**[ BOCA Dialog 說明文檔]**

1. Progressbar dialog

2. Alarm dialog

3. Text dialog

4. Confirmation dialog

5. Dialog with input field

6. Dialog softkey 設定方式

7. References

**1. Progressbar dialog**

(1) Open window by text ptr

PUBLIC void MMIPUB\_OpenProgressWinByTextPtr(

MMI\_STRING\_T \*text1\_ptr, //text 1

MMI\_WIN\_ID\_T \*win\_id\_ptr, //win id,default is MMIPUB\_ALERT\_WIN\_ID

MMI\_WIN\_PRIORITY\_E \*win\_priority\_ptr, //win priority,default is WIN\_ONE\_LEVEL

MMIPUB\_SOFTKEY\_STYLE\_E softkey\_style, //softkey style

MMIPUB\_HANDLE\_FUNC handle\_func //handle function

);

(2) Open window by text id

PUBLIC void MMIPUB\_OpenProgressWinByTextId(

MMI\_TEXT\_ID\_T text1\_id, //text 1,no default TXT\_NULL

MMI\_WIN\_ID\_T \*win\_id\_ptr, //win id,default is MMIPUB\_ALERT\_WIN\_ID

MMI\_WIN\_PRIORITY\_E \*win\_priority\_ptr, //win priority,default is WIN\_ONE\_LEVEL

MMIPUB\_SOFTKEY\_STYLE\_E softkey\_style, //softkey style

MMIPUB\_HANDLE\_FUNC handle\_func //handle function

);

(3) Set total size

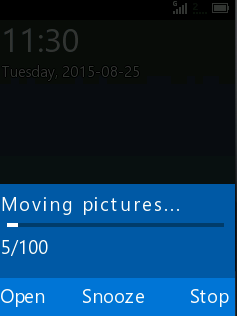
PUBLIC void MMIPUB\_SetProgressTotalSize(MMI\_WIN\_ID\_T \*win\_id\_ptr, uint32 total\_size);

(4) Set finished size

PUBLIC void MMIPUB\_UpdateProgressBarEx(MMI\_WIN\_ID\_T \*win\_id\_ptr, uint32 finished\_size, BOOLEAN is\_flash );

[Sample code]

範例代碼圖示



範例代碼

|  |
| --- |
| PUBLIC MMI\_HANDLE\_T MMIHELLO\_CreateTestProgressDialogWin(void)  {  MMI\_STRING\_T progress\_str = {0};  wchar progress\_text[] = L"Moving pictures...";  MMIPUB\_DIALOG\_SOFTKEY\_T dialog\_softkey = {0};  progress\_str.wstr\_ptr = progress\_text;  progress\_str.wstr\_len = MMIAPICOM\_Wstrlen(progress\_text);  MMIPUB\_OpenProgressWinByTextPtr(&progress\_str, PNULL, PNULL, MMIPUB\_SOFTKEY\_DIALOG\_ALARM, HandleProgressbarDialogWindow);  } |
| LOCAL MMI\_RESULT\_E HandleProgressbarDialogWindow(  MMI\_WIN\_ID\_T win\_id,  MMI\_MESSAGE\_ID\_E msg\_id,  DPARAM param  )  {  MMI\_RESULT\_E recode = MMI\_RESULT\_TRUE;  MMI\_STRING\_T title\_str = {0};  wchar title\_text[] = L"Dialog Title";    title\_str.wstr\_ptr = title\_text;  title\_str.wstr\_len = MMIAPICOM\_Wstrlen(title\_text);  switch(msg\_id)  {  case MSG\_OPEN\_WINDOW:  {  MMIPUB\_SetProgressTotalSize(&win\_id,100);  MMIPUB\_UpdateProgressBarEx(&win\_id,5,TRUE);  }  break;  case MSG\_CTL\_MIDSK:  case MSG\_APP\_WEB:  case MSG\_APP\_OK:  case MSG\_CTL\_OK:  case MSG\_CTL\_CANCEL:  case MSG\_APP\_CANCEL:  MMK\_CloseWin(win\_id);  break;  case MSG\_CLOSE\_WINDOW:  default:  recode = MMIPUB\_HandleProgressWinMsg(win\_id, msg\_id, param);  break;  }  return (recode);  } |

