Qualcomm Technologies, Inc.

Modem Software Configuration Overview

80-N5576-96 L

September 28, 2016

Confidential and Proprietary – Qualcomm Technologies, Inc.

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

Revision history

Revision	Date	Description
A	November 2012	Initial release
B	December 2012	Updated
C	July 2013	Numerous changes were made to this document revision; it should be read in its entirety
D	February 2014	Numerous changes were made to this document revision; it should be read in its entirety
E	November 2014	Restructured and rewritten; this document should be read in its entirety
F	February 2015	Numerous changes were made to this document; it should be read in its entirety
G	April 2015	Updated Section 3.5 and added new Chapter 7
H	May 2015	Updated Section 4.4.1 and added a description of mcfgVariant to Appendix A
J	June 2015	Added Section 7.3
K	January 2016	Updated Section 4.2.6 and Chapter 7
L	September 2016	Added new features of MSM8998 and Multi MBN details

Note: There is no Rev. I, O, Q, S, X, or Z per Mil. standards.

Contents

Figures

Tables

1 Introduction

1.1 Purpose

This document provides information about the Modem Configuration (MCFG) framework and the modem configuration binary (MBN) files. It provides instructions on building a software MBN and loading it onto a device by:

- Using the macro-enabled workbook/spreadsheet (MCFG_SW_Items_List_Macro.xlsm) to generate software MBNs
- Modifying the source XML to generate software MBNs
- Integrating MBNs onto Android
- Loading, activating, and deactivating MBNs using QPST
- Enabling the autoselect mechanism for automatically selecting the correct carrier configuration based on the UICC

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 Technical assistance

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

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

2 MCFG framework overview

This chapter provides an overview on modem configuration MBNs and the modem configuration framework. For step-by-step procedures for generating, loading, and activating configurations, see Chapters 2 through 9.

2.1 Introduction to MCFG MBNs

The goal of the MCFG framework is to enable a single binary image to be paired with configuration data/image that can support multiple software/hardware configurations. This goal is primarily accomplished by using *mcfg_hw.mbn* and *mcfg_sw.mbn* files.

The MBNs are essentially a set of critical NV/EFS and policy manager settings that properly configure the UE to meet the operational requirements of a carrier's network. MBNs are also used to configure the UE to comply with lab testing and certification requirements.

Table 2-1 provides a brief description of each MBN type.

Table 2-1 Description of MBN types

MBN type	Purpose and details
Hardware MBN (*mcfg_hw.mbn*)	▪ Modem platform configuration data (e.g., RFC, *.dat files) ▪ Prepares the UE for accepting a software MBN ▪ Must be loaded to the UE before loading the software MBNs ▪ Separate hardware MBNs for single SIM devices and for dual SIM devices
Software MBN (*mcfg_sw.mbn*)	▪ Configures the UE to comply with lab testing and certification requirements ▪ Contains the necessary NV/EFS configuration for proper operation of the UE ▪ Separate software MBNs for single SIM devices and for dual SIM devices ▪ Separate software MBNs for each UE variant

The MBNs allow OEMS to configure the modem for various technologies (e.g., CDMA2000, GSM-UMTS, LTE, etc.), software features, and carrier-specific customizations. The licensee groups the carrier-specific settings into an MBN that is selected based on the carrier SIM (automatic) or manual methods.

NOTE: 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.

2.2 Introduction to the MCFG framework

Figure 2-1 explains how the MCFG framework provides and uses the default (A), sample (B), and OEM-generated MBNs (C).

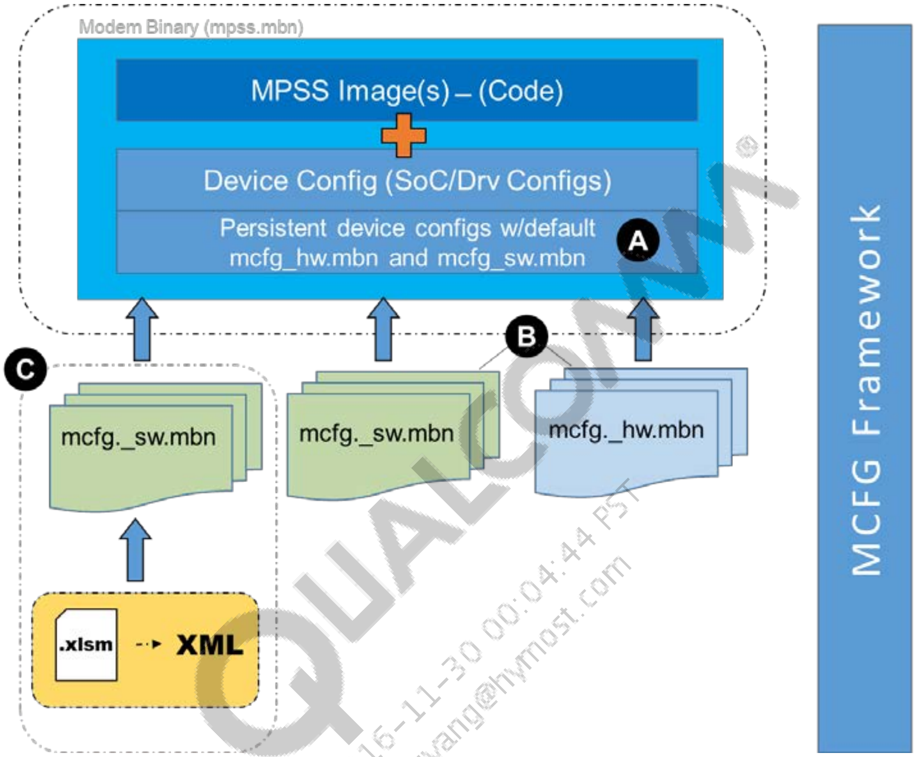


Figure 2-1 MCFG framework feature concept

Ⓐ Default MBNs are embedded into the modem image. Typically, the NV/EFS settings persistent to a device and common across a product line are expected to be included in the default configuration.

Ⓑ Reference MBNs that contain the QTI-recommended settings and the necessary configurations for proper operation on specific commercial networks or for lab testing. Provision the UE by locating, loading, and activating the carrier-specific MBN that corresponds to the device type and SIM capabilities of the device.

Ⓒ OEMs can generate their own MBNs, if necessary, to meet product-specific requirements. There are two ways to generate MBNs:

- Modifying the XML source code
- Using the MCFG_SW_Items_List_Macro.xlsm, which in turn generates XML

The MCFG framework involves:

- Loading and activating MBNs
- Authenticating the MBNs
- Processing the MBN types (the hardware MBN for RFC, UIM configurations; the software MBN for NV/EFS, carrier-specific settings)
- Selecting/downloading the MBNs using QPST or UICC-based (automatic) mechanisms

- Generating custom MBNs (primarily the software MBNs although the hardware MBNs are also modified)

The MCFG framework brings in a new modem configuration task that runs before any of the other modem tasks are started. It configures the modem according to the active hardware MBN and the active software MBN. If the autoselection feature is enabled, the correct carrier software MBN is automatically selected based on the Issuer Identification Number (IIN) field of the ICCID in the SIM. See Chapter 9 for more information on autoselection.

2.3 OEM-generated MBNs

OEMs typically generate their own MBNs by modifying one of the reference MBNs provided by QTI to meet to their product requirements. There are two ways to generate MBNs:

- Using the macro-enabled spreadsheet
- Modifying the source XML

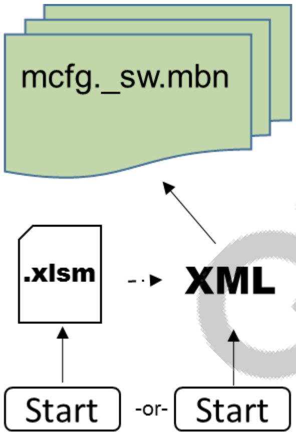


Figure 2-2 Two ways to generate MBNs

Each method uses the same XML schema as the default MBN. This allows any carrier XML to be used as a substitute for the default XML, and then compiled into the modem image as the default configuration.

2.3.1 Prerequisites for OEM-generated MBNs

The following software packages are required to generate MBNs.

- For Nickel, Dime, Triton, and Bolt products:
  - Perl 5.6
  - Python 2.7.5
  - A complete MPSS build
  - Qualcomm® Hexagon™ Toolset (version specific to the modem build)
  - Microsoft Excel 2010 or 2013 (if using the macro-enabled spreadsheet)

- For Jolokia 1.0 and later:
  - Perl 5.18 or later
  - Microsoft Excel 2010 or 2013 (if using the macro-enabled spreadsheet)
  - Complete *modem_proc\mcfg folder** from MPSS source code
  - Complete *modem_proc\mmcp folder** from MPSS source code
- For MSM8998 and later:
  - Perl 5.18 or later
  - Complete *modem_proc\mcfg folder** from MPSS source code
  - Complete *modem_proc\mmcp folder** from MPSS source code

NOTE: *The mmcp and mcfg folders must both be present in a parent directory titled *modem_proc*.

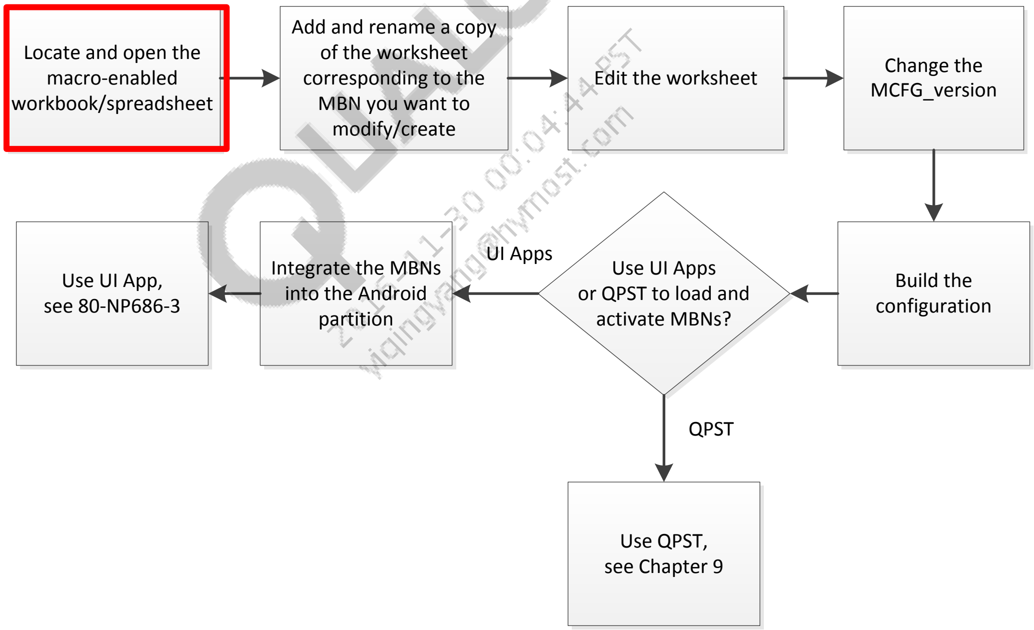
2.3.2 Security

The configuration data is authenticated before flashing into the modem EFS and every time it is processed. The signing infrastructure for the current implementation of MCFG images is the Code Signing Management System (CSMS) mechanism, which is also available for main images. See *Presentation: Secured MSM™ Code Signing Service* (80-V9807-1) and *Presentation: Code Signing Management System Overview* (80-V3999-1) for more information.

3 Generating MBNs using the macro-enabled workbook

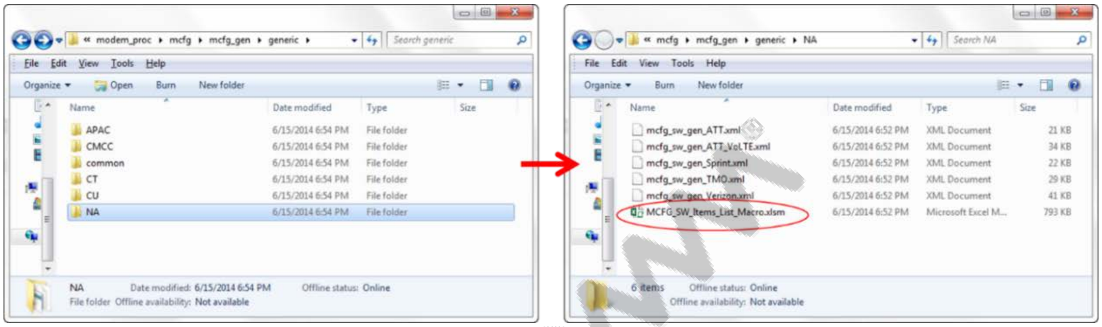
NOTE: Chapter 3 and 4 are applicable for targets prior to MSM8998 and MDM9x50. See Chapter 5 for MBN changes implemented with MSM8998 and MDM9x50.

The MCFG software items list workbook provides OEMs with a user-friendly mechanism for viewing and updating the NV/EFS settings in software MBNs. The macros in the workbook enable OEMs to generate updated MBNs with the click of a button. This section provides step-by-step procedures for using the workbook to generate software MBNs. See Chapter 10 for an overview of the workbook.


Locate and open the macro-enabled workbook/spreadsheet
Add and rename a copy of the worksheet corresponding to the MBN you want to modify/create
Edit the worksheet
Change the MCFG_version
Build the configuration
Use UI Apps or QPST to load and activate MBNs?
UI Apps
Integrate the MBNs into the Android partition
Use UI App, see 80-NP686-3
QPST
Use QPST, see Chapter 9

3.1 Locate and open the MCFG software items workbook

There are separate workbooks for each region and carrier. Figure 3-1 shows the directories (APAC, CMCC, etc.) for each of the regions and the contents of NA directory. The other directories contain the workbooks and source files specific to that region or carrier.


« modem_proc ▸ mcfg ▸ mcfg_gen ▸ generic ▸
Search generic
File Edit View Tools Help
Organize ▾ Open Burn New folder
« mcfg ▸ mcfg_gen ▸ generic ▸ NA
Search NA
File Edit View Tools Help
Organize ▾ Burn New folder

Name	Date modified	Type	Size
APAC	6/15/2014 6:54 PM	File folder	
CMCC	6/15/2014 6:54 PM	File folder	
common	6/15/2014 6:54 PM	File folder	
CT	6/15/2014 6:54 PM	File folder	
CU	6/15/2014 6:54 PM	File folder	
NA	6/15/2014 6:54 PM	File folder	

NA Date modified: 6/15/2014 6:54 PM Offline status: Online
File folder Offline availability: Not available

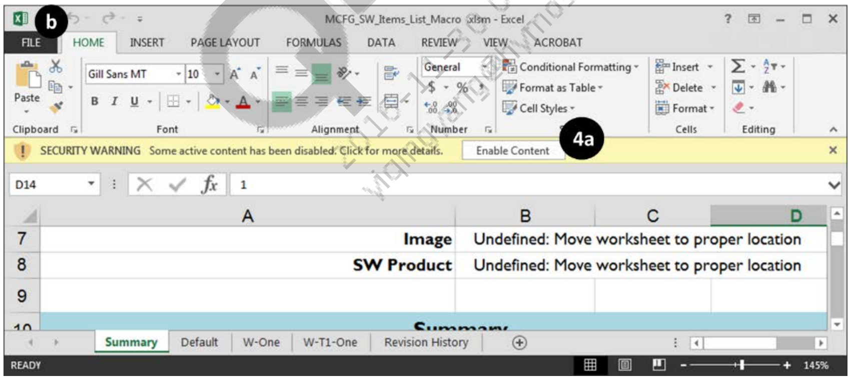
Name	Date modified	Type	Size
mcfg_sw_gen_ATT.xml	6/15/2014 6:52 PM	XML Document	21 KB
mcfg_sw_gen_ATT_VoLTE.xml	6/15/2014 6:52 PM	XML Document	34 KB
mcfg_sw_gen_Sprint.xml	6/15/2014 6:52 PM	XML Document	22 KB
mcfg_sw_gen_TMO.xml	6/15/2014 6:52 PM	XML Document	29 KB
mcfg_sw_gen_Verizon.xml	6/15/2014 6:52 PM	XML Document	41 KB
MCFG_SW_Items_List_Macro.xlsm	6/15/2014 6:52 PM	Microsoft Excel M...	793 KB

6 items Offline status: Online
Offline availability: Not available

Figure 3-1 Location of the MCFG software items workbook

To locate and open the source files, do the following:

1. Move to *$BUILD_ROOT\ modem_proc\mcfg\mcfg_gen\generic*.
2. Open the directory for your region or carrier.
3. Open the file named MCFG_SW_Items_List_Macro.xlsm.

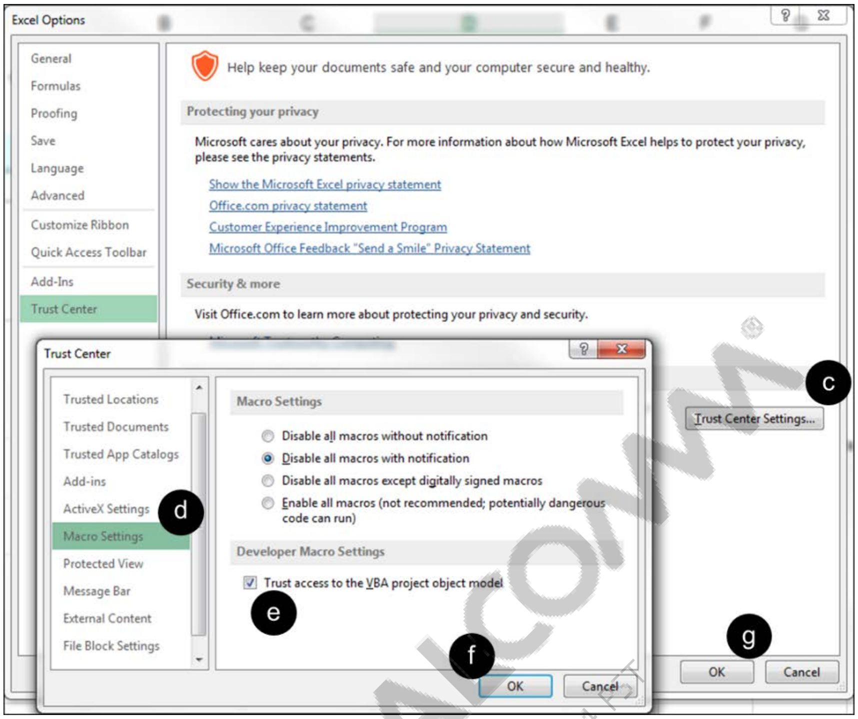

b
MCFG_SW_Items_List_Macro .xlsm - Excel
FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW ACROBAT
Paste
Gill Sans MT
10
General
Conditional Formatting
Format as Table
Cell Styles
Insert
Delete
Format
Clipboard Font Alignment Number Cells Editing
4a
Summary Default W-One W-T1-One Revision History
READY
145%

SECURITY WARNING Some active content has been disabled. Click for more details. Enable Content

D14 1

	A	B	C	D
7	Image	Undefined: Move worksheet to proper location		
8	SW Product	Undefined: Move worksheet to proper location		
9				
10	Summary			

4. Allow editing and macros to run, if prompted by Excel.
   a. If a SECURITY WARNING appears in the workbook, click **Enable Content**.
   b. Select File > Options > Trust Center.
   c. Click Trust Center Settings.
   d. In the Trust Center window, select **Macro Settings**.
   e. Select the Trust access to the VBA project object model check box.
   f. Click **OK**.
   g. Click **OK**.



Help keep your documents safe and your computer secure and healthy.

Protecting your privacy

Microsoft cares about your privacy. For more information about how Microsoft Excel helps to protect your privacy, please see the privacy statements.

Show the Microsoft Excel privacy statement

Office.com privacy statement

Customer Experience Improvement Program

Microsoft Office Feedback "Send a Smile" Privacy Statement

Security & more

Visit Office.com to learn more about protecting your privacy and security.

Macro Settings

○ Disable all macros without notification

● Disable all macros with notification

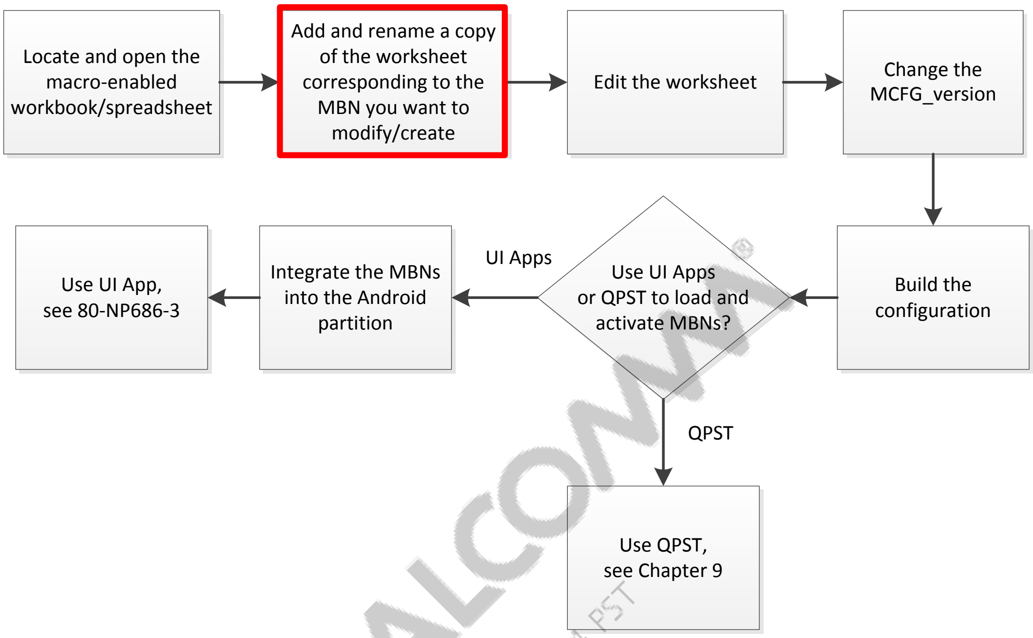
○ Disable all macros except digitally signed macros

○ Enable all macros (not recommended; potentially dangerous code can run)

Developer Macro Settings

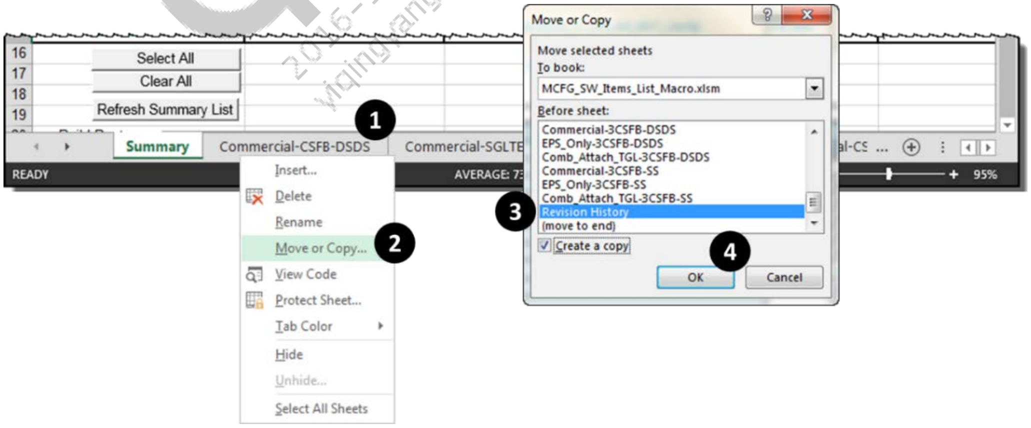
☑ Trust access to the VBA project object model

3.2 Add and rename worksheets


Locate and open the macro-enabled workbook/spreadsheet
Add and rename a copy of the worksheet corresponding to the MBN you want to modify/create
Edit the worksheet
Change the MCFG_version
Build the configuration
Use UI Apps or QPST to load and activate MBNs?
UI Apps
Integrate the MBNs into the Android partition
Use UI App, see 80-NP686-3
QPST
Use QPST, see Chapter 9

To add and rename worksheets, do the following:

1. After you have determined which MBN to use as a base for your new configuration, right-click the corresponding tab name at the bottom of the workbook. The Commercial-CSFB-DSDS is used as an example.

