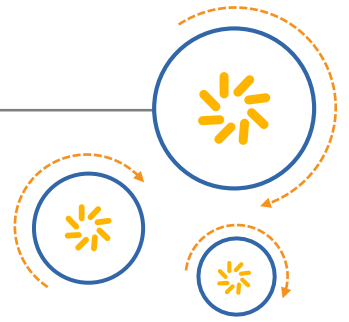




Qualcomm Technologies, Inc.



Modem Software Configuration Overview

80-N5576-96 F

February 20, 2015

Confidential and Proprietary – Qualcomm Technologies, Inc.

© 2012-2015 Qualcomm Technologies, Inc. and/or its affiliated companies. All rights reserved.

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

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



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

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

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

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

Revision history

Revision	Date	Description
A	Nov 2012	Initial release
B	Dec 2012	Updated
C	Jul 2013	Numerous changes were made to this document revision; it should be read in its entirety
D	Feb 2014	Numerous changes were made to this document revision; it should be read in its entirety
E	Nov 2014	Restructured and rewritten; this document should be read in its entirety
F	Feb 2015	Numerous changes were made to this document; it should be read in its entirety

Contents

1 Introduction.....	7
1.1 Purpose.....	7
1.2 Conventions	7
1.3 Technical assistance.....	7
2 MCFG Framework Overview	8
2.1 Introduction to MCFG MBNs.....	8
2.2 Introduction to the MCFG framework.....	9
2.3 OEM-generated MBNs	10
2.3.1 Prerequisites for OEM-generated MBNs.....	10
2.3.2 Security	11
3 Generating MBNs Using the Macro-Enabled Workbook	12
3.1 Locate and open the MCFG software items workbook	12
3.2 Add and rename worksheets	15
3.3 Edit the worksheet	18
3.3.1 Add an NV item.....	19
3.3.2 Remove an NV item	19
3.3.3 Edit an EFS item.....	19
3.3.4 Include a multisubscription NV	20
3.3.5 Add an EFS file	20
3.3.6 Include an updated EFS file.....	20
3.4 Change the MCFG_version	21
3.5 Build the configuration	22
3.5.1 Build signed MBN files.....	23
3.5.2 About the MBN generation backend	23
3.5.3 Schema of XMLs generated by the workbook	24
3.6 Load and activate MBNs	24
4 Generating MBNs by Modifying the XML Source	25
4.1 Locate and open the source.....	25
4.2 Edit the source	26
4.2.1 Add an NV item.....	27
4.2.2 Remove an NV item	27
4.2.3 Edit an EFS item.....	28
4.2.4 Include a multisubscription NV item.....	29
4.2.5 Add an EFS file	29
4.3 Change the MCFG_version	30
4.4 Build the configuration	31
4.4.1 Generate a specific, single-carrier configuration.....	32

4.4.2 Generate a default configuration using carrier settings	33
4.4.3 Procedure for multiple images with default configuration	33
4.5 Load and activate MBNs	34
5 Integrating MBNs into Android.....	35
6 Loading, Activating, and Deactivating MBNs Using QPST	37
6.1 Load and activate MBNs	38
6.2 Switch between MBNs	40
6.2.1 Deactivate an MBN	40
6.3 Delete an MBN from the UE	41
6.4 Working with hardware/platform configurations	41
6.4.1 Switch between single-SIM and dual-SIM configurations while using both hardware and software MBNs	41
7 Autoselection Mechanism	42
8 Workbook Overview	43
8.1 Summary worksheet	43
8.2 SW items list worksheets	44
8.2.1 NV Items details	45
8.2.2 EFS Files details	46
8.2.3 Trailer Record details	47
8.2.4 Attributes column	48
8.2.5 Formatting	49
9 Factory Provisioning and OTA Update	50
10 Troubleshooting	51
10.1 ELF size build error	51
11 FAQs	52
A XML Schema of Generated Source Files	53
B References.....	56
B.1 Related documents	56
B.2 Acronyms and terms	56

Figures

Figure 2-1 MCFG framework feature concept	9
Figure 2-2 Two ways to generate MBNs	10
Figure 3-1 Location of the MCFG software items workbook.....	13
Figure 8-1 Example Summary worksheet.....	43

Tables

Table 2-1 Description of MBN types.....	8
Table 6-1 MBN subdirectories by carrier and region	39
Table 8-1 Description of NV Item types in SW Items List.....	44
Table 8-2 Description of NV Items section of worksheet.....	45
Table 8-3 Description of EFS Files section of worksheet.....	46
Table 8-4 Description of Trailer Record section of worksheet.....	47
Table 8-5 Description of attribute fields and values	48
Table 8-6 Formatting legend for worksheets	49

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://support.cdmatech.com/>.

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

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 3 through 6.

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 (<i>mcfg_hw.mbn</i>)	<ul style="list-style-type: none">▪ 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 (<i>mcfg_sw.mbn</i>)	<ul style="list-style-type: none">▪ 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).

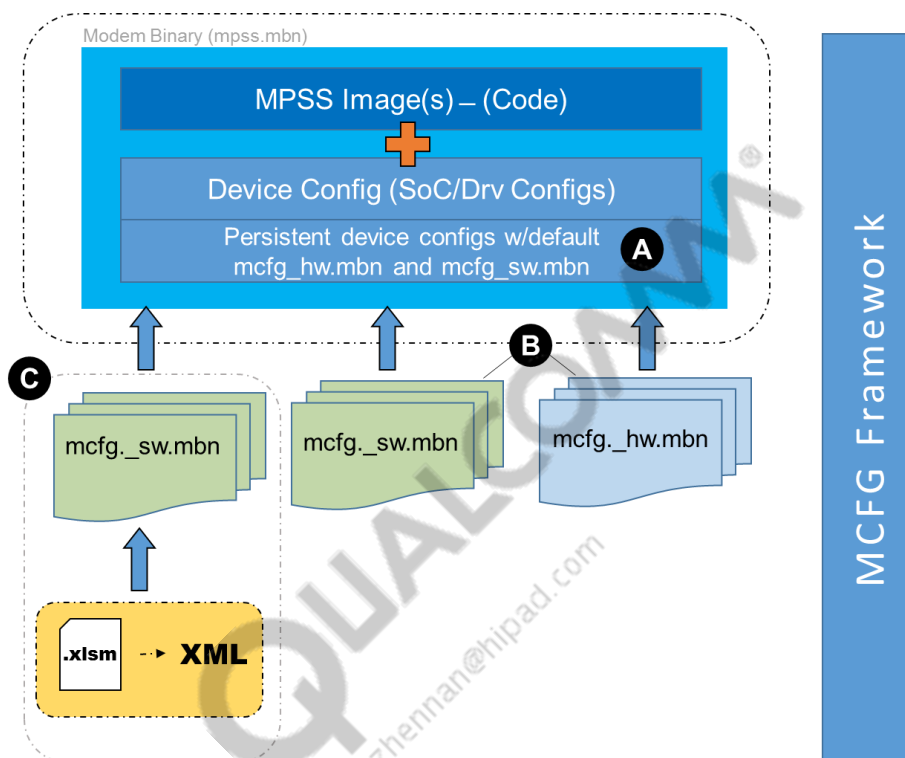


Figure 2-1 MCFG framework feature concept

A	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.
B	Additional reference MBNs are provided that contain the QTI-recommended settings and the necessary configurations for proper operation on specific commercial networks or for lab testing. You provision the UE by locating, loading, and activating the carrier-specific MBN that corresponds to the device type and SIM capabilities of the device.
C	OEMs can generate their own MBNs, if necessary, to meet product-specific requirements. There are two ways to generate MBNs: <ul style="list-style-type: none"> ▪ 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 UIM-based auto-selection feature is enabled, this selects the correct carrier software MBN based on the Issuer Identification Number (IIN) field of the ICCID in the UIM.

