



## Customization in NVRAM

Doc No: CS6000-BC4B-CGD-V1.0EN

Version: V1.0

Release date: 2017-02-14

Classification: internal

© 2008 - 2017 MediaTek Inc.

This document contains information that is proprietary to MediaTek Inc.

Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.

Specifications are subject to change without notice.

Keywords  
Customization Guide

**MediaTek Inc.**

Postal address

No. 1, Dusing 1st Rd. , Hsinchu Science  
Park, Hsinchu City, Taiwan 30078

MTK support office address

No. 1, Dusing 1st Rd. , Hsinchu Science  
Park, Hsinchu City, Taiwan 30078

Internet

<http://www.mediatek.com/>



Document Revision History

Revision	Date	Author	Description
V1.0	2017-02-14	Koshi Chiu	
V1.1	2017-02-14	Yuchi Xu	
V1.2	2017-02-14	Jian Lin	
V1.3	2017-02-14	Jiajie Hao	

MediaTek Confidential

© 2016 - 2017 MediaTek Inc.

Classification:internal

This document contains information that is proprietary to MediaTek Inc.  
Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.

## Table of Contents

1	NvRAM Rule .....	5
2	Cusomization in NVRAM.....	6
3	Reset to Default.....	8
4	Step by Step to Add NvRAM Data .....	9
5	Add NvRAM File to Backup List.....	11
6	NvRAM Library .....	Error! Bookmark not defined.
7	How to change structure in customer file .....	Error! Bookmark not defined.
8	Step by Step to Reset data(Example).....	Error! Bookmark not defined.
9	Step by Step to Convert data(Example).....	Error! Bookmark not defined.
10	Sync your thread with Nvram Daemon .....	20

## **1 NVRAM Rule**

---

- 1.1 NvRAM bin region is for storing very important data like calibration.**
- 1.2 The available NvRAM size is about 3MB.**
- 1.3 The size of every NvRAM file should be keep small.**
- 1.4 The NVRAM operations need 'system' and 'nvram' group rights.**
- 1.5 The implement in NvRAM custom callback should be finished in short time.**
- 1.6 Be careful use sleep function in customized implement.**

## 2 Customization in NvRAM

**2.1 For the different requirements of projects, NvRAM modules also need to provide the supports of customization configurations, including default value and record data structure of NvRAM files.**

**2.2 There are two parts of NvRAM data**

- Common
  - For MTK platform NvRAM used
  - Customer can see the definition of related NVRAM record structure
  - But should not modify them
- Customized for different projects
  - For customer NvRAM used
  - Customer can see the definition of related NVRAM record structure
  - Can modify them according to the requirements

**2.3 The folder of NvRAM customization is located in the path**

- mediatek\custom\[*\$PROJECT*]\cgen
- vendor\mediatek\proprietary\custom\[*\$PROJECT*]\cgen(L/KK.AOSP use)

**2.4 There are three folders in this customization folder**

- Cfgdefault
  - Used to define the default value of NvRAM files
- Cfgfileinc
  - Used to define the record data structure of NvRAM file
- Inc
  - Used to support general NvRAM module functionalities

**2.5 Should modify the file**

- mediatek\custom\[*\$PROJECT*]\cgen\inc\CFG\_file\_info\_custom.h

- vendor\mediatek\proprietary\custom\[PROJECT]\cgen\inc\CFG\_file\_info\_custom.h (L/KK.AOSP use)
- Data structure of *g\_akCFG\_File\_Custom*

### 2.6 The information of NvRAM file

- File path
  - The file path that the NvRAM files should be store
- File version
- Record size
- Record numbers
- The type of the default value
- The default value
- type of processing when version not match (Convert/Reset)
- Data-convert function