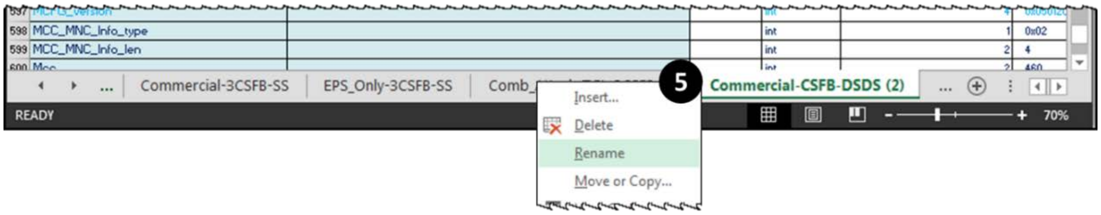


2. Select Move or Copy.

3. The copy must be placed before the Revision History worksheet. Select **Revision History** and the **Create a copy** check box.

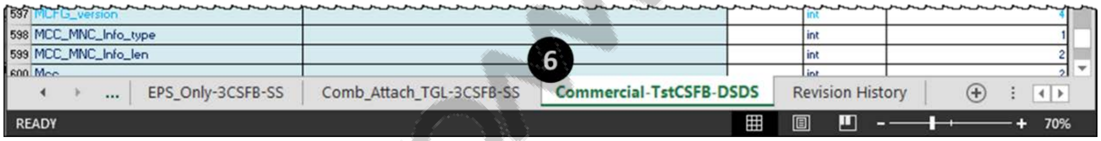
4. Click **OK**. Excel creates and displays the copy.

5. Right-click the new worksheet and select **Rename**.



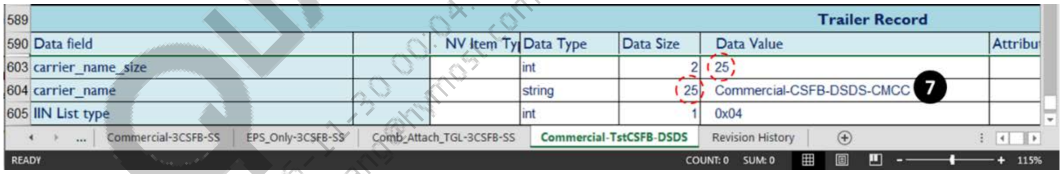
598	MCC_MNC_Info_type			int	1	0x02
599	MCC_MNC_Info_len			int	2	4

6. Type the new name for the MBN using the following naming conventions:
   - □ Add two or three characters before the first hyphen in the name.
   - □ Limit the Excel tab name to 31 characters.
   - □



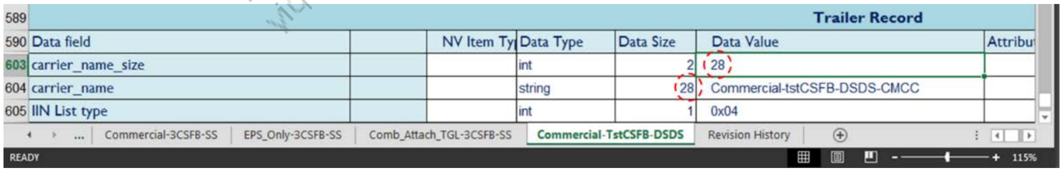
598	MCC_MNC_Info_type		int	1
599	MCC_MNC_Info_len		int	2

7. Change the carrier_name in the Trailer Record portion of the worksheet. The carrier_name_size row and the Data Size column of the carrier_name are automatically updated when you change the name.
   - □ Before



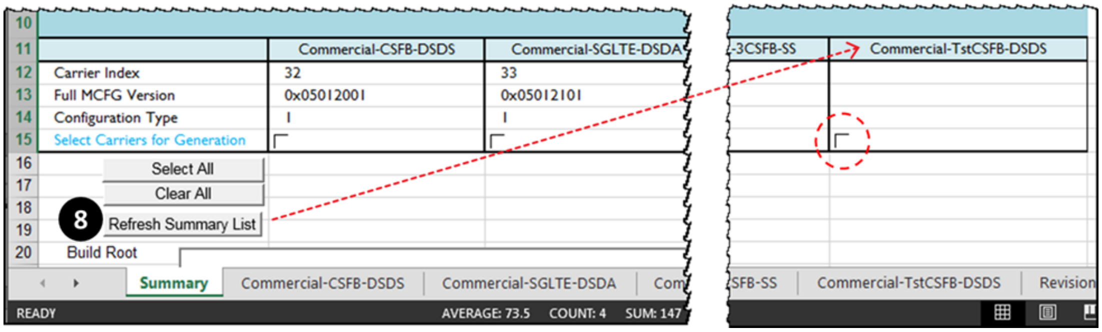
589						Trailer Record	
590	Data field		NV Item Ty	Data Type	Data Size	Data Value	Attribu
603	carrier_name_size			int	2	25	
604	carrier_name			string	25	Commercial-CSFB-DSDS-CMCC	
605	IIN List type			int	1	0x04	

   - □ After



589						Trailer Record	
590	Data field		NV Item Ty	Data Type	Data Size	Data Value	Attribu
603	carrier_name_size			int	2	28	
604	carrier_name			string	28	Commercial-tstCSFB-DSDS-CMCC	
605	IIN List type			int	1	0x04	

8. Click the Summary sheet tab and click **Refresh Summary List**. A new column displaying the name of the new worksheet is inserted as the last column on the right side of the summary sheet. The new column includes a check box.



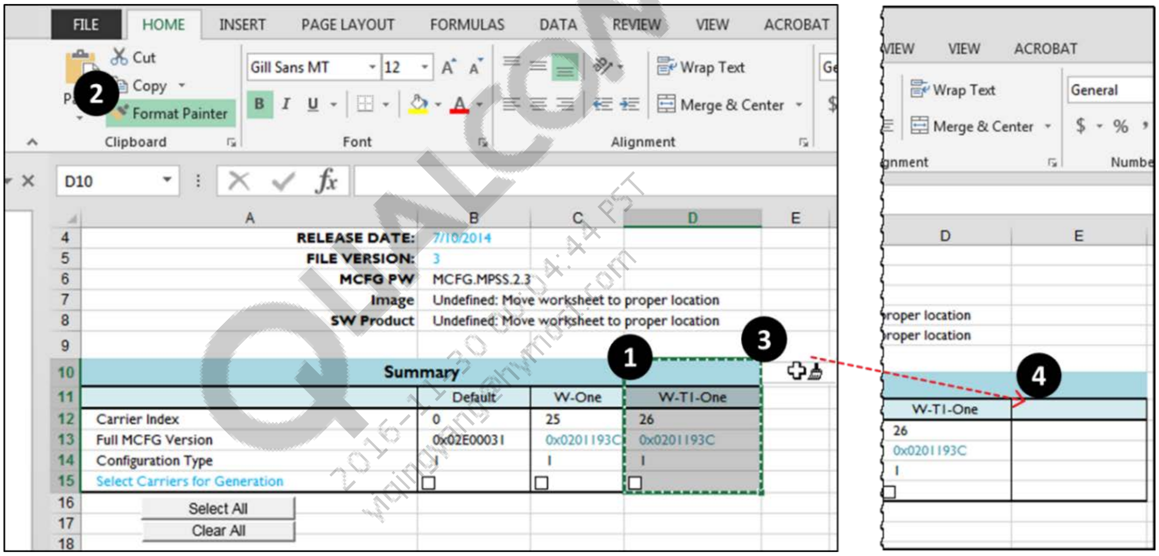
		Commercial-CSFB-DSDS	Commercial-SGLTE-DSDA	-3CSFB-SS	Commercial-TstCSFB-DSDS
12	Carrier Index	32	33		
13	Full MCFG Version	0x05012001	0x05012101		
14	Configuration Type	1	1		
15	Select Carriers for Generation	□	□		□

9. Type the values for the new column.
   - Carrier index – Type the same value as the adjacent column.
   - Full MCFG Version – Go to the new worksheet that corresponds to the new column. Copy the value for the MCFG_version in the Trailer Record portion of the worksheet (near the bottom). Return to the summary sheet and paste the value.
   - Configuration type – 1 = software configuration, 0 = hardware configuration

After the Summary table is up to date, generate source files and MBNs.

Process for when there is no Refresh Summary List button

The workbooks for older PLs do not have the Refresh Summary List button. In such cases, manually add the new column.

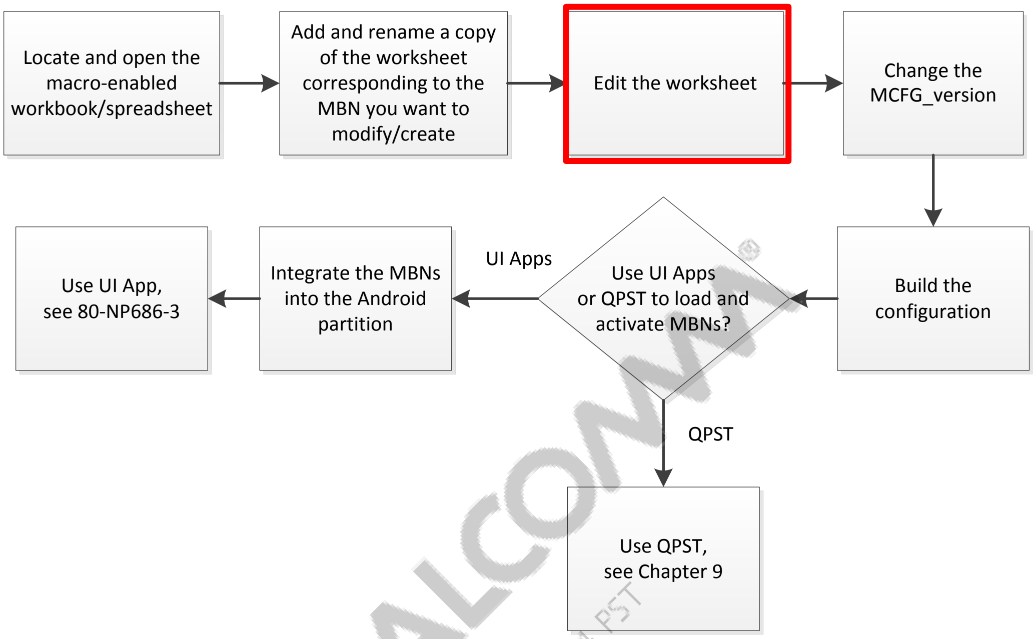


	A	B	C	D	E
4	RELEASE DATE:	7/10/2014			
5	FILE VERSION:	3			
6	MCFG PW	MCFG.MPSS.2.3			
7	Image	Undefined: Move worksheet to proper location			
8	SW Product	Undefined: Move worksheet to proper location			
9					
10	Summary				
11		Default	W-One	W-TI-One	
12	Carrier Index	0	25	26	
13	Full MCFG Version	0x02E00031	0x0201193C	0x0201193C	
14	Configuration Type	1	1	1	
15	Select Carriers for Generation	□	□	□	

D	E
W-TI-One	
26	
0x0201193C	
1	
□	

1. Select one of the existing configuration columns from the summary sheet.
2. Select **Format Painter**. The cursor turns into a plus sign with a paintbrush.
3. Paste it next to the last configuration column as shown. A new column appears.
4. Type the same name that you used in Step 6 of the previous section.
5. Save the file, close it, and reopen it. A macro runs when you reopen the file and functions, including the check box, are activated for the cells pasted in Step 3.

3.3 Edit the worksheet


Locate and open the macro-enabled workbook/spreadsheet
Add and rename a copy of the worksheet corresponding to the MBN you want to modify/create
Edit the worksheet
Change the MCFG_version
Build the configuration
Use UI Apps or QPST to load and activate MBNs?
UI Apps
Integrate the MBNs into the Android partition
Use UI App, see 80-NP686-3
QPST
Use QPST, see Chapter 9

A complete review of the NV/EFS items in the worksheets is beyond the scope of this document. Nevertheless, it is useful to understand that the worksheets are arranged into three sections:

- NV Items at the top of the worksheet
- EFS Files in the center
- Trailer Record information at the bottom

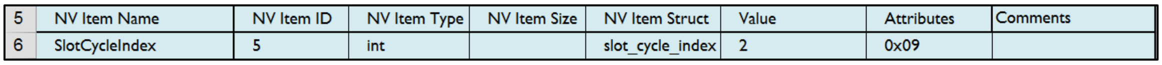
See Chapter 12 for additional information regarding the format and entries in the worksheets. This section provides examples of the following common use cases:

- Adding an NV item
- Removing an NV item
- Editing an EFS item
- Including a multisubscription NV item
- Add an EFS file
- Including an updated EFS file

3.3.1 Add an NV item

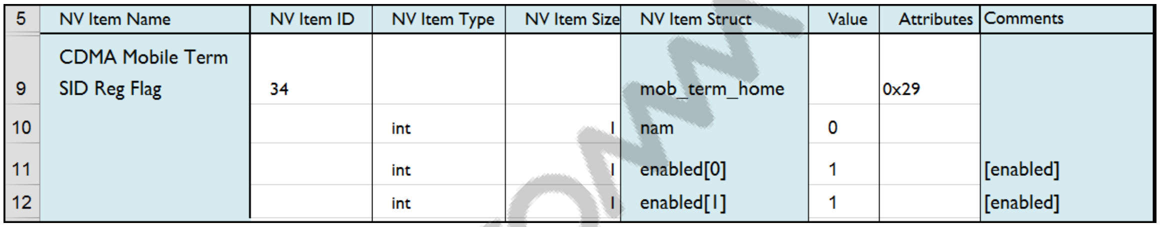
To add an NV item, move to the NV Items section of the worksheet and insert a row using basic Excel functionality.

If the item has only one member, then list the member data on the same line as the item's metadata. NV 5 is an example of an item with only one member.



5	NV Item Name	NV Item ID	NV Item Type	NV Item Size	NV Item Struct	Value	Attributes	Comments
6	SlotCycleIndex	5	int		slot_cycle_index	2	0x09	

If multiple members are present, then each member needs its own line, e.g., NV 34.

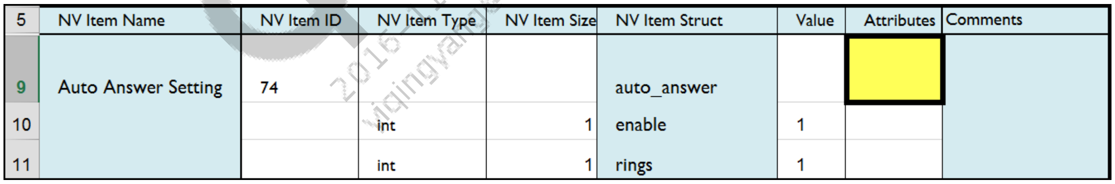


5	NV Item Name	NV Item ID	NV Item Type	NV Item Size	NV Item Struct	Value	Attributes	Comments
9	CDMA Mobile Term SID Reg Flag	34			mob_term_home		0x29	
10			int	I	nam	0		
11			int	I	enabled[0]	1		[enabled]
12			int	I	enabled[I]	1		[enabled]

3.3.2 Remove an NV item

To remove an NV item, do one of the following:

- Delete the rows containing the item.
- Remove the item's attributes. Items with no attributes are considered placeholders in the spreadsheet and are not used.

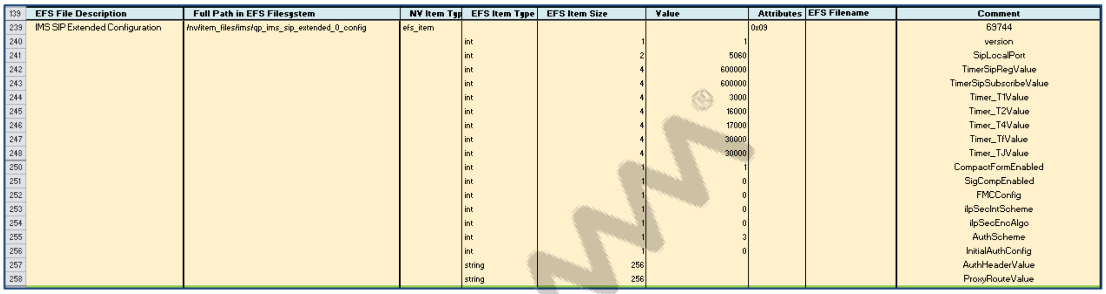


5	NV Item Name	NV Item ID	NV Item Type	NV Item Size	NV Item Struct	Value	Attributes	Comments
9	Auto Answer Setting	74			auto_answer			
10			int	1	enable	1		
11			int	1	rings	1		

3.3.3 Edit an EFS item

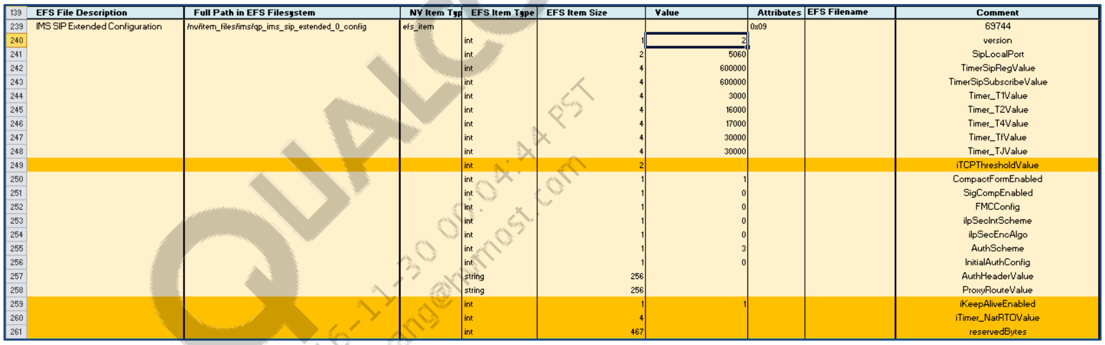
Add or remove new members within an EFS item by adding or removing rows within the item's structure. The before and after screenshots highlight NV 69744 with four new fields to reflect a new version of the item.

- 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

- After



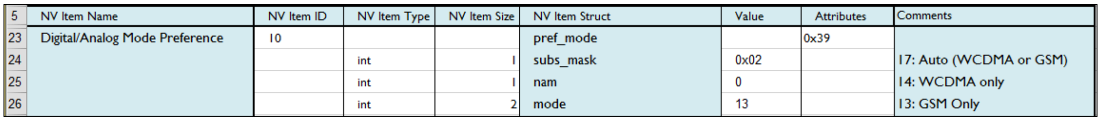
139	EFS File Description	Full Path in EFS Filesystem	NV Item Typ	EFS Item Type	EFS Item Size	Value	Attributes	EFS Filename	Comment
239	IMS SIP Extended Configuration	/nv/item_files/ims/qp_ims_sip_extended_0_config	efs_item				0x09		69744
240				int	1	2			version
241				int	2	5060			SipLocalPort
242				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 contains <EFS Item Size> bytes filled with 0x00.

ReservedBytes were added, because 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.

3.3.4 Include a multisubscription NV

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



5	NV Item Name	NV Item ID	NV Item Type	NV Item Size	NV Item Struct	Value	Attributes	Comments
23	Digital/Analog Mode Preference	10			pref_mode		0x39	
24			int	1	subs_mask	0x02		17: Auto (WCDMA or GSM)
25			int	1	nam	0		14: WCDMA only
26			int	2	mode	13		13: GSM Only

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

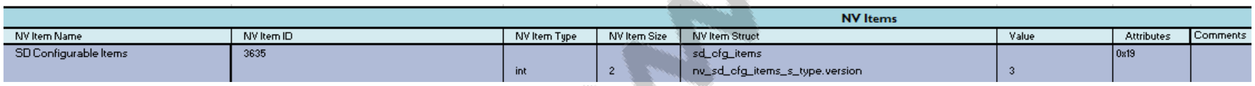
3.3.5 Making a multiplexed NV item

For carriers in which specific NV/EFS could be updated through an OTA process and which the updated NV/EFS values need to be preserved, the NV/EFS must be defined with the multiplexed attribute.

An example for the need to use this multiplexed attribute is for devices that are going to Sprint. They use OMADM to update some NV/EFS parameters.