**2. Progressbar dialog – waiting indicator**

(1) Open window by text ptr

PUBLIC void MMIPUB\_OpenProgressWaitingIndWinByTextPtr(

MMI\_STRING\_T \*text1\_ptr, //text 1

MMI\_WIN\_ID\_T \*win\_id\_ptr, //win id,default is MMIPUB\_ALERT\_WIN\_ID

MMI\_WIN\_PRIORITY\_E \*win\_priority\_ptr, //win priority,default is WIN\_ONE\_LEVEL

MMIPUB\_SOFTKEY\_STYLE\_E softkey\_style, //softkey style

MMIPUB\_HANDLE\_FUNC handle\_func //handle function

);

(2) Open window by text ptr

PUBLIC void MMIPUB\_OpenProgressWaitingIndWinByTextId(

MMI\_TEXT\_ID\_T text1\_id, //text 1,no default TXT\_NULL

MMI\_WIN\_ID\_T \*win\_id\_ptr, //win id,default is MMIPUB\_ALERT\_WIN\_ID

MMI\_WIN\_PRIORITY\_E \*win\_priority\_ptr, //win priority,default is WIN\_ONE\_LEVEL

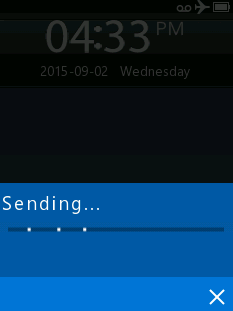
MMIPUB\_SOFTKEY\_STYLE\_E softkey\_style, //softkey style

MMIPUB\_HANDLE\_FUNC handle\_func //handle function

);

[Sample code]

範例代碼圖示



範例代碼

|  |
| --- |
| PUBLIC void TestProgressDialogWaitingIndWin(void)  {  MMI\_STRING\_T progress\_str = {0};  wchar progress\_text[] = L"Sending...";    progress\_str.wstr\_ptr = progress\_text;  progress\_str.wstr\_len = MMIAPICOM\_Wstrlen(progress\_text);  MMIPUB\_OpenProgressWaitingIndWinByTextPtr(&progress\_str, PNULL, PNULL, MMIPUB\_SOFTKEY\_DIALOG\_CLOSE, PNULL);  } |

**3. Alarm dialog**

(1) Open window by text ptr

PUBLIC void MMIPUB\_OpenAlarmDialogWin(

uint8 text\_num, //text number

MMI\_STRING\_T \*text1\_ptr, //text 1

MMI\_STRING\_T \*text2\_ptr, //text 2

MMI\_STRING\_T \*text3\_ptr, //text 3

MMI\_WIN\_ID\_T win\_id, //win id

MMI\_IMAGE\_ID\_T image\_id, //image id

MMI\_ANIM\_ID\_T anim\_id, //anim id

MMI\_WIN\_PRIORITY\_E win\_priority, //priority

MMIPUB\_SOFTKEY\_STYLE\_E softkey\_style, //softkey style

MMIPUB\_HANDLE\_FUNC handle\_func //handle function

);

**4. Text dialog**

(1) Open window by text ptr

PUBLIC void MMIPUB\_OpenTextDialogByTextPtr(

uint32 \*time\_period\_ptr, //time period,default is 3s

MMI\_STRING\_T \*text1\_ptr, //text 1

MMI\_STRING\_T \*text2\_ptr, //text 2

MMI\_IMAGE\_ID\_T image\_id, //alert image id

MMI\_WIN\_ID\_T \*win\_id\_ptr, //win id,default is MMIPUB\_ALERT\_WIN\_ID

MMI\_WIN\_PRIORITY\_E \*win\_priority\_ptr, //win priority,default is WIN\_ONE\_LEVEL

MMIPUB\_SOFTKEY\_STYLE\_E softkey\_style, //softkey style

MMIPUB\_HANDLE\_FUNC handle\_func, //handle function

);