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

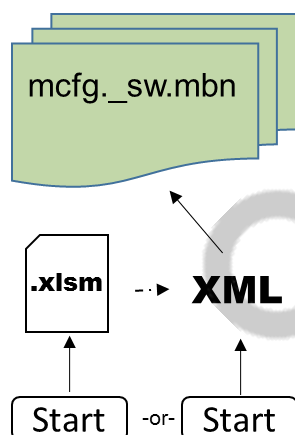


Figure 2-2 Two ways to generate MBNs

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 subsequently 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
 - 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

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

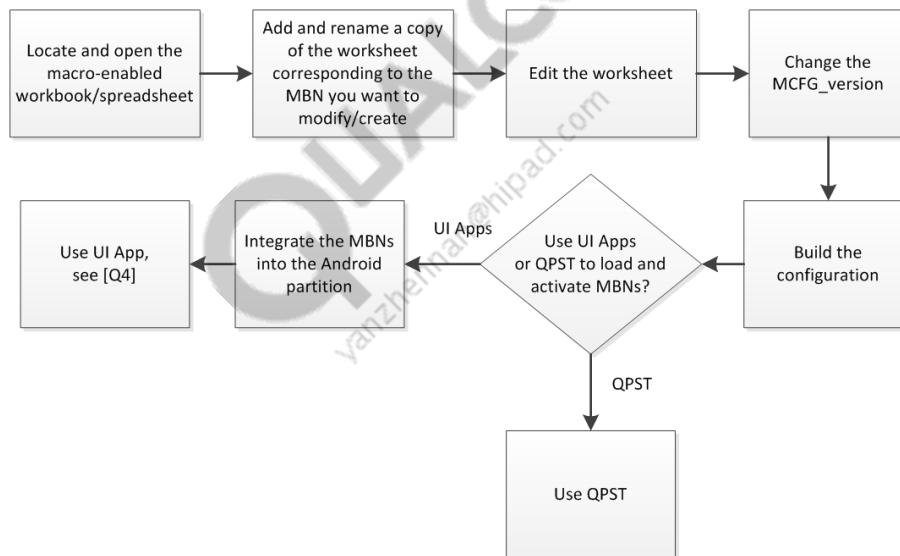
2.3.2 Security

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

Software configuration supports the concept of multiple subscriptions via the use of a subscription bitmask. This mask determines for which subscription slots an NV/EF1 setting will be active.

3 Generating MBNs Using the Macro-Enabled Workbook

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 8 for an overview of the workbook.



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.

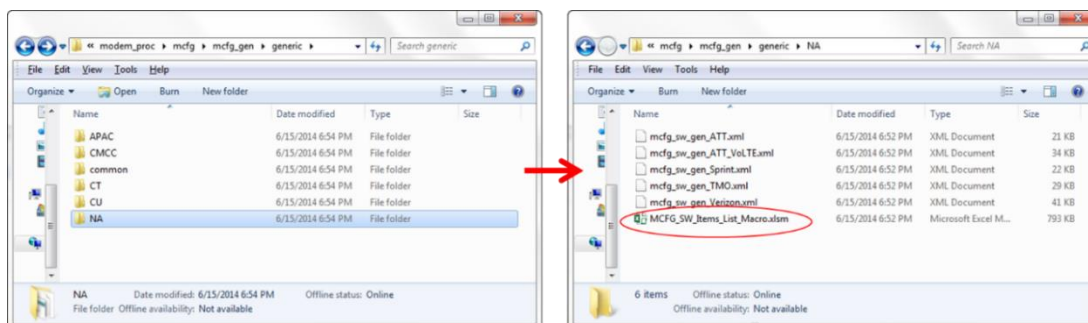
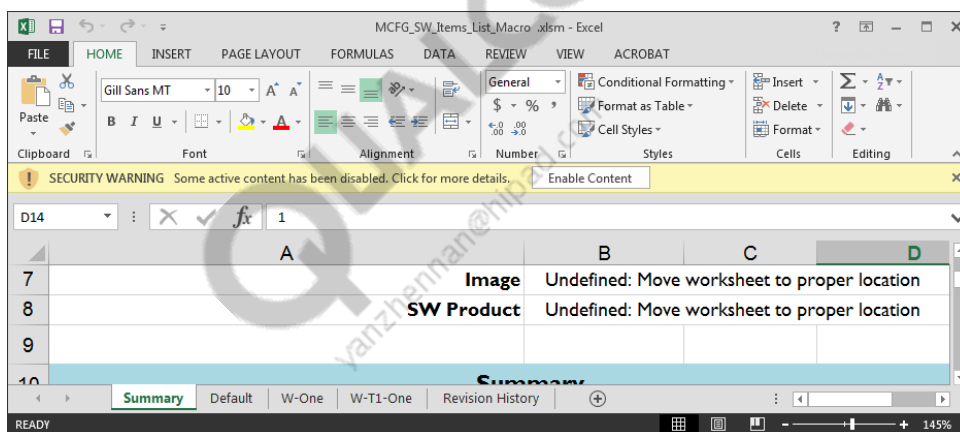


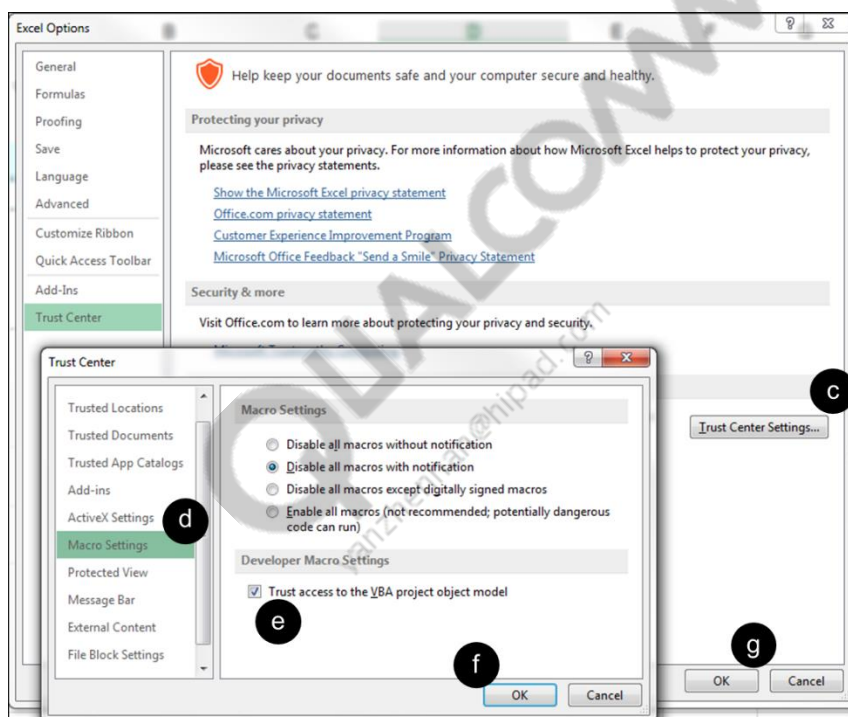
Figure 3-1 Location of the MCFG software items workbook

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

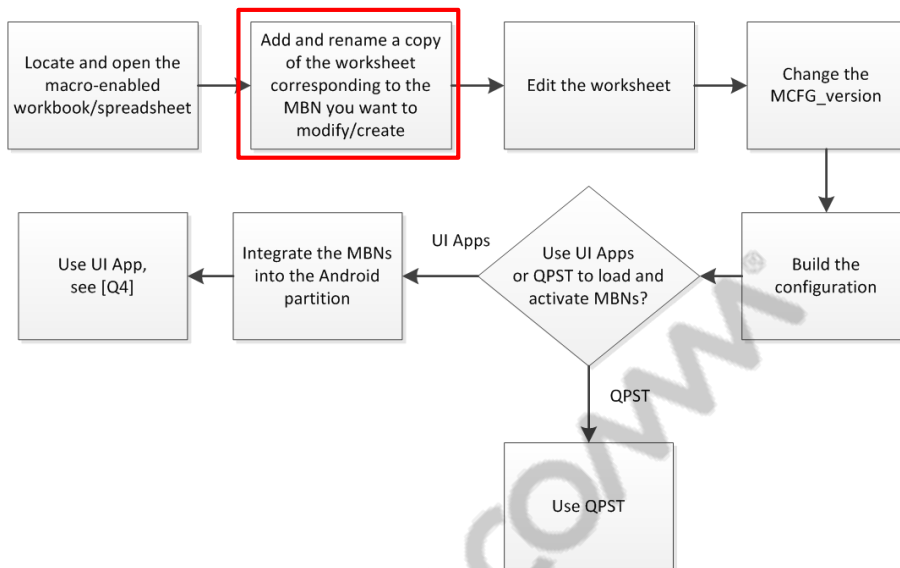
1. Move to `$BUILD_ROOT\modem_proc\mcfg\mcfg_gen\generic`.
2. Open the directory for your region or carrier.
3. Open the file named `MCFG_SW_Items_List_Macro.xlsm`.



