MEDIATEK

VPI UI Component



Introduction and Overview

Copyright © MediaTek Inc. All rights reserved.

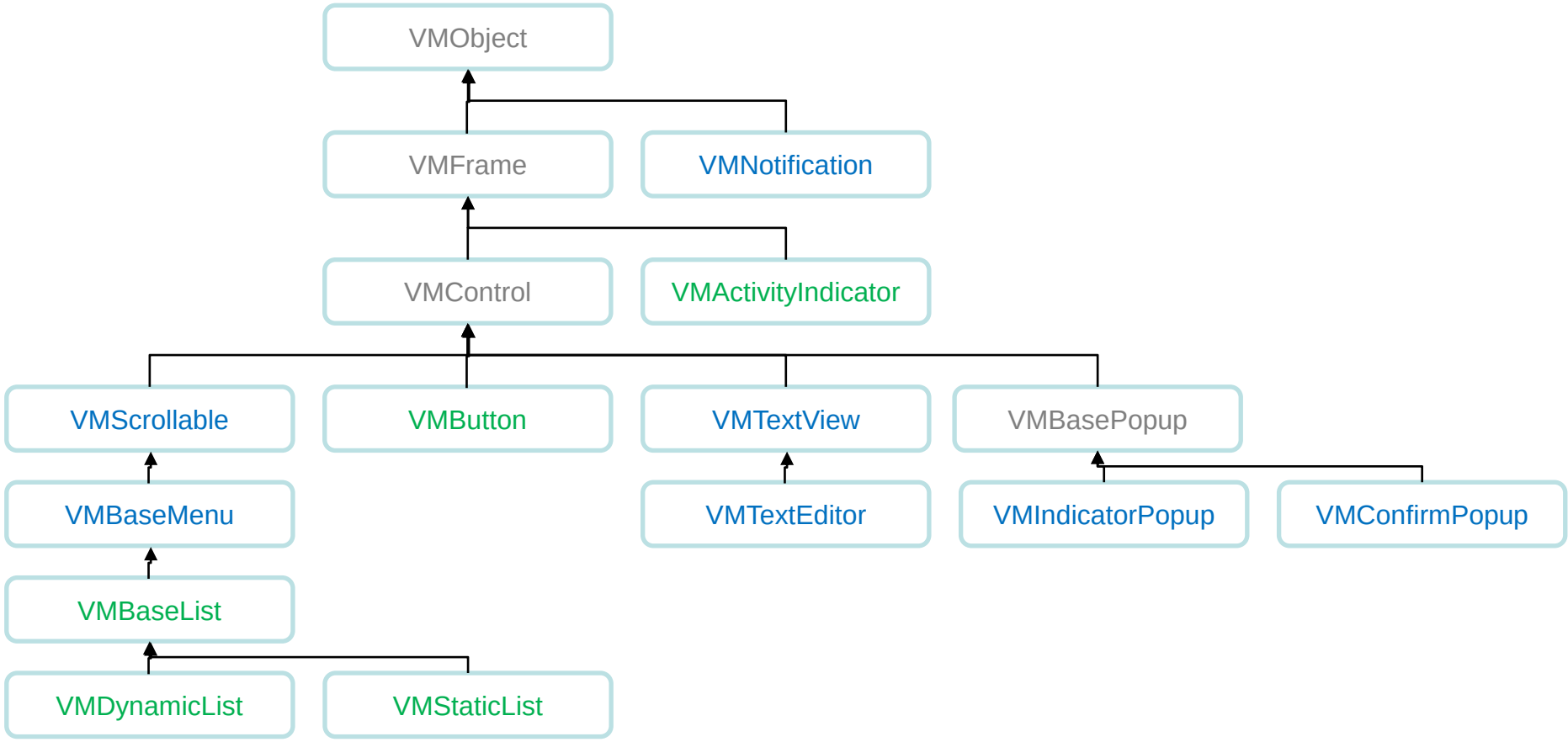
Introduction

- VPI CP provides a set of UI components for developers to compose their own screens, including
  - Scrollable
  - Base menu
  - Confirm popup
  - Indicator popup
  - Notification
  - Text view
  - Text editor
  - Button
  - Activity indicator
  - List menu

*exported from native, the UI look and feel follows native CP
*created in MRE only, independent from native CP

Introduction

- Class hierarchy



MEDIATEK

VMScrollable



Copyright © MediaTek Inc. All rights reserved.

VMScrollable

- VMScrollable is to make a bunch of components to be scrollable in a view. It enables users to scroll within the content area by swiping and implements deceleration of smooth scrolling movement.