(2) Open window by text id

PUBLIC void MMIPUB\_OpenTextDialogByTextId(

uint32 \*time\_period\_ptr, //time period,default is 2s

MMI\_TEXT\_ID\_T text1\_id, //text 1,no default TXT\_NULL

MMI\_TEXT\_ID\_T text2\_id, //text 2,no default TXT\_NULL

MMI\_IMAGE\_ID\_T image\_id, //alert image id

MMI\_WIN\_ID\_T \*win\_id\_ptr, //win id,default is MMIPUB\_ALERT\_WIN\_ID

MMI\_WIN\_PRIORITY\_E \*win\_priority\_ptr, //win priority,default is WIN\_ONE\_LEVEL

MMIPUB\_SOFTKEY\_STYLE\_E softkey\_style, //softkey style

MMIPUB\_HANDLE\_FUNC handle\_func, //handle function

);

[Set title]

PUBLIC void MMIPUB\_SetWinTitleText(

MMI\_WIN\_ID\_T win\_id,

MMI\_STRING\_T \*text\_ptr,

BOOLEAN is\_fresh

);

**5. Confirmation dialog**

(1) Open window by text ptr

PUBLIC void MMIPUB\_OpenConfirmationDialogByTextPtr(

uint32 \*time\_period\_ptr, //time period,default is 3s

MMI\_STRING\_T \*text1\_ptr, //text 1

MMI\_STRING\_T \*text2\_ptr, //text 2

MMI\_IMAGE\_ID\_T image\_id, //alert image id

MMI\_WIN\_ID\_T \*win\_id\_ptr, //win id,default is MMIPUB\_ALERT\_WIN\_ID

MMI\_WIN\_PRIORITY\_E \*win\_priority\_ptr, //win priority,default is WIN\_ONE\_LEVEL

MMIPUB\_SOFTKEY\_STYLE\_E softkey\_style, //softkey style

MMIPUB\_HANDLE\_FUNC handle\_func //handle function

);

(2) Open window by text id

PUBLIC void MMIPUB\_OpenConfirmationDialogByTextId(

uint32 \*time\_period\_ptr, //time period,default is 2s

MMI\_TEXT\_ID\_T text1\_id, //text 1,no default TXT\_NULL

MMI\_TEXT\_ID\_T text2\_id, //text 2,no default TXT\_NULL

MMI\_IMAGE\_ID\_T image\_id, //alert image id

MMI\_WIN\_ID\_T \*win\_id\_ptr, //win id,default is MMIPUB\_ALERT\_WIN\_ID

MMI\_WIN\_PRIORITY\_E \*win\_priority\_ptr, //win priority,default is WIN\_ONE\_LEVEL

MMIPUB\_SOFTKEY\_STYLE\_E softkey\_style, //softkey style

MMIPUB\_HANDLE\_FUNC handle\_func //handle function

);

(3) Open window by text ptr (for applet)

PUBLIC void MMIPUB\_OpenConfirmationDialogWinByTextPtrEx(

MMI\_HANDLE\_T applet\_handle,

uint32 \*time\_period\_ptr, //time period,default is 3s

MMI\_STRING\_T \*text1\_ptr, //text 1

MMI\_STRING\_T \*text2\_ptr, //text 2

MMI\_IMAGE\_ID\_T image\_id, //alert image id

MMI\_WIN\_ID\_T \*win\_id\_ptr, //win id,default is MMIPUB\_ALERT\_WIN\_ID

MMI\_WIN\_PRIORITY\_E \*win\_priority\_ptr, //win priority,default is WIN\_ONE\_LEVEL

MMIPUB\_SOFTKEY\_STYLE\_E softkey\_style, //softkey style

MMIPUB\_HANDLE\_FUNC handle\_func, //handle function

uint32 user\_data //user\_data

);

(4) Open window by text id (for applet)

PUBLIC void MMIPUB\_OpenConfirmationDialogWinByTextIdEx(

MMI\_HANDLE\_T applet\_handle,

uint32 \*time\_period\_ptr, //time period,default is 2s

MMI\_TEXT\_ID\_T text1\_id, //text 1,no default TXT\_NULL

MMI\_TEXT\_ID\_T text2\_id, //text 2,no default TXT\_NULL

MMI\_IMAGE\_ID\_T image\_id, //alert image id

MMI\_WIN\_ID\_T \*win\_id\_ptr, //win id,default is MMIPUB\_ALERT\_WIN\_ID

MMI\_WIN\_PRIORITY\_E \*win\_priority\_ptr, //win priority,default is WIN\_ONE\_LEVEL