### 2.7 The data structure of *g\_akCFG\_File\_Custom*

```
TCFG_FILE g_akCFG_File_Custom[] =
(
    "/data/nvram/APCFG/APRDC/ Audio_Sph", VER(AP_CFG_RDCL_FILE_AUDIO_LID), CFG_FILE_SPEECH_REC_SIZE,
    CFG_FILE_SPEECH_REC_TOTAL, SINGLE_DEFAULT_REC, (char *) &speech_custom_default, DataReset, NULL),
(
    "/data/nvram/APCFG/APRDEB/GPS", VER(AP_CFG_CUSTOM_FILE_GPS_LID), CFG_FILE_GPS_CONFIG_SIZE,
    CFG_FILE_GPS_CONFIG_TOTAL, SINGLE_DEFAULT_REC, (char *) &stGPSConfigDefault, DataReset, NULL),

```

### 2.8 The default value of *stGPSConfigDefault*

```
ap_nvram_gps_config_struct stGPSConfigDefault =
{
    /* "/dev/ttyMT1" */
    {'/', 'd', 'e', 'v', '/', 't', 't', 'y', 'M', 'T', '1', 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0},
    /* 0:s/w, 1:none, 2:h/w */
    1,
    /* 16.368MHz */
    16368000,
    /* 500ppb */
    500,
    /* 0:16.368MHz TCXO */
    0,
    /* 0:mixer-in, 1:internal-LNA */
    0,
    /* 0:none */
    0
};
```

Note: the sequence of *a\_akCFG\_File\_Custom* must be the same as LID definition table(Page

## 3 Reset to Default

Type	Descriptions
<b><i>SINGLE_DEFAULT_REC</i></b>	<p>If multiple records have same default value, this type should be used to minimize the Ram size.</p> <p>It only need define the default value of one record, NvRAM module will use the default value of this record to initialize all of records</p>
<b><i>MULTIPLE_DEFAULT_REC</i></b>	<p>If NvRAM has different default value for different records, this type should be used.</p> <p>It will use default value which is define in the cfg_file, then writes to NvRAM file</p>
<b><i>DEFAULT_ZERO</i></b>	<p>The default value is 0, the property of default value will not be cared</p>
<b><i>DEFAULT_FF</i></b>	<p>The default value is 0xff, the property of default value will not be cared</p>

☞ [Random filler text. Not intended for actual reading.] Must keep the chapter even it have empty content.



## 4 Step by Step to Add NvRAM Data

### 4.1 Add one header file which describes the definition of its record data structure, record size and record numbers

- Path:mediatek\custom\[*\$PROJECT*]\cgen\cfgfileinc
- Path:vendor\mediatek\proprietary\custom\[*\$PROJECT*]\cgen\cfgfileinc (L/KK.AOSP use)

```
#ifndef _CFG_CUSTOM1_FILE_H
#define _CFG_CUSTOM1_FILE_H

typedef struct
{
    unsigned int Array[1];
} File_Custom1_Struct;

#define CFG_FILE_CUSTOM1_REC_SIZE    sizeof(File_Custom1_Struct)
#define CFG_FILE_CUSTOM1_REC_TOTAL  1

#endif
```

### 4.2 Add header file which define its default value of NvRAM file

- Path:mediatek\custom\[*\$PROJECT*]\cgen\cfgdefault
- Path:of vendor\mediatek\proprietary\custom\[*\$PROJECT*]\cgen\cfgdefault (L/KK.AOSP use)

```
#ifndef _CFG_CUSTOM1_D_H
#define _CFG_CUSTOM1_D_H

File_Custom1_Struct stCustom1Default =
{
    1
};

#endif
```

### 4.3 Add one lid in the enum definition of “CUSTOM\_CFG\_FILE\_LID” and define the version number of NvRAM file

- Path:mediatek\custom\[*\$PROJECT*]\cgen\inc\Custom\_NvRAM\_LID.h
- Path:vendor\mediatek\proprietary\custom\[*\$PROJECT*]\cgen\inc\Custom\_NvRAM\_LID.h (L/KK.AOSP use)

```
/* the definition of file LID */
typedef enum
{
    AP_CFG_RDCL_FILE_AUDIO_LID=AP_CFG_CUSTOM_BEGIN_LID, //AP_CFG_CUSTOM_BEGIN_LID: this lid must not be changed, it is reserved for system.
    AP_CFG_CUSTOM_FILE_GPS_LID,
    AP_CFG_RDCL_FILE_META_LID,
    AP_CFG_CUSTOM_FILE_CUSTOM1_LID,
    AP_CFG_CUSTOM_FILE_CUSTOM2_LID,

    AP_CFG_CUSTOM_FILE_MAX_LID,
} CUSTOM_CFG_FILE_LID;

/* verno of data items */
/* audio file version */
#define AP_CFG_RDCL_FILE_AUDIO_LID_VERNO "001"
/* META log and com port config file version */
#define AP_CFG_RDCL_FILE_META_LID_VERNO "000"

/* custom2 file version */
#define AP_CFG_CUSTOM_FILE_CUSTOM1_LID_VERNO "000"
/* custom2 file version */
#define AP_CFG_CUSTOM_FILE_CUSTOM2_LID_VERNO "000"
/* GPS file version */
#define AP_CFG_CUSTOM_FILE_GPS_LID_VERNO "000"
```

The sequence of lids in enum definition can't change. The newest lid must add the definition table.(before MAX\_LID)

#### 4.4 Add one include path which added in the step 1

- Path:mediatek\custom\[PROJECT] cgen\inc\custom\_cfg\_module\_file.h
- Path:vendor\mediatek\proprietary\custom\[PROJECT] cgen\inc\custom\_cfg\_module\_file.h (L/KK.AOSP use)

#### 4.5 Add one include path which added in the step 2

- Path:mediatek\custom\[PROJECT] cgen\inc\custom\_cfg\_module\_default.h
- Path:vendor\mediatek\proprietary\custom\[PROJECT] cgen\inc\custom\_cfg\_module\_default.h (L/KK.AOSP use)

#### 4.6 Add the related information of NvRAM file into the definition of "g\_akCFG\_File\_Custom"

- Path:mediatek\custom\[PROJECT] cgen\inc\CFG\_file\_info\_custom.h
- Path:vendor\mediatek\proprietary\custom\[PROJECT] cgen\inc\CFG\_file\_info\_custom.h (L/KK.AOSP use)

#### 4.7 Add its related information, including record structure, NvRAM lid, and record number

- Path:mediatek\custom\[PROJECT] cgen\inc\Custom\_NvRAM\_data\_item.h
- Path:vendor\mediatek\proprietary\custom\[PROJECT] cgen\inc\Custom\_NvRAM\_data\_item.h (L/KK.AOSP use)

## 5 Add NvRAM File to Backup List

### 5.1 *If your NvRAM File Need to Backup to BinRegion, add your module in aBackupToBinRegion[]*

Path:mediatek\custom\[*\$PROJECT*]\cgen\CFG\_file\_info.c

Path:vendor\mediatek\proprietary\custom\[*\$PROJECT*]\cgen\CFG\_file\_info.c (L/KK.AOSP use)

```
FileName aBackupToBinRegion[] =
{
    {"FILE_VER",AP_CFG_FILE_VER_INFO_LID},
    {"BT_Addr",AP_CFG_RDEB_FILE_BT_ADDR_LID},
    {"WIFI",AP_CFG_RDEB_FILE_WIFI_LID},
    {"AUXADC",AP_CFG_RDCL_FILE_AUXADC_LID},
    {"FACTORY",AP_CFG_RDCL_FACTORY_LID},
    {"BWCS",AP_CFG_RDCL_BWCS_LID},
    {"HWMON_ACC",AP_CFG_RDCL_HWMON_ACC_LID},
    {"HWMON_GYRO",AP_CFG_RDCL_HWMON_GYRO_LID},
    {"WIFI_CUSTOM",AP_CFG_RDEB_WIFI_CUSTOM_LID},
    {"GPS",AP_CFG_CUSTOM_FILE_GPS_LID},
    ← {"FileName In NvRAM",THE_LID_OF_YOUR_FILE}
};
```

**Note:**This Backup Mechanism can be triggered by Meta tool

## 6 NvRAM Library

### 6.1 NvRAM Interface For Module

- To provide related interface functions for modules to read, write NvRAM file
- Interface of opening and closing nvram file is provided by Nvram library

