the spreadsheet

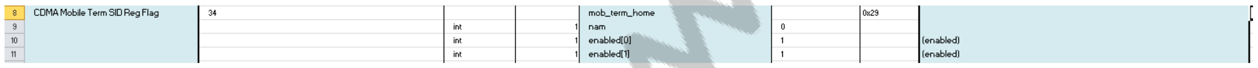
Add an NV item

NV 5 has only one member, so it can be listed with a special form that has member data inline with the NV's metadata.



5	NV Item Name	NV Item ID	NV Item Type	NV Item Size	NV Item Struct	Value	Attributes	Comments
6	Slot Cycle Index	5	int	1	slot_cycle_index	2	0x09	

NOTE: If multiple members were present, then each member would need its own line, e.g., NV 34.

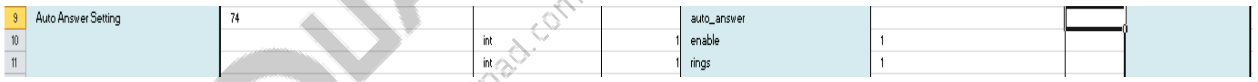


8	CDMA Mobile Term SID Reg Flag	34			mob_term_home		0x29	
9			int	1	nam	0		
10			int	1	enabled[0]	1		(enabled)
11			int	1	enabled[1]	1		(enabled)

Remove an NV item

An item can be removed from the spreadsheet by either:

- Deleting the rows containing the item, or
- Only removing the item's attributes; items with no attributes are considered placeholders in the spreadsheet and are left unused.



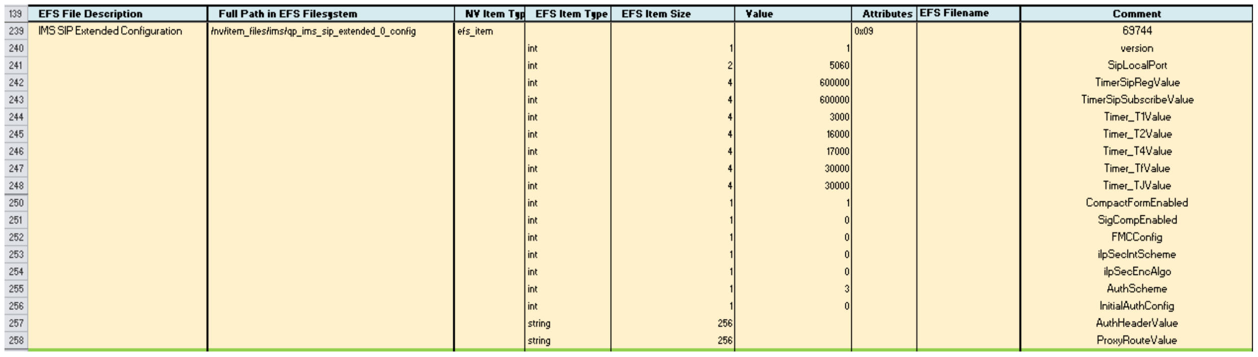
9	Auto Answer Setting	74			auto_answer			
10			int	1	enable	1		
11			int	1	rings	1		

QUALCOMM zhangnan@hipad.com

Edit an EFS item

Add/remove new members within an EFS item by adding/removing rows within the item's structure. The before and after screenshots highlight NV 69744 with four new fields to reflect this new version of the item.