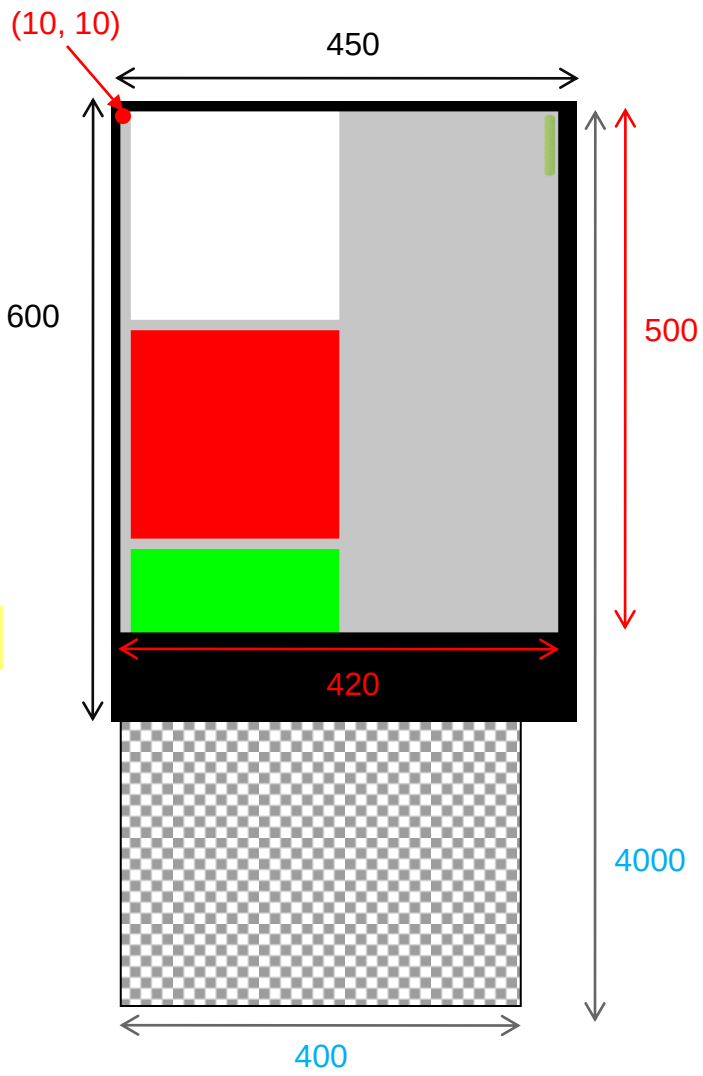
Typical Use Case

```
VMScrollable *scrollable;

// create a scrollable
VM_OBJ_CREATE(scrollable, VMScrollable, this);
scrollable->setBounds(VfxRect(0, 0, 450, 600));
scrollable->setContentSize(VfxSize(400, 4000));
// set view rect (x, y, width, height)
scrollable->setViewRect(VfxRect(10, 10, 420, 500));

// make some childs scrollable
VMFrame *child;
for (VfxS32 i = 0; i < 40; i++)
{
    VM_OBJ_CREATE(child, VMFrame, scrollable);
     child->setPos(VfxPoint(10, i * 210));
     child->setBounds(VfxRect(0, 0, 200, 200));
    child->setBgColor(colorlst[i]);
}
```

Set scrollable as parent



MEDIATEK

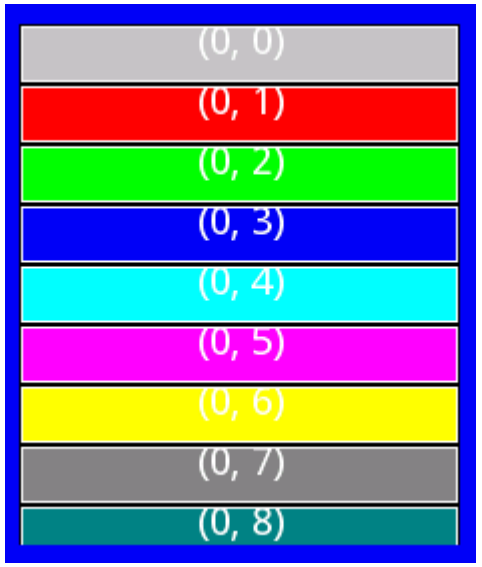
VMBaseMenu



Copyright © MediaTek Inc. All rights reserved.

VMBaseMenu

- Base menu is only a framework that can help to display a huge number of items in a scrolling frame.
- It creates and closes each menu items dynamically on demand.
- You can not create a base menu instance directly, but you can only inherit it and create your own menu.



Typical Use Case

```
// before using base menu, you have to create a layoutor
class MyLayout : public VMBaseMenuLayout
{
public:
    // set scrolling direction of the menu
    virtual vm_base_menu_layout_scroll_dir_enum getScrollDir() {
        return VM_BASE_MENU_LAYOUT_SCROLL_DIR_VERT;
    }
    // set number menu items
    virtual VMINT32 getCellCount(VMINT32 group) {return 5;}
    // set the total length of the menu
    virtual VMINT32 getContentLength() {
        return getGroupCount() * (GROUP_HEIGHT + getCellCount(0) * CELL_HEIGHT);
    }
    // let menu know the detail layout information of each menu item
    virtual vm_base_menu_region_struct queryRegion(const vm_rect_struct &r) {}
    virtual vm_base_menu_pos_struct getCellByPos(vm_point_struct pt) {}
    virtual vm_rect_struct queryCellRect(vm_base_menu_pos_struct pos) {}
};
```

Typical Use Case

```
// create our own menu
class MyMenu : public VMBaseMenu
{
public:
    // create the instance of our layout
    virtual IVpiBaseMenuLayout* createLayout() {}
    // create each cell, which is a frame
    virtual IVpiFrame* createCell(VMINT32 group, VMINT32 pos)
    // close each cell
    virtual void onCloseCell(VMINT32 group, VMINT32 pos, IVpiFrame* cellFrame)
    // update each cell
    virtual void onUpdateCell(
        const vm_rect_struct &rect, VMINT32 group, VