Confidential B

MEDIATEK

How to Write an MMI Application

Copyright © MediaTek Inc. All rights reserved.

Part 1. Add Resources

- Step 1. Add new folders and files for new application.
- Step 2. Add new applications files to make file list
- Step 3. Declare the application and its resource ID range.
- Step 4. Declare IDs of screens, strings and images within the application.
- Step 5. Add new entry into menu tree.
- Step 6. Resource Gen

Copyright © MediaTek Inc. All rights reserved.

Part2. Application Implementation

- Step 1. Write an initialization function to register event handlers.
- Step 2. Write event handler for processing received primitives
- Step 3. Write highlight handlers for function registration.
- Step 4. Write an entry function to enter the application.
- Step 5. Write an exit function to exit the application.
- Step 6. Write the logic to call between various entry functions.

Copyright © MediaTek Inc. All rights reserved.

Res Step1. New Folders and Files

- Create a new folder under \mcu\plutommi\mtkapp\
  - \mcu\plutommi\mtkapp\MyApp
  - \mcu\plutommi\mtkapp\MyApp\MyAppInc\ (header files)
    - MyAppResDef.h: resource ID of MyApp.
    - MyAppProt.h: enum, structure, constant, variables, prototypes of functions used by MyApp only.
    - MyAppGprot.h: enum, structure, constant, variables, prototypes of functions exported by MyApp.
  - \mcu\plutommi\mtkapp\MyApp\MyAppSrc\ (source files)
    - MyApp.c: implementation of MyApp.
  - In general, applications relative to GSM/GPRS would be put in \mcu\plutommi\mmi\. Other applications would be put in \mcu\plutommi\mtkapp, such as multimedia applications.

Copyright © MediaTek Inc. All rights reserved.

Res Step2. Make File List

- Add new files to make file list therefore these files would be compiled during build load process.

- Make file list folder
  - make\plutommi\conn_app, make\plutommi\inet_app, make\plutommi\media_app, make\plutommi\mmi_app, and make\plutommi\mmi_framework.

- plutommi.def
    - add Macro
- Take mmi_app as example.
  - mmi_app.inc
    - add include file path
  - mmi_app.pth
    - add file path
  - mmi_app.lis
    - add file name

Copyright © MediaTek Inc. All rights reserved.

Res Step3.
Application Resource ID Range Declaration (1/2)

- Filename: mmi_res_range_def.h
- Directory: \mcu\plutommi\mmi\inc\
- Description: declare MyApp into the resource ID range used for MyApp's in RESOURCE_BASE_ENUM.

```
RESOURCE_BASE_ENUM_BEGIN()
        …
        RESOURCE_BASE_RANGE(ALARM,   300),
        RESOURCE_BASE_RANGE(MYAPP,   200),
RESOURCE_BASE_ENUM_END()
```

Copyright © MediaTek Inc. All rights reserved.

Res Step3.
Application Resource ID Range Declaration (2/2)

- Filename: mmi_res_range_def.h
- Directory: \mcu\plutommi\mmi\inc\
- Description: define resource ID base between RESOURCE_BASE_TABLE_BEGIN() and RESOURCE_BASE_TABLE_END()

```
RESOURCE_BASE_TABLE_BEGIN()
……
#define     MYAPP_BASE          ((U16) GET_RESOURCE_BASE(MYAPP))
#define     MYAPP_BASE_MAX      ((U16) GET_RESOURCE_MAX(MYAPP))
RESOURCE_BASE_TABLE_ITEM(MYAPP)
……
RESOURCE_BASE_TABLE_END()
```

Copyright © MediaTek Inc. All rights reserved.

Res Step4.
IDs of Screens, Strings and Images Declaration

- Filename: MyAppResDef.h
- Directory: \mcu\plutommi\mtkapp\MyAppInc\
- Description: to define enumerations for screens, strings and images.

```
// Screen ID
typedef enum {
        SCR_ID_MYAPP_MAIN = (MYAPP_BASE + 1),
        SCR_ID_MYAPP_TOTAL
} myapp_screen_enum;
// String ID
typedef enum {
        STR_ID_MYAPP_MENUITEM = (MYAPP_BASE + 1),
        STR_ID_MYAPP_UNDER_CONSTRUCTION,
        STR_ID_MYAPP_TOTAL
} myapp_string_enum;
// Image ID
typedef enum {
        IMG_ID_MYAPP_MAIN = (MYAPP_BASE + 1),
        IMG_ID_MYAPP_TOTAL
} myapp_image_enum;
        …
```

Copyright © MediaTek Inc. All rights reserved.