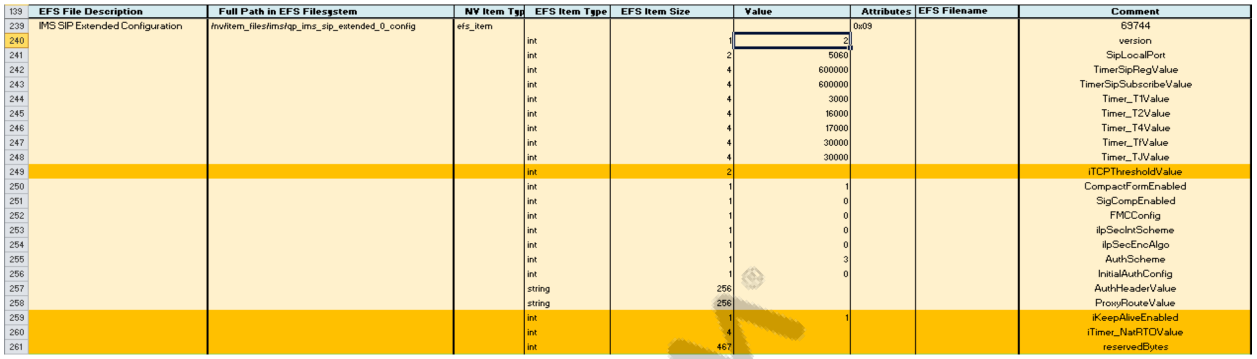
- Before



139	EFS File Description	Full Path in EFS Filesystem	NV Item Typ	EFS Item Type	EFS Item Size	Value	Attributes	EFS Filename	Comment
239	IMS SIP Extended Configuration	/nv/item_files/ims/qp_ims_sip_extended_0_config	efs_item				0x09		69744
240				int	1	1			version
241				int	2	5060			SipLocalPort
242				int	4	600000			TimerSipRegValue
243				int	4	600000			TimerSipSubscribeValue
244				int	4	3000			Timer_T1Value
245				int	4	16000			Timer_T2Value
246				int	4	17000			Timer_T4Value
247				int	4	30000			Timer_TfValue
248				int	4	30000			Timer_TJValue
250				int	1	1			CompactFormEnabled
251				int	1	0			SigCompEnabled
252				int	1	0			FMCConfig
253				int	1	0			ilpSecIntScheme
254				int	1	0			ilpSecEncAlgo
255				int	1	3			AuthScheme
256				int	1	0			InitialAuthConfig
257				string	256				AuthHeaderValue
258				string	256				ProxyRouteValue

Confidential and Proprietary – Qualcomm Technologies, Inc.

- After



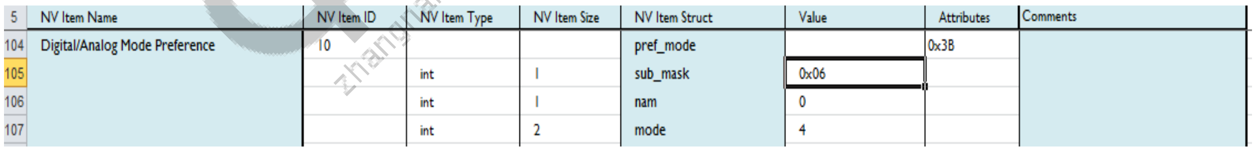
139	EFS File Description	Full Path in EFS Filesystem	NV Item Typ	EFS Item Type	EFS Item Size	Value	Attributes	EFS Filename	Comment
239	IMS SIP Extended Configuration	/nv/item_files/ims/qp_ims_sip_extended_0_config	efs_item				0x09		69744
240				int	1	2			version
241				int	2	5060			SipLocalPort
242				int	4	600000			TimerSipRegValue
243				int	4	600000			TimerSipSubscribeValue
244				int	4	3000			Timer_T1Value
245				int	4	16000			Timer_T2Value
246				int	4	17000			Timer_T4Value
247				int	4	30000			Timer_TfValue
248				int	4	30000			Timer_TJValue
249				int	2				iTCPThresholdValue
250				int	1	1			CompactFormEnabled
251				int	1	0			SigCompEnabled
252				int	1	0			FMCConfig
253				int	1	0			ilpSecIntScheme
254				int	1	0			ilpSecEncAlgo
255				int	1	3			AuthScheme
256				int	1	0			InitialAuthConfig
257				string	256				AuthHeaderValue
258				string	256				ProxyRouteValue
259				int	1	1			iKeepAliveEnabled
260				int	4				iTimer_NatRTOValue
261				int	467				reservedBytes

Any member with a blank value will contain <EFS Item Size> bytes filled with 0x00.