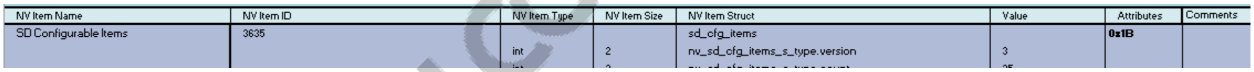
As defined in Section 4.2.5, the bit corresponding to multiplexed attribute is bit 2. This bit will need to set to 1. Please see example below:

Original NV item attribute value (attribute = 0x19):



NV Items							
NV Item Name	NV Item ID	NV Item Type	NV Item Size	NV Item Struct	Value	Attributes	Comments
SD Configurable Items	3635			sd_cfg_items		0x19	
		int	2	nv_sd_cfg_items_s_type.version	3		

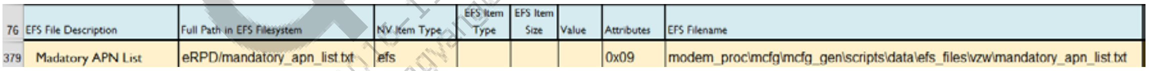
Updating the NV's attribute to make it a multiplexed item (attribute changed to 0x1B):



NV Item Name	NV Item ID	NV Item Type	NV Item Size	NV Item Struct	Value	Attributes	Comments
SD Configurable Items	3635			sd_cfg_items		**0x1B**	
		int	2	nv_sd_cfg_items_s_type.version	3		

3.3.6 Add an EFS file

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



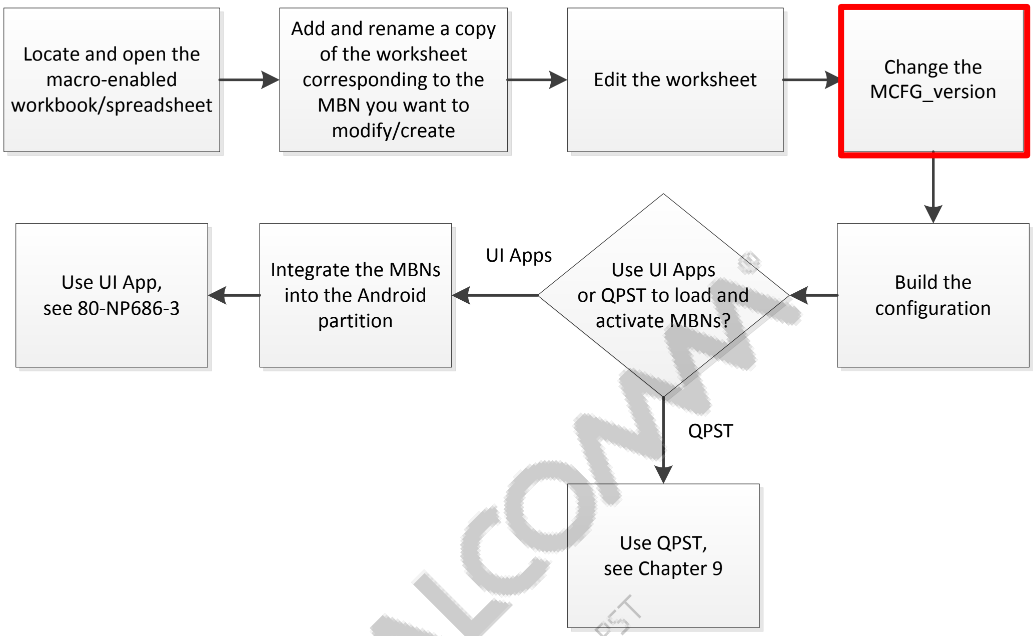
76	EFS File Description	Full Path in EFS Filesystem	NV Item Type	EFS Item Type	EFS Item Size	Value	Attributes	EFS Filename
379	Madatory APN List	eRPD/mandatory_apn_list.txt	efs				0x09	modem_proc\mcfg\mcfg_gen\scripts\data\efs_files\vzw\mandatory_apn_list.txt

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.

3.3.7 Include an updated EFS file

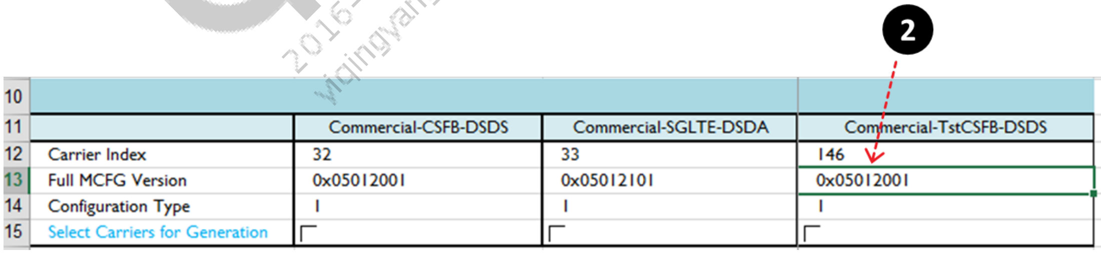
If contents within an EFS file are updated, there is no need to change anything in the spreadsheet except the version listed in the Trailer Record.

3.4 Change the MCFG_version


Locate and open the macro-enabled workbook/spreadsheet
Add and rename a copy of the worksheet corresponding to the MBN you want to modify/create
Edit the worksheet
Change the MCFG_version
Build the configuration
Use UI Apps or QPST to load and activate MBNs?
UI Apps
Integrate the MBNs into the Android partition
Use UI App, see 80-NP686-3
QPST
Use QPST, see Chapter 9

Before you build the configuration, you must change the MCFG_version of the configuration you updated. 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 is not loaded to the target. Avoid this scenario by updating the MCFG_version.

1. Move to the summary worksheet.



10				
11		Commercial-CSFB-DSDS	Commercial-SGLTE-DSDA	Commercial-TstCSFB-DSDS
12	Carrier Index	32	33	146
13	Full MCFG Version	0x05012001	0x05012101	0x05012001
14	Configuration Type	1	1	1
15	Select Carriers for Generation	☐	☐	☐

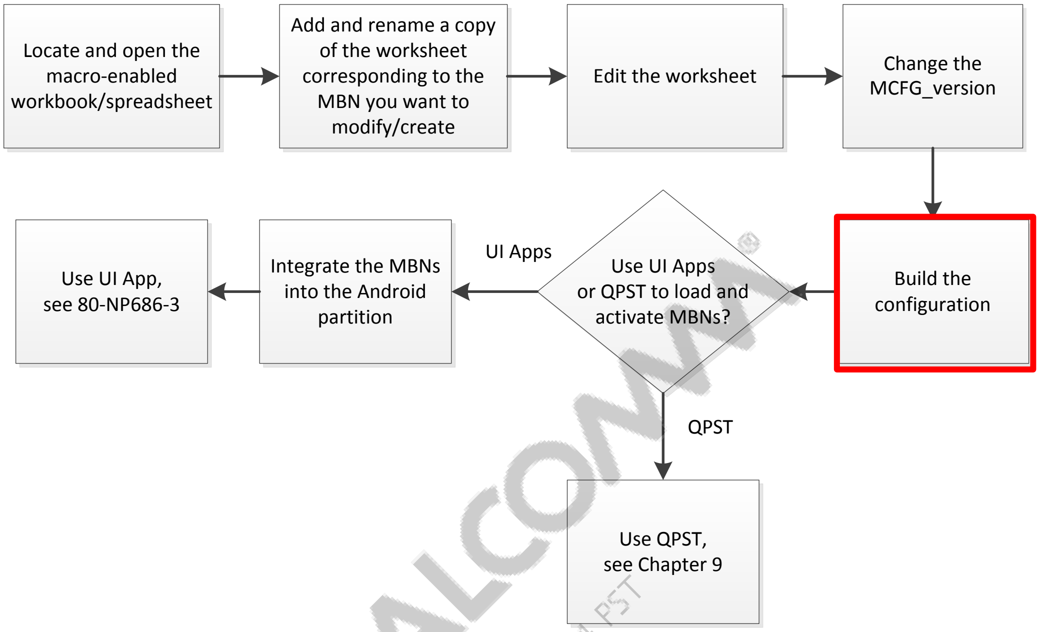
2. Increment the second byte of the Full MCFG Version by 1 (for example, change 0x05012001 to 0x0502201).

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

3. Move to the worksheet for this column and verify that the MCFG_version in the Trailer Record was also updated (the cells are linked). If it did not update, change the version in the Trailer Record section so that it matches the version in the summary.

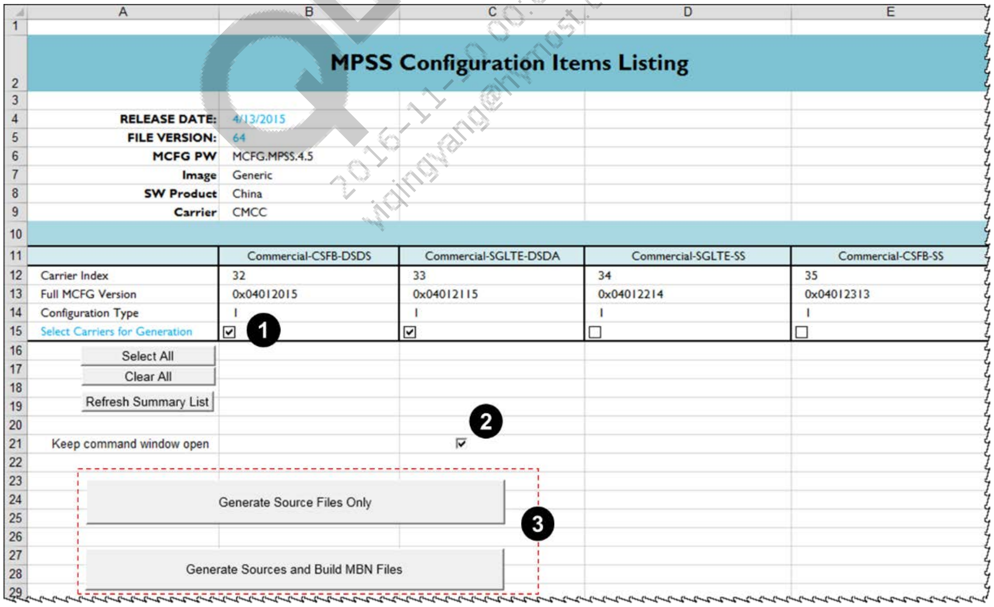
4. Save the file.

3.5 Build the configuration



Locate and open the macro-enabled workbook/spreadsheet
Add and rename a copy of the worksheet corresponding to the MBN you want to modify/create
Edit the worksheet
Change the MCFG_version
Build the configuration
Use UI Apps or QPST to load and activate MBNs?
UI Apps
Integrate the MBNs into the Android partition
Use UI App, see 80-NP686-3
QPST
Use QPST, see Chapter 9

To build the configuration, click the Summary tab and do the following:



MPSS Configuration Items Listing

RELEASE DATE:	4/13/2015
FILE VERSION:	64
MCFG PW	MCFG.MPSS.4.5
Image	Generic
SW Product	China
Carrier	CMCC

	Commercial-CSFB-DSDS	Commercial-SGLTE-DSDA	Commercial-SGLTE-SS	Commercial-CSFB-SS
Carrier Index	32	33	34	35
Full MCFG Version	0x04012015	0x04012115	0x04012214	0x04012313
Configuration Type	1	1	1	1
Select Carriers for Generation	☑	☑	☐	☐

Keep command window open ☑

Generate Source Files Only
Generate Sources and Build MBN Files
1
2
3

1. Select the checkboxes for each configuration you want to build. Use the **Select All** and **Clear All** buttons to select or clear all checkboxes. Optionally, manually select or deselect configurations one at a time by clicking the checkbox.

2. Ensure that the **Keep command window open** checkbox is selected so that errors encountered during generation, if any, are displayed. If this option is unchecked, the command window closes immediately after the generation process terminates even if it terminates with errors.

3. There are two different options for building the configuration.
   - 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.
   - Generate Source Files 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.

3.5.1 Signing MBN files

On a secure device with an eFuse chip, MBNs are authenticated and only secure signed MBNs are loaded. Use the sectools component of the CSMS to generate secure signed MBNs from unsigned MBNs. See *Sectools SecImage User Guide* (80-NM248-1) for more information.

To sign MBN files, do the following:

1. If you are using security tools v3.10 or later, see step 2. If you are using previous versions of security tools, update the MCFG sign_ID in the XML as follows:

```
<image sign_id="mcfg_hw" name="mcfg_hw.mbn" image_type="elf_has_ht">
    <general_properties_overrides>
        <sw_id>0x0000000000000002</sw_id>
    </general_properties_overrides>
</image>

<image sign_id="mcfg_sw" name="mcfg_sw.mbn" image_type="elf_has_ht">
    <general_properties_overrides>
        <sw_id>0x0000000000000002</sw_id>
    </general_properties_overrides>
</image>
```

NOTE: The actual format of the XML may vary due to the version.

2. To sign the MBN, `cd` to `<Meta_path>\common\tools` then issue one of the following commands:
   - For software MBNs
     ```
     python sectools.py secimage -s -g mcfg_sw -i <unsigned_mbn_path>
     -o <signed_mbn_path>
     ```
   - For hardware MBNs
     ```
     python sectools.py secimage -s -g mcfg_hw -i <unsigned_mbn_path>
     -o <signed_mbn_path>
     ```

3.5.2 About the 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.

The buttons in the workbook 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.

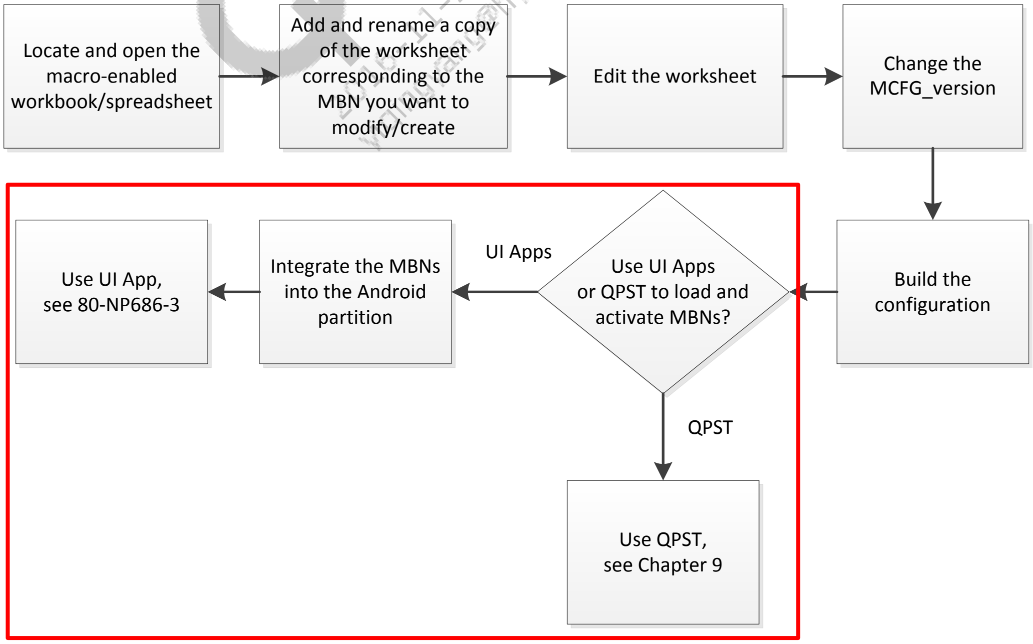
Spreadsheet button	Equivalent command
Generate Source Files Only	>build_mcfgs.exe --build_id=TAAANAA --configs=mcfg_sw:all --force-regenerate --sources-only
Generate Sources and Build MBN Files	>build_mcfgs.exe --build_id=TAAANAA --configs=mcfg_sw:all --force-regenerate --force-rebuild

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

3.5.3 Schema of XMLs generated by the workbook

The XML generated by MCFG_SW_Items_List_Macro.xlsm uses a schema with tags that closely parallel the listing structure within the workbook itself. See Appendix A for more information.

3.6 Load and activate MBNs


Locate and open the macro-enabled workbook/spreadsheet
Add and rename a copy of the worksheet corresponding to the MBN you want to modify/create
Edit the worksheet
Change the MCFG_version
Build the configuration
Use UI Apps or QPST to load and activate MBNs?
UI Apps
Integrate the MBNs into the Android partition
Use UI App, see 80-NP686-3
QPST
Use QPST, see Chapter 9

The next step of the process is to load and activate the MBNs. The tools you use to perform these tasks depends on the type of release.

- Windows customers must use QPST.
- Android customers can use either QPST or UI applications.

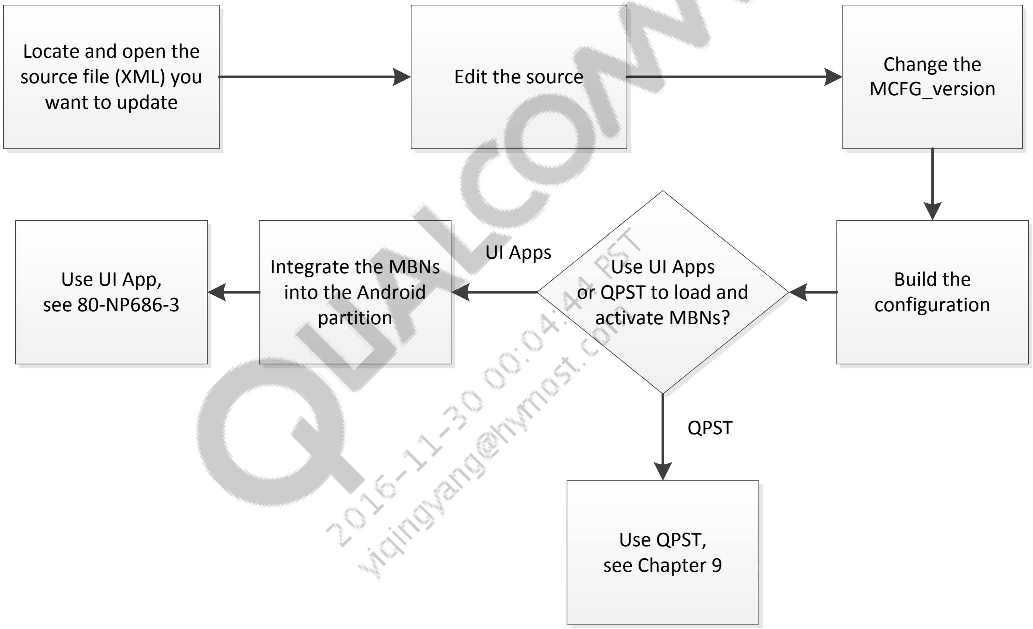
Where available, UI applications are the recommended method to use for Android images during testing because they enable faster switching between MBNs. QPST is recommended for factory use but can also be used during testing.

To use UI applications to load and activate the newly generated MBNs, integrate the new MBNs into the Android partition.

- For information on integrating MBNs into the Android partition, see Chapter 8.
- For information on using QPST, see Chapter 9.
- For information on using UI applications, see *Application Note: Configuring a UE Using Binary Modem Configuration* (80-NP686-1).

4 Generating MBNs by modifying the XML source

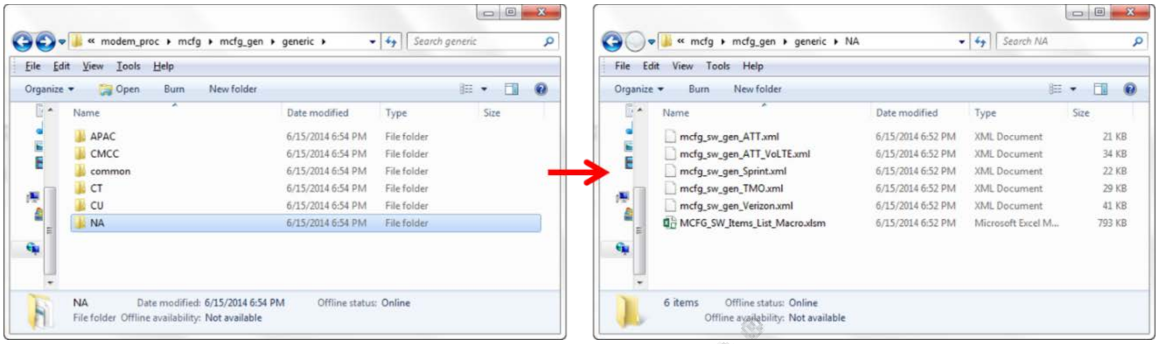
NOTE: Chapter 3 and 4 are applicable for targets prior to MSM8998 and MDM9x50. See Chapter 5 for MBN changes implemented with MSM8998 and MDM9x50.


Locate and open the source file (XML) you want to update
Edit the source
Change the MCFG_version
Build the configuration
Use UI Apps or QPST to load and activate MBNs?
UI Apps
Integrate the MBNs into the Android partition
Use UI App, see 80-NP686-3
QPST
Use QPST, see Chapter 9

MBNs can also be updated or created by modifying the XML source files. Keep in mind that 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.

4.1 Locate and open the source

The XML source files are located in the same directories as the workbooks. The following figure shows the directories for each of the regions and the contents of NA directory. The other directories, e.g., APAC, CMCC, etc., contain the source files specific to that region or carrier.