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** checkbox.
 - f. Click **OK**.
 - g. Click **OK**.

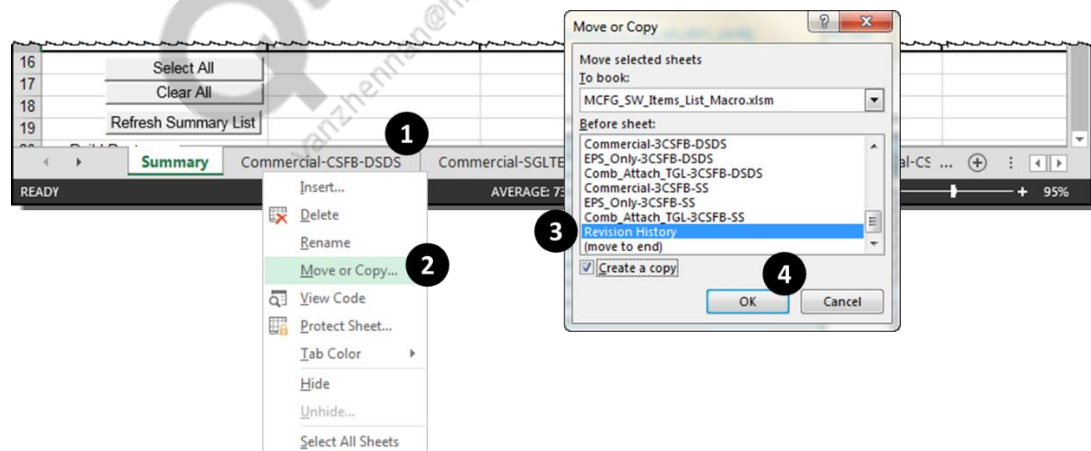


3.2 Add and rename worksheets

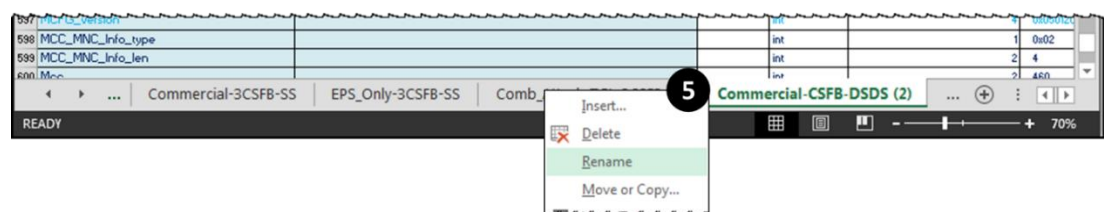


To add and rename worksheets, do the following:

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.

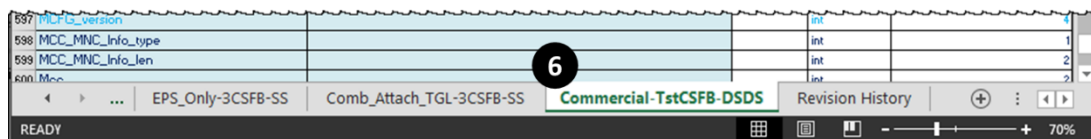


2. Select **Move or Copy**.
3. The copy must be placed before the Revision History worksheet. Select **Revision History** and the **Create a copy** checkbox.
4. Click **OK**. Excel creates and displays the copy.
5. Right-click the new worksheet and select **Rename**.



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.



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

Trailer Record						
589	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

Trailer Record						
589	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 checkbox.

10	Commercial-CSFB-DSDS		Commercial-SGLTE-DSDA	Commercial-TstCSFB-DSDS
11	Carrier Index	32	33	
12	Full MCFG Version	0x05012001	0x05012101	
13	Configuration Type	1	1	
14	Select Carriers for Generation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15	Select All			
16	Clear All			
17	Refresh Summary List			
18	Build Root			
19				
20				

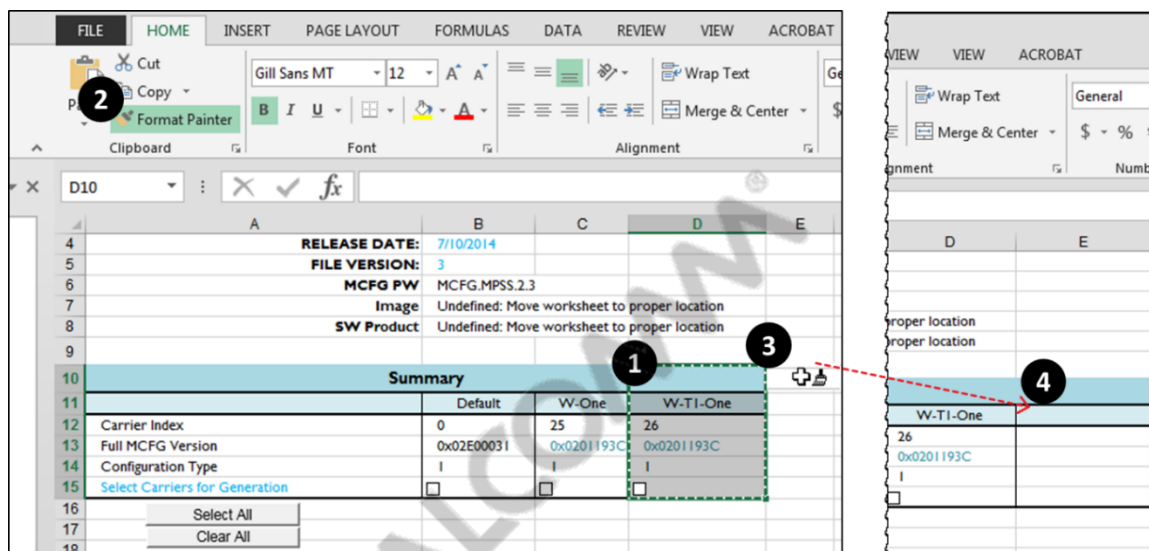
9. Type the values for the new column.

- 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.

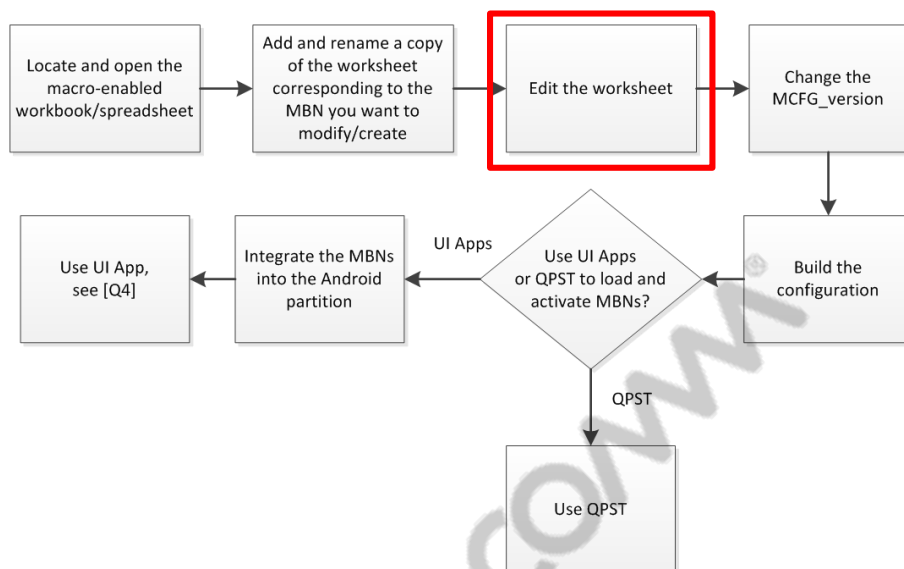
Process for when there is no Refresh Summary List button

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



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 checkbox, are activated for the cells pasted in Step 3.

3.3 Edit the worksheet



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 (this is the bulk of the sheet)
- Trailer Record information at the bottom

See Chapter 8 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.