Interface Function	Description
NVM_GetFileDesc	Get the description of nvram file and the information of record size and number
NVM_CloseFileDesc	close the file description

- Example

```
#include "libnvram.h"

F_ID fid ;

int rec_size = 0;

int rec_num = 0;

Int your_file_lid = YOUR_FILE_LID;

bool isread = true;

YOUR_LID_STRUCT *your_lid_struct = NULL;

your_lid_struct=(YOUR_LID_STRUCT *) malloc(sizeof(YOUR_LID_STRUCT ));

If(your_lid_struct == NULL)
    return false;

fid = NVM_GetFileDesc(your_file_lid, &rec_size, &rec_num, isread );

If(fd<0)
    return false;

If(rec_size != read(fid.iFileDesc,your_lid_struct,rec_size)){
    free(your_lid_struct );
    return false;}

.....

free(your_lid_struct);

if(!NVM_CloseFileDesc(fid))
    return false ;

return true;
```

## 7.1 Sometimes you maybe just want to change structure in your customer file

- Add some item
- Delete some item
- Change some item type
- **example**

```
typedef struct _WIFI_CUSTOM_PARAM_STRUCT
{
    UINT_32          u4Resrv;          /* Reserved */
} WIFI_CUSTOM_PARAM_STRUCT;
```



```
typedef struct _WIFI_CUSTOM_PARAM_STRUCT
{
    UINT_32 u4Resv; /* Reserved */
    UINT_32 u4ResvAdd;
} WIFI_CUSTOM_PARAM_STRUCT;
```


- When first boot up, Restore data from Binregion(Nvram partition) and Reset updated file to default data
- When first boot up, Restore data from Binregion(Nvram partition) then Convert date in updated file to anything you want it to be

## 8 Step by Step to Reset data(Example)

### 8.1 Update the version of file in which data structure you want to change

- Path:mediatek\custom\[*\$PROJECT*]\cgen\inc\Custom\_Nvram\_LID.h
- Path:vendor\mediatek\proprietary\custom\[*\$PROJECT*]\cgen\inc\Custom\_Nvram\_LID.h  
(L/KK.AOSP use)

```
#define AP_CFG_RDEB_WIFI_CUSTOM_LID_VERNO "000"
```



```
#define AP_CFG_RDEB_WIFI_CUSTOM_LID_VERNO "001"
```

## 8.2 Change header file which describes the definition of data structure

- Path:mediatek\custom\[*\$PROJECT*]\cgen\cfgfileinc\CFG\_wifi\_File.h
- Path:vendor\mediatek\proprietary\custom\[*\$PROJECT*]\cgen\cfgfileinc\CFG\_wifi\_File.h (L/KK.AOSP use)

```
typedef struct _WIFI_CUSTOM_PARAM_STRUCT
{
    UINT_32          u4Resv;          /* Reserved */
} WIFI_CUSTOM_PARAM_STRUCT;
```

```
typedef struct _WIFI_CUSTOM_PARAM_STRUCT
{
    UINT_32 u4Resv; /* Reserved */
    UINT_32 u4ResvAdd;
} WIFI_CUSTOM_PARAM_STRUCT;
```

### 8.3 Change header file which define its default value of NvRAM file

- Path:mediatek\custom\[PROJECT]\cgen\cfgdefault\CFG\_WIFI\_default.h
- Path:vendor\mediatek\proprietary\custom\[PROJECT]\cgen\cfgdefault\CFG\_WIFI\_default.h (L/KK.AOSP use)

```
WIFI_CUSTOM_PARAM_STRUCT stWifiCustomDefault =
{
    0x0, // Reserved
};
```