MMIPUB\_SOFTKEY\_STYLE\_E softkey\_style, //softkey style

MMIPUB\_HANDLE\_FUNC handle\_func, //handle function

uint32 user\_data //user\_data

);

**6. Dialog with input field**

(1) Open window by text ptr

PUBLIC void MMIPUB\_OpenDialogInputFieldWinByTextPtr(

MMI\_STRING\_T \*text1\_ptr, //text1\_ptr for title

MMI\_WIN\_ID\_T \*win\_id\_ptr, //win id,default is MMIPUB\_ALERT\_WIN\_ID

MMI\_WIN\_PRIORITY\_E \*win\_priority\_ptr, //win priority,default is WIN\_ONE\_LEVEL

MMIPUB\_SOFTKEY\_STYLE\_E softkey\_style, //softkey style

MMIPUB\_HANDLE\_FUNC handle\_func //handle function

);

(2) Open window by text ptr (for applet)

PUBLIC void MMIPUB\_OpenDialogInputFieldWinByTextPtrEx(

MMI\_HANDLE\_T applet\_handle,

MMI\_STRING\_T \*text1\_ptr, //text 1

MMI\_WIN\_ID\_T \*win\_id\_ptr, //win id,default is MMIPUB\_ALERT\_WIN\_ID

MMI\_WIN\_PRIORITY\_E \*win\_priority\_ptr, //win priority,default is WIN\_ONE\_LEVEL

MMIPUB\_SOFTKEY\_STYLE\_E softkey\_style, //softkey style

MMIPUB\_HANDLE\_FUNC handle\_func, //handle function

uint32 user\_data //user\_data

)

(3) Get input field ctrl id

PUBLIC MMI\_CTRL\_ID\_T MMIPUB\_GetInputFieldDialogCtrlId(

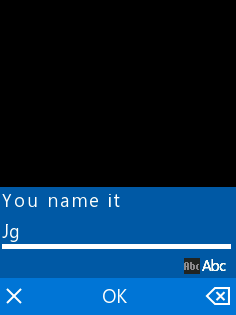
MMI\_WIN\_ID\_T win\_id

);

[Sample code]

1. 開啟預設窗口

範例代碼圖示



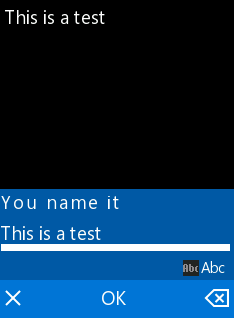
範例代碼

|  |
| --- |
| PUBLIC void MMIHELLO\_CreateDialogInputFieldWin(void)  {  MMI\_STRING\_T title\_str = {0};  wchar title\_text[] = L"You name it";  //DIALOG\_CUSTOM\_SOFTKEY init\_data;  GUI\_LCD\_DEV\_INFO lcd\_dev\_info\_ptr = {GUI\_MAIN\_LCD\_ID, GUI\_BLOCK\_MAIN};  GUI\_RECT\_T win\_rect = {0};  uint16 logic\_width = 0;  uint16 logic\_height = 0;  MMIPUB\_DIALOG\_SOFTKEY\_T dialog\_softkey = {0};    title\_str.wstr\_ptr = title\_text;  title\_str.wstr\_len = MMIAPICOM\_Wstrlen(title\_text);  MMIPUB\_OpenDialogInputFieldWinByTextPtr(&title\_str, PNULL, PNULL, PNULL, PNULL);  } |