« modem_proc ▸ mcfg ▸ mcfg_gen ▸ generic ▸

Search generic

Name	Date modified	Type	Size
APAC	6/15/2014 6:54 PM	File folder	
CMCC	6/15/2014 6:54 PM	File folder	
common	6/15/2014 6:54 PM	File folder	
CT	6/15/2014 6:54 PM	File folder	
CU	6/15/2014 6:54 PM	File folder	
NA	6/15/2014 6:54 PM	File folder	

NA Date modified: 6/15/2014 6:54 PM Offline status: Online
File folder Offline availability: Not available

« mcfg ▸ mcfg_gen ▸ generic ▸ NA

Search NA

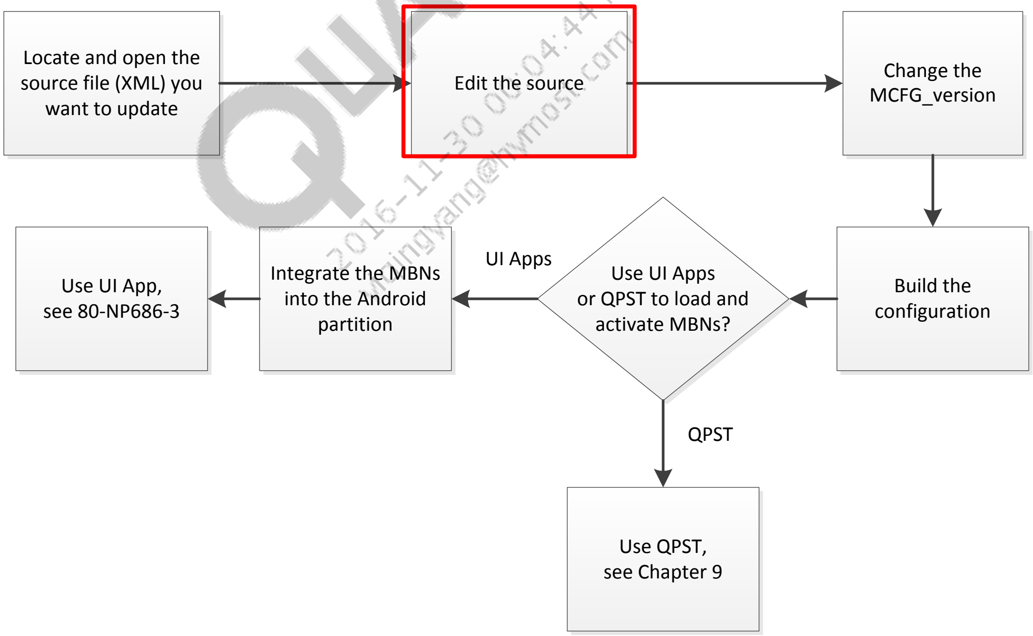
Name	Date modified	Type	Size
mcfg_sw_gen_ATT.xml	6/15/2014 6:52 PM	XML Document	21 KB
mcfg_sw_gen_ATT_VoLTE.xml	6/15/2014 6:52 PM	XML Document	34 KB
mcfg_sw_gen_Sprint.xml	6/15/2014 6:52 PM	XML Document	22 KB
mcfg_sw_gen_TMO.xml	6/15/2014 6:52 PM	XML Document	29 KB
mcfg_sw_gen_Verizon.xml	6/15/2014 6:52 PM	XML Document	41 KB
MCFG_SW_Items_List_Macro.xlsm	6/15/2014 6:52 PM	Microsoft Excel M...	793 KB

6 items Offline status: Online
Offline availability: Not available

To locate and open the source files, do the following:

1. Move to *$BUILD_ROOT\ modem_proc\mcfg\mcfg_gen\generic*.
2. Open the directory for your region or carrier.
3. Use the XML editor of your choice to open the XML file for the MBN that you want to modify.

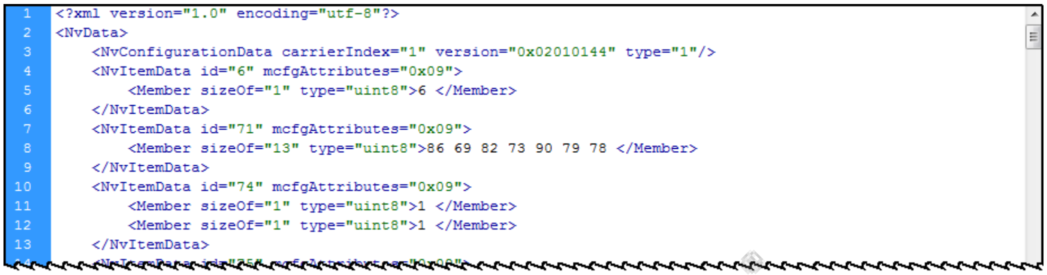
4.2 Edit the source


Locate and open the source file (XML) you want to update
Edit the source
Change the MCFG_version
Build the configuration
Use UI Apps or QPST to load and activate MBNs?
UI Apps
Integrate the MBNs into the Android partition
Use UI App, see 80-NP686-3
QPST
Use QPST, see Chapter 9

A complete review of the NV/EFS items in the XML is beyond the scope of this document. This section details how to modify the XML to make the same updates made with the macro-enabled workbook in the previous chapter. It provides examples of the following common use cases:

- Adding an NV item
- Removing an NV item
- Editing an EFS item
- Including a multisubscription NV item
- Adding an EFS file

The following figure shows an example of source XML:



```
<?xml version="1.0" encoding="utf-8"?>
<NvData>
    <NvConfigurationData carrierIndex="1" version="0x02010144" type="1"/>
    <NvItemData id="6" mcfgAttributes="0x09">
        <Member sizeOf="1" type="uint8">6 </Member>
    </NvItemData>
    <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">
```

4.2.1 Add an NV item

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

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

2. Insert the item contents somewhere below it:

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

4.2.2 Remove an NV item

To remove an NV/EFS item, remove the NvItemData element from the XML. In this example, item 74 is removed.

- 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>
```

4.2.3 Edit an EFS item

To edit an EFS item, locate the item you want to edit and make the desired changes. Note that the full pathname in these entries is the location on the target where the EFS file is to be placed.

- Before

```
<NvEfsItemData mcfgAttributes="0x09"
fullpathname="/nv/item_files/ims/qp_ims_sip_extended_0_config">
 <Member sizeOf="1" type="uint8">1 </Member>
 <Member sizeOf="1" type="uint16">5060 </Member>
 <Member sizeOf="1" type="uint32">600000 </Member>
 <Member sizeOf="1" type="uint32">600000 </Member>
 <Member sizeOf="1" type="uint32">3000 </Member>
 <Member sizeOf="1" type="uint32">16000 </Member>
 <Member sizeOf="1" type="uint32">17000 </Member>
 <Member sizeOf="1" type="uint32">30000 </Member>
 <Member sizeOf="1" type="uint32">30000 </Member>
 <Member sizeOf="1" type="uint8">1 </Member>
 <Member sizeOf="1" type="uint8">0 </Member>
 <Member sizeOf="1" type="uint8">0 </Member>
 <Member sizeOf="1" type="uint8">0 </Member>
 <Member sizeOf="1" type="uint8">0 </Member>
 <Member sizeOf="1" type="uint8">3 </Member>
 <Member sizeOf="1" type="uint8">0 </Member>
 <Member sizeOf="256" type="uint8"></Member>
 <Member sizeOf="256" type="uint8"></Member>
</NvEfsItemData>
```

- After

```
<NvEfsItemData mcfgAttributes="0x09"
fullpathname="/nv/item_files/ims/qp_ims_sip_extended_0_config">
 <Member sizeOf="1" type="uint8">2 </Member>
 <Member sizeOf="1" type="uint16">5060 </Member>
 <Member sizeOf="1" type="uint32">600000 </Member>
 <Member sizeOf="1" type="uint32">600000 </Member>
 <Member sizeOf="1" type="uint32">3000 </Member>
 <Member sizeOf="1" type="uint32">16000 </Member>
 <Member sizeOf="1" type="uint32">17000 </Member>
 <Member sizeOf="1" type="uint32">30000 </Member>
 <Member sizeOf="1" type="uint32">30000 </Member>
 <Member sizeOf="1" type="uint16">1500 </Member>
 <Member sizeOf="1" type="uint8">1 </Member>
 <Member sizeOf="1" type="uint8">0 </Member>
 <Member sizeOf="1" type="uint8">0 </Member>
 <Member sizeOf="1" type="uint8">0 </Member>
 <Member sizeOf="1" type="uint8">0 </Member>
 <Member sizeOf="1" type="uint8">3 </Member>
 <Member sizeOf="1" type="uint8">0 </Member>
 <Member sizeOf="256" type="uint8"></Member>
 <Member sizeOf="256" type="uint8"></Member>
 <Member sizeOf="1" type="uint8">1 </Member>
 <Member sizeOf="1" type="uint32"> </Member>
 <Member sizeOf="467" type="uint8"></Member>
</NvEfsItemData>
```

4.2.4 Include a multisubscription NV item

1. To include a multisubscription NV item, locate the following line:

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

2. Insert the following contents 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>
```

4.2.5 Making a multiplexed NV item

For carriers in which specific NV/EFS could be updated through an OTA process and which the updated NV/EFS values need to be preserved, the NV/EFS must be defined with the Multiplexed attribute.

An example for the need to use this is multiplexed attribute is for devices that are going to Sprint. They use OMADM to update some NV/EFS parameters.

The bit corresponding to multiplexed attribute is bit 2. This bit will need to set to 1. Please see example below:

Original NV item attribute value (attribute = 0x19):

```
<NvItemData id="3635" mcfgAttributes="0x19" mcfgVariant="1">
        <Member sizeOf="1" type="uint8">7 </Member>
        ::
        <Member sizeOf="1" type="uint32">12 </Member>
</NvItemData>
```

Updating the NV's attribute to make it a multiplexed item (attribute changed to 0x1B):

```
<NvItemData id="3635" mcfgAttributes="0x1B" mcfgVariant="1">
    <Member sizeOf="1" type="uint8">7 </Member>
    ::
    <Member sizeOf="1" type="uint32">12 </Member>
</NvItemData>
```

4.2.6 Add an EFS file

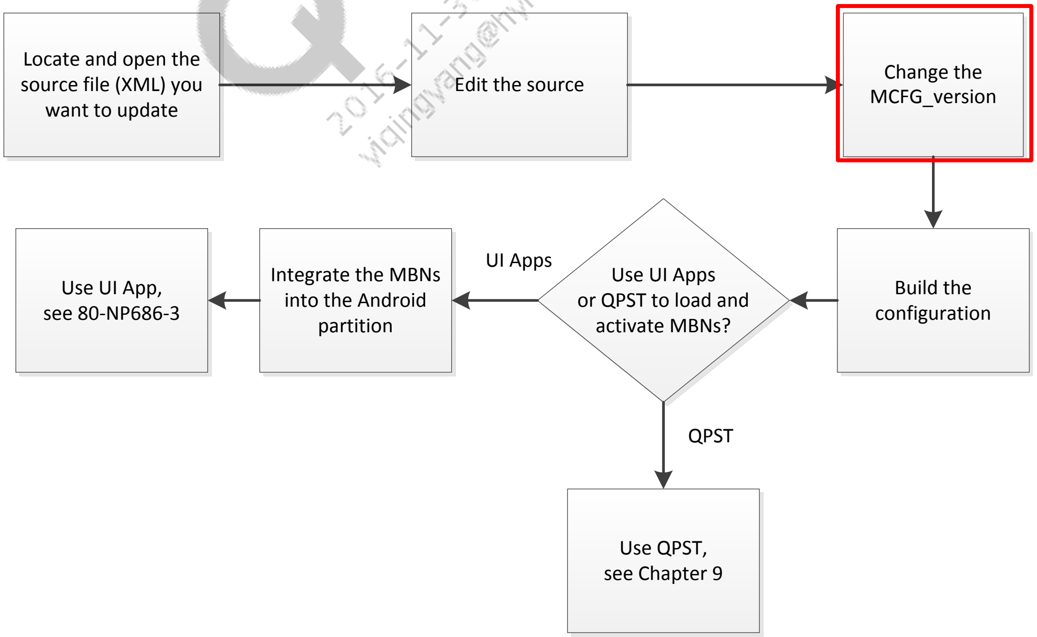
To include a new carrier_policy.xml file in the carrier XML, add the following line:

```
<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.

4.3 Change the MCFG_version


Locate and open the source file (XML) you want to update
Edit the source
Change the MCFG_version
Build the configuration
Use UI Apps or QPST to load and activate MBNs?
UI Apps
Integrate the MBNs into the Android partition
Use UI App, see 80-NP686-3
QPST
Use QPST, see Chapter 9

Before you build the configuration, you must change the MCFG_version of the configuration you updated. 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 is not loaded to the target. Avoid this scenario by updating the MCFG_version.

The version is listed in two places within the .xml file. Update the version in both places.

First location of version in XML

1. Find the following line:

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

2. Increment the version by 1.

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

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

Second location of version in XML

The second update is in decimal and is located in the Trailer Record near the bottom of the file. It is identified by the following tag:

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

1. Locate the version in the NVTrlRecord tag. The version is the sixth member down in the trailer record and looks similar to the following:

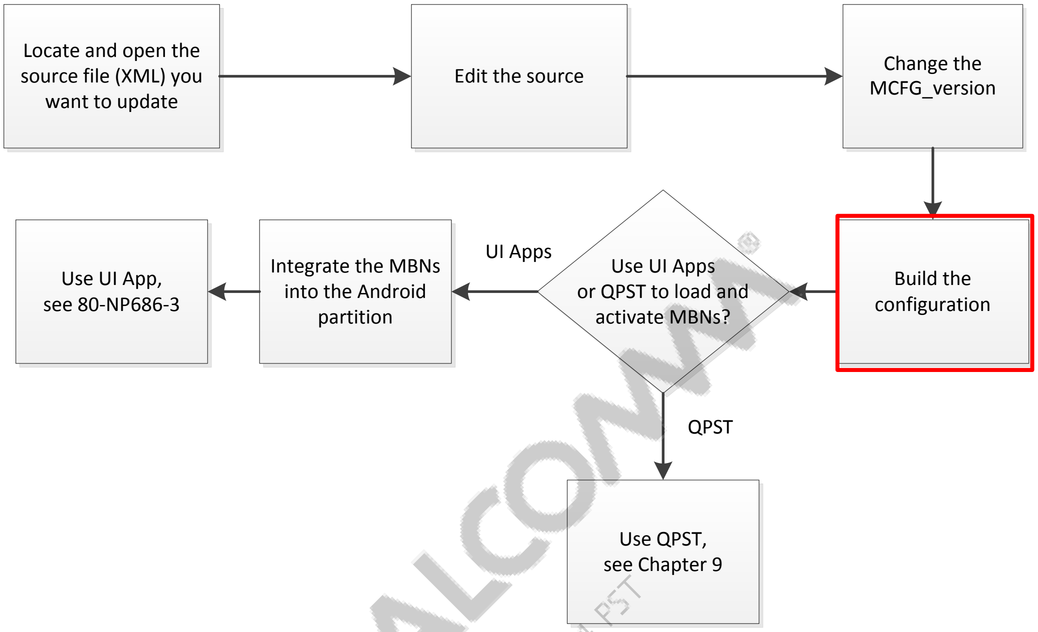
```
<NvTrlRecord mcfgAttributes="0x00">
 <Member sizeOf="1" type="uint8">0 </Member>
 <Member sizeOf="1" type="uint16">2 </Member>
 <Member sizeOf="1" type="uint16">256 </Member>
 <Member sizeOf="1" type="uint8">1 </Member>
 <Member sizeOf="1" type="uint16">4 </Member>
 <Member sizeOf="1" type="uint32">33620280 </Member>
…
</NvTrlRecord>
```

2. Increment the version by 1.

```
…
<Member sizeOf="1" type="uint32">33620281 </Member>
…
```

3. Save the file.

4.4 Build the configuration

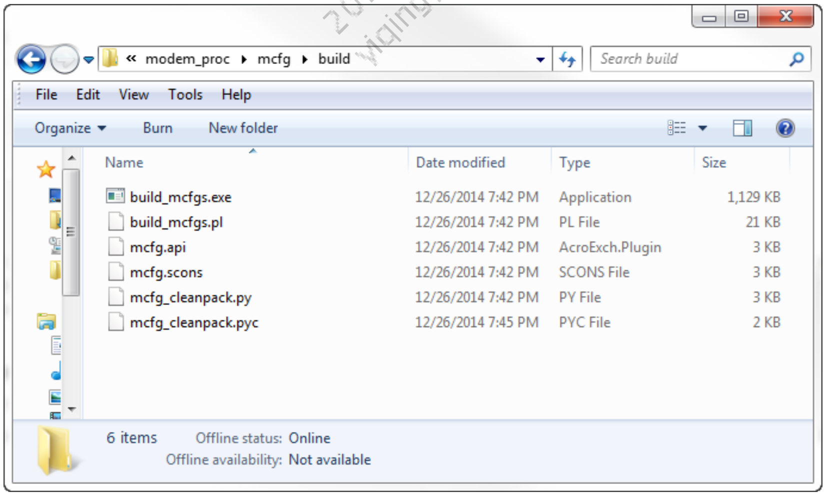

Locate and open the source file (XML) you want to update
Edit the source
Change the MCFG_version
Build the configuration
Use UI Apps or QPST to load and activate MBNs?
UI Apps
Integrate the MBNs into the Android partition
Use UI App, see 80-NP686-3
QPST
Use QPST, see Chapter 9

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.

The following figure shows the location of the *build.mcfgs.exe* file:



Name	Date modified	Type	Size
build_mcfgs.exe	12/26/2014 7:42 PM	Application	1,129 KB
build_mcfgs.pl	12/26/2014 7:42 PM	PL File	21 KB
mcfg.api	12/26/2014 7:42 PM	AcroExch.Plugin	3 KB
mcfg.scons	12/26/2014 7:42 PM	SCONS File	3 KB
mcfg_cleanpack.py	12/26/2014 7:42 PM	PY File	3 KB
mcfg_cleanpack.pyc	12/26/2014 7:45 PM	PYC File	2 KB

6 items Offline status: Online
Offline availability: Not available

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

4.4.1 Generate a specific, single-carrier configuration

The general form of the command used to build a single source and MBN is as follows:

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

Explanation of command arguments

- `<BUILD ID>` – The build variant ID of the image; listed in the summary spreadsheet of the macro-enabled Excel workbook. This is only needed for products prior to DPM 1.x.
- `<PLATFORM>:<CONFIG_NAME>` – The MBN platform and configuration name being generated
  - `PLATFORM` – Use `mcfg_sw` for carrier configurations and `mcfg_hw` for hardware configurations; these are the only two values currently accepted
  - `CONFIG_NAME` – For software configurations, use the carrier name (e.g., Verizon), for hardware, use the 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)
- [--source-dir=<CARRIER_DIR_PATH] – Option path to select which carrier settings to generate. Path spans from the generic folder down to a specific carrier (e.g. generic/NA/ATT). If none is supplied then the default configuration is generated (i.e. generic/common/Default).

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

4.4.2 Generate a default configuration using carrier settings

To specify a specific carrier as the default configuration, do the following:

- For PLs prior to DPM1.0 – Add the MCFG_SW_TYPE option to the build command.

  ```
  build.cmd 9x25.geni BUILD_ID=TAAAANAA BUILD_VER=0001_scons_change
  MCFG_SW_TYPE=Verizon
  ```

- For PLs later than DPM 1.0 but prior to DPM 2.0 – Add the MCFG_SW_TYPE, MCFG_IMAGE, and MCFG_SW PRODUCT options to the build command.

  ```
  build.cmd 8916.gen.test BUILD_ID=EAAAANVA BUILD_VER=0001_scons_change
  MCFG_IMAGE=generic MCFG_SW_PRODUCT=NA MCFG_SW_TYPE=Verizon
  ```

- For PLs later than DPM 2.0 – Add the MCFG_SW_TYPE and MCFG_SW_PRODUCT options to the build command where the MCFG_SW_PRODUCT is the full path between *modem_proc/mcfg/mcfg_gen* and the XML source file

  ```
  build.cmd 9x35.gen.test BUILD_ID=EAAAANVA BUILD_VER=0001_scons_change
  MCFG_SW_PRODUCT=generic/NA/Verizon MCFG_SW_TYPE=Verizon_hVoLTE
  ```

Each of the examples above builds an image with Verizon as the default carrier.

NOTE: While the form of the build command is predominantly static, it is subject to change between PLs.

If no carrier is specified in the image build command, XML data from the default tab of MCFG_SW_Items_List_Macro.xlsm is used as the default configuration in the modem image.