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

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

NOTE: 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	1	nam	0		
11			int	1	enabled[0]	1		[enabled]
12			int	1	enabled[1]	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 Tag	EFS Item Type	EFS Item Size	Value	Attributes	EFS Filename	Comment
239	IMS SIP Extended Configuration	nvitem_files\msl\ip_ims_sip_extended_0_config	efs_item						69744
240				int	1	1	0x09		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	60000			Timer_T2Value
246				int	4	17000			Timer_T4Value
247				int	4	30000			Timer_T3Value
248				int	4	30000			Timer_T3Value
250				int	1	1			CompactFormEnabled
251				int	1	0			SigCompEnabled
252				int	1	0			FMCCConfig
253				int	1	0			IpSecScheme
254				int	1	0			IpSecEncAlgo
255				int	1	3			AuthScheme
256				int	1	0			InitialAuthConfig
257				string	256				AuthHeaderValue
258				string	256				PriorityRouteValue

■ After

139	EFS File Description	Full Path in EFS Filesystem	NV Item Type	EFS Item Type	EFS Item Size	Value	Attributes	EFS Filename	Comment
239	IMS SIP Extended Configuration	nvitem_files\imaps_ims_sip_extended_0_config	efs_item				0x09		63744
240				int	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	6000			Timer_T2Value
246				int	4	17000			Timer_T4Value
247				int	4	30000			Timer_T1Value
248				int	4	30000			Timer_T3Value
249				int	2				ITCPThresholdValue
250				int	1	1			CompactFormEnabled
251				int	1	0			SigCompEnabled
252				int	1	0			FMDCConfig
253				int	1	0			IpSecInScheme
254				int	1	0			IpSecEncAlgo
255				int	1	3			AuthScheme
256				int	1	0			InitialAuthConfig
257				string	256				AuthHeaderValue
258				string	256				ProxyRouteValue
259				int	1	1			KeepAliveEnabled
260				int	4				ITime_NaRTTValue
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

- 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.
- 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 Add an EFS file

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

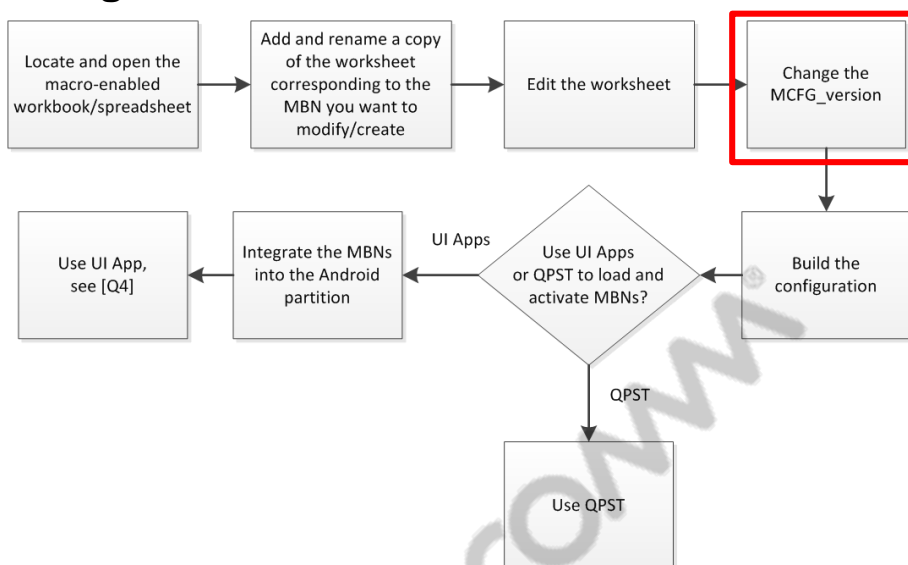
76	EFS File Description	Full Path in EFS Filesystem	NV Item Type	EFS Item Type	EFS Item Size	Value	Attributes	EFS Filename
379	Mandatory APN List	eRPD/mandatory_apn_list.txt	efs			0x09		modem_proc\mcfg\mcfg_gen\scripts\data\efs_files\vwz\mandatory_apn_list.txt

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.

3.3.6 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



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

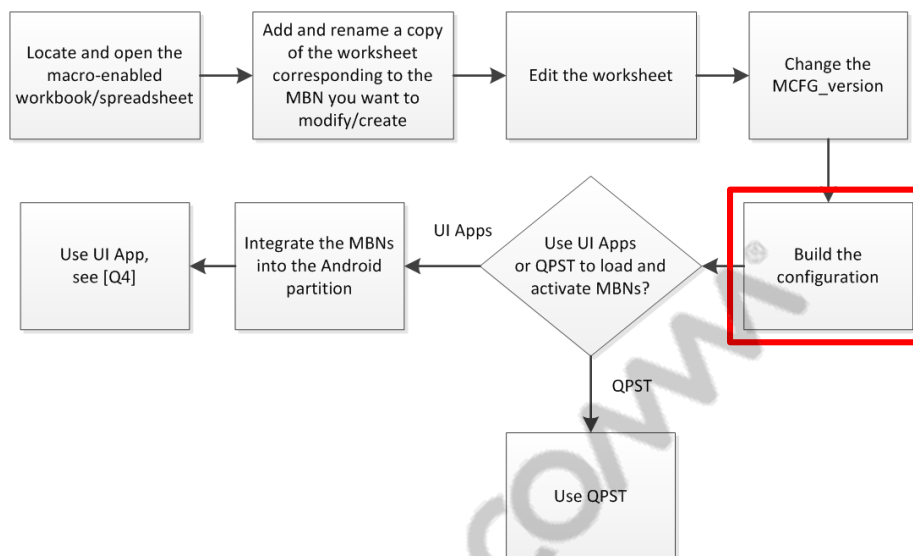
NOTE: Although there is no rule about version numbering, it is best that you increment the configuration version by one for each set of changes.

2

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	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

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



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

	Commercial-CSFB-DSDS	Commercial-SGLTE-DSDA	Commercial-TstCSFB-DSDS
Carrier Index	32	33	146
Full MCFG Version	0x05012001	0x05012101	0x05022001
Configuration Type	I	I	I
Select Carriers for Generation	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> 1
Select All			
Clear All			
Refresh Summary List			
2 Build Root			
Build ID			
Build Version	0571		
Keep command window open		<input checked="" type="checkbox"/>	
3 Generate Source Files Only			
Generate Sources and Build MBN Files			
Build Signed MBN Files			

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. Enter the build root, build ID, and build version.
3. There are three 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.

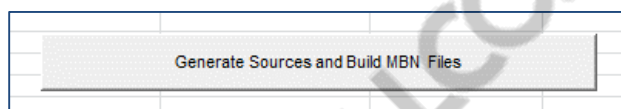
□ **Build Signed MBN Files**

This option is used for creating signed MBN files. It is used in conjunction with the Generate Sources and Build MBN Files option.

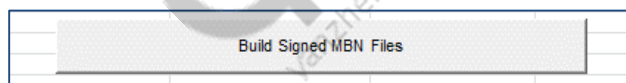
3.5.1 Build signed MBN files

To generate secboot-enabled builds, do the following:

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



2. Use Code Signing Management System (CSMS) to generate a signature with Secure Boot Software Type as AMSS_HASH_TABLE (0002) and save the certificate. CSMS is available to you as a security feature. See *Presentation: Secured MSM™ Code Signing Service* (80-V9807-1) and *Presentation: Code Signing Management System Overview* (80-V3999-1).
3. Click **Build Signed MBN Files** to make the final signed images.



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.

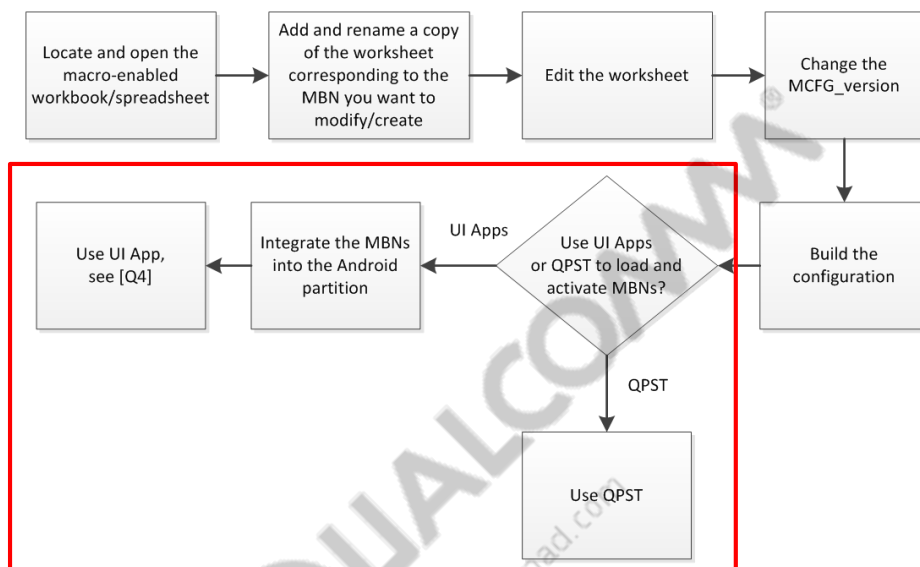
Spreadsheet button	Equivalent command
Generate Source Files Only	<code>>build_mcfgs.exe -- build_id=TAAANAA -- configs=mcfg_sw:all --force-regenerate --sources-only</code>
Generate Sources and Build MBN Files	<code>>build_mcfgs.exe --build_id=TAAANAA -- configs=mcfg_sw:all --force-regenerate --force-rebuild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

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



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.