Res Step 5. Add New Entry into Menu Tree (1/3)

- Filename: GlobalMenuItems.h
- Directory: \mcu\plutommi\mmi\inc\
- Description: to add new menu items IDs into enum "GLOBALMENUITEMSID"
- DO NOT use compile option

```
enum  GLOBALMENUITEMSID
{
        IDLE_SCREEN_MENU_ID=1,
        MAIN_MENU_PHONEBOOK_MENUID,
        MAIN_MENU_MESSAGES_MENUID,
        MAIN_MENU_FUNANDGAMES_MENUID,
        …
        MENU_ID_MYAPP_MAIN,
        …
        MAX_MENU_ITEMS_VALUE
};
```

Copyright © MediaTek Inc. All rights reserved.

Res Step 5. Add New Entry into Menu Tree (2/3)

- Filename: Res_MyApp.c
- Directory: \mcu\plutommi\Customer\CustResource\[PRJ]\Res_MMI\
  - For internal project, [PRJ] would be “plutommi”.

- Make sure the total number of children are updated.

Copyright © MediaTek Inc. All rights reserved.

Res Step 5. Add New Entry into Menu Tree (3/3)

```
    MAIN_MENU_FUNANDGAMES_MENUID,
    IDLE_SCREEN_MENU_ID,
    FUN_ENUM_TOTAL+1,
#if defined(__J2ME__)
    MENU3108_JAVA,
#endif /* !__J2ME__ */
    ……
#ifdef __TEST_MYAPP__
    MENU_ID_MYAPP_MAIN,
#endif /* __TEST_MYAPP__ */

    SHOW,
    MOVEABLEACROSSPARENT,
    1,
    MAIN_MENU_FUNANDGAMES_TEXT,
    MAIN_MENU_FUNANDGAMES_ICON));
```

Copyright © MediaTek Inc. All rights reserved.

Res Step 6. Populate Function (1/3)

- Filename: Res_MyApp.c
- Directory: \mcu\plutommi\Customer\CustResource\[PRJ]\Res_MMI\
  - For internal project, [PRJ] would be “plutommi”.
- Description: to implement functions to populate resources including strings, images, menu items used by application MyApp. Highlight handler and hint handler of menu item are also populated here.

```
void PopulateMyAppRes(void)
{
        ADD_APPLICATION_STRING2(…);
        ADD_APPLICATION_IMAGE2(…);
        ADD_APPLICATION_MENUITEM2(…);
        ADD_APPLICATION_MENUITEM_HILITE_HANDLER (…);
        ADD_APPLICATION_MENUITEM_HINT_HANDLER (…);
}
```

Copyright © MediaTek Inc. All rights reserved.

Res Step 6. Populate Function (2/3)

- Filename: PopulateRes.h
- Directory: \mcu\plutommi\Customer\CustomerInc\
- Description: to declare the definition of MyApp's resource population function here.

```
…
#ifdef __TEST_MYAPP__
extern void PopulateMyAppRes(void);
#endif
…
```

Copyright © MediaTek Inc. All rights reserved.

Res Step 6. Populate Function (3/3)

- Filename: PopulateRes.c
- Directory: \mcu\plutommi\MMI\Resource\
- Description: to add extern reference of MyApp's resource population function here and call it inside function

```
…
#ifdef __TEST_MYAPP__
extern void PopulateMyAppRes(void);
#endif
…
void PopulateResData(void)
{
        …
    #ifdef __TEST_MYAPP__
        PopulateMyAppRes();
    #endif
        …
}
```

Copyright © MediaTek Inc. All rights reserved.

Res Step 6. Resource Gen Tips

- Path of resource ID declaration file (e.g. MyAppDef.h) should be added to this file: plutommi\Customer\ResGenerator\Makefile
  - Add one line:
    - -I “\mcu\plutommi\mtkapp\MyAppInc” \

Copyright © MediaTek Inc. All rights reserved.

AP Step 1. Initialization Function (1/3)

- Filename: MyApp.c
- Directory: \mcu\plutommi\mtkapp\MyAppSrc
- Description: When the phone boots up, the application has to register its event handlers.

- Example

```
void mmi_myapp_init(void)
{
        /* Set Event Handler, for example : for Incoming Call */
        SetProtocolEventHandler(psCBackCallIncoming, PRT_INCOMINGCALL_EVENT);

        ……
};
```

Copyright © MediaTek Inc. All rights reserved.

AP Step 1. Initialization Function (2/3)

- Event Handler
  - to implement the event handler that would be called while MMI task receives the correspondent primitive.
  - registered in application initialization function or registered when needed.

Copyright © MediaTek Inc. All rights reserved.

Slide 17

llh1 这页是把原来后面的对eventhandle and highlight handle 的描述综合在这一页
llh, 12/18/2006

AP Step 1. Initialization Function (3/3)

- The initialization function of the application should be called from InitializeAll() function if the application has to be initialized whether there is SIM inserted or not.
- If the application is not required when there is no SIM, it should be called from InitAllApplications().

```
void InitializeAll(void)
{
        …
    #ifdef __TEST_MYAPP__
        mmi_myapp_init ();
    #endif
        …
}
```

Copyright © MediaTek Inc. All rights reserved.