ReservedBytes were added, since they are a part of the item's structure. Some EFS items are partially validated by whether the bytes read in match the structure size, so it is best to always include any reservedBytes from the structure definition.

Include a multisubscription NV

1. For multi-SIM items, an additional member is added before any other member data to indicate to which subscriptions this NV/EFS setting will apply.
   - Bit 4 of the attributes indicates whether an NV/EFS will be treated as a multi-SIM item.
2. To set NV 10 = 13 (GSM only) for the second and third subscriptions, the following lines are added to the spreadsheet:

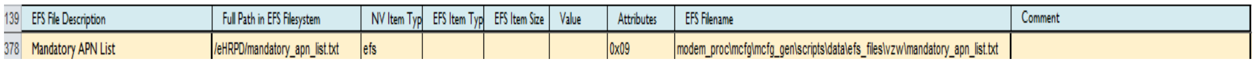


5	NV Item Name	NV Item ID	NV Item Type	NV Item Size	NV Item Struct	Value	Attributes	Comments
104	Digital/Analog Mode Preference	10			pref_mode		0x3B	
105			int	1	sub_mask	0x06		
106			int	1	nam	0		
107			int	2	mode	4		

   - The bitmask 0x06 sets bits 1 and 2, which correspond to subscriptions 2 and 3, since subscription IDs are 0-based indices, i.e., subID 0 = subscription 1.

Add an EFS file

To include a new *carrier_policy.xml* file in the carrier spreadsheet, add a line similar to the following:



139	EFS File Description	Full Path in EFS Filesystem	NV Item Typ	EFS Item Typ	EFS Item Size	Value	Attributes	EFS Filename	Comment
378	Mandatory APN List	/eHRPD/mandatory_apn_list.txt	efs				0x09	modem_proc\mcfg\mcfg_gen\scripts\data\efs_files\vzw\mandatory_apn_list.txt	

The EFS destination must be in UNIX format. The EFS source path can be in either Windows or UNIX format, and must be located somewhere within the build root.

Include an updated EFS file

If contents within an EFS file are updated, there is no need to change anything in the spreadsheet except the version listed in the trailer record.

QUALCOMM zhangyan@hipad.com

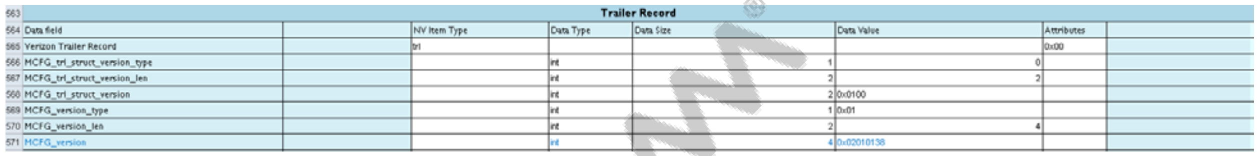
Update the configuration version

Any updates to the .mbn file will require an update to the version listed in the trailer record. Suggested protocol is to increment the carrier version by 1 for each set of NV/EFS updates.

Update version via the spreadsheet

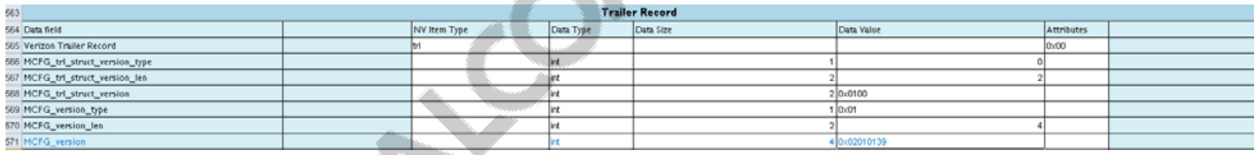
Add 1 to the version listed in the trailer record.