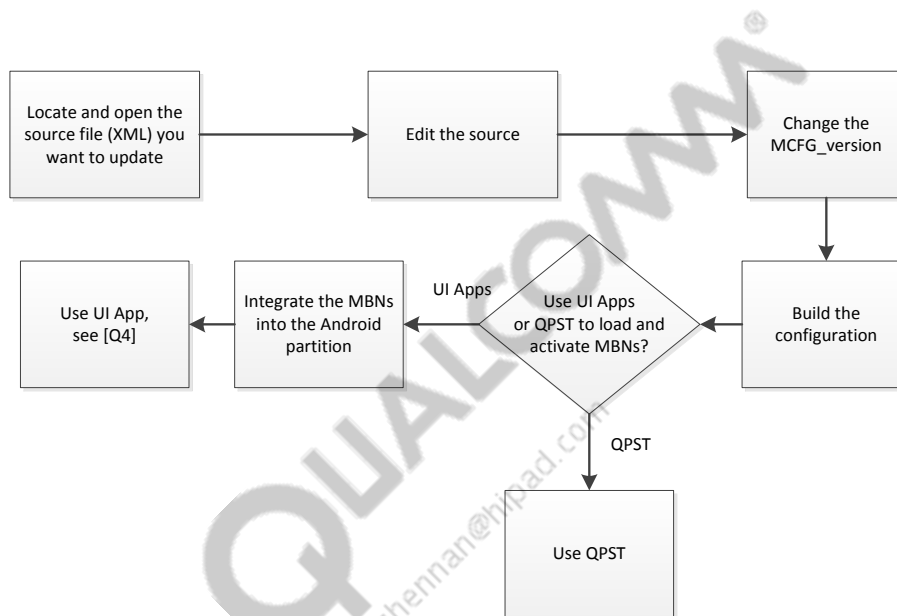
- Windows customers must use QPST.
- Android customers can use either QPST or UI applications.

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 5.
- For information on using QPST, see Chapter 6.
- For information on using UI applications, see *Application Note: Configuring a UE Using Binary Modem Configuration* (80-NP686-1).

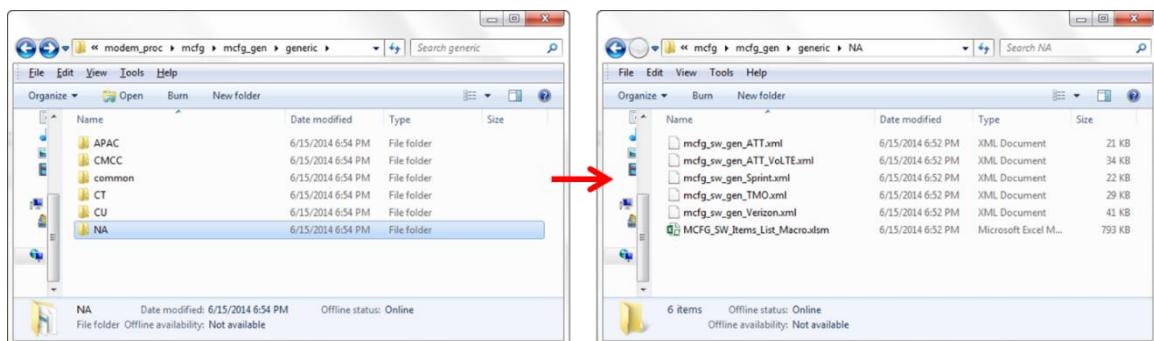
4 Generating MBNs by Modifying the XML Source



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

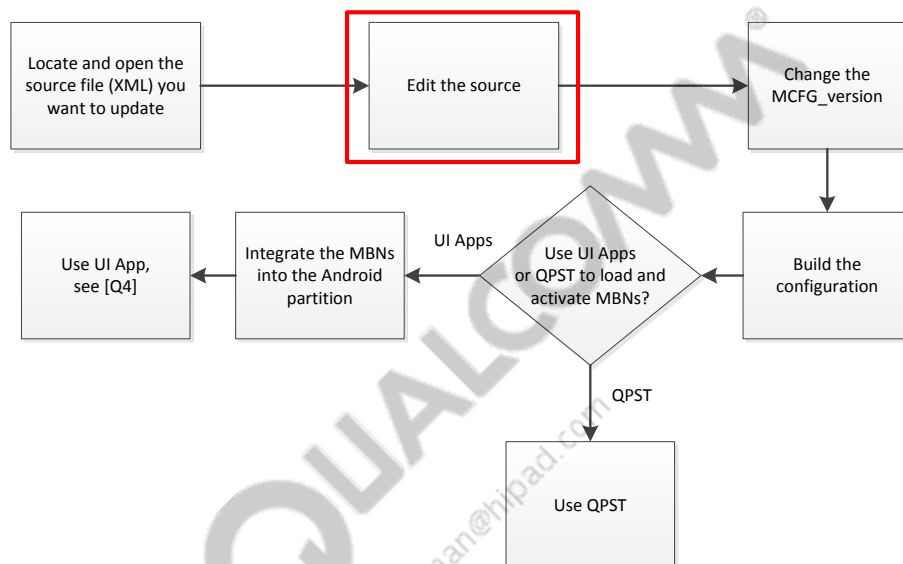
The XML source files are located in the same directories as the workbooks. The following figure shows the directories for each of the regions and the contents of NA directory. The other directories, e.g., APAC, CMCC, etc., contain the source files specific to that region or carrier.



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



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:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <NvData>
3   <NvConfigurationData carrierIndex="1" version="0x02010144" type="1"/>
4   <NvItemData id="6" mcfgAttributes="0x09">
5     <Member sizeOf="1" type="uint8">6 </Member>
6   </NvItemData>
7   <NvItemData id="71" mcfgAttributes="0x09">
8     <Member sizeOf="13" type="uint8">86 69 82 73 90 79 78 </Member>
9   </NvItemData>
10  <NvItemData id="74" mcfgAttributes="0x09">
11    <Member sizeOf="1" type="uint8">1 </Member>
12    <Member sizeOf="1" type="uint8">1 </Member>
13  </NvItemData>

```

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>
</NvEfsItemData>
```

```

<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 Add an EFS file

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

