

修改 1 个接口

```
/**
 * Description : 增加一个动态 menu
 * Global resource dependence :
 * Author:jin.wang
 * Note:成功返回 TRUE，失败返回 FALSE
 */
PUBLIC MMIMENU_SYN_VALUE_E MMIMENU_DyAppendItem(
    GUIMAINMENU_DYNAMIC_INFO_T *dynamic_item_ptr,
    GUIMENU_FILE_DEVICE_E store_idx,
    BOOLEAN is_set_menu_id
);
```

【函数功能】

主菜单增加一个动态菜单项

【参数说明】

is_set_menu_id: 菜单项 menu_id 是否由系统自动生成(TRUE: 自动生成, FALSE: 应用传入)

【返回值】

增加结果, 请见 MMIMENU_SYN_VALUE_E 枚举定义

【使用说明】

1. 预置应用、非预置应用增加至主菜单都是调用该接口, 区别只是动态菜单结构体变量的填充方式不同。
2. 该接口用法可参考原移植层接口 `MMIJAVA_AddSuiteRecordToMainmenu`。扩展参数 `is_set_menu_id` 是考虑到预置应用需要将菜单项 `menu_id` 作为运行函数参数, 如果 `menu_id` 只能由系统自动生成, 在填充动态菜单结构体变量时无法获取。
3. 如果应用填充动态菜单结构体变量时需要使用 `menu_id`, 请先调用 `MMIMENU_DyGenMenuId` 接口生成 `menu_id`, 并赋值给 `GUIMAINMENU_DYNAMIC_INFO_T` 结构体成员变量 `menu_id`, 之后调用 `MMIMENU_DyAppendItem` 时, `is_set_menu_id` 传入 `FALSE`; 如果不需要使用, 调用 `MMIMENU_DyAppendItem` 时, `is_set_menu_id` 传入 `TRUE` 就可以了。
4. `GUIMAINMENU_DYNAMIC_INFO_T` 结构体新增了一个成员变量 `suite_id`, 在调用 `MMIMENU_DyAppendItem` 前需要赋值为正确的 `suite_id`。由于预置应用增加至主菜单时未安装, 无法获取 `suite_id`, 请将该成员变量赋值为非法 `suite_id` (`0xffffffff`), 安装好后再调用后面提到的接口 `MMIMENU_DySetItemSuiteId` 修改其值。

新增 4 个接口

```

/*****/

// Description : Set Item Param
// Global resource dependence :
// Author:
// Note:成功返回 TRUE，失败返回 FALSE
/*****/

PUBLIC BOOLEAN MMIMENU_DySetItemParam(
                                uint32 menu_id,
                                void* param1ptr,
                                uint32 parm1_size,
                                void* param2ptr,
                                uint32 parm2_size
                                );
```

【函数功能】

修改动态菜单项运行函数参数（运行函数支持两个入参）

【参数说明】

menu_id: 菜单项 menu_id

param1ptr, param2ptr: 记录参数值的指针

parm1_size, parm2_size: 参数 size

【返回值】

是否修改成功

【使用说明】

1. 该接口支持单独修改一个参数。比如只需修改 param1，则 param1ptr, parm1_size 正确赋值，param2ptr 赋值为 PNULL 就可以了。
2. 预置应用安装后可以调用该接口修改其运行函数参数。比如在生成预置应用对应的菜单项时，运行函数参数传入 suite_id，但由于此时该应用尚为安装，suite_id 是非法值。安装好后调用该接口，可将运行函数参数改为正确的 suite_id。

```

/*****/

// Description : Set Item Suite Id
// Global resource dependence :
// Author:
// Note:成功返回 TRUE，失败返回 FALSE
/*****/
```

```
PUBLIC BOOLEAN MMIMENU_DySetItemSuiteId(
    uint32 menu_id,
    int32 suite_id
);
```

【函数功能】

修改动态菜单项 suite_id

【参数说明】

menu_id: 菜单项 menu_id

suite_id: suite_id 改为的值

【返回值】

是否修改成功

【使用说明】

如前所述，该函数用于预置应用安装后修改 suite_id。

```

/*****
// Description : Set State
// Global resource dependence :
// Author:
// Note:
*****/

```

```
PUBLIC BOOLEAN MMIMAINMENU_SetState(
    MMIMAINMENU_STATE_T state,
    BOOLEAN is_true
);
```

【函数功能】

开始/结束安装应用时，设置/取消安装状态

【参数说明】

state: 设置的状态（MMIMAINMENU_INSTALL_STATE: 非预置应用安装状态，
MMIMAINMENU_PRE_INSTALL_STATE: 预置应用安装状态）

is_true: 状态值（TRUE: 设置状态，FALSE: 取消状态）

【返回值】

状态设置是否成功

【使用说明】

1. 开始安装需要显示进度时将相应状态置为 TRUE，安装完成不需要显示进度时将相应状态置为 FALSE。

2. 由于主菜单对预置应用、非预置应用安装图标的处理方式不同，请务必正确设置相应状态。

```
/*
*****
// Description : Set Install Progress
// Global resource dependence :
// Author:
// Note:
*****
PUBLIC BOOLEAN MMIMAINMENU_SetInstallProgress(
                                uint8 install_progress
                                );

```

【函数功能】

更新主菜单安装进度显示

【参数说明】

install_progress: 安装进度（由于总进度设为 100，此处进度值为 0~100）

【返回值】

进度设置是否成功

【使用说明】

需要主菜单更新进度显示时调用该接口。比如以前在 MMI_NAMS_EVENT_INSTALL_PROGRESS 消息响应中调用更新 progress pubwin 进度条的接口，现在改为调用这个主菜单接口。

建议：

1. 根据之前讨论的结果，预置应用在开机初始化时添加至主菜单。在将预置应用添加至主菜单的函数中，menu_id 调用 MMIMENU_DyGenMenuId 生成，suite_id 赋值为非法值（0xffffffff），运行函数的两个参数分别传入 suite_id 和 menu_id。之后调用 MMIMENU_DyAppendItem 时，最后一个参数 is_set_menu_id 传入 FALSE，不要再由系统自动生成 menu_id。
2. 预置应用安装好后，调用 MMIMENU_DySetItemParam、MMIMENU_DySetItemSuiteId 将运行函数参数和 suite_id 修改为正确值。
3. 预置应用安装前，其运行函数只有一个有效参数 menu_id，调用 MMIMENU_DySlideGetDispInfo 接口可以根据 menu_id 获取应用名称。
4. 非预置应用运行函数参数以前为应用名称，由于应用名称存在重复的可能，建议也改为 suite_id。
5. 根据主菜单 spec 对移植层代码做了一点改动，如果与 java 修改不冲突，建议保留。

6. 微软目前提出直接在主菜单上管理 java 应用，不要那个 java 应用列表窗口了，所以主菜单已没有 java 入口。如果调试需要，可以自行添加，添加方法如下：

s_mainmenu_item_name 中添加一个字串 L"Java"

s_mainmenu_item_data 中添加一个菜单项(务必添加在 ID_PRIMO_INSTALL 菜单项之前)

```
{ID_ENTERTAINMENT_GOLDHOUSE, TXT_JAVA, IMAGE_MAINMENU_ICON_GOLDHOUSE,  
IMAGE_MAINMENU_ICON_SELECT, MMIMAINMENU_StartJava, 0}
```

这两处修改都在 mmi_mainmenu_data_240X320.c 文件中