- Before



563	Trailer Record							
564	Data field		NV Item Type	Data Type	Data Size	Data Value	Attributes	
565	Verizon Trailer Record		trl				0x00	
566	MCFG_trl_struct_version_type			int	1	0		
567	MCFG_trl_struct_version_len			int	2	2		
568	MCFG_trl_struct_version			int	2	0x0100		
569	MCFG_version_type			int	1	0x01		
570	MCFG_version_len			int	2	4		
571	MCFG_version			int	4	0x02010138		

- After



563	Trailer Record							
564	Data field		NV Item Type	Data Type	Data Size	Data Value	Attributes	
565	Verizon Trailer Record		trl				0x00	
566	MCFG_trl_struct_version_type			int	1	0		
567	MCFG_trl_struct_version_len			int	2	2		
568	MCFG_trl_struct_version			int	2	0x0100		
569	MCFG_version_type			int	1	0x01		
570	MCFG_version_len			int	2	4		
571	MCFG_version			int	4	0x02010139		

7.1.2 Update source files via XML file

XML updates equivalent to those made in the spreadsheet are described and displayed in the following sections.

NOTE: Updates made directly in an .xml file are not reflected in the spreadsheet.

Also, if an XML is regenerated while using the spreadsheet as a source file, then any updates made directly within the XML will be overwritten and lost.

Add an NV item

1. To add an NV, locate the following line:

```
<NvConfigurationData carrierIndex="1" version="0x02010138" type="1"/>
```

2. Also, insert the item contents somewhere below it:

```
<NvItemData id="5" mcfgAttributes="0x09">    <Member sizeOf="1"
type="uint8">2 </Member></NvItemData>
```

Remove an NV/EFS item

- Before

```
<NvItemData id="71" mcfgAttributes="0x09">   <Member sizeOf="13"
type="uint8">86 69 82 73 90 79 78 </Member></NvItemData><NvItemData id="74"
mcfgAttributes="0x09">   <Member sizeOf="1" type="uint8">1 </Member>
<Member sizeOf="1" type="uint8">1 </Member></NvItemData><NvItemData id="75"
mcfgAttributes="0x09">   <Member sizeOf="1" type="uint8">1 </Member>
<Member sizeOf="1" type="uint8">1 </Member></NvItemData>
```

- After

```
<NvItemData id="71" mcfgAttributes="0x09">   <Member sizeOf="13"
type="uint8">86 69 82 73 90 79 78 </Member></NvItemData><NvItemData id="75"
mcfgAttributes="0x09">   <Member sizeOf="1" type="uint8">1 </Member>
<Member sizeOf="1" type="uint8">1 </Member></NvItemData>
```

Add a multisubscription NV

1. Locate the following line:

```
<NvConfigurationData carrierIndex="1" version="0x02010138" type="1"/>
```

2. Insert these contents some place within the NVConfigurationData element:

```
<NvItemData id="10" mcfgAttributes="0x2B"> <Member sizeOf="1"
type="uint8">0x06 </Member> <Member sizeOf="1" type="uint8">0 </Member>
<Member sizeOf="1" type="uint16">4 </Member></NvItemData>
```

Add an EFS file

To include a new carrier_policy.xml file in the carrier XML, add the following line:

```
<NvEfsFile mcfgAttributes="0x09" targetPath="/policyman/carrier_policy.xml"
buildPath="modem_proc/mmcp/policyman/configurations/Carrier/ATT/carrier_pol
icy.xml"/>
```

The source location (buildPath) can be located anywhere within the build root of the modem image.

NOTE: The EFS destination must be in Unix format. The EFS source path can be in either Windows or Unix format, and must be located somewhere within the build root. The sample buildPath does not currently exist in the modem build.

QUALCOMM zhangmin@chipad.com

Update a version via the XML file

The version is listed in two places within the .xml file. Both will need to be updated.

1. In the corresponding carrier XML, find the following line:

```
<NvConfigurationData carrierIndex="1" version="0x02010138" type="1"/>
```

2. Increment it by 1.

```
<NvConfigurationData carrierIndex="1" version="0x02010139" type="1"/>
```

This first edit is in hexadecimal so the next version update will go to 0x0201013A.

The second update location is in the TRL record, located near the bottom.