```

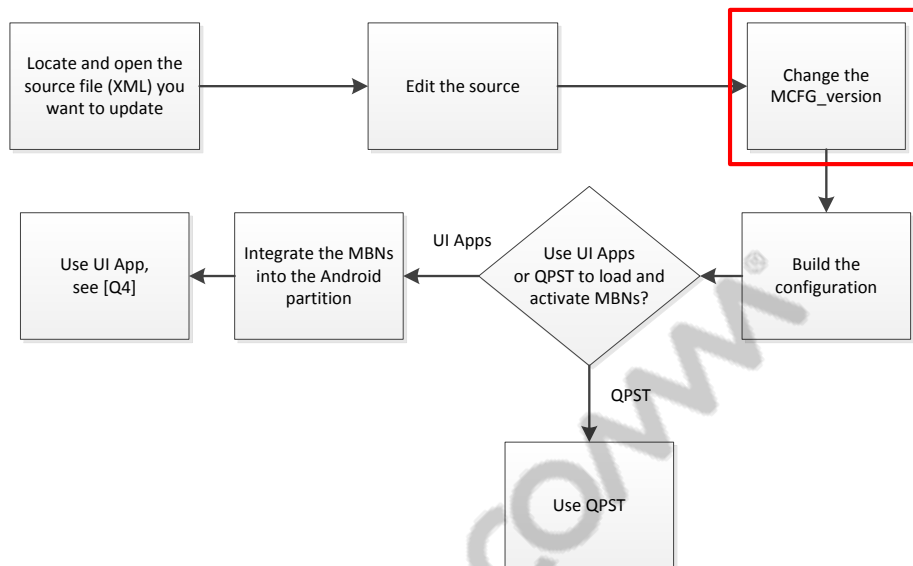
<NvEfsFile mcfgAttributes="0x09" targetPath="/policyman/carrier_policy.xml"
buildPath="modem_proc/mmcp/policyman/configurations/Carrier/ATT/carrier_policy.xml"/>

```

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



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:

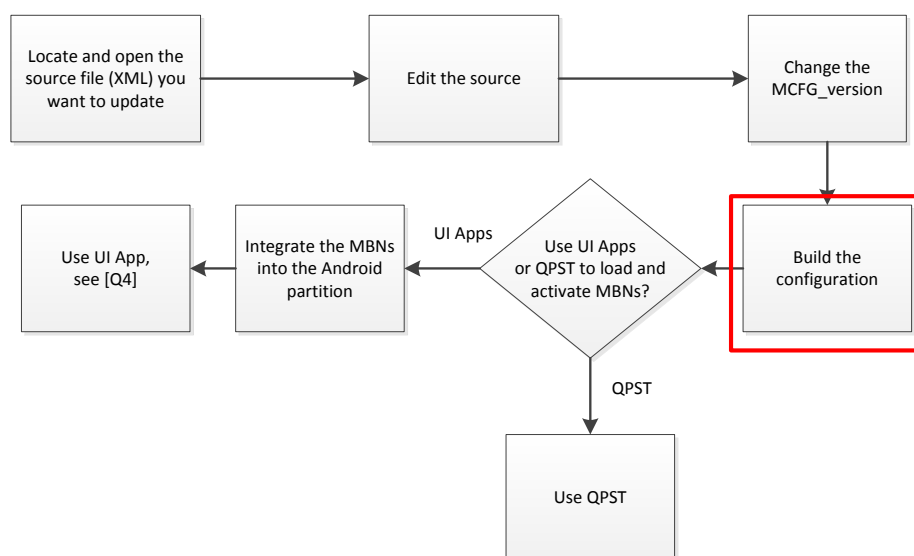
```
<NvTrlRecord mcfgAttributes="0x00">
  <Member sizeOf="1" type="uint8">0 </Member>
  <Member sizeOf="1" type="uint16">2 </Member>
  <Member sizeOf="1" type="uint16">256 </Member>
  <Member sizeOf="1" type="uint8">1 </Member>
  <Member sizeOf="1" type="uint16">4 </Member>
  <Member sizeOf="1" type="uint32">33620280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

2. Increment the version by 1.

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

3. Save the file.

4.4 Build the configuration

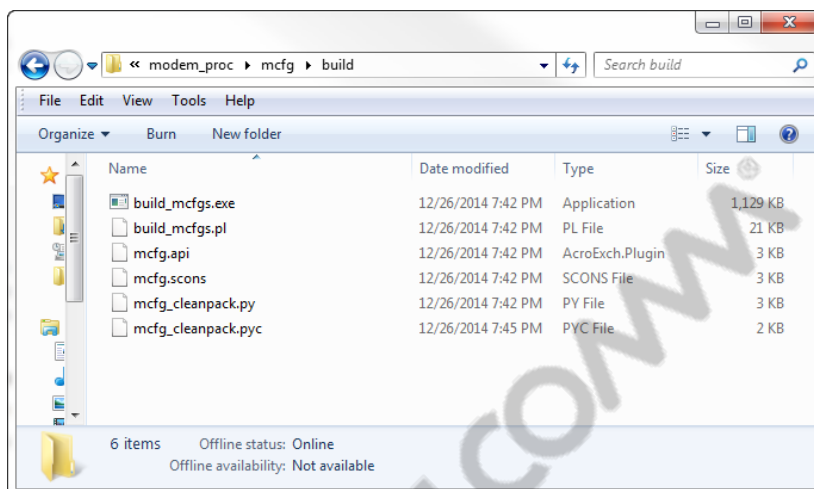


The executable used to generate configurations is *build_mcfgs.exe*.

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:



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]
```

Explanation of command arguments

- `<BUILD ID>` – The build variant ID of the image; listed in the summary spreadsheet of the macro-enabled Excel workbook
- `<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)

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=TAAANAA 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=TAAANAA 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`

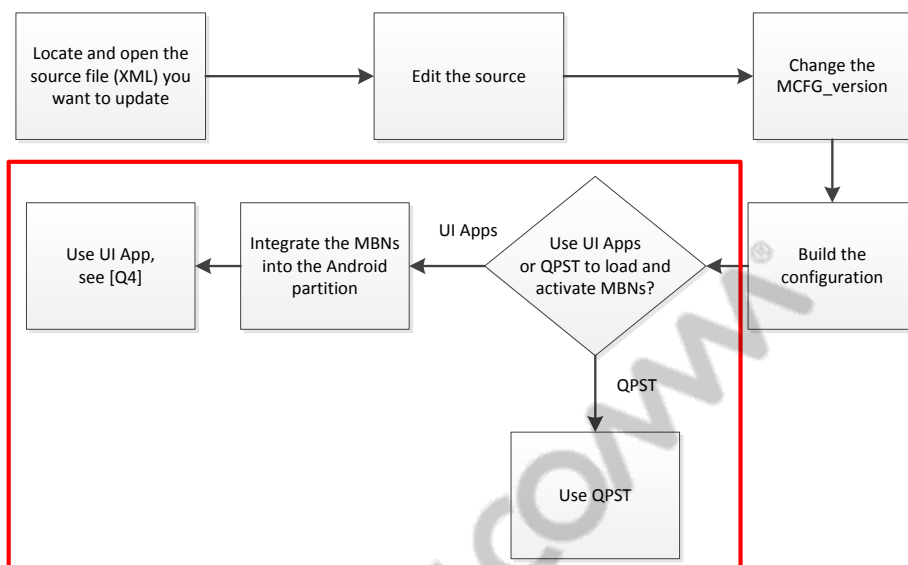
4.4.3 Procedure for multiple images with default configuration

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



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.

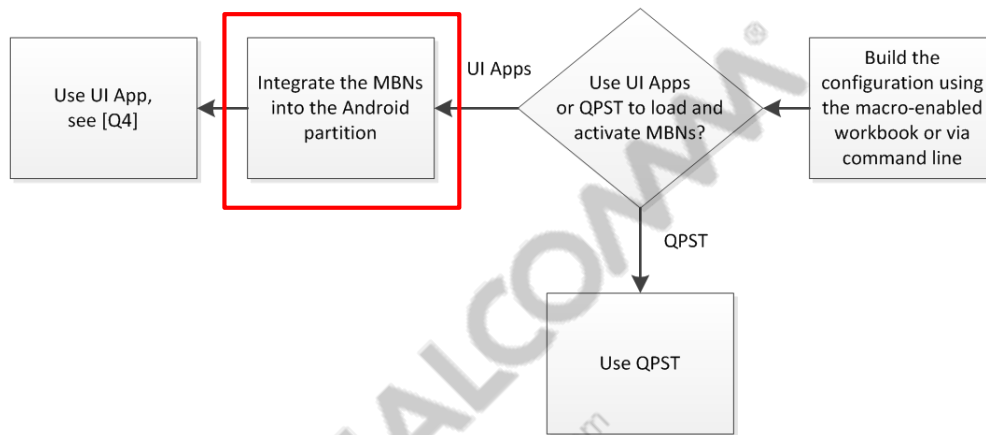
- Windows customers must use QPST.
- Android customers can use either QPST or the UI applications.

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 5](#).
- For information on using QPST, see [Chapter 6](#).
- For information on using the UI applications, see *Application Note: Configuring a UE Using Binary Modem Configuration (80-NP686-1)*.

5 Integrating MBNs into Android



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 remount;
```

- d. Push the local MBN to the Android partition.

```
adb push "local mbn file name"  
/system/vendor/modemconfig/CMCC/SGLTE/SS/new name.mbn
```

In this case, a new name could be EPS_Only_XYZ-CMCC_SGLTE_SS.mbn.