2. 設定預設字串與取得輸入字串

- Set/Get editbox string

範例代碼圖示



範例代碼

|  |
| --- |
| PUBLIC void MMIHELLO\_CreateDialogInputFieldWin(void)  {  MMI\_STRING\_T title\_str = {0};  wchar title\_text[] = L"You name it";  //DIALOG\_CUSTOM\_SOFTKEY init\_data;  GUI\_LCD\_DEV\_INFO lcd\_dev\_info\_ptr = {GUI\_MAIN\_LCD\_ID, GUI\_BLOCK\_MAIN};  GUI\_RECT\_T win\_rect = {0};  uint16 logic\_width = 0;  uint16 logic\_height = 0;  MMIPUB\_DIALOG\_SOFTKEY\_T dialog\_softkey = {0};  title\_str.wstr\_ptr = title\_text;  title\_str.wstr\_len = MMIAPICOM\_Wstrlen(title\_text);  MMIPUB\_OpenDialogInputFieldWinByTextPtr(&title\_str, PNULL, PNULL, PNULL, HandleInputFieldDialogWindow);  } |
| LOCAL MMI\_RESULT\_E HandleInputFieldDialogWindow(  MMI\_WIN\_ID\_T win\_id,  MMI\_MESSAGE\_ID\_E msg\_id,  DPARAM param  )  {  MMI\_RESULT\_E recode = MMI\_RESULT\_TRUE;  wchar input\_default\_text[] = L"This is a test";  GUI\_LCD\_DEV\_INFO lcd\_dev\_info\_ptr = {GUI\_MAIN\_LCD\_ID, GUI\_BLOCK\_MAIN};  GUI\_RECT\_T win\_rect = {0};  GUI\_RECT\_T text\_rect = {0};  uint16 logic\_width = 0;  uint16 logic\_height = 0;  GUISTR\_STYLE\_T text\_style = {0};  MMI\_CTRL\_ID\_T input\_field\_ctrl\_id = MMIPUB\_GetInputFieldDialogCtrlId(win\_id);  MMI\_STRING\_T inputstr = {0};  switch(msg\_id)  {    case MSG\_OPEN\_WINDOW:  {  GUIEDIT\_SetString(input\_field\_ctrl\_id,input\_default\_text,100);  }  break;    case MSG\_CTL\_MIDSK:  {  GUIEDIT\_GetString(input\_field\_ctrl\_id,&inputstr);    text\_style.align = ALIGN\_LEFT;  text\_style.font = SONG\_FONT\_20;  text\_style.font\_color = MMI\_WHITE\_COLOR;    GUILCD\_GetLogicWidthHeight(lcd\_dev\_info\_ptr.lcd\_id, &logic\_width, &logic\_height);  text\_rect.left = 10;  text\_rect.right = text\_rect.left + logic\_width;  text\_rect.top = 10;  text\_rect.bottom = text\_rect.top + logic\_height;  GUISTR\_DrawTextToLCDInRect((const GUI\_LCD\_DEV\_INFO \*)&lcd\_dev\_info\_ptr, &text\_rect, &text\_rect, &inputstr, &text\_style, GUISTR\_STATE\_ALIGN | GUISTR\_STATE\_SINGLE\_LINE | GUISTR\_STATE\_ELLIPSIS, GUISTR\_TEXT\_DIR\_AUTO);  }  break;    case MSG\_APP\_OK:  case MSG\_CTL\_OK:  case MSG\_CTL\_CANCEL:  case MSG\_APP\_CANCEL:  case MSG\_CLOSE\_WINDOW:  MMK\_CloseWin(win\_id);  break;    default:  recode = MMIPUB\_HandleDialogInputFieldWinMsg(win\_id, msg\_id, param);  break;  }  return (recode);  } |

**7. Dialog with password editor**

(1) Open window by text ptr

PUBLIC void MMIPUB\_OpenDialogPasswordWinByTextPtr(

MMI\_STRING\_T \*text1\_ptr, //text 1

MMI\_WIN\_ID\_T \*win\_id\_ptr, //win id,default is MMIPUB\_ALERT\_WIN\_ID

MMI\_WIN\_PRIORITY\_E \*win\_priority\_ptr, //win priority,default is WIN\_ONE\_LEVEL

MMIPUB\_SOFTKEY\_STYLE\_E softkey\_style, //softkey style

MMIPUB\_HANDLE\_FUNC handle\_func //handle function

);

(2) Open window by text ptr (for applet)