This record is identified by the following tag:

```
<NvTrlRecord mcfgAttributes="0x00"> … </NvTrlRecord>
```

The version is the sixth member down in the trailer record, and will look something like this:

```
<NvTrlRecord mcfgAttributes="0x00">
   …
   <Member sizeOf="1" type="uint32">33620281 </Member>
   …
</NvTrlRecord>
```

NOTE: This second update is in decimal.

7.2 Save the source and build the configurations

7.2.1 Save the source

There are no special instructions. Once updates are complete, save the source file.

7.2.2 Build the configuration

- (Option) – Generate source files and MBNs.

  This rebuilds all the configurations listed in the spreadsheet. Time can be saved by passing configuration names directly to the executable used to generate the configuration files.
- (Option) – Use *build_mcfgs.exe* to only build configurations with updates.

  The following build command can be used to regenerate AT&T and Verizon only:
  - CD to the *<build_root>\modem_proc\mcfg\build* directory.
  - Run the following command to rebuild the Verizon and AT&T configurations from the spreadsheet. TAAAANAA is used in the sample, but in practice this will match the build ID of the modem image variant:
    ```
    build_mcfgs.exe --build_id=TAAAANAA --force-regenerate --force-rebuild --configs=mcfg_sw:Verizon,mcfg_sw:ATT
    ```
  - To generate configurations from the XML's sources that were edited directly, append `--xml` to the command: `build_mcfgs.exe --build_id=TAAAANAA --force-regenerate --force-rebuild --configs=mcfg_sw:Verizon,mcfg_sw:ATT –xml`.
  - Force-regenerate and force-rebuild switches are needed to overwrite the carriers' preexisting *.xml and *.mbn files.
- At this point, the MBNs have been regenerated and copied to the configuration directory, *build_root>\modem_proc\mcfg\configs\mcfg_sw\<image>\<product_line>*).

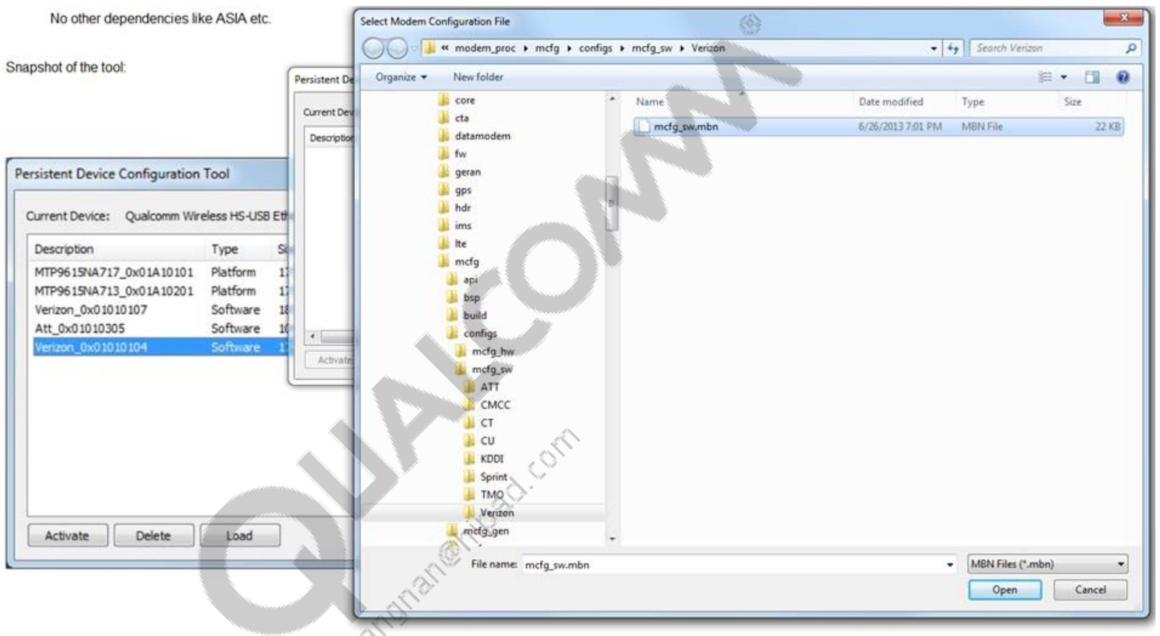
Confidential and Proprietary – Qualcomm Technologies, Inc.