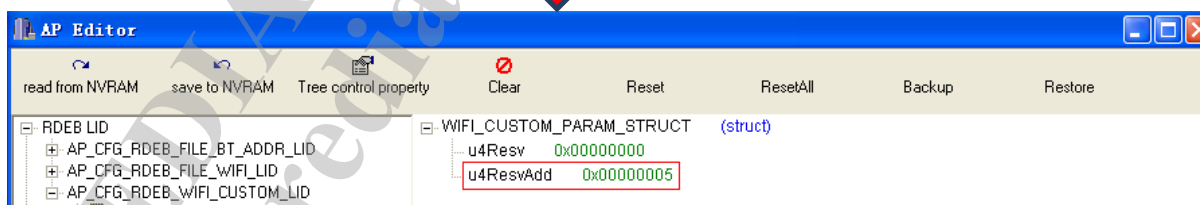
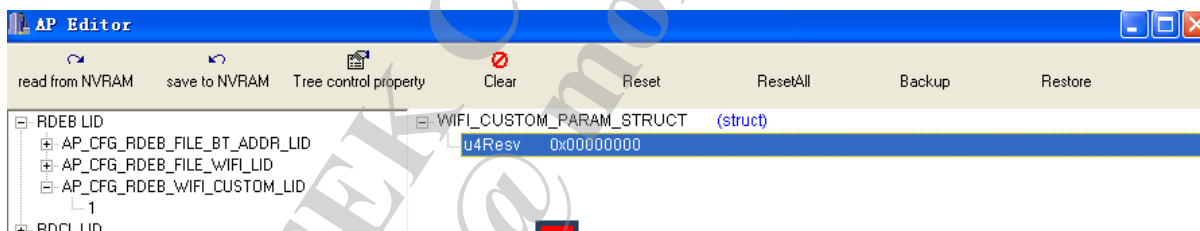


```
WIFI_CUSTOM_PARAM_STRUCT stWifiCustomDefault =
{
    0x0, // Reserved
    0x5,
};
```

### 8.4 Change the related information of NvRAM file into the definition of "g\_akCFG\_File\_Custom"

- Path:mediatek\custom\[PROJECT]\cgen\inc\CFG\_file\_info\_custom.h
- Path:vendor\mediatek\proprietary\custom\[PROJECT]\cgen\inc\CFG\_file\_info\_custom.h (L/KK.AOSP use)

### 8.5 Result of Reset



## 9 Step by Step to Convert data(Example)

### 9.1 Update the version of file in which data structure you want to change

- Path:mediatek\custom\[*\$PROJECT*]\cgen\inc\Custom\_Nvram\_LID.h
- Path:vendor\mediatek\proprietary\custom\[*\$PROJECT*]\cgen\inc\Custom\_Nvram\_LID.h (L/KK.AOSP use)

```
#define AP_CFG_RDEB_WIFI_CUSTOM_LID_VERN0 "000"
↓
#define AP_CFG_RDEB_WIFI_CUSTOM_LID_VERN0 "001"
```

### 9.2 Change header file which describes the definition of data structure and declaration of convert function

- Path:mediatek\custom\[*\$PROJECT*]\cgen\cfgfileinc\CFG\_wifi\_File.h
- Path:vendor\mediatek\proprietary\custom\[*\$PROJECT*]\cgen\cfgfileinc\CFG\_wifi\_File.h (L/KK.AOSP use)

```
typedef struct _WIFI_CUSTOM_PARAM_STRUCT
{
    UINT_32 u4Resv; /* Reserved */
} WIFI_CUSTOM_PARAM_STRUCT;
↓
typedef struct _WIFI_CUSTOM_PARAM_STRUCT
{
    UINT_32 u4Resv; /* Reserved */
    UINT_32 u4ResvAdd;
} WIFI_CUSTOM_PARAM_STRUCT;
```

### 9.3 Change header file which define its default value of NvRAM file

- Path:mediatek\custom\[*\$PROJECT*]\cgen\cfgdefault\CFG\_WIFI\_default.h
- Path:vendor\mediatek\proprietary\custom\[*\$PROJECT*]\cgen\cfgdefault\CFG\_WIFI\_default.h (L/KK.AOSP use)



```
WIFI_CUSTOM_PARAM_STRUCT stWifiCustomDefault =
{
    0x0, // Reserved
};
```



```
WIFI_CUSTOM_PARAM_STRUCT stWifiCustomDefault =
{
    0x0, // Reserved
    0x5,
};
```