AP Step 3. Highlight Handler

- Filename: MyApp.c
- Directory: \mcu\plutommi\mtkapp\MyAppSrc\
- Description: to implement the highlight function that would be called while menu item is highlighted
- Please use ADD_APPLICATION_MENUITEM_HILITE_HANDLER add your highlight handler in res_xxx.c

```
void mmi_myapp_highlight_menu(void)
{
        /*1 Change left soft key icon and label */
        ChangeLeftSoftkey(STR_GLOBAL_OK, 0);

        /*2 Change right soft key icon and label */
        ChangeRightSoftkey(STR_GLOBAL_BACK,0);

        /*3 Register function for left soft key */
        SetLeftSoftkeyFunction(mmi_myapp_entry_menu,KEY_EVENT_UP);

        /*4 Register function for right soft key */
        SetRightSoftkeyFunction(GoBackHistory,KEY_EVENT_UP);
}
```

Copyright © MediaTek Inc. All rights reserved.

AP Step 4. Entry Function

- Filename: MyApp.c
- Directory: \mcu\plutommi\mtkapp\MyAppSrc\
- Description: When the user selects an menu item, the highlight handler registers the entry function of that item, which will be called if the user presses the correspondent key .
- General procedure
  - Save the contents of previous screen.
  - Get GUI buffer of current screen to display the contents of screen.
  - Retrieve number of sub-menu items to be displayed.
  - Get display attribute for the menu item to be displayed, i.e. item to be displayed as lists, matrix, or circular menu etc.
  - Set parent ID of current menu items.
  - Register highlight handler to be called in menu screen.
  - Call ShowCategoryxxxScreen to display screen.
  - Register function with left and right soft key.

Copyright © MediaTek Inc. All rights reserved.

```
void mmi_myapp_entry_menu( void)
{
/*1 EntryNewScreen would call the exit handler of the previous screen and register exit and entry handler for this screen . This must be the first function to be called for all entry functions. */
EntryNewScreen(SCR_ID_MYAPP_MAIN , mmi_myapp_exit_menu, mmi_myapp_entry_menu, NULL)
/*2 Get the GUI buffer of the current screen stored in the history.*/
guiBuffer = GetCurrGuiBuffer (SCR_ID_MYAPP_MAIN );
/*3. Retrieve the number of child of menu item to be displayed, if this entry function is to display a list menu. */
nNumofItem = GetNumOfChild (MENU_ID_MYAPP_01);
/* 4. Get attribute of this menu item to be displayed */
nDispAttribute = GetDispAttributeOfItem (MENU_ID_MYAPP_01);
/* 5. Retrieve string ids, in sequence, of given menu item to be displayed. The sequence of strings is defined as per the registration of string items in the populate function.*/
GetSequenceStringIds (MENU_ID_MYAPP_01, nStrItemList);
```

Copyright © MediaTek Inc. All rights reserved.

```
/* 6 Set the current menu items' parent Id, so that the framework knows where this menu item appears*/
SetParentHandler (MENU_ID_MYAPP_01);
/*7 Register highlight handler to be called in menu screen */
RegisterHighlightHandler (ExecuteCurrHiliteHandler);
/*8 Display Category1 Screen by passing all the information collected in above steps.*/
ShowCategory1Screen(STR_ID_MyApp_CAPTION, IMG_ID_MyApp_CAPTION,
                    STR_ID_MyApp_LSK, IMG_ID_MyApp_LSK,
                    STR_ID_MyApp_RSK, IMG_ID_MyApp_RSK,
                     nNumofItem, nStrItemList, guiBuffer);
/*9.Register function with right softkey*/
SetRightSoftkeyFunction (GoBackHistory, KEY_EVENT_UP)
```

Copyright © MediaTek Inc. All rights reserved.

AP Step 5. Exit Function

- Filename: MyApp.c
- Directory: \mcu\plutommi\mtkapp\MyAppSrc\
- Description:When a screen exits, the function is called back, which typically saves state of the current screen or release resource.
- Note. If in entry function, EntryNewScreen is called with parameter entry function, which is the 3rd parameter of EntryNewScreen, MMI framework would automatically save the screen data to history when the screen is covered by other screen, therefore we do not need to save the screen history in exit function by ourselves.

```
void mmi_myapp_exit_menu(void)
{
   /* mainly release the malloced memory,etc */
}
```

Copyright © MediaTek Inc. All rights reserved.

MEDIATEK

MEDIATEK Confidential Release for

www.mediatek.com





Copyright © MediaTek Inc. All rights reserved.