```
build.cmd 9x25.geni BUILD_ID=TAAAANAA BUILD_VER=0001_def_config.
```

For later PLs this spreadsheet has been moved to its own workbook under *<build_root>/modem_proc/mcfg/mcfg_gen/generic/common/Default/ MCFG_SW_Items_List_Macro.xlsm*

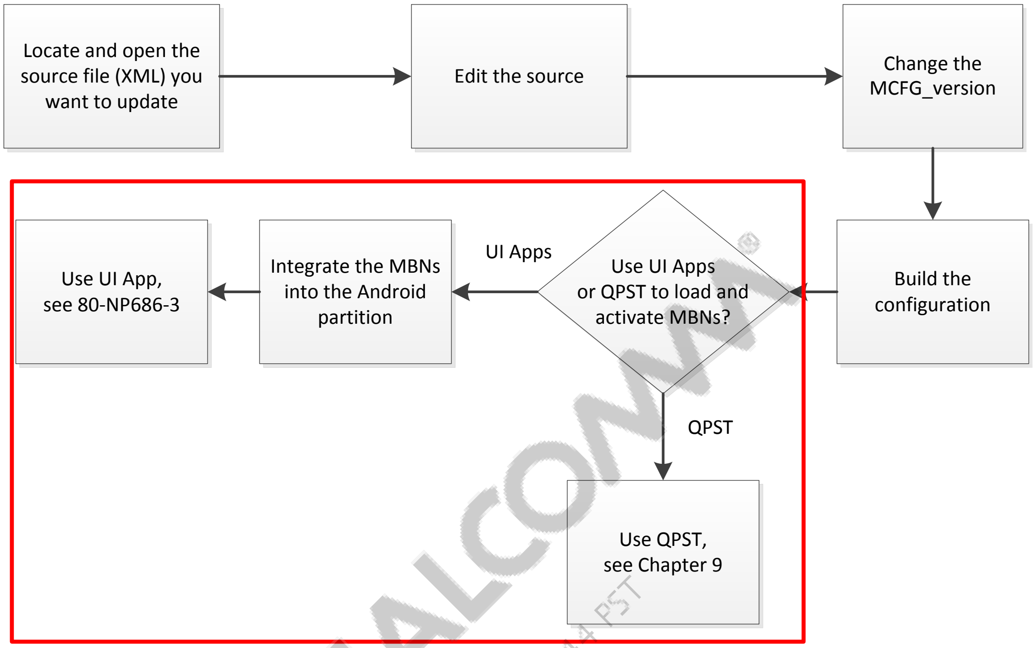
4.4.3 Procedure for multiple images with default configuration

This procedure involves using the same default configuration to build separate images for multiple carriers or for updating the same carrier (e.g., from Verizon to Verizon_2). The key to this procedure is to save each image in different locations so as not to overwrite the previous builds. To build multiple images with the default configuration, do the following:

1. Build a single MPSS image with the default carrier configuration as described in Section 4.4.2.
2. Save the firmware image to a location other than the original. The original location is *<build_root>\modem_proc\build\ms\bin\<build_id>\qdsp6sw.mbn.*
3. Repeat Steps 1-2 for each carrier ensuring that you save the file to a different location.

NOTE: If you are building images to update the same carrier (e.g., from Verizon to Verizon_2), update the MCFG_Version as described in Section 4.3.

4.5 Load and activate MBNs


Locate and open the source file (XML) you want to update
Edit the source
Change the MCFG_version
Build the configuration
Use UI Apps or QPST to load and activate MBNs?
UI Apps
Integrate the MBNs into the Android partition
Use UI App, see 80-NP686-3
QPST
Use QPST, see Chapter 9

The next step of the process is to load and activate the MBNs. The tools you use to perform these tasks depends on the type of release.

- Windows customers must use QPST.
- Android customers can use either QPST or the UI applications.

Where available, the UI applications are the recommended method to use for Android images during testing because they enable faster switching between MBNs. QPST is recommended for factory use, but can be used during testing as well.

To use the UI apps to load and activate the newly generated MBNs, you must first integrate the new MBNs into the Android partition.

- For information on integrating MBNs into the Android partition, see Chapter 8.
- For information on using QPST, see Chapter 9.
- For information on using the UI applications, see Application Note: Configuring a UE Using Binary Modem Configuration (80-NP686-1).

5 Changes from MSM8998 and MDM9x50

From MSM8998 and MDM9x50, MCFG has the following changes:

- Deprecation of Excel workbook
- XMLs are not generated from the Excel workbook
- Added new XML schema: Root/Carrier config XML and Group XML.
- Storing of MBNs in remote file storage (RFS) and auto discovery
- MBN compression

With the new XML schema, there is one root XML (also called as Carrier Config XML) that has references to each technology-specific XML (also called as Group XML). This is the only way to create MBNs in MSM8998 and MDM9x50.

- **Group XML** – Individual technology-specific settings. Group XML files are at: *"modem_proc\mcfg\mcfg_gen\groups"*
    - Group XMLs are linked together by referencing them within the Root XML files.
    - Groups can contain as many subgroups as needed. Settings within groups can be organized in any logical manner
- **Root XML or carrier configuration XML** – Main XML that contains links to group XMLs. These files are at *"\modem_proc\mcfg\mcfg_gen\generic\"*
    - Root XML files are used by the SCons framework to generate MBNs at build time.
    - Location of Root XML is same as legacy path (i.e., before MSM8998). Instead of the Excel workbook and legacy XMLs, you now see Root XMLs.

Example: Root XML has reference to IMS group XML (contains all IMS settings), Data group XML (contains all Data Services settings) etc.

5.1 XML structure

XMLs consist of original configuration XML data and the following changes:

- Name, description, comment tags for each NV and its sub elements
- Technology area categories labeled to each item
- NV numbers are assigned as per QXDM database
- Subscription mask (listed as an attribute for multi-SIM items)

The following is an example of the XML structure:

```
<NvData McfgXmlVersion="2.0">
<NvConfigurationData carrierIndex="8" version="0x08010807" type="1"/>
 <GroupFile name="mcfg_sw_gen_MMCP_IR92.xml" description="MCFG Group XML"
comment="" category="MCFG"  version="mmcp.ir92.0" mcfgAttributes="0x09"
buildPath="modem_proc/mcfg/mcfg_gen/groups/mmcp/mcfg_sw_gen_MMCP_IR92.xml"/
>
<NvItemData name="Banner" id="71" description="" comment=""
category="Display" subscription_mask="0x07" mcfgAttributes="0x19"
mcfgVariant="1"> <Member name="letters" description="" comment=""
sizeOf="13" type="string">ROW_Gen_3GPP</Member> </NvItemData>
```

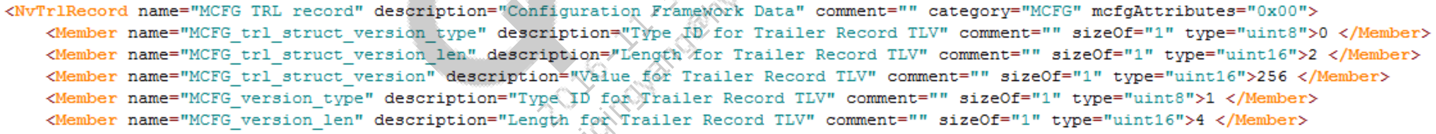
Table 5-1 Description of XML member attributes

Tag name	Example value	Description	Mandatory/Optional
Name	mcfg_sw_gen_IMS_IR92.xml	Filename - can be any combination of alphanumeric characters, '+', '_', or '-'.	Mandatory
Description	MCFG Group XML	Single-line description of file.	Optional. Can be blank
Comment	<blank>	Comments can be seen in XML by customer. Comments, like description should be on single line	Optional. Can be blank
Category	MCFG	Currently all group files are categorized MCFG because groups are MCFG specific. Category value may change at later point to reflect tech area owning NV group	Optional. Can be blank
Version	ims.ir92.0	Version string is applicable to group XML references. This is for informational purposes only. Teams can use this field to track a particular set of changes.	Optional. Can be blank
mcfgAttributes	0x09	You can refer to the attributes in section-8.2.4 Value is irrelevant for group items. All that matters is some hex value is present for a group XML.	Mandatory

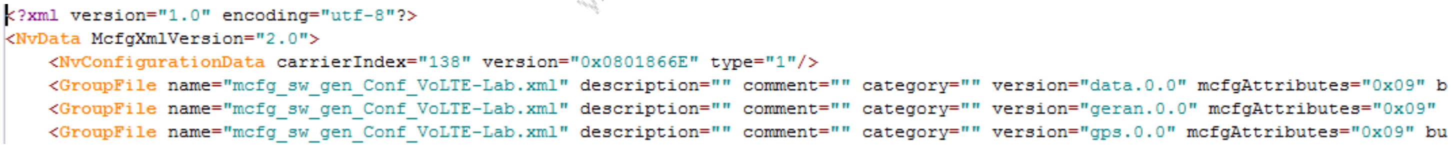
Tag name	Example value	Description	Mandatory/Optional
subscription_ mask	0x07	Only required if multiSIM bit of mcfgAttributes is set. Each bit position represents the corresponding subscription, i.e. bit position 0 represents subscription0. Setting a particular bit position to '1' enables an NV on the corresponding subscription. Value listed is decimal representation of bitmask. Refer to Table-1	Mandatory if multiSIM bit set is set in mcfgAttributes. Should not be present otherwise
buildPath	modem_proc/mcfg/ mcfg_gen/groups/ims/ mcfg_sw_gen_IMS_IR92.xml	Relative build location of group XML starting from modem_proc directory	Mandatory for EFS and group types. For all other types, buildPath is not listed.
targetPath / fullpathname	/nv/item_files/ims/ qp_ims_reg_config	Destination path of file in target EFS	Mandatory for PRL, EFS, and EFS item types. For all other types fullpathname is not needed

The following is an example of the Root XML:

Sample XML: *modem_proc\mcfg\mcfg_gen\generic\China\CMCC\mcfg_sw_gen_Conf_VoLTE-Lab.xml*



```
<NvTrlRecord name="MCFG TRL record" description="Configuration Framework Data" comment="" category="MCFG" mcfgAttributes="0x00">
    <Member name="MCFG_trl_struct_version_type" description="Type ID for Trailer Record TLV" comment="" sizeOf="1" type="uint8">0 </Member>
    <Member name="MCFG_trl_struct_version_len" description="Length for Trailer Record TLV" comment="" sizeOf="1" type="uint16">2 </Member>
    <Member name="MCFG_trl_struct_version" description="Value for Trailer Record TLV" comment="" sizeOf="1" type="uint16">256 </Member>
    <Member name="MCFG_version_type" description="Type ID for Trailer Record TLV" comment="" sizeOf="1" type="uint8">1 </Member>
    <Member name="MCFG_version_len" description="Length for Trailer Record TLV" comment="" sizeOf="1" type="uint16">4 </Member>
```



```
<?xml version="1.0" encoding="utf-8"?>
<NvData McfgXmlVersion="2.0">
    <NvConfigurationData carrierIndex="138" version="0x0801866E" type="1"/>
    <GroupFile name="mcfg_sw_gen_Conf_VoLTE-Lab.xml" description="" comment="" category="" version="data.0.0" mcfgAttributes="0x09" b
    <GroupFile name="mcfg_sw_gen_Conf_VoLTE-Lab.xml" description="" comment="" category="" version="geran.0.0" mcfgAttributes="0x09"
    <GroupFile name="mcfg_sw_gen_Conf_VoLTE-Lab.xml" description="" comment="" category="" version="gps.0.0" mcfgAttributes="0x09" bu
```

5.2 Creation of group XML file

Do the following to create or modify the Group XML:

To create the Group XML, do the following:

1. Refer to the sample XML files in MPSS at *\modem_proc\mcfg\mcfg_gen\groups*.
2. Copy a sample XML file and rename it. The name must end with the .xml file extension.
3. Save and close the XML file.

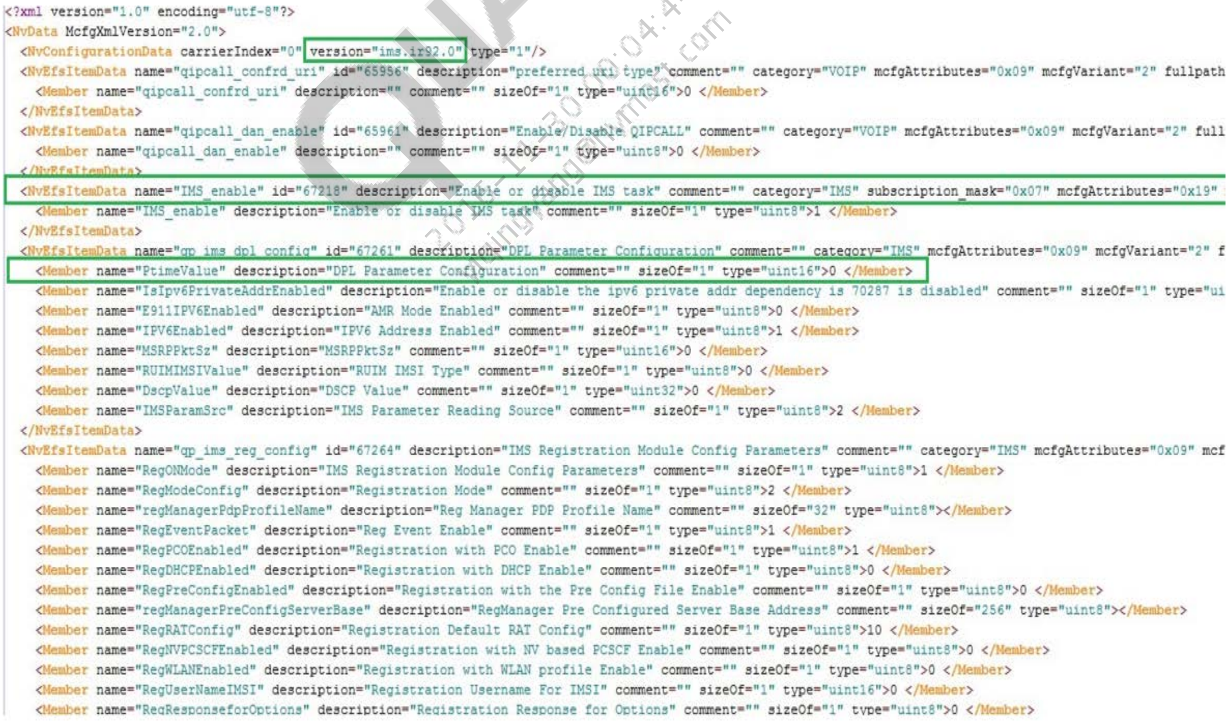
NOTE: To modify the values in the Group XML, open the existing XML file and save the changes.

To migrate the settings from the Carrier configuration/Root XML to the Group XML, do the following:

1. Open the Carrier configuration XML file (source) and the new Group XML file (destination).
2. Copy settings from the source file and paste in the destination file.

   Insert the copied contents between the NvConfigurationData tag at the top, and the NvData tag at the bottom.

   The following is an example of a Group XML file:



```
<?xml version="1.0" encoding="utf-8"?>
<NvData McfgXmlVersion="2.0">
  <NvConfigurationData carrierIndex="0" version="ims.ir92.0" type="1"/>
  <NvEfsItemData name="qipcall_confrd_uri" id="65956" description="preferred uri type" comment="" category="VOIP" mcfgAttributes="0x09" mcfgVariant="2" fullpath
    <Member name="qipcall_confrd_uri" description="" comment="" sizeOf="1" type="uint16">0 </Member>
  </NvEfsItemData>
  <NvEfsItemData name="qipcall_dan_enable" id="65961" description="Enable/Disable QIPCALL" comment="" category="VOIP" mcfgAttributes="0x09" mcfgVariant="2" full
    <Member name="qipcall_dan_enable" description="" comment="" sizeOf="1" type="uint8">0 </Member>
  </NvEfsItemData>
  <NvEfsItemData name="IMS_enable" id="67218" description="Enable or disable IMS task" comment="" category="IMS" subscription_mask="0x07" mcfgAttributes="0x19"
    <Member name="IMS_enable" description="Enable or disable IMS task" comment="" sizeOf="1" type="uint8">1 </Member>
  </NvEfsItemData>
  <NvEfsItemData name="qp_ims_dpl_config" id="67261" description="DPL Parameter Configuration" comment="" category="IMS" mcfgAttributes="0x09" mcfgVariant="2" f
    <Member name="PtimeValue" description="DPL Parameter Configuration" comment="" sizeOf="1" type="uint16">0 </Member>
    <Member name="IsIpv6PrivateAddrEnabled" description="Enable or disable the ipv6 private addr dependency is 70287 is disabled" comment="" sizeOf="1" type="ui
    <Member name="E911IPV6Enabled" description="AMR Mode Enabled" comment="" sizeOf="1" type="uint8">0 </Member>
    <Member name="IPV6Enabled" description="IPV6 Address Enabled" comment="" sizeOf="1" type="uint8">1 </Member>
    <Member name="MSRPPktSz" description="MSRPPktSz" comment="" sizeOf="1" type="uint16">0 </Member>
    <Member name="RUIMIMSIValue" description="RUIM IMSI Type" comment="" sizeOf="1" type="uint8">0 </Member>
    <Member name="DscpValue" description="DSCP Value" comment="" sizeOf="1" type="uint32">0 </Member>
    <Member name="IMSParamSrc" description="IMS Parameter Reading Source" comment="" sizeOf="1" type="uint8">2 </Member>
  </NvEfsItemData>
  <NvEfsItemData name="qp_ims_reg_config" id="67264" description="IMS Registration Module Config Parameters" comment="" category="IMS" mcfgAttributes="0x09" mcf
    <Member name="RegONMode" description="IMS Registration Module Config Parameters" comment="" sizeOf="1" type="uint8">1 </Member>
    <Member name="RegModeConfig" description="Registration Mode" comment="" sizeOf="1" type="uint8">2 </Member>
    <Member name="regManagerPdpProfileName" description="Reg Manager PDP Profile Name" comment="" sizeOf="32" type="uint8"></Member>
    <Member name="RegEventPacket" description="Reg Event Enable" comment="" sizeOf="1" type="uint8">1 </Member>
    <Member name="RegPCOEnabled" description="Registration with PCO Enable" comment="" sizeOf="1" type="uint8">1 </Member>
    <Member name="RegDHCPEnabled" description="Registration with DHCP Enable" comment="" sizeOf="1" type="uint8">0 </Member>
    <Member name="RegPreConfigEnabled" description="Registration with the Pre Config File Enable" comment="" sizeOf="1" type="uint8">0 </Member>
    <Member name="regManagerPreConfigServerBase" description="RegManager Pre Configured Server Base Address" comment="" sizeOf="256" type="uint8"></Member>
    <Member name="RegRATConfig" description="Registration Default RAT Config" comment="" sizeOf="1" type="uint8">10 </Member>
    <Member name="RegNVPCSCFEnabled" description="Registration with NV based PCSCF Enable" comment="" sizeOf="1" type="uint8">0 </Member>
    <Member name="RegWLANEnabled" description="Registration with WLAN profile Enable" comment="" sizeOf="1" type="uint8">0 </Member>
    <Member name="RegUserNameIMSI" description="Registration Username For IMSI" comment="" sizeOf="1" type="uint16">0 </Member>
    <Member name="RegResponseforOptions" description="Registration Response for Options" comment="" sizeOf="1" type="uint8">0 </Member>
```

NOTE: For more example XML files, see *MCFG/MBN Changes in MSM8998* (80-P2484-45).

5.3 Build commands

Run the build command with the MCFG_MBN_LOCATION flag pointing to source XMLs (include quotes). Not including MCFG_MBN_LOCATION builds all MBNs by default.

The following table lists and describes the build commands.

Description	Command
Builds both MPSS and MCFG, Default build command	`build.sh 8998.gen.test`
Build MCFG with all MBNs only, no MPSS	`build.sh 8998.gen.test image=mcfg_sw,mcfg_hw`

Build only the particular carrier MBNs	`build.sh 8998.gen.test image=mcfg_sw,mcfg_hw`
	`MCFG_MBN_LOCATION="mcfg\mcfg_gen\generic\<carrier>"`

The following is an example command for building CT MBNs on 8998 target:

```
build.sh 8998.gen.test image=mcfg_sw,mcfg_hw
MCFG_MBN_LOCATION="mcfg\mcfg_gen\generic\China\CT"
```

Build log messages

The build log is in modem_proc\build\ms. The following are sample build log messages printed while generating MBNs.