4. Reboot the phone:

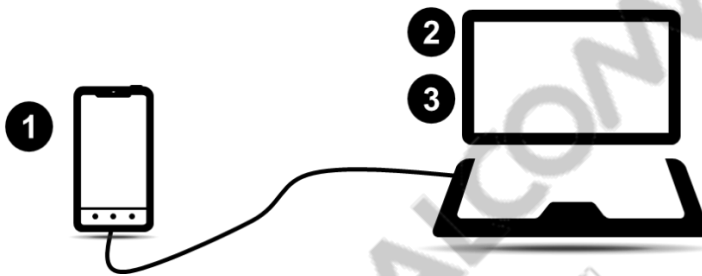
```
adb reboot
```

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

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

6 Loading, Activating, and Deactivating MBNs Using QPST

Setup and prerequisites

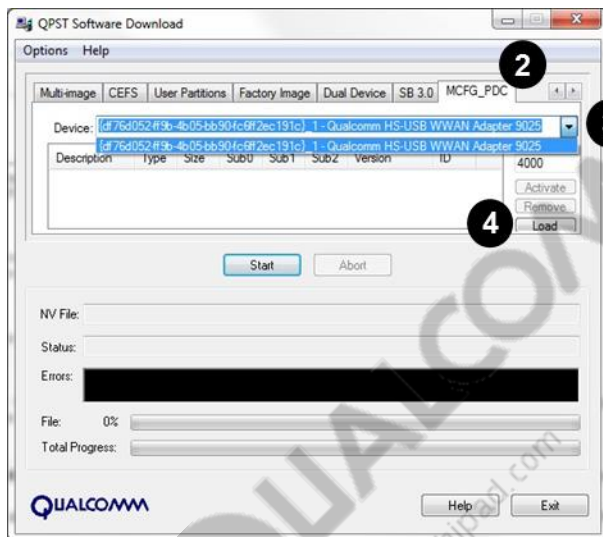


1	UE is connected to a PC
2	PC is running QPST v2.7.421 or later
3	Know your <MODEM_BUILD> path and be able to access it from the PC

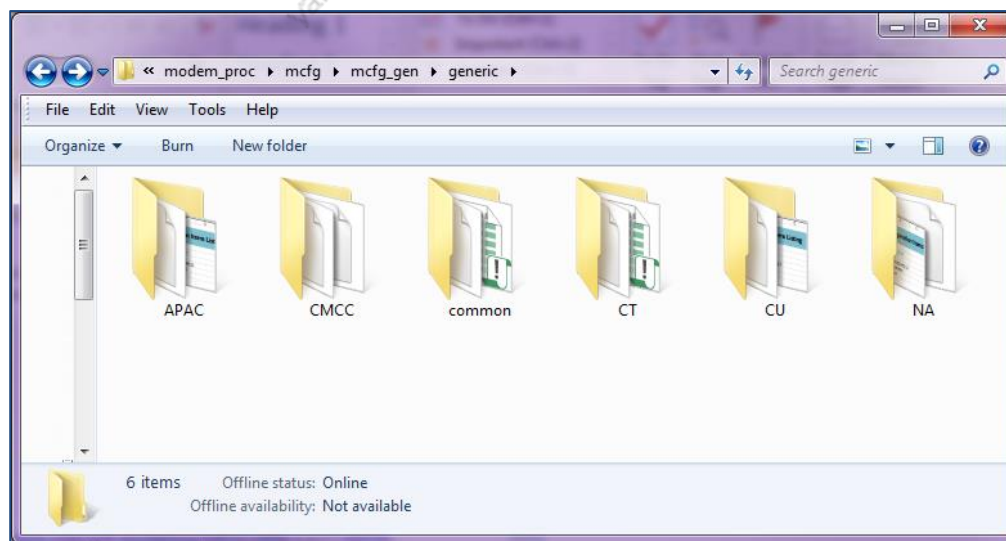
NOTE: If you disabled the RmNet port on the UE, ensure that it is enabled before performing these steps.

6.1 Load and activate 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**.



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



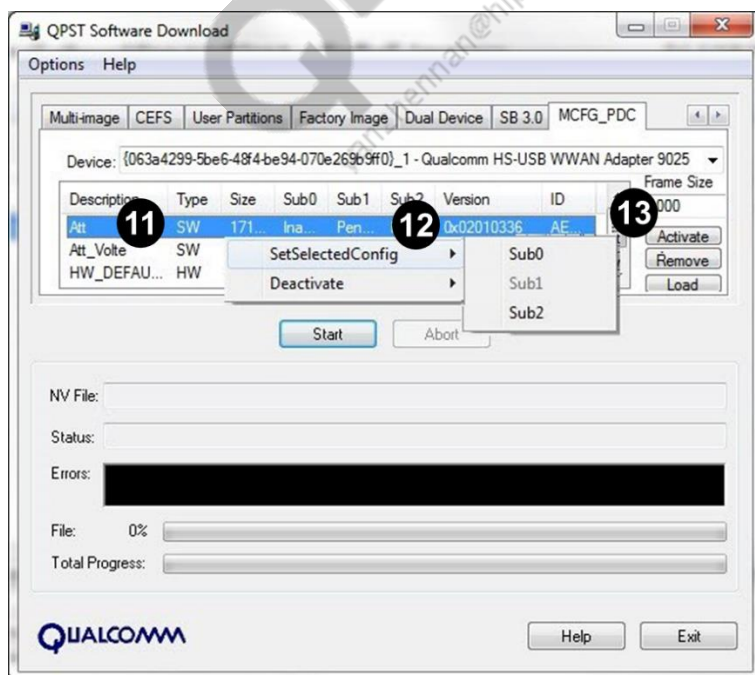
There are directories in this path that organize the MBNs by geographic region or by carrier.

Table 6-1 identifies the appropriate subdirectory for each carrier or region.

Table 6-1 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.
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 10 and 11, and select **Sub1**.

The configuration state changes to Pending after the selection.

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

6.2 Switch between MBNs

To switch between MBNs using QPST, do the following:

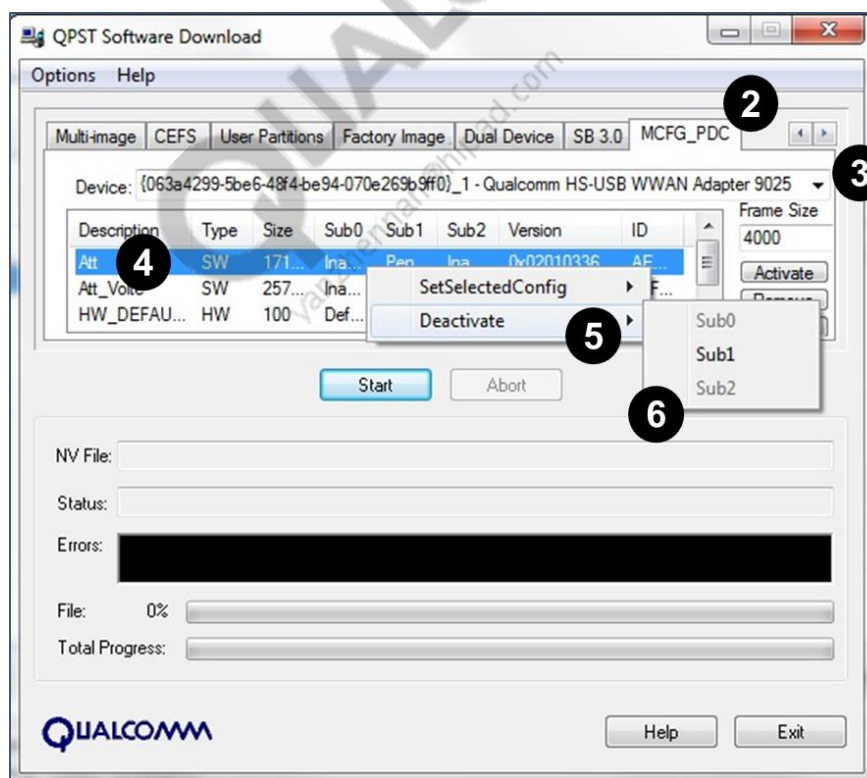
1. Deactivate the currently active MBN. See Section 6.2.1 for more information.
2. Load and activate the new MBN. See Section 6.1 for more information.

6.2.1 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.
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.

6.3 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.

6.4 Working with hardware/platform configurations

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

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

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

7 Autoselection Mechanism

For software configurations, it is also possible to automatically select the correct carrier configuration based on the UICC. The feature is disabled by default.

To enable the autoselect mechanism, load the commercial MBNs using QPST, but do not activate any subscription:

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

See Chapter 6 for details.

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

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

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

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

ITU maintains lists of registered telecom IINs

See Chapter 11 for more information on IINs.

8 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.

