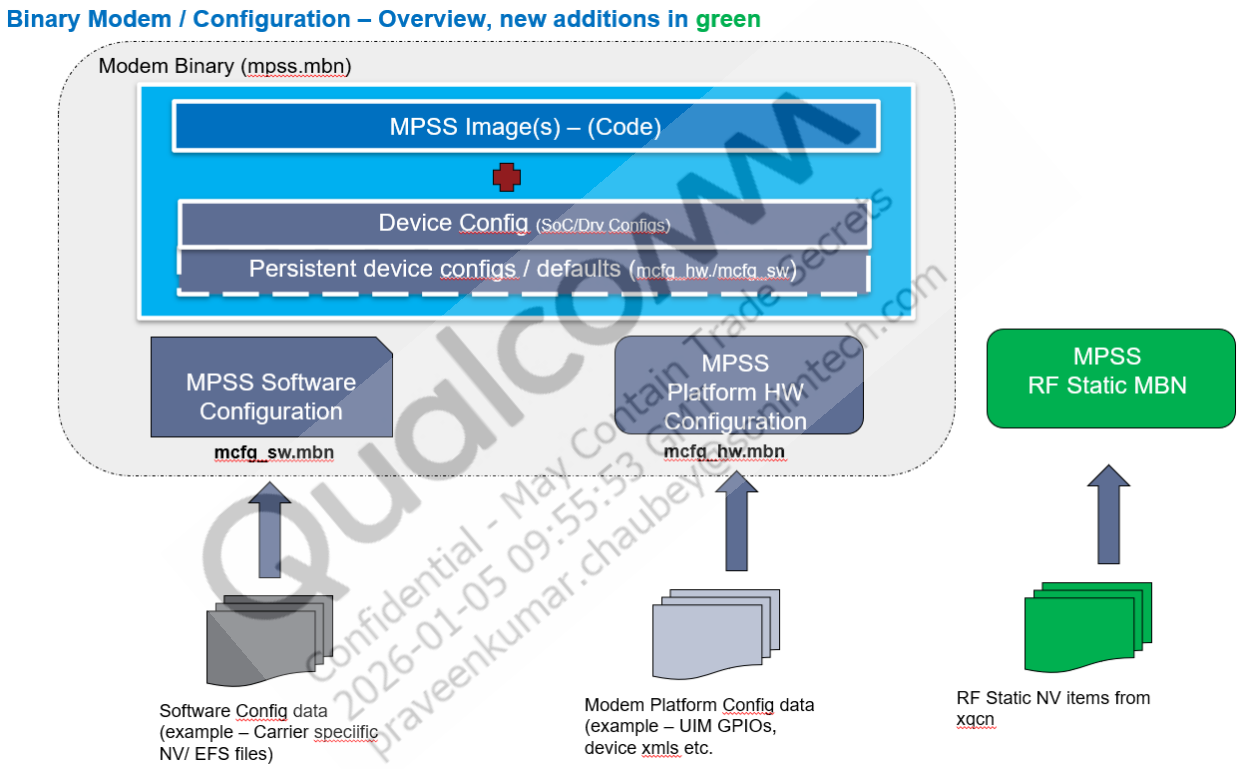
How to create and use rf_static.mbn

Motivation:

- In general RF static NVs occupy more than 50% of EFS space
- As EFS space is limited, it's advantageous to store the static RF NVs on Apps side (RFS) to save EFS space.
- This new feature is introduced from DE3.0 (SM8550) to store the RF static NVs as new 'rf_static_*.mbn'
- This file is stored on RFS side and read by MCFG at power up.



How the feature works:

- OEMs create a new XML file to store the RF static NV items
- MCFG module will create a rf_static.mbn with the values from this XML
- MCFG also authenticates this rf_static.mbn and saves in DDR.
- When modem boots up, MCFG module will read RF NV(/nv/item_files/rfnv/28967,/nv/item_files/rfnv/29652) and get to know the HWID and BID and will fetch the right RF static MBN and put that in DDR memory.
- On power-up, RF module will first read from EFS (for the NVs loaded via QCN).
    - If the item is not found (EFS read fails), then RF will query MCFG to get the NV value.

- This NV value is what OEM has stored in the XML/rf_static.mbn

NOTE:

- RF reads the QCN/EFS first. If the NV value is found in there, then RF will not query MCFG rf_static.mbn.
- So OEMs should keep the values in QCN vs rf_static.mbn diligently. There should not be overlapping NVs, as QCN is always given priority.