```
=== Generating  mcfg_sw/qdsp6/9655.gen.test/mcfg_sw.mbn
/local/mnt/workspace/at/crs/mcfg.mpss.8.0.as_PW80_CR970456_script_updates_i
mage_13419/modem_proc/mcfg/mcfg_gen/generic/China/CMCC/mcfg_sw_gen_TGL_Comb
_Attach-Lab.xml
mcfg_sw_gen_TGL_Comb_Attach-Lab.xml
perl
/local/mnt/workspace/at/crs/mcfg.mpss.8.0.as_PW80_CR970456_script_updates_i
mage_13419/modem_proc/mcfg/build/build_mcfgs.pl --
configs=mcfg_sw:TGL_Comb_Attach-Lab --force-rebuild --source-
dir=generic/China/CMCC/ -xml done
=== Generating  mcfg_sw/qdsp6/9655.gen.test/mcfg_sw.mbn
/local/mnt/workspace/at/crs/mcfg.mpss.8.0.as_PW80_CR970456_script_updates_i
mage_13419/modem_proc/mcfg/mcfg_gen/generic/China/CMCC/mcfg_sw_gen_Volte_Op
enMktCommercial.xml
mcfg_sw_gen_Volte_OpenMkt-Commercial.xml
perl
/local/mnt/workspace/at/crs/mcfg.mpss.8.0.as_PW80_CR970456_script_updates_i
mage_13419/modem_proc/mcfg/build/build_mcfgs.pl --
configs=mcfg_sw:Volte_OpenMkt-Commercial --force-rebuild --source-
dir=generic/China/CMCC/ -xml
18:Building MBN for mcfg_sw:Commercial...
  36874 18:Building MBN for mcfg_sw:Conf_VoLTE-Lab...
  36880 18:Building MBN for mcfg_sw:EPS_Only-Lab...
  36883 18:Building MBN for mcfg_sw:Commercial...
```

Table 5-2 Subscription mask details:

Multi-SIM bit (bit-4)	Subs_mask (hex)	Subs_mask (binary)	Enabled subscriptions	Note
0	-	-	Sub-0	Subscription mask is not used if multiSIM bit is not set. For these cases NV is only applied when activating MBN on subscription0
1	0x03	0011	Sub-0, Sub-1	NV is only applied during MBN activation on subscription0 or subscription1
1	0x06	0110	Sub-1, Sub-2	NV is only applied during MBN activation on subscription1 or subscription2
1	0x07	0111	Sub-0, Sub-1, Sub-2	NV is only applied during MBN activation on subscription0 or subscription1 or subscription2

6 Remote file storage (RFS) and auto discovery

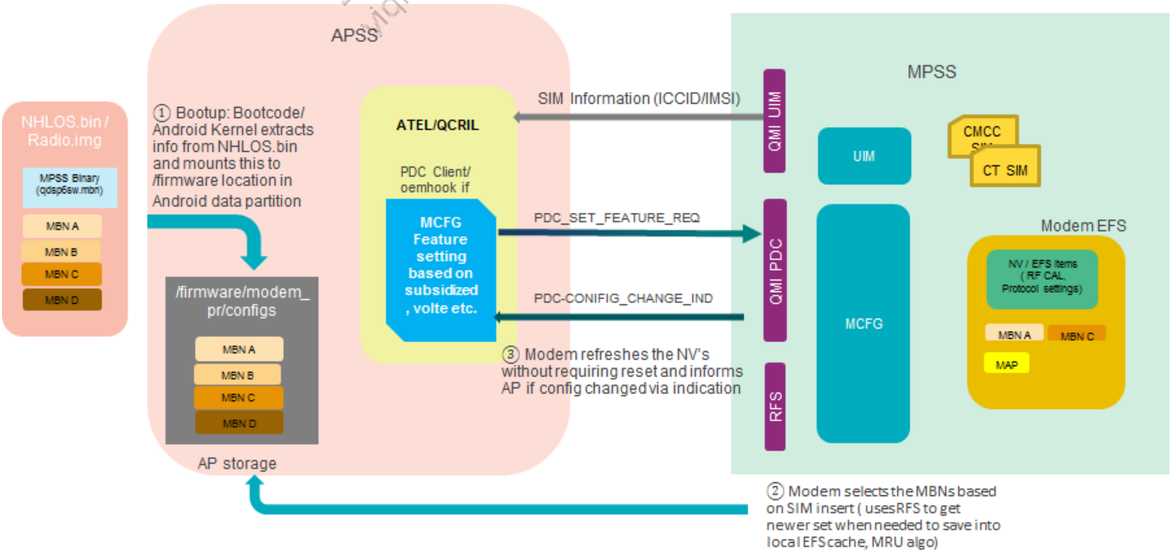
Storing MBNs in the modem takes significant EFS space. Because the apps processor has more space than the modem, it is better to store MBNs on the apps processor and then load them into the modem as needed. We have introduced the usage of remote file storage (RFS) procedure for this purpose.

With RFS, OEMs do not need to use the ‘Multi MBN’ feature to store multiple MBNs on the modem side.

With the existing implementation, the UE stores the MBNs in the non-HLOS bin and these MBNs will be mounted to AP fold */firmware/modem_proc/configs*.

The following image provides an overview of how RFS works. More details are provided in Section 6.1

1. Bootup: bootcode/Android Kernel extracts info from NHLOS.bin and mounts this to /firmware location in Android data partition.
2. From MSM8998, the RFS server is available in the apps image. The RFS client on the modem talks to the server and fetches MBNs from non-HLOS bin to the modem memory, then activates the MBNs.
3. Modem refreshes the NVs without requiring reset and informs AP if the config changed.


APSS
NHLOS.bin / Radio.img
MPSS Binary (qdsp6sw.mbn)
MBN A
MBN B
MBN C
MBN D
① Bootup: Bootcode/ Android Kernel extracts info from NHLOS.bin and mounts this to /firmware location in Android data partition
/firmware/modem_pr/configs
MBN A
MBN B
MBN C
MBN D
AP storage
ATEL/QCRIL
PDC Client/ oemhook if
MCFG Feature setting based on subsidized , volte etc.
SIM Information (ICCID/IMSI)
PDC_SET_FEATURE_REQ
PDC-CONIFIG_CHANGE_IND
③ Modem refreshes the NV's without requiring reset and informs AP if config changed via indication
MPSS
QMI UIM
QMI PDC
RFS
UIM
MCFG
CMCC SIM
CT SIM
Modem EFS
NV / EFS items (RF CAL, Protocol settings)
MBN A
MBN C
MAP
② Modem selects the MBNs based on SIM insert (uses RFS to get newer set when needed to save into local EFS cache, MRU algo)

6.1 How RFS works

1. On power-up, the apps processor mounts the MBNs included in the non-HLOS.bin to */firmware location*.
2. In auto discovery, the modem automatically discovers all the MBNs through RFS path */readonly/firmware*. The most recently selected/loaded MBN is stored in modem EFS depending on the EFS usage.

   The following steps describe how auto discover works:

   a. A text file called *mbn_sw.txt* is available in the MPSS image at *modem_proc\mcfg\configs\*. The file lists all available MBNs.

      NOTE: The filename is case-sensitive.

   b. If there are no changes, then all of the carrier MBNs listed in the file are compiled and available in the build.

   c. If the OEM wants to compile only a subset of carrier MBNs, they can create a test file called ‘oem_sw.txt’ and only list the specific carrier MBN names.

   d. If oem_sw.txt file is present, it is used to decide which carrier MBNs are generated by MPSS.

   e. The hash file of the MBNs is called Digest and is in the same location with the filename *mbn_sw.dig*. MCFG uses this Digest to compare if any changes were done in the MBNs.

   f. When the Digest is changed, only MBNs added as ‘Remote’ type are deleted.

   g. All the MBNs included in *oem_sw.txt* or *mbn_sw.txt* are automatically added to modem.
3. For the MBNs loaded through PDC interface, a copy is stored in RFS path /readwrite.
4. Trailer records of all the MBNs are stored in modem EFS for faster selection and list query.
5. Based on the SIM card inserted, MCFG picks the MBN based on the trailer record. If the MBN is not already stored in local, it deletes some least recent used MBNs from modem EFS to make room, and copy the needed MBN from remote to local. See Table 6-1 for load options.
6. This new FR has no impact on the QMI PDC interface and made transparent to PDC clients. The MCFG mechanism to load, select, and activate MBNs remains the same as legacy targets.

NOTE: There is no change in the MBN signing procedure with RFS mechanism.

Table 6-1 Different load options and how they work

Load Option	Loading process	Modem EFS status after loading	RFS status after loading
PDC Load Config Req (Storage TLV = **LOCAL**)	PDC client transfers entire MBN as binary through QMI.	If there is enough space, MBN gets saved in EFS.	MBN always gets saved in RFS readwrite folder.

Load Option	Loading process	Modem EFS status after loading	RFS status after loading
PDC Load Config Req (Storage TLV = **REMOTE**)	▪ PDC client passes MBN path only ▪ MCFG loads it directly from RFS location	If there is enough space, MBN gets copied from RFS into EFS.	MBN exists in RFS prior loading.
Auto Discovery	Through auto-discover with path specified in MBN list file.	If there is enough space, MBN gets copied from RFS into EFS.	In auto-discover scenario, it is under readonly/firmware/ folder.

7 MBN compression

To store more MBNs in the same available EFS space, MCFG now supports MBN compression. The MBNs are compressed before storing them in EFS.

Two advantages of MBN compression are as follows:

- It saves memory. Compressed MBNs take less than 20% of the space, thus enabling you to save more MBNs in the same available memory space.
- OEMS can use compression even with Multi MBN solution.

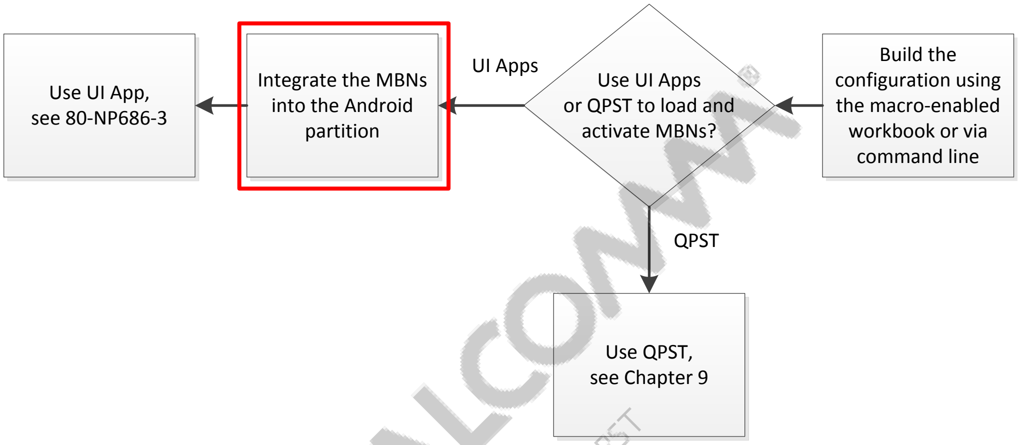
How compression works

To store more MBNs, UE uses ‘zlib’ utility to compress the MBN and then store it in the modem EFS location. The‘zlib’ utility comes as part of MPSS image.

Compression is enabled by default from MSM8998. It is transparent to MCFG/PDC client. All the load/select/activation/delete/list QMI commands remain the same as before.

If it is added through PDC, then the MBN is compressed when it is stored in modem EFS. It is also saved in the compressed format in the RFS readwrite partition.

8 Integrating MBNs into Android


Use UI App,
see 80-NP686-3
Integrate the MBNs into the Android partition
UI Apps
Use UI Apps or QPST to load and activate MBNs?
Build the configuration using the macro-enabled workbook or via command line
QPST
Use QPST,
see Chapter 9

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

1. Generate and build the MBNs as described in Chapter 3 or Chapter 4.
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 shell mount -o remount,rw /firmware
   ```
   d. Push the local MBN to a subdirectory within the Android partition at:
   ```
   /firmware/image/modem_pr/mcfg/configs/mcfg_sw/generic
   ```
   For example:
   ```
   adb push "mcfg_sw.mbn"
   /firmware/image/modem_pr/mcfg/configs/mcfg_sw/generic/China/CMCC/sglt
   e/ss/mcfg_sw.mbn
   ```
   To push MBN to a new subdirectory on the target first use the command
   ```
   adb shell mkdir -p
   /firmware/image/modem_pr/mcfg/configs/mcfg_sw/generic/new/subdirectory/path
   ```

Notes:

- APPS config directory is a mirror of *modem_proc\mcfg\configs* folder from modem image. (APPS config directory = */firmware/image/modem_pr/mcfg/configs/mcfg_sw/generic*)
- Subdirectory and file names need to be truncated to 8 characters (due to FAT16 limitation)
- Early implementations of MBN test app only display configurations from China region.

4. Reboot the phone.

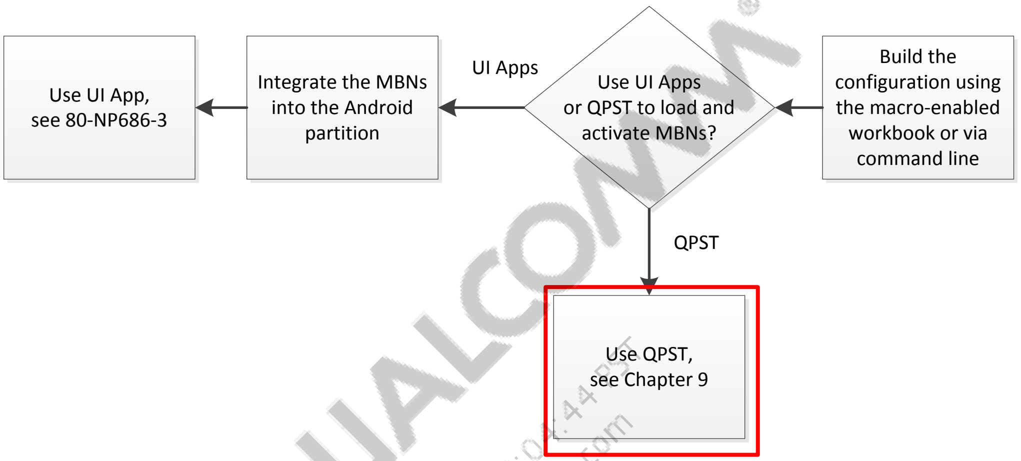
```
adb reboot
```

Browsing MBN files

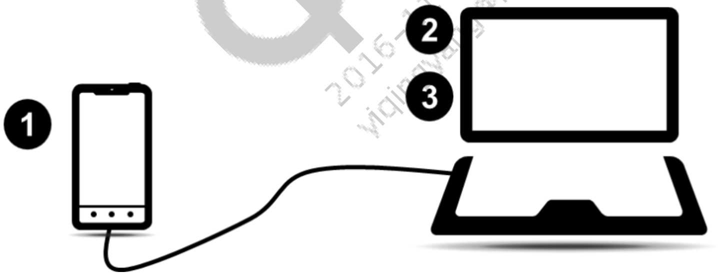
To browse the MBN files in the phone, use the following commands:

```
adb root; // get the root permission
cd /firmware/image/modem_pr/mcfg/configs;  // change directory to
/firmware/image/modem_pr/mcfg/configs
ls -R // list all files under the current directory
```

9 Loading, activating, and deactivating MBNs using QPST


Use UI App,
see 80-NP686-3
Integrate the MBNs into the Android partition
UI Apps
Use UI Apps or QPST to load and activate MBNs?
Build the configuration using the macro-enabled workbook or via command line
QPST
Use QPST,
see Chapter 9

Setup and prerequisites



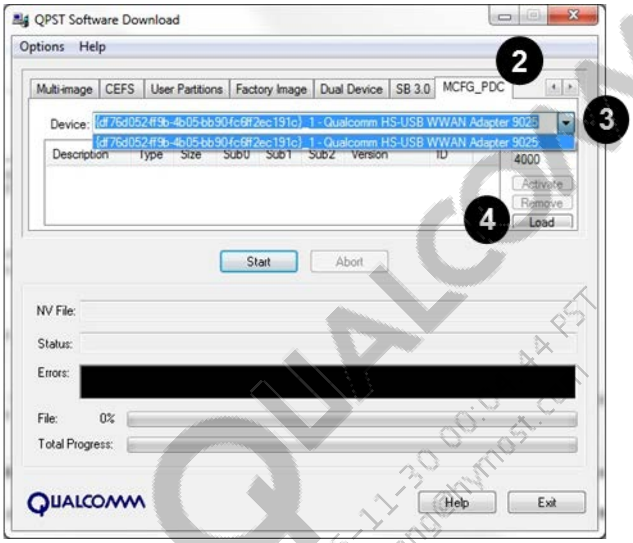
❶	UE is connected to a PC
❷	PC is running QPST v2.7.421 or later
❸	Know your <MODEM_BUILD> path and be able to access it from the PC

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

9.1 Load and activate MBNs

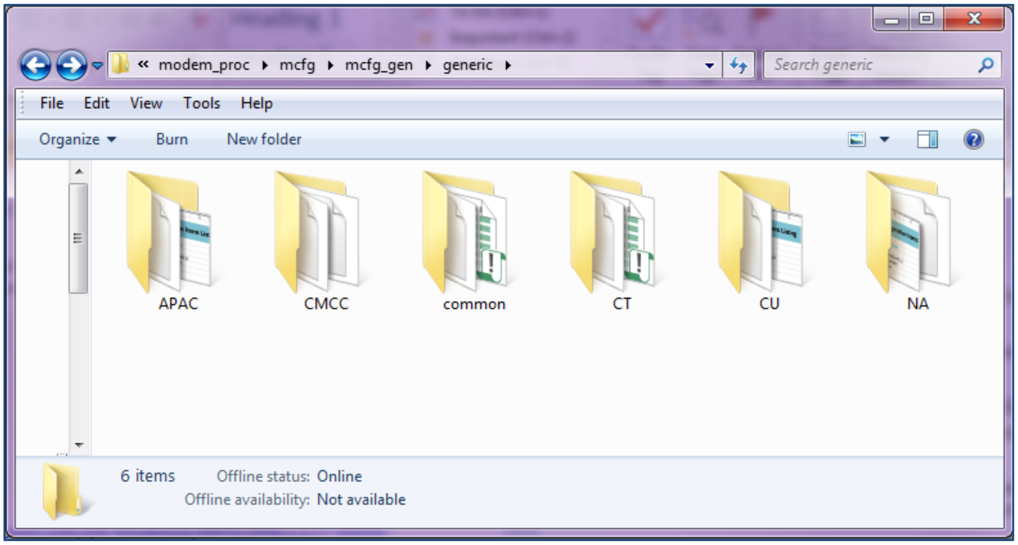
This section provides details on loading software MBNs. As a reminder, hardware MBNs must be loaded to the UE before loading the software MBNs.

1. Open the QPST Software Download module on the PC.
2. Click 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. Click **Load**.



Device: {df76d052-ff9b-4b05-bb90-fc6ff2ec191c}_1 - Qualcomm HS-USB WWAN Adapter 9025

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



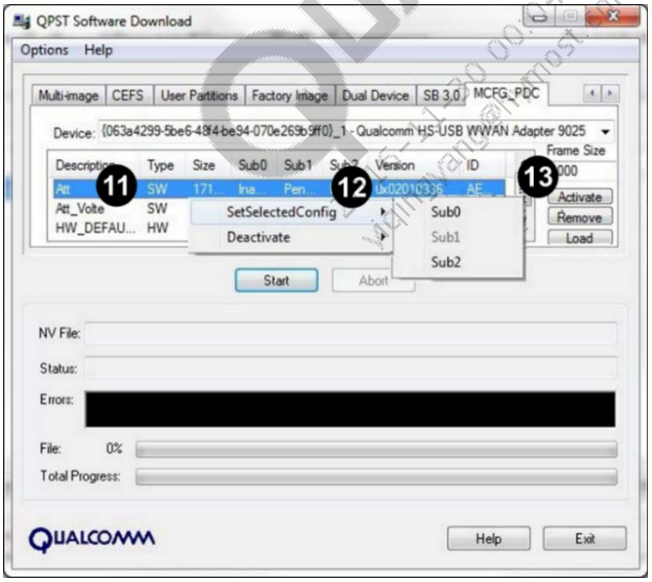
There are directories in this path that organize the MBNs by geographic region or by carrier.

Table 9-1 identifies the appropriate subdirectory for each carrier or region.

Table 9-1 MBN subdirectories by carrier and region

Carrier or region	Subdirectory containing 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 and paste it in the QPST pop-up window.
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						

Frame Size: 000

NV File:

Status:

Errors:

File: 0%

Total Progress:

11. Select and right-click the MBN that is appropriate for the UE that you are configuring.
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**. Repeat Steps 11 and 12, this time select **Sub1** in Step 12 for the second subscription.

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

9.2 Switch between MBNs

To switch between MBNs using QPST, do the following:

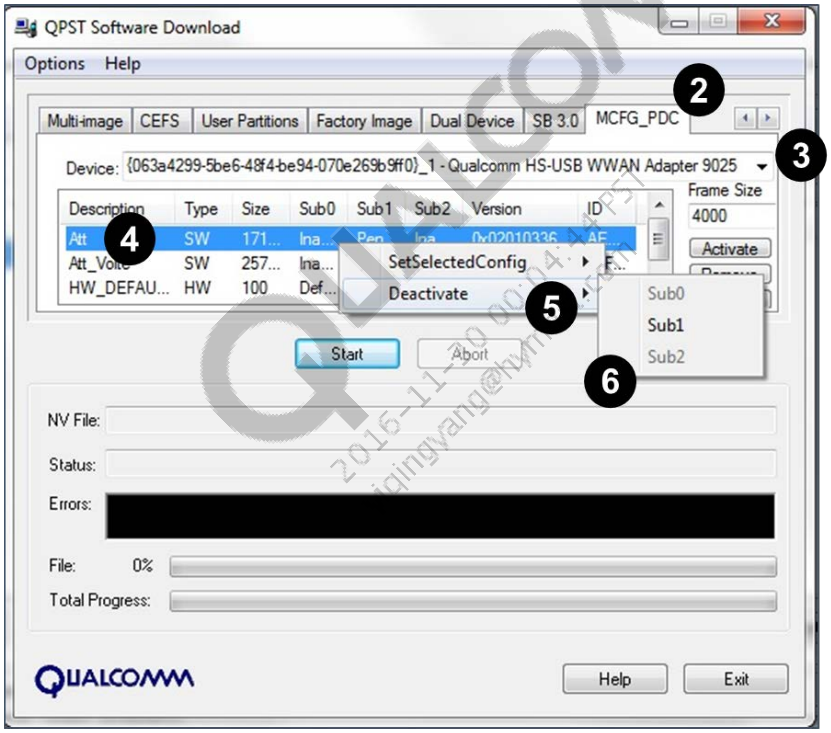
1. Deactivate the currently active MBN. See Section 9.2.1 for more information.
2. Load and activate the new MBN. See Section 9.1 for more information.

9.2.1 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, do the following:

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.
3. Click the drop-down arrow and select any RmNet port available from the UE.
4. Select and right-click the MBN that you want to deactivate.
5. Select **Deactivate**. A list of subscriptions appears.
6. 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**, repeat Steps 4 and 5, and select **Sub1**.

   The status of the MBN changes to Inactive.

9.3 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. Click the MCFG-PDC tab.
3. Select an inactive/pending configuration and click **Remove**. This removes the MBN from the Flash memory of the UE.

9.4 Working with hardware/platform configurations

Unlike software/carrier configurations, you must deactivate the current hardware configuration before activating a different hardware configuration. The following is a use case for switching between the single-SIM and the multi-SIM hardware configurations, while also using the single-SIM and multi-SIM carrier software configurations.

9.4.1 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. The target is ready for single-SIM testing.
2. To begin testing dual SIM, deactivate the single-SIM hardware configuration.
3. Load and activate the dual-SIM hardware and software configurations. The target is ready for dual-SIM testing.

To switch from dual-SIM to single-SIM testing, perform these steps in reverse but this time deactivate the dual-SIM hardware configuration in Step 2.

10 Autoselection

10.1 Autoselection methods and MCFG-related functions

UEs can simultaneously store more than one software MBN. It is important to be able to switch configurations automatically to take advantage of the multiple configurations. There are various autoselection methods used to achieve this functionality.

NOTE: Not all autoselection methods and related functions are available on all product lines

Use NV 71546 to control the various autoselection methods. Table 10-1 and Table 10-2 list the values for NV71546, identify which method of autoselect is invoked for each value, and list related functions like flex-mapping, MCFG refresh (SSR-less), and AP-based autoselection.

Table 10-1 MCFG NV 71546 settings

Bit	Value	Description
7	0x80	Disable MCFG Refresh without reset feature (FR 20298)
5	0x20	Enable MCFG Refresh MBN update only feature
2	0x04	Enable MCFG Auto-Selection Based on IMSI/MCC-MNC
1	0x02	Enable MCFG MBN Switching for Flex Mapping (bit 5 or 7 has to be enabled as well)
0	0x01	Enable MCFG “Auto-Selection Based on ICCID”

NOTE: All other bits are reserved.

Table 10-2 Description of autoselection methods

Autoselect method	Description	NV 71546 setting	
IIN-based autoselection	MBN selection done by matching IIN of ICCID as determined by the modem code.	0x01 for all SIM slots	See Section 7.2 for more information.
IMSI-based autoselection	MBN selection done by matching MCC-MNC of the IMSI in the UIM as determined by the modem code.	0x04 for all SIM slots	See Section 7.4 for more information.
IIN and IMSI-based autoselection	Enables both autoselection methods. See previous rows for description.	0x05 for all SIM slots	See Section 7.4 for more information.
AP-controlled autoselection	MBN selection is done by IIN of ICCID, but AP determines the match not the modem.	0x00 or 0x80 for all SIM slots	See Section 7.5 for more information.