PUBLIC void MMIPUB\_OpenDialogPasswordWinByTextPtrEx(

MMI\_HANDLE\_T applet\_handle,

MMI\_STRING\_T \*text1\_ptr, //text 1

MMI\_WIN\_ID\_T \*win\_id\_ptr, //win id,default is MMIPUB\_ALERT\_WIN\_ID

MMI\_WIN\_PRIORITY\_E \*win\_priority\_ptr, //win priority,default is WIN\_ONE\_LEVEL

MMIPUB\_SOFTKEY\_STYLE\_E softkey\_style, //softkey style

MMIPUB\_HANDLE\_FUNC handle\_func, //handle function

uint32 user\_data //user\_data

);

(3) Get password editor ctrl id

PUBLIC MMI\_CTRL\_ID\_T MMIPUB\_GetPasswordDialogCtrlId(

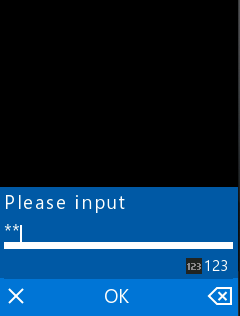
MMI\_WIN\_ID\_T win\_id

);

[Sample code]

1. 開啟預設窗口

範例代碼圖示



範例代碼

|  |
| --- |
| PUBLIC void MMIHELLO\_CreateDialogPasswordWin(void)  {  MMI\_STRING\_T title\_str = {0};  wchar title\_text[] = L"Please input:";  //DIALOG\_CUSTOM\_SOFTKEY init\_data;  GUI\_LCD\_DEV\_INFO lcd\_dev\_info\_ptr = {GUI\_MAIN\_LCD\_ID, GUI\_BLOCK\_MAIN};  GUI\_RECT\_T win\_rect = {0};  uint16 logic\_width = 0;  uint16 logic\_height = 0;  MMIPUB\_DIALOG\_SOFTKEY\_T dialog\_softkey = {0};    title\_str.wstr\_ptr = title\_text;  title\_str.wstr\_len = MMIAPICOM\_Wstrlen(title\_text);  MMIPUB\_OpenDialogPasswordWinByTextPtr(&title\_str, PNULL, PNULL, PNULL, PNULL);  } |

**5. Dialog with date editor**

(1) Open window by text ptr

PUBLIC void MMIPUB\_OpenDialogDateEditorWinByTextPtr(

MMI\_STRING\_T \*text1\_ptr, //text 1

MMI\_WIN\_ID\_T \*win\_id\_ptr, //win id,default is MMIPUB\_ALERT\_WIN\_ID

MMI\_WIN\_PRIORITY\_E \*win\_priority\_ptr, //win priority,default is WIN\_ONE\_LEVEL

MMIPUB\_SOFTKEY\_STYLE\_E softkey\_style, //softkey style

MMIPUB\_HANDLE\_FUNC handle\_func //handle function

)

(2) Open window by text ptr (for applet)

PUBLIC void MMIPUB\_OpenDialogDateEditorWinByTextPtrEx(

MMI\_HANDLE\_T applet\_handle,

MMI\_STRING\_T \*text1\_ptr, //text 1

MMI\_WIN\_ID\_T \*win\_id\_ptr, //win id,default is MMIPUB\_ALERT\_WIN\_ID

MMI\_WIN\_PRIORITY\_E \*win\_priority\_ptr, //win priority,default is WIN\_ONE\_LEVEL

MMIPUB\_SOFTKEY\_STYLE\_E softkey\_style, //softkey style

MMIPUB\_HANDLE\_FUNC handle\_func, //handle function

uint32 user\_data //user\_data

)

(3) Get password editor ctrl id

PUBLIC MMI\_CTRL\_ID\_T MMIPUB\_GetDateDialogCtrlId(

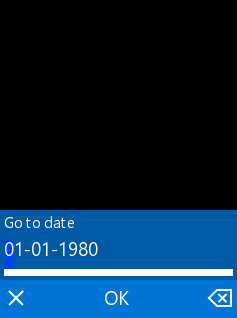
MMI\_WIN\_ID\_T win\_id

);

[Sample code]