7.3 Load and activate MBNs

7.3.1 Load a carrier configuration

Before a carrier configuration can be flashed to the device, it has to be loaded.

1. Click **Load** on the MCFG_PDC interface from within the QPST Software Download tool and navigate to your carrier configuration.



No other dependencies like ASIA etc.

Snapshot of the tool:

Persistent Device Configuration Tool

Current Device: Qualcomm Wireless HS-USB Eth

Description	Type
MTP9615NA717_0x01A10101	Platform
MTP9615NA713_0x01A10201	Platform
Verizon_0x01010107	Software
Att_0x01010305	Software
Verizon_0x01010104	Software

Select Modem Configuration File

« modem_proc ▸ mcfg ▸ configs ▸ mcfg_sw ▸ Verizon

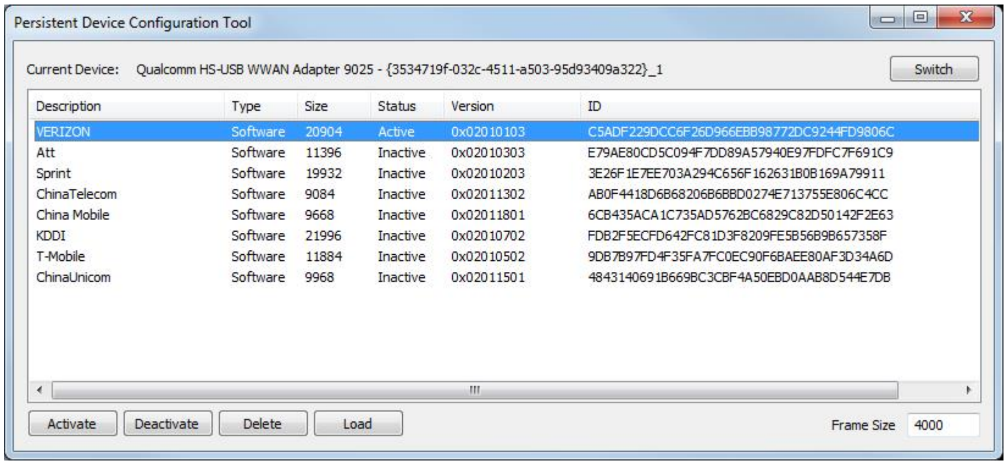
Name	Date modified	Type	Size
mcfg_sw.mbn	6/26/2013 7:01 PM	MBN File	22 KB

File name: mcfg_sw.mbn

MBN Files (*.mbn)

QUALCOMM
zhangnan@...d.com

2. Select the configuration you would like and then click **Activate**. The target will reset and then process the carrier configuration you have chosen.



Persistent Device Configuration Tool

Current Device: Qualcomm HS-USB WWAN Adapter 9025 - {3534719f-032c-4511-a503-95d93409a322}_1

Description	Type	Size	Status	Version	ID
VERIZON	Software	20904	Active	0x02010103	C5ADF229DCC6F26D966EBB98772DC9244FD9806C
Att	Software	11396	Inactive	0x02010303	E79AE80CD5C094F7DD89A57940E97FDFC7F691C9
Sprint	Software	19932	Inactive	0x02010203	3E26F1E7EE703A294C656F162631B0B169A79911
ChinaTelecom	Software	9084	Inactive	0x02011302	AB0F4418D6B68206B6BBD0274E713755E806C4CC
China Mobile	Software	9668	Inactive	0x02011801	6CB435ACA1C735AD5762BC6829C82D50142F2E63
KDDI	Software	21996	Inactive	0x02010702	FDB2F5ECFD642FC81D3F8209FE5B56B9B657358F
T-Mobile	Software	11884	Inactive	0x02010502	9DB7B97FD4F35FA7FC0EC90F6BAEE80AF3D34A6D
ChinaUnicom	Software	9968	Inactive	0x02011501	4843140691B669BC3CBF4A50EBD0AAB8D544E7DB

Frame Size 4000

3. Validate that the MBN has been activated.

   The target should reset during the activation phase. Once it has completed, power cycle to check any of the NV items that were previously edited to verify their settings are correct.