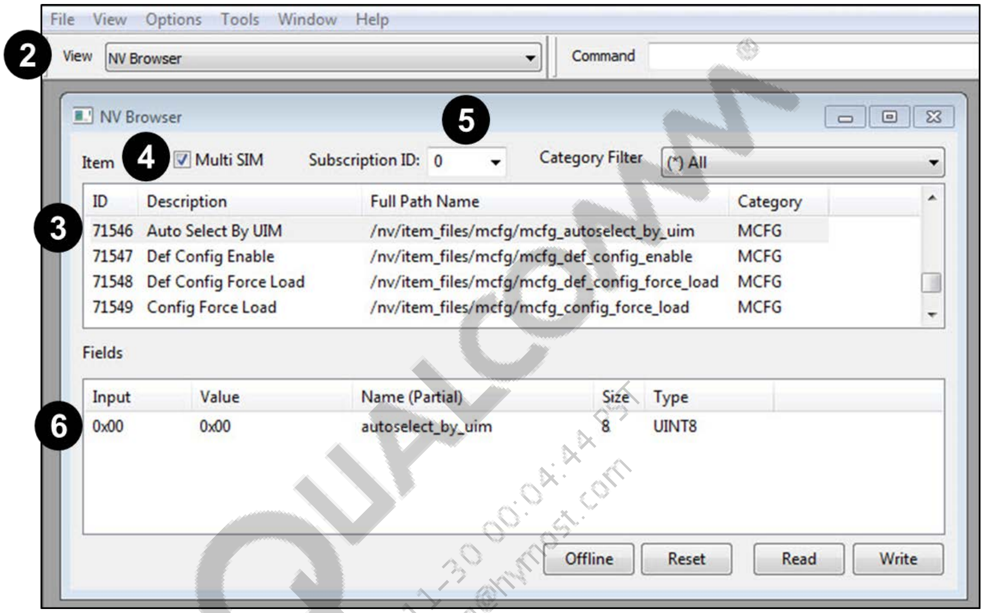
You can only enable either modem-based autoselection or AP-controlled autoselection. You cannot enable both.

10.1.1 How to change the value of NV 71546

To enable autoselection or autoselection with flex mapping, do the following:

1. Use the QPST Software Download tool as described in Chapter 9 to load all of the commercial MBNs intended for the device. For example, if the device is for the China open market, then load the MBNs for CMCC, CT, CU, and ROW.

   Do not activate any subscription.



View: NV Browser

Item ☑ Multi SIM Subscription ID: 0 Category Filter: (*) All

ID	Description	Full Path Name	Category
71546	Auto Select By UIM	/nv/item_files/mcfg/mcfg_autoselect_by_uim	MCFG
71547	Def Config Enable	/nv/item_files/mcfg/mcfg_def_config_enable	MCFG
71548	Def Config Force Load	/nv/item_files/mcfg/mcfg_def_config_force_load	MCFG
71549	Config Force Load	/nv/item_files/mcfg/mcfg_config_force_load	MCFG

Input	Value	Name (Partial)	Size	Type
0x00	0x00	autoselect_by_uim	8	UINT8

2. In QXDM Professional™ (QXDM Pro), select **NV Browser** from the View bar.
3. Locate and select 71546 Auto Select By UIM.
4. Select **Multi SIM** at the top of the NV Browser window.
5. Select **0** from the Subscription ID drop-down menu.
6. Double-click in the Input column and change the value of NV71546 based on Table 10-1.
7. Repeat Steps 3 through 6, but this time select **1** as the Subscription ID (you may also want to enable the autoselect mechanism for SIM 2).
8. Reset the UE.
9. Insert the SIM that you want into the device. Assuming that SIM does not match the currently loaded MBN, the modem resets. This is expected behavior.

10.2 IIN-based autoselection

When autoselection is enabled, the device automatically selects and activates a software MBN based on the SIM cards inserted in the device. This feature is disabled by default.

There are two autoselection options:

- Autoselection
- Autoselection with flex mapping

Autoselection works by reading the IIN from the ICCID on the SIM card. The IIN for a particular carrier is also in the metadata of each software MBN. When the SIM is inserted in the device, a modem configuration task loads and activates the software MBN that corresponds to the IIN on the SIM card. If there is no IIN match, then the MBN with “IIN_flag =1” is selected (e.g., ROW MBN). The ROW MBN includes the generic 3GPP settings and IR92 IMS configurations.

Autoselection with flex mapping works like regular autoselection but with an extra feature. If the device UI is used to change the network mode of each SIM slot, then it also automatically switches the active MBNs from one slot to another.

NOTE: Flex mapping is only available for DSDS devices.

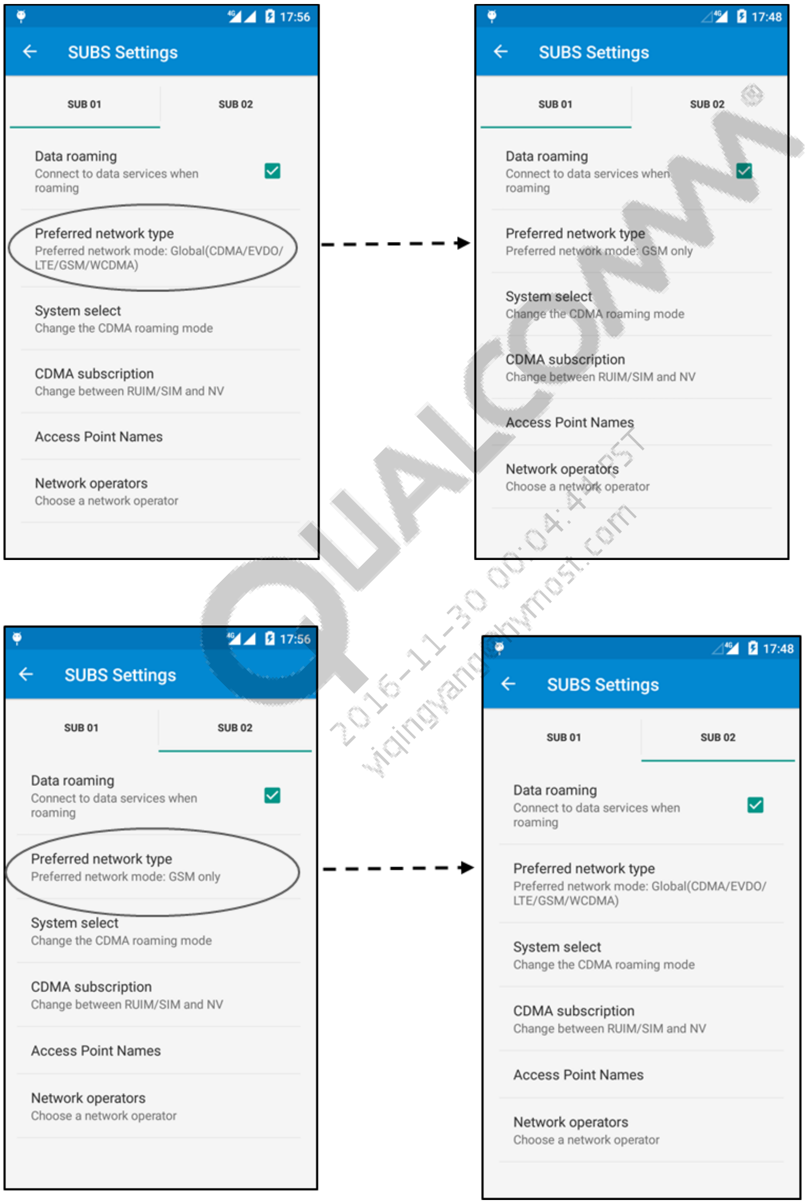
Regular autoselection is available in all modem PLs. Autoselection with flex mapping is available in the following modem PLs:

- DI4.0.1
- DPM2.0 and later
- JO
- BO2.0.1, 2.5, and 2.6

10.3 UI operation for flex mapping and MBN switching

When flex mapping is enabled, the active MBNs are automatically switched from one slot to the other if the device UI is used to change the network mode of the slots.

In this example, the preferred network type is changed from Global to GSM only for SUB 01 and then changed from GSM only to Global for SUB 02.



17:56

SUBS Settings

SUB 01 SUB 02

Data roaming
Connect to data services when roaming ☑

Preferred network type
Preferred network mode: Global(CDMA/EVDO/LTE/GSM/WCDMA)

System select
Change the CDMA roaming mode

CDMA subscription
Change between RUIM/SIM and NV

Access Point Names

Network operators
Choose a network operator

17:48

SUBS Settings

SUB 01 SUB 02

Data roaming
Connect to data services when roaming ☑

Preferred network type
Preferred network mode: GSM only

System select
Change the CDMA roaming mode

CDMA subscription
Change between RUIM/SIM and NV

Access Point Names

Network operators
Choose a network operator

17:56

SUBS Settings

SUB 01 SUB 02

Data roaming
Connect to data services when roaming ☑

Preferred network type
Preferred network mode: GSM only

System select
Change the CDMA roaming mode

CDMA subscription
Change between RUIM/SIM and NV

Access Point Names

Network operators
Choose a network operator

17:48

SUBS Settings

SUB 01 SUB 02

Data roaming
Connect to data services when roaming ☑

Preferred network type
Preferred network mode: Global(CDMA/EVDO/LTE/GSM/WCDMA)

System select
Change the CDMA roaming mode

CDMA subscription
Change between RUIM/SIM and NV

Access Point Names

Network operators
Choose a network operator

This change triggers the UI/RIL to send a change provisioning request to the modem with the updated mapping. When the modem restarts, slot 1 on the device is remapped to GSM only and slot 2 is remapped to Global. In addition, the MBNs that were active for each slot before the remapping are also switched.

10.4 IMSI-based autoselection

MBN selection done by matching MCC-MNC of the IMSI in the UIM as determined by the modem code. If IMSI-based autoselection is enabled, IIN-based autoselection is still enabled. However, IMSI-based selection takes precedence over IIN-based selection if the two methods have different MBN matches.

10.5 AP-controlled autoselection

The AP-controlled autoselection is a means for selecting the correct MBN using the QMI PDC interface provided by the modem. As a part of this process, the AP deactivates and deletes the current MBN and then loads and activates the MBN.

10.5.1 Enabling AP-controlled autoselection

In addition to the required NV 71546 settings, do the following adb commands:

```
adb root
adb shell setprop persist.radio.sw_mbn_update 1
adb shell sync
adb reboot
```

10.5.2 Determine the carrier configuration

The carrier-related storage items are built in separate software MBNs.

During boot up, QCRIL parses through software MBNs and selects the software MBN to load based on the ICCID of the SIM card.

If there is no specific matching MBN, then QCRIL selects a wildcard MBN.

Using the QMI PDC interface, QCRIL deactivates and deletes the current MBN and then loads and activates the selected MBN

10.6 Flex mapping and 7+5 devices

Except for a special use case described later in this section, there is no need to perform flex mapping on devices that support 7+5. This is because both subscriptions in 7+5 devices can support multiple RATs.

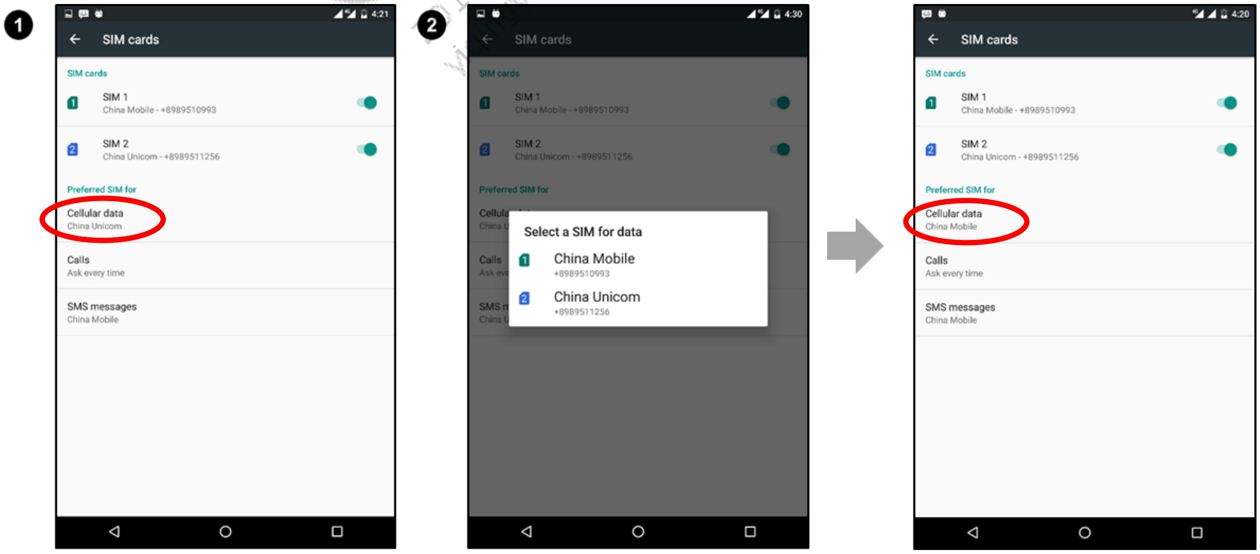
You are required to set DDS through the UI for data to be enabled, and to enable data-capable RATs like LTE, TD-SCDMA, and WCDMA on the appropriate SIM slot. To enable data and additional RATs for slot 2, first disable DDS on slot 1 and then enable it on slot 2.

Table 10-3 gives an example of the RAT capabilities of each slot in a 7+5 device. First, a CT SIM is inserted in SIM slot 1 and a CU SIM is inserted in SIM slot 2. Then DDS is switched.

Table 10-3 Switching DDS from SIM 1 to SIM 2 in 7+5 devices

	SIM slot 1	SIM slot 2	Comment
Subscription	sub0	sub1	Straight mapping of SIM slot 1 to sub 0 and SIM slot 2 to sub 1.
Carrier SIM card	CT (DDS)	CU	
Max capability	L/W/T/G/Do/1x	G	
Current capability	1xSRLTE (home)	G (home)	
	L/W/G (roaming)	G (roaming)	
Full RAT search capability	L/W/G/Do/1x	G	
Emergency call capability	L/W/T/G/Do/1x	G	
Now switch DDS from SIM 1 to SIM 2.			
Subscription	sub0	sub1	Notice that sub0 is still in slot 1 and sub 1 is still in slot 2, but multiple RATs now enabled for sub 1. No flex mapping was done here.
Carrier SIM card	CT	CU (DDS)	
Max capability	1x/G/DO	L/T/W/G	
Current capability	1x (home)	L/W/G (home)	
	G (roaming)	L/W/G (roaming)	
Full RAT search capability	1x/G	L/W/G	
Emergency call capability	1x/G	L/T/W/G	

10.6.1 How to switch DDS for 7+5 devices



SIM cards

SIM cards

SIM 1
China Mobile - +8989510993

SIM 2
China Unicom - +8989511256

Preferred SIM for

Cellular data
China Unicom

Calls
Ask every time

SMS messages
China Mobile

Select a SIM for data

China Mobile
+8989510993

China Unicom
+8989511256

SIM cards

SIM cards

SIM 1
China Mobile - +8989510993

SIM 2
China Unicom - +8989511256

Preferred SIM for

Cellular data
China Mobile

Calls
Ask every time

SMS messages
China Mobile

To switch DDS for 7+5 devices, go to the menu item displaying the current SIM card setup and do the following:

1. Tap **Cellular data**. In this example, data is on the China Unicom card.
2. Select the SIM to which you want to switch for data. In this example, China Mobile is selected.

The data is now switched to China Mobile

NOTE: DDS switch does not trigger MBN reloading. In addition, there is no flex mapping.

10.6.2 Special use case for flex mapping and 7+5

In 7+5 devices, CDMA is only available in the primary subscription (sub0). Because of this, if the CT card is used, then whichever slot the CT card is placed is where the primary service (sub0) must be. If CT card is in slot 1, no action is required because straight mapping keeps SIM slot 1 for sub0. But if the CT card is in slot 2, then use flex mapping to associate slot 2 with sub0. This flex mapping can be done manually or automatically depending on the AP software design. Set the DDS as discussed in the previous section.

NOTE: Flex mapping triggers MBN reloading on both subscriptions.

10.7 MCFG refresh

The MCFG refresh feature allows for NV/EFS to be activated without having to reset the modem.

11 Multi MBN

Available beginning with MSM8996, the Multi MBN feature bundles MBNs (up to 512K in total) into the MPSS image. When the MPSS image is loaded, all of the existing MBNs are replaced with the bundled MBNs. The 512K RAM used for MBN bundling is returned to the modem heap when MCFG is finished processing Multi MBN at rcinit task group 2 and before all of the other modem tasks start.

- Multi MBN is processed when upgrading MPSS image with a new Multi MBN version.
- If the Multi MBN versions are the same between MPSS images, Multi MBN is not re-processed. All of the existing MBNs loaded in the MPSS image will remain the same.

The following table provides three examples of how the hardware and software MBNs on a device are updated based on the Multi MBNs in the MPSS image. The **red bold** indicates the active MBNs in each scenario.

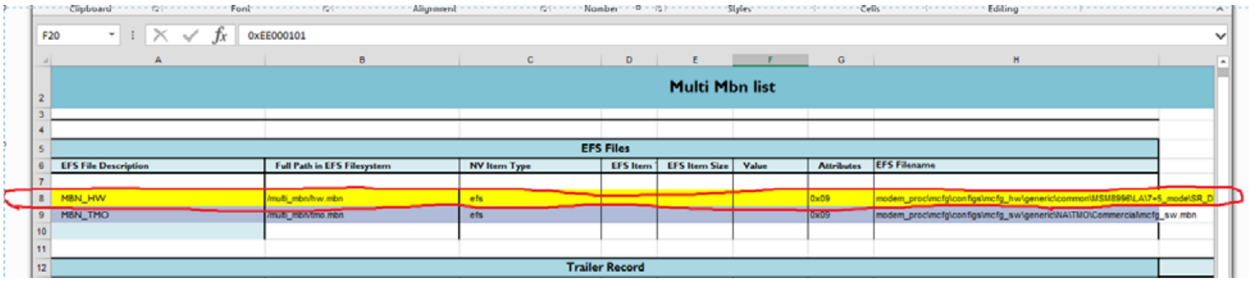
Condition	Example A	Example B	Example C
If these are the MBNs on the device,	**HW_V1**, **VZW_v1**, SPRINT_v1, ROW_v1	**HW_V1**, **VZW_v1**, SPRINT_v1, TMO_v1	VZW_v1, SPRINT_v1, TMO_v1 (nothing active)
and these are in the new MPSS image…	HW_V2, SPRINT_v2, ROW_v2	VZW_v2, SPRINT_v2, ROW_v2	HW_V1, VZW_v2, SPRINT_v2, ROW_v2
then these are the MBNs on device after upgrade.	**HW_V2**, **VZW_v1**, SPRINT_v2, ROW_v2	**HW_V1**, **VZW_v2**, SPRINT_v2, ROW_v2	**HW_V1**, VZW_v2, SPRINT_v2, ROW_v2

Create Multi MBN using Excel workbook

The following procedure is applicable for MSM8996. For later releases, create Mult MBNs by modifying the XML.

To create Multi MBNs using the Excel workbook, do the following:

1. Add or replace the MBN entries in the spreadsheet located at: *modem_proc\mcfg\mcfg_gen\generic\common\MultiMbn.*



F20 0xEE000101

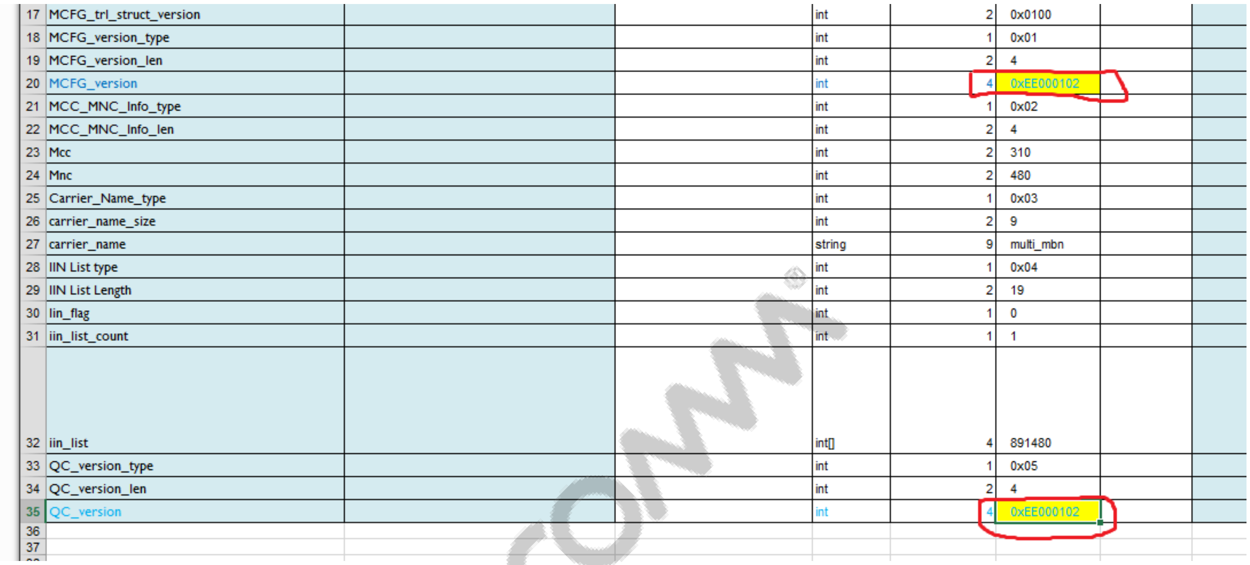
Multi Mbn list

EFS Files

EFS File Description	Full Path in EFS Filesystem	NV Item Type	EFS Item	EFS Item Size	Value	Attributes	EFS Filename
MBN_HW	/multi_mbn/hw.mbn	efs				0x09	modem_proc\mcfg\configs\mcfg_hw\generic\common\MSM8996\LA\7+5_mode\SR_D
MBN_TMO	/multi_mbn/tmo.mbn	efs				0x09	modem_proc\mcfg\configs\mcfg_sw\generic\NA\TMO\Commercial\mcfg_sw.mbn

Trailer Record

2. Update the Multi MBN version.



17	MCFG_trl_struct_version			int	2	0x0100		
18	MCFG_version_type			int	1	0x01		
19	MCFG_version_len			int	2	4		
20	MCFG_version			int	4	0xEE000102		
21	MCC_MNC_Info_type			int	1	0x02		
22	MCC_MNC_Info_len			int	2	4		
23	Mcc			int	2	310		
24	Mnc			int	2	480		
25	Carrier_Name_type			int	1	0x03		
26	carrier_name_size			int	2	9		
27	carrier_name			string	9	multi_mbn		
28	IIN List type			int	1	0x04		
29	IIN List Length			int	2	19		
30	Iin_flag			int	1	0		
31	iin_list_count			int	1	1		
32	iin_list			int[]	4	891480		
33	QC_version_type			int	1	0x05		
34	QC_version_len			int	2	4		
35	QC_version			int	4	0xEE000102		
36								
37								

3. Click “generate source files” only for new MultiMBN xml.
4. Rebuild MPSS image to include a new set of MBNs.

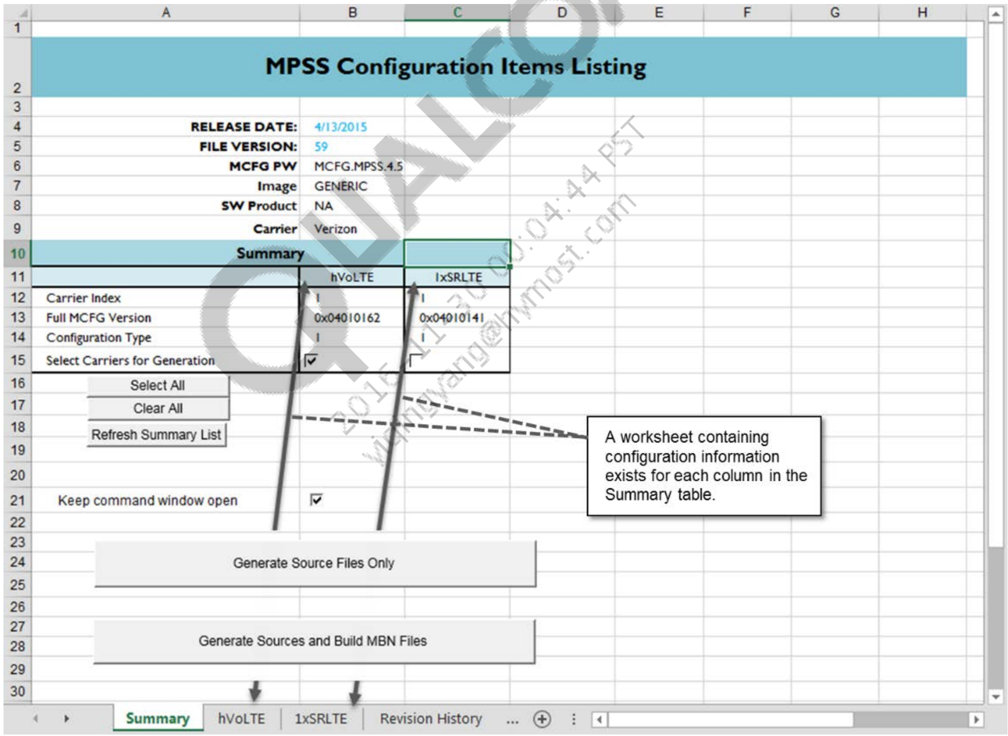
NOTE: With Auto discovery/RFS, OEM need not use the Multi MBN feature.