### 9.4 Change the related information of NvRAM file into the definition of "g\_akCFG\_File\_Custom"

- Path:mediatek\custom\[PROJECT]\cgen\inc\CFG\_file\_info\_custom.h
- Path:vendor\mediatek\proprietary\custom\[PROJECT]\cgen\inc\CFG\_file\_info\_custom.h (L/KK.AOSP use)

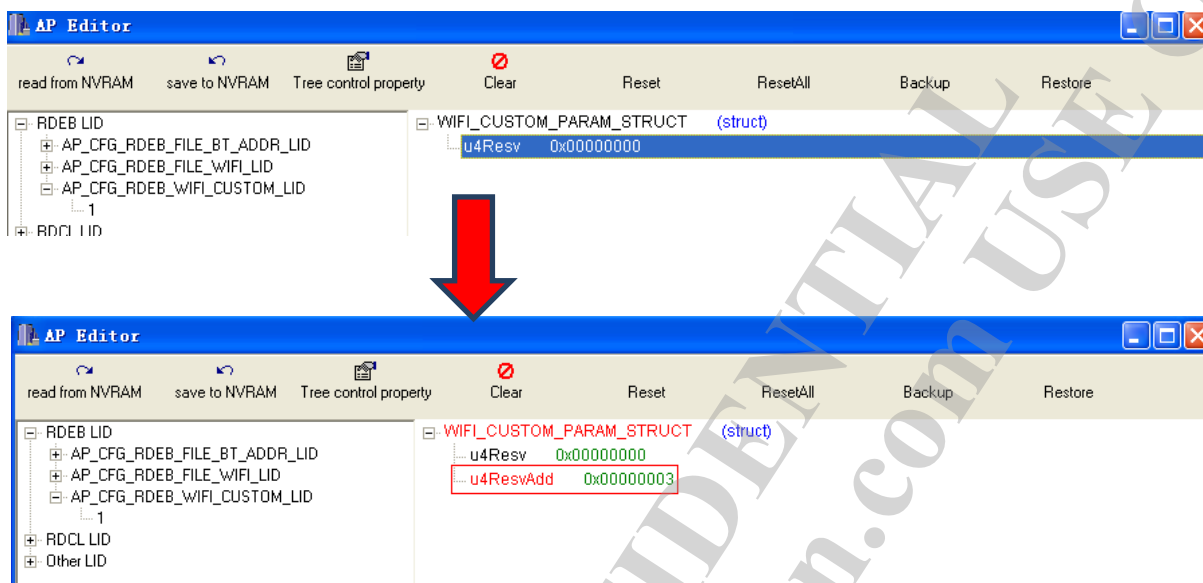
```
{
    "/data/nvram/APCFG/APRDEB/WIFI_CUSTOM", VER(AP_CFG_RDEB_WIFI_CUSTOM_LID), CFG_FILE_WIFI_CUSTOM_REC_SIZE,
    CFG_FILE_WIFI_CUSTOM_REC_TOTAL, SIGNLE_DEFAULT_REC, ((char *)&stWifiCustomDefault, DataConvert, WifiCustom_ConvertFunction)
},
```

### 9.5 Add definition of convert function in C source file

- Path:mediatek\custom\common\cgen\CFG\_file\_info.c
- Path:vendor\mediatek\proprietary\custom\[PROJECT]\cgen\CFG\_file\_info.c (L/KK.AOSP use)