Advantages of this approach:

- QCN can be light and the static RF NV items that can be modified in future (via OTA) can be done easily by placing them in rf_static.mbn
- Less EFS space consumption

How to configure rf_static.mbn:

- OEM need to create an XML and store the XML in the path (e.g.):
  …/modem_proc/rf/build/mcfg_config/mcfg_rf_gen_hwid895_fsid00.xml
  NOTES:
  - The XML file name should follow the below nomenclature :
    mcfg_rf_gen_<HW ID number>_<feature set ID number><BID>.xml
    Ex: mcfg_rf_gen_hwid895_fsid00.xml
  - Please use 00 for BID always
  - RF NVs 28967 and 29652 are configured in the QCNs which can provide the hardware_ID and featureset_ID for RFNV_RF_CARD_CMW_CONFIG and RFNV_RF_CARD_MMW_CONFIG respectively
- Sample XML can be found at: modem_proc/rf/build/mcfg_config/
- The data of the above XML can be edited like below:

```
<?xml version="1.0" encoding="utf-8"?>
<NvData McfgXmlVersion="2.0">
    <NvConfigurationData carrierIndex="0" version="mmcp.0.0" type="2" />
    <!-- Please fill in the needed data for targetPath and buildPath before
uncommenting the entry below -->
    <!--
    <NvEfsFile name="name of NV item " id="none" description="RF Static"
comment="" category="Unknown" subscription_mask="0x07" mcfgAttributes="0x19"
mcfgVariant="2" targetPath="/location/file/should/reside/in/target_efs"
buildPath="modem_proc/rf/location/of/file/in/modem_build" />
    -->
    <NvEfsFile name="name of NV item" id="none" description="RF Static"
comment="" category="Unknown" subscription_mask="0x07" mcfgAttributes="0x19"
mcfgVariant="2"
targetPath="/rfc/proto/rfc_hwid895_de_2xmagnus_global_et_v2_cmn_ag.dat"
buildPath="modem_proc/rf/card/config/target/olympic/rf_card/rfc_hwid895_de_2x
```

```
magnus_global_et_v2_ag/build/default/src/rfc_hwid895_de_2xmagnus_global_et_v2
_cmn_ag.dat" />

    <NvEfsFile name="name of NV item" id="none" description="RF Static"
comment="" category="Unknown" subscription_mask="0x07" mcfgAttributes="0x19"
mcfgVariant="2"
targetPath="/rfc/proto/rfc_hwid895_de_2xmagnus_global_et_v2_res_ag.dat"
buildPath="modem_proc/rf/card/config/target/olympic/rf_card/rfc_hwid895_de_2x
magnus_global_et_v2_ag/build/default/rfc_hwid895_de_2xmagnus_global_et_v2_res
_ag.dat" />

    <!-- Example of how to pull data from an xQCN ( in case it's needed at a
later time )
    <NvEfsFile name="29619" id="none" description="RF Static" comment=""
category="Unknown" subscription_mask="0x07" mcfgAttributes="0x19"
mcfgVariant="21" targetPath="/nv/item_files/rfnv/00029619"
buildPath="modem_proc/rf/rftarget_denali/mtp/qcn/hwid895_bid0_pid255/hwid895_
bid0_pid255_reference.xqcn" />
    -->
</NvData>
```

- OEM must add 'USES_FLAGS=USES_MCFG_RF_MBN_BUILDER' to the build command for the RF static MBN generation to happen (do a clean build).
- scons file OEM need to edit to create new xml -> rf mbn pairings (path): …/modem_proc/rf/build/mcfg/mcfg_rf_gen.scons (contents of the file must be in the following format):

```
Import('env')
env = env.Clone()

#-------------------------------------------------------------------------------
--
# MCFG Image Variables
#-------------------------------------------------------------------------------
--
MCFG_RF_IMG = env.get('MCFG_RF_IMG')
MCFG_IMGS   = env.get('MCFG_IMGS')

#-------------------------------------------------------------------------------
--
# Add MCFG groups to accepted MBN configurations
#-------------------------------------------------------------------------------
--
if 'USES_MCFG_XML_BUILDER' in env:
    if MCFG_IMGS is not None and env.get('IMAGE_NAME') in MCFG_IMGS:
        env.AddMcfgInfo(MCFG_RF_IMG,
        {

'modem_proc/rf/build/mcfg_config/mcfg_rf_gen_hwid895_fsid00.xml':['rf_static_
895_00_00'],
        })
```