1. 開啟預設窗口

範例代碼圖示



範例代碼

|  |
| --- |
| PUBLIC void MMIHELLO\_CreateDialogDateEditorWin(void)  {  MMI\_STRING\_T title\_str = {0};  wchar title\_text[] = L"Go to date";  GUI\_LCD\_DEV\_INFO lcd\_dev\_info\_ptr = {GUI\_MAIN\_LCD\_ID, GUI\_BLOCK\_MAIN};  GUI\_RECT\_T win\_rect = {0};  uint16 logic\_width = 0;  uint16 logic\_height = 0;  MMIPUB\_DIALOG\_SOFTKEY\_T dialog\_softkey = {0};    title\_str.wstr\_ptr = title\_text;  title\_str.wstr\_len = MMIAPICOM\_Wstrlen(title\_text);  MMIPUB\_OpenDialogDateEditorWinByTextPtr(&title\_str, PNULL, PNULL, PNULL, PNULL);  } |

**6. Dialog softkey 設定方式**

1. 在開啟窗口時設定softkey\_style,如果傳入PNULL或傳入非dialog之softkey\_style則會設定為預設風格。

MMIPUB\_OpenProgressWinByTextPtr(&progress\_str, PNULL, PNULL, MMIPUB\_SOFTKEY\_DIALOG\_CLOSE,

HandleProgressbarDialogWindow);

可供設置之風格如下:

MMIPUB\_SOFTKEY\_DIALOG\_NONE

MMIPUB\_SOFTKEY\_DIALOG\_OK

MMIPUB\_SOFTKEY\_DIALOG\_ALARM

MMIPUB\_SOFTKEY\_DIALOG\_CLOSE

MMIPUB\_SOFTKEY\_DIALOG\_CLOSE\_OK\_RETURN

MMIPUB\_SOFTKEY\_DIALOG\_OK\_CLOSE

2. 客製化softkey

(接口)

PUBLIC void MMIPUB\_SetDialogSoftkey(

MMI\_WIN\_ID\_T \*win\_id\_ptr,

DIALOG\_TYPE dialog\_type,

MMIPUB\_DIALOG\_SOFTKEY\_T \*dialog\_softkey

);

(使用方式)

在開啟窗口後設置MMIPUB\_DIALOG\_SOFTKEY\_T型態變數,並調用接口MMIPUB\_SetDialogSoftkey。

|  |
| --- |
| index - 左鍵為0, 中鍵為1, 右鍵為2  content - text\_id 或image\_id  type - DIALOG\_SOFTKEY\_TEXT\_ID 或DIALOG\_SOFTKEY\_IMAGE\_ID |

(範例)

|  |
| --- |
| MMIPUB\_DIALOG\_SOFTKEY\_T dialog\_softkey = {0};  MMIPUB\_OpenProgressWinByTextPtr(&progress\_str, PNULL, PNULL, PNULL, HandleProgressbarDialogWindow);    dialog\_softkey.index = 1;  dialog\_softkey.content = TXT\_COMMON\_OK;  dialog\_softkey.type = DIALOG\_SOFTKEY\_TEXT\_ID;  MMIPUB\_SetDialogSoftkey(PNULL,DIALOG\_WITH\_PROGRESSBAR,&dialog\_softkey);    dialog\_softkey.index = 2;  dialog\_softkey.content = IMAGE\_PRIMO\_COMMON\_CANCEL;  dialog\_softkey.type = DIALOG\_SOFTKEY\_IMAGE\_ID;  MMIPUB\_SetDialogSoftkey(PNULL,DIALOG\_WITH\_PROGRESSBAR,&dialog\_softkey); |

(注意項目)

- MMIPUB\_DIALOG\_SOFTKEY\_T型態變數的content與type需相符合,否則會造成無法顯示softkey或顯示異常.

- MMIPUB\_SetDialogSoftkey接口的調用需在窗口開啟後,否則設定無效。

- 接口MMIPUB\_SetDialogSoftkey若第一個參數win\_id\_ptr沒有設置,將依據第二個參數使用窗口類型設定預設win\_id。

- MMIPUB\_SOFTKEY\_DIALOG\_NONE 為無softkey的情況,布局不會有softkey bar。

**7. References**

Please refer to sample code for details.

- sample\_dialog\_ctrl.c

- sample\_dialog\_ctrl.h