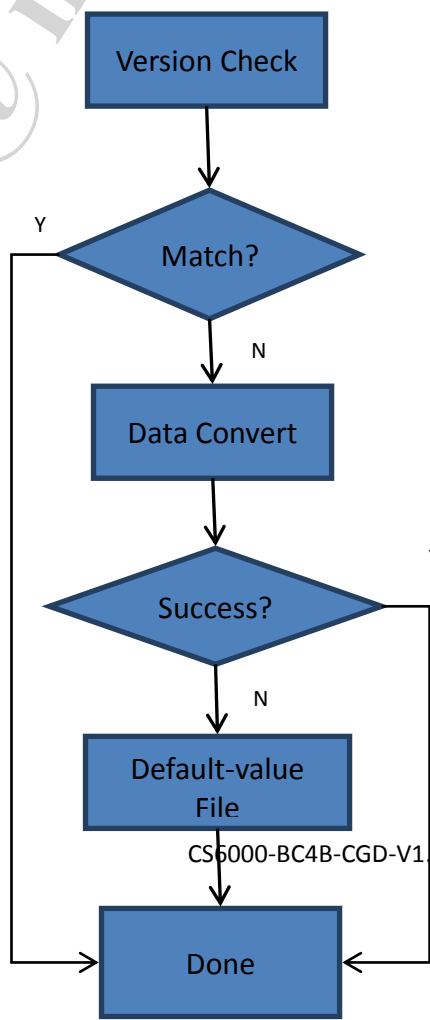
```
int WifiCustom_ConvertFunction(int CurrentVerID, int NewVerID, char* pSrcMem, char*pDstMem)
{
    if(NULL == pSrcMem || NULL == pDstMem) {
        return 0;
    }
    else if (0 == CurrentVerID && 1 == NewVerID) {
        memcpy(pDstMem, pSrcMem, sizeof(unsigned int));
        *((unsigned int*)(pDstMem + 4)) = 0x3;
        return 1;
    }
    else {
        return 0;
    }
}
```

### 9.6 Result of Convert



## 9.8 Introduction of Convert function(1/n)

Workflow:



## 9.9 Introduction of Convert function(2/n)

### ▪ Prototype

- Int XXXX\_ConvertFunction(int CurrentVerID,int NewVerID,char \*p SrcMem, char\* pDstMem)
  - *CurrentVerID: The file version which restore from BinRegion in first boot up.*
  - *NewVerID: The file version which you have updated*
  - *SrcMem: the memory saved date in file which restore from Binregion when first boot up.*
  - *pDstMem: the memory saved data which you want the file to be after convert.(must match the structure in new version)*

```

WIFI_CUSTOM_PARAM_STRUCT stWifiCustomDefault =
{
    0x0, // Reserved
};

typedef struct _WIFI_CUSTOM_PARAM_STRUCT
{
    UINT_32 u4Resv; /* Reserved */
    UINT_32 u4ResvAdd;
} WIFI_CUSTOM_PARAM_STRUCT;
    
```

## 10 Sync your thread with Nvram Daemon

### 10.1 Why your private thread should sync with nvram daemon

- After bring up, Nvram daemon will check /data/nvram folder and do some initialization. After that, other threads can access /data/nvram correctly. If other thread access /data/nvram before nvram\_daemon ready, there may bring about unpredictable results. So your thread must sync with nvram daemon.

### 10.2 How your private thread should sync with nvram daemon

- When nvram daemon is ready, it will set system variable "nvram\_init" to "Ready" by property\_set("nvram\_init", "Ready").
- You can get status of "nvram\_init" by property\_get, then check if nvram daemon is "Ready".
- When nvram daemon is "Ready", you can access /data/nvram safely. Otherwise your private thread should wait.

### 10.3 Sample Code

```
#define MAX_RETRY_COUNT 20

int read_nvram_ready_retry = 0;

while(read_nvram_ready_retry < MAX_RETRY_COUNT)
{
    read_nvram_ready_retry++;
    property_get("nvram_init", nvram_init_val, NULL);
    if(strcmp(nvram_init_val, "Ready") == 0)
    {
        break;
    }
    else
    {
        usleep(500*1000);
    }
}

NVRAM_LOG("Get nvram restore ready retry cc=%d\n", read_nvram_ready_retry);

if(read_nvram_ready_retry >= MAX_RETRY_COUNT)
```

```
{  
  
    printf("Get nvram restore ready failed\n");  
  
    NVRAM_LOG("Get nvram restore ready failed!!!\n");  
  
}
```