8 Integrating MBNs into Android

This chapter describes the steps to load the user-modified/generated MBN files onto the Android partition so they are available on the UI application for easy user selection.

1. Generate MBNs within the modem environment.

   Generate the MBN with a name that satisfies the naming convention. The name modification should happen only before the first "-" in the name of an existing MBN.

   For example, if you want to modify "EPS_Only-CMCC-CS-SS" and rename the modified MBN, then the modified MBN should be renamed "EPS_Only_XYZ-CMCC-CS-SS" where "XYZ" could be anything the you choose.
2. Copy the generated MBN to a local drive directory on your PC. For example, copy the file to C:\MBNapp.
3. Connect the phone to the PC via USB. Enter the following commands in the command window:
   a. Go to the local directory where the new MBN is copied (C:\MBNapp).
   ```
   cd c:\MBNApp
   ```
   b. Get the root permission.
   ```
   adb root;
   ```
   c. Remount the Android system partition with write permission.
   ```
   adb remount;
   ```
   d. Push the local MBN to the Android partition.
   ```
   adb push "local mbn file name"
   /system/vendor/modemconfig/CMCC/SGLTE/SS/new name.mbn
   ```
   (in this case, new name could be EPS_Only_XYZ-CMCC_SGLTE_SS.mbn)
4. Reboot the phone:
   ```
   adb reboot
   ```

If you want to browse the MBN files in the phone, use the following command:

```
adb root; // get the root permission
cd /system/vendor/modemconfig // enter the /system/vendor/ directory
find // list all files under the current directory
```

QUALCOMM® zhangnan@hipad.com

9 Troubleshooting

9.1 ELF size build error

When the size of the default configuration is larger than the memory space allotted to the device configuration section, an error similar to the following will appear.

```
** Build
errors...<build_root>\modem_proc\core\bsp\devcfg_img\build\devcfg_img\qdsp6
\AAAAANAA\M8974AAAAANAAQ0005_elfparsutil.py_edit.elf failed:
RuntimeError : Error: ELF file
D:\Builds\8974\DI.2.0.c26\latest\as_M8974AAAAANAAM1026020.1_10022013\modem_
proc\build\ms\M8974AAAAANAAQ0005_elfparsutil.py_edit_NODEVCFG.elf's
Section: ".8974_DEVCFG_DATA" not big enough to contain the secondaryELF's
section(s). out_
shdr.sh_size: 349440, total_sec_size: 359528
```

Resolution

1. Navigate to
   `<build root>/modem_proc/core/bsp/build/tbc_core.builds.`
2. Change define
   `DEVCFG_DATA_SEG_SIZE 0x55500`

   to define
   `DEVCFG_DATA_SEG_SIZE 0x59500.`
3. Clean the build by suffixing `--clean` to the build command.
4. Rebuild the image.

QUALCOMM® zhangnan@hipad.com

A Debug Table

Use this table to assist in debugging problems with the logs.

Table A-1 Debug table

	logcat	BTSnoop	kmsg	QXDM Pro	OTA
General IOT	Required	Required	Not required	Not required	Recommended
BT SoC	Not required	Recommended	Not required	Required	Required
BT on/off	Required	Not required	Not required	Required	Not required
SCO Audio	Recommended	Not required	Not required	Required	Required
Android crash	Required	Not required	Required	Not required	Not required
WCNSS crash	Not required	Not required	Not required	Required	Recommended

QUALCOMM®
zhangnan@hipad.com

For an Android/WCNSS crash, a RAM dump log is needed.

For an Android crash, additional logs are needed:

- Tombstone log (/data/tombstones/*)
- anr log (/data/anr/*)
- vmlinux binary file
- Symbols files (<Source code>/out/target/product/msm*/symbols/*)