8.1 Summary worksheet

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

MPSS Configuration Items Listing

RELEASE DATE: 4/24/2014
FILE VERSION: 61.2
MCFG PW: MCFG.MPSS.2.1
Image: Generic
SW Product: Common

	Default	Verizon	Sprint	ATT	TMO	KDDI	DCM	CT	CU	CMCC	CMCC_Lab	W-One
Carrier Index	0	1	2	3	5	7	13	19	21	24	24	25
Full MCFG Version	0x02E00002	0x02010118	0x0201020D	0x0201030E	0x02010506	0x02010707	0x02010D01	0x02011306	0x02011504	0x02011808	0x02011803	0x02011115
Configuration Type	I	I	I	I	I	I	I	I	I	I	I	I

Build Root: [Select Directory]
Build ID: TAAAAA
Build Version: 13721
Keep command window open: ☒
Generate Source Files Only
Generate Sources and Build MBN Files
Build Signed MBN Files

Figure 8-1 Example Summary 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 is the starting point for the script, where it determines which other worksheets to process.

Figure 8-1 shows a summary worksheet where the columns are the column headers since there are Verizon and Sprint columns, spreadsheets in the workbook named Verizon and Sprint would be expected.

- These are typical carrier names for software configurations, but they can be anything as long as they match the spreadsheet names.
- These names are used to differentiate the various output .xml files. A spreadsheet named Verizon would yield an output .xml file named <PREFIX>Verizon.xml, where <PREFIX> is whatever output prefix is specified in the command line arguments.

8.2 SW items list worksheets

The worksheets are arranged into three sections.

	A	B	C	D	E	F	G	H
1	Verizon MCFG SW Items List							
2								
3	NV Items							
4	NV Item Name	NV Item ID	NV Item Type	NV Item Size	NV Item Struct	Value	Attributes	Comments
5	Mobile CAI Revision Number	6	int	1	mob_cai_rev	6	0x09	
6	Banner	71			banner		0x09	
7			string	13	letters	VERIZON		
8	Auto Answer Setting	74			auto_answer		0x09	
9			int	1	enable	1		
10								
11	EFS Files							
12	EFS File Description	Full Path in EFS filesystem	NV Item Type	EFS Item Type	EFS Item Size	Value	Attributes	EFS Filename
13	PDN Throttling Config	/m/item_files/modem/data/efs	efs			0x09		modem_proc/mcfg/mcfg_gen/scripts/data/efs_
14	Thermal Config	/m/item_files/therm_mon/efs	efs			0x09		modem_proc/mcfg/mcfg_gen/scripts/data/efs_
15	LTE NAS temp forbidden PLMN fe	/m/item_files/modem/nas/efs_item	efs_item			0x09		
16								
17	3GPP2 Max PPP Inactivity TimerEn	/m/item_files/data/3gpp2/efs_item	efs_item	int	4	0xFFFFFFFF	0x09	
18								
19			int		1	1		
20								
21	Trailer Record							
22	Data field		NV Item Type	Data Type	Data Size	Data Value	Attributes	
23	Verizon Trailer Record		tr				0x00	
24								
25	MCFG_trl_struct_version_type			int		1	0	
26								
27	MCFG_trl_struct_version_len			int		2	2	
28								
29	MCFG_trl_struct_version			int		2	0x0100	
30								
31	MCFG_version_type			int		1	0x01	

- A** NV Items
- B** EFS Files (this is the bulk of the worksheet)
- C** Trailer Record

Each section 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 8-1 provides a description of the NV item types that appear in the EFS Files section of the worksheet.

Table 8-1 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"

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

8.2.1 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 additional 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		enable	1		

Table 8-2 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: <ul style="list-style-type: none"> int – An integer value int [] – An integer array of unspecified length; its size is dependent on how many elements are listed in the value column int [n] – An integer value of length n string – A character string
NV Item Size	One of the following: <ul style="list-style-type: none"> 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 the length of the string should be the number of characters in the value + 1 for the NULL terminator An integer means the string should be a fixed size; it will be padded with zeros if the value does not have this many characters
Value	One of the following: <ul style="list-style-type: none"> If type is an int array – Value should be a list of integers or hex values delimited by commas If type is an int value – Value should be an integer or hex value If type is a string – Value can be any string
Attributes	See Section 8.2.4 for details

8.2.2 EFS Files details

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

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

Table 8-3 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: <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> efs efs_dir efs_item prl trl
EFS Item Type	One of the following: <ul style="list-style-type: none"> int – Integer value int [] – Integer array of unspecified length; its size is dependent on 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: <ul style="list-style-type: none"> 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 the length of the string should be the number of characters in the value + 1 for the NULL terminator An integer means the string should be a fixed size; it will be padded with zeros if the value does not have this many characters
Value	One of the following: <ul style="list-style-type: none"> If type is an int array – Value should be a list of integers or hex values delimited by commas If type is an int value – Value should be an integer or hex value If type is a string – Value can be any string
Attributes	See Section 8.2.4 for details
EFS Filename	Local path to file relative to build root

8.2.3 Trailer Record details

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

Trailer Record						
Data field	NV Item Type	Data Type	Data Size	Data Value	Attributes	
Verizon Trailer Record	trl				0x00	
MCFG_trl_struct_version_type		int	1	0		
MCFG_trl_struct_version_len		int	2	2		
MCFG_trl_struct_version		int	2	0x0100		
MCFG_version_type		int	1	0x01		

Table 8-4 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: <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> 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 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 will be padded with zeros if the value does not have this many characters
Data Value	One of the following: <ul style="list-style-type: none"> If type is an int array – Value should be a list of integers or hex values delimited by commas If type is an int value – Value should be an integer or hex value If type is a string – Value can be any string
Attributes	0x00

8.2.4 Attributes column

The Attributes column (column G) of the worksheet is designed to contain hex values representing one byte of data. The bits are set as described in [Table 8-5](#).

Table 8-5 Description of attribute fields and values

Attribute field	Value	Description
#define U_ITEM_ATTRIB_CFG	0x01	/* C */ – This is a configuration bit. If this is set, the values for the MCFG item are taken from the spreadsheet.
#define U_ITEM_ATTRIB_MUXD	0x02	/* M */ – This is a multiplexed item. If this is set, the MCFG item becomes a symbolic link to the filename. <MCFG_Item>_S<carrier_index> – This attribute is used to support multiple subscriptions. Typical MCFG items that fall into this category include items that are set by the configuration and can also be modified via OTA or connection managers, e.g., PRL item. Note: This is 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 prior to activation. Not many items fall into this category.
#define U_ITEM_ATTRIB_REST_FACT	0x08	/* R */ – This is always set to 1.
#define U_ITEM_ATTRIB_MULTISIM	0x10	/* S */ – This is for multi-SIM-related NVs.
#define U_ITEM_ATTRIB_INDEXED	0x20	/* I */ – This is set to 1 for legacy, old-style indexed items.
#define U_ITEM_ATTRIB_DELETE	0x40	/* D */ – This is used if the set deletes the MCFG item from the device. Use caution and only if required.
#define U_ITEM_ATTRIB_UPDATE_ONLY	0x80	This bit allows for more optimization. If this is set, the values are read first and written only if different. This saves time in case of slow access flash devices like SPI-NOR where write times are significant.

8.2.5 Formatting

Table 8-6 describes the formatting of the worksheets. Any color that may appear in the worksheets that is not in the legend is an artifact and irrelevant with regard to the user.

Table 8-6 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
Grey background	Placeholder (listing present in spreadsheet only; data is not a part of MBN)
Blue Font (any background)	Recently changed item

9 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.

QUALCOMM
vanzhennan@hipad.com

10 Troubleshooting

10.1 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.

11 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:

- a. Close all Office applications.
- b. 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 *.exd files and not *.exe.
- c. Obtain the following:
 - C:\users\username\AppData\Local\Temp\Excel8.0\MSForms.exd
 - C:\users\username\AppData\Local\Temp\VBE\MSForms.exd
- d. Reboot the computer.
- e. 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. Below is a description of each XML element type and their associated attributes.

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 to be read in 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_files\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

QUALCOMM
vanzhennan@hipad.com

B References

B.1 Related documents

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

B.2 Acronyms and terms

Acronym or term	Definition
CSFB	Circuit Switched Fallback
DSDS	Dual Sim Dual Standby
EFS	Embedded File System
MBN	Modem configuration binary
MCFG	Modem Configuration
NV	Nonvolatile
OTA	Over the Air
PL	Product Line
QPST	Qualcomm Product Support Tool