The Multi MBN feature is supported in both MSM and MDMs.

12 Workbook overview

There are separate workbooks for each MPSS image. The workbooks contain a worksheet named Summary along with worksheets for each configuration to be generated. Depending on the image version and software product, the worksheets are either for separate carriers, separate regions, or different configurations for the same carrier.

12.1 Summary worksheet



MPSS Configuration Items Listing

RELEASE DATE:	4/13/2015
FILE VERSION:	59
MCFG PW	MCFG.MPSS.4.5
Image	GENERIC
SW Product	NA
Carrier	Verizon

Summary

	hVoLTE	1xSRLTE
Carrier Index	1	1
Full MCFG Version	0x04010162	0x04010141
Configuration Type	1	1
Select Carriers for Generation	☑	☐

Keep command window open ☑

A worksheet containing configuration information exists for each column in the Summary table.

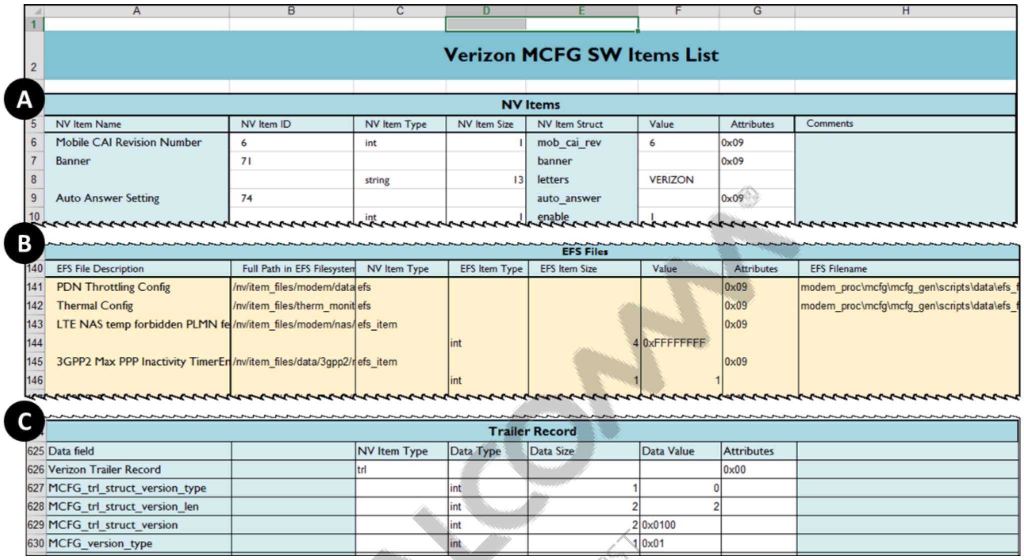
Figure 12-1 Example Summary worksheet

The Summary worksheet contains a Summary table with columns for each of the configurations in the workbook. A corresponding worksheet containing configuration information exists for each of the columns in the table. The column headers must match the name of the corresponding worksheet.

NOTE: Do not change the name of the Summary tab and do not delete the word Summary from the title of the Summary table. The title of the table is actually in column A and formatted to be centered across the table. This location is the starting point for the script and is used to determine which other worksheets to process.

12.2 SW Items List worksheets

The worksheets are arranged into three sections.



Verizon MCFG SW Items List

A

NV Items

	NV Item Name	NV Item ID	NV Item Type	NV Item Size	NV Item Struct	Value	Attributes	Comments
5	NV Item Name	NV Item ID	NV Item Type	NV Item Size	NV Item Struct	Value	Attributes	Comments
6	Mobile CAI Revision Number	6	int	1	mob_cai_rev	6	0x09	
7	Banner	71			banner		0x09	
8			string	13	letters	VERIZON		
9	Auto Answer Setting	74			auto_answer		0x09	
10			int	1	enable	1		

B

EFS Files

140	EFS File Description	Full Path in EFS Filesystem	NV Item Type	EFS Item Type	EFS Item Size	Value	Attributes	EFS Filename
141	PDN Throttling Config	/nv/item_files/modem/data	efs				0x09	modem_proc\mcfg\mcfg_gen\scripts\data\efs_f
142	Thermal Config	/nv/item_files/therm_monit	efs				0x09	modem_proc\mcfg\mcfg_gen\scripts\data\efs_f
143	LTE NAS temp forbidden PLMN fe	/nv/item_files/modem/nas/	efs_item				0x09	
144				int	4	0xFFFFFFFF		
145	3GPP2 Max PPP Inactivity TimerEn	/nv/item_files/data/3gpp2/r	efs_item				0x09	
146				int	1	1		

C

Trailer Record

625	Data field		NV Item Type	Data Type	Data Size	Data Value	Attributes	
626	Verizon Trailer Record		trl				0x00	
627	MCFG_trl_struct_version_type			int	1	0		
628	MCFG_trl_struct_version_len			int	2	2		
629	MCFG_trl_struct_version			int	2	0x0100		
630	MCFG_version_type			int	1	0x01		

A NV Items

B EFS Files

C Trailer Record

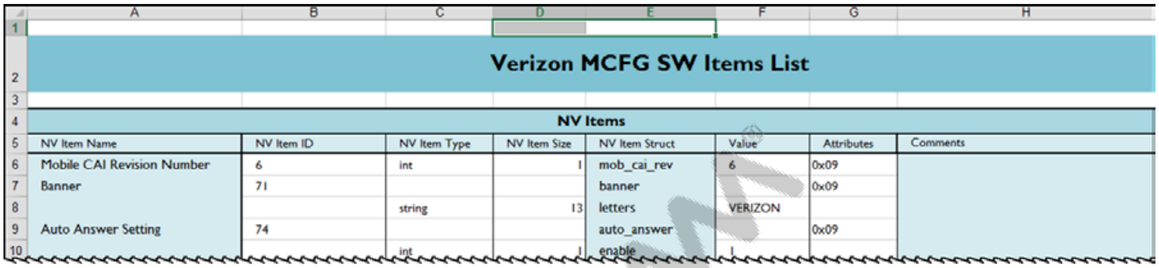
Each section of the worksheet contains various MCFG software items that can be one of several types as indicated by the NV Item Type column. All of the items in the NV Items section are numbered NV items, as indicated by the NV Item ID column, and are mapped to the /nvm/num directory of the EFS file system. Table 12-1 provides a description of the NV item types that appear in the EFS Files section of the worksheet.

Table 12-1 Description of NV Item types in SW Items List

NV item type	Description
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	Similar to 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 copied from a file on the local machine
prl	Indicates a PRL file
trl	Indicates the Trailer Record of the configuration

12.2.1 NV Items details

All of the items in the NV Items section are numbered NV items and are mapped to the /nvm/num directory of the EFS file system. This section provides details on the layout of the NV Items portion of the worksheet.



Verizon MCFG SW Items List

NV Items

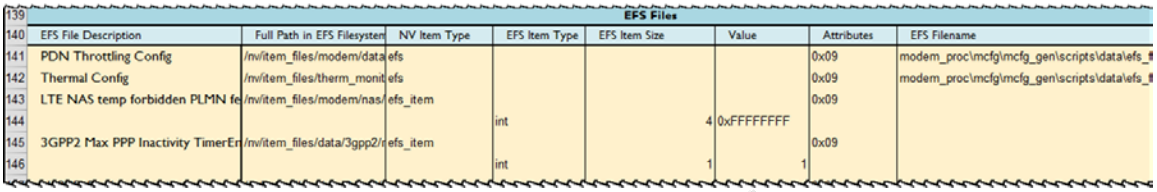
NV Item Name	NV Item ID	NV Item Type	NV Item Size	NV Item Struct	Value	Attributes	Comments
Mobile CAI Revision Number	6	int	1	mob_cai_rev	6	0x09	
Banner	71			banner		0x09	
		string	13	letters	VERIZON		
Auto Answer Setting	74			auto_answer		0x09	
		int	1	enable	1		

Table 12-2 Description of NV Items section of worksheet

Column	Description
NV Item Name	Not used by the generation script; used as an internal note or comment to identify the NV item
NV Item ID	Some integer value
NV Item Type	One of the following: ▪ int – An integer value ▪ int [ ] – An integer array of unspecified length; its size depends how many elements are listed in the value column ▪ int [n] – An integer value of length n ▪ string – A character string
NV Item Size	One of the following: ▪ If type is int – Size in bytes of this integer value ▪ If type is int [ ] or int [n] – Size in bytes of each element in the array ▪ If type is string – Blank size means that the length of the string should be the number of characters in the value + 1 for the NULL terminator ▪ An integer means that the string should be a fixed size; it is padded with zeros if the value does not have this many characters
Value	One of the following: ▪ If type is an int array – Value should be a list of integers or hexadecimal values delimited by commas ▪ If type is an int value – Value should be an integer or hexadecimal value ▪ If type is a string – Value can be any string
Attributes	See Section 12.2.4 for details

12.2.2 EFS Files details

This section provides details on the layout of the EFS Files portion of the worksheet.



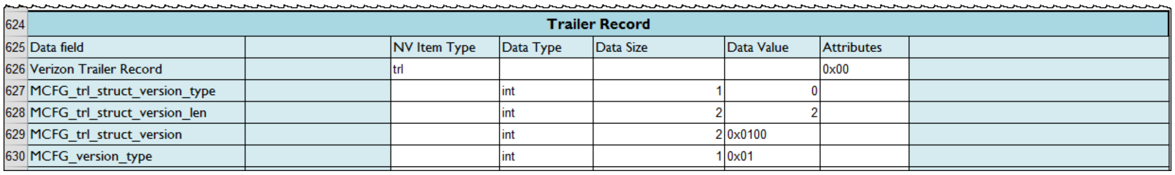
139	EFS Files							
140	EFS File Description	Full Path in EFS Filesysten	NV Item Type	EFS Item Type	EFS Item Size	Value	Attributes	EFS Filename
141	PDN Throttling Config	/nv/item_files/modem/data	efs				0x09	modem_proc\mcfg\mcfg_gen\scripts\data\efs_f
142	Thermal Config	/nv/item_files/therm_monit	efs				0x09	modem_proc\mcfg\mcfg_gen\scripts\data\efs_f
143	LTE NAS temp forbidden PLMN fe	/nv/item_files/modem/nas/	efs_item				0x09	
144				int	4	0xFFFFFFFF		
145	3GPP2 Max PPP Inactivity TimerEr	/nv/item_files/data/3gpp2/r	efs_item				0x09	
146				int	1	1		

Table 12-3 Description of EFS Files section of worksheet

Column	Description
EFS File Description	Not used by the generation script; used as an internal note or comment to identify the EFS file
Full Path in EFS Filesystem	One of the following based on EFS Item Type: ▪ EFS Item type = efs – Path to which the data should be copied ▪ EFS Item type = efs_dir – Source efs files are copied to this path ▪ EFS Item type = efs_item – Path to which the data should be copied ▪ EFS Item type = prl – 257
NV Item Type	One of the following: ▪ efs ▪ efs_dir ▪ efs_item ▪ prl ▪ trl
EFS Item Type	One of the following: ▪ int – Integer value ▪ int [ ] – Integer array of unspecified length; its size depends how many elements are listed in the value column ▪ int [n] – Integer value of length n ▪ string – Character string
EFS Item Size	One of the following: ▪ If EFS item type is int – Size in bytes of this integer value ▪ If EFS item type is int [ ] or int [n] – Size in bytes of each element in the array ▪ If EFS item type is string – Blank size means that the length of the string should be the number of characters in the value + 1 for the NULL terminator ▪ An integer means that the string should be a fixed size; it is padded with zeros if the value does not have this many characters
Value	One of the following: ▪ If type is an int array – Value should be a list of integers or hexadecimal values delimited by commas ▪ If type is an int value – Value should be an integer or hexadecimal value ▪ If type is a string – Value can be any string
Attributes	See Section 12.2.4 for details
EFS Filename	Local path to file relative to build root

12.2.3 Trailer Record details

This section provides details on the layout of the Trailer Record portion of the worksheet.



624	Trailer Record							
625	Data field		NV Item Type	Data Type	Data Size	Data Value	Attributes	
626	Verizon Trailer Record		trl				0x00	
627	MCFG_trl_struct_version_type			int	1	0		
628	MCFG_trl_struct_version_len			int	2	2		
629	MCFG_trl_struct_version			int	2	0x0100		
630	MCFG_version_type			int	1	0x01		

Table 12-4 Description of Trailer Record section of worksheet

Column	Description
Data Field	Not used by the generation script; used as an internal note or comment to identify the EFS file
NV Item Type	trl
Data Type	One of the following: ▪ int – Integer value ▪ int [ ] – Integer array of unspecified length; its size depends on how many elements are listed in the value column ▪ int [n] – Integer value of length n ▪ string – Character string
Data Size	One of the following: ▪ If data type is int – Size in bytes of this integer value ▪ If data type is int [ ] or int [n] – Size in bytes of each element in the array ▪ If data type is string – Blank size means that the length of the string should be the number of characters in the value + 1 for the NULL terminator ▪ An integer means that the string should be a fixed size; it is padded with zeros if the value does not have this many characters
Data Value	One of the following: ▪ If type is an int array – Value should be a list of integers or hexadecimal values delimited by commas ▪ If type is an int value – Value should be an integer or hexadecimal value ▪ If type is a string – Value can be any string
Attributes	0x00

12.2.4 Attributes column

The Attributes column (column G) of the worksheet is designed to contain hexadecimal values representing one bye of data. The bits are set as described in Table 12-5.

Table 12-5 Description of attribute fields and values

Attribute field	Value	Description
#define U_ITEM_ATTRIB_CFG	0x01	/* C */ – Configuration bit. If this is set, the values for the MCFG item are taken from the spreadsheet.
#define U_ITEM_ATTRIB_MUXD	0x02	/* M */ – 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. **Note**: Currently not used in QTI MBNs.
#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 before activation. Not many items fall into this category.
#define U_ITEM_ATTRIB_REST_FACT	0x08	/* R */ – Always set to 1.
#define U_ITEM_ATTRIB_MULTISIM	0x10	/* S */ – For multi-SIM-related NVs.
#define U_ITEM_ATTRIB_INDEXED	0x20	/* I */ – 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 if there are slow access flash devices like SPI-NOR where write times are significant.

12.2.4.1 Formatting

Table 12-6 describes the formatting of the worksheets. Any color in the worksheets that is not in the legend is an artifact and irrelevant the user.

Table 12-6 Formatting legend for worksheets

Background color	Description
No background	NV columns containing necessary data
Blue background	NV/EFS columns containing notes/comments
Yellow background	EFS listing (as opposed to NV)
Purple background	MultiSIM listing
Gray background	Placeholder (listing present in spreadsheet only; data is not a part of MBN)
Blue Font (any background)	Recently changed item

13 Factory provisioning and OTA update

See *Updating Modem Configurations in Factory and Over the Air* (80-NV514-1) for details on how to update modem configurations in factory environments and OTA.

14 Troubleshooting

14.1 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 appears:

```
** 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.

15 FAQs

QUESTION 1 The modem crashes when I activate the configuration with the PDC tool with the crash signature `- MODEM - mcfg_utils.c:148 MCFG:Modem Initiated Reset`. Is this expected?

ANSWER This is a known crash (intentional crash) initiated by the modem. It is done to allow the modem subsystem to reboot following an activate configuration.

If the QPST->MCFG_PDC tool is used to activate the modem configuration, reboot only the modem for the new settings to take effect. With the current SSR architecture in older modems, the only way for modem to initiate a reboot is to crash. If SSR is enabled (end-user scenario), this manifests itself as a modem subsystem reboot. Because it is not enabled by default in apps processor builds, the MTP enters Download mode instead.

There is no need to collect a crash dump in this scenario. The tester can power-cycle the phone after the configuration is listed as pending in the PDC tool.

QUESTION 2 What are IINs and why are they important?

ANSWER Issuer Identification Numbers (IINs) identify the carrier to which a SIM card belongs. MCFG requires at least one IIN in a configuration's Trailer Record for autoselection.

IINs are the first six to seven digits from the ICCID of a carrier UIM.

QUESTION 3 Why am I seeing "Error 438: Object reference not set" when I open an Excel workbook?

ANSWER One of the reasons for this is related to a recent Microsoft Office update. Follow these steps to resolve it:

1. Close all Microsoft Office applications.
2. Do a search in Windows Explorer for *.exd files and delete them. Ensure that your search includes hidden and system files and folders. Also note that the search is for *.*ex**d*** files and not *.*ex**e***.
3. Obtain the following:
   - C:\users\username\AppData\Local\Temp\Excel8.0\MSForms.exd
   - C:\users\username\AppData\Local\Temp\VBE\MSForms.exd
4. Reboot the computer.
5. Restart the Microsoft Office applications and retest the controls.

A XML Schema of Generated Source Files

The XML generated by MCFG_SW_Items_List_Macro.xlsm uses a schema with tags that closely parallel the listing structure within the spreadsheet itself. This section contains a description of each XML element type and their associated attributes.

mcfgVariant

```
<NvItemData id="4631" mcfgAttributes="0x09" mcfgVariant="1">
   <Member sizeOf="1" type="uint8">0 </Member>
</NvItemData>
```

This tag is used to reflect the various item types possible in the spreadsheet. Table A-1 lists valid mcfgVariant values for each item type.

Table A-1 Valid mcfgVariant values

Item type (spreadsheet)	mcfgVariant value
nv_item	1
efs	2
efs_item	2
efs_dir	2
slot_nv_item	3
slot_efs	4
slot_efs_item	4
slot_efs_dir	4
data_profile	5
config	6
active_config	7

NvItemData

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

- Used for normal NV items
- id – Specifies the numerical item ID.
- mcfgAttributes – 1-byte value used to specify attribute flags
- Member list – List of the various structs in the item; each member has a sizeOf attribute and a type attribute

- sizeOf – Length of the list of numbers inside the member tag
- type – Size of each number in the list; type can be uint8, uint16, uint32, uint64, int8, int16, int32, or int64
- int types – Used when the data in the member tag is signed; uint is used otherwise

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>
```

- Used for EFS items whose data is listed explicitly in the XML file
- fullpathname – Location on the target where the EFS file is to be placed
- Member list – For information on the member list, see NvItemData

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/>
```

- Used for EFS items whose data is contained in a file on the build system; this data is read and added to the final configuration when it is built
- targetPath – Location on the target where the EFS file is to be placed
- buildPath – Location on the build system relative to the root where the file resides

NvEfsDir

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

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

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

This specifies the location of the prl file on the build system to be copied to the target.

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>
```

Contains data about the configuration’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.

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

B References

B.1 Related documents

Title	Number
Qualcomm Technologies, Inc.	
Presentation: Secured MSM Code Signing Service	80-V9807-1
Presentation: Code Signing Management System Overview	80-V3999-1
Application Note: Configuring a UE Using Binary Modem Configuration	80-NP686-1
Updating Modem Configurations in Factory and Over the Air	80-NV514-1
MCFG/MBN Changes in MSM8998	80-P2484-45

B.2 Acronyms and terms

Acronym or term	Definition
CSFB	circuit switched fallback
DSDS	dual sim dual standby
EFS	embedded file system
IIN	issuer identification number
MBN	modem configuration binary
MCFG	modem configuration
NV	nonvolatile
OTA	over the air
PL	product line
QPST	Qualcomm® Product Support Tool
ROW	rest of world
RFS	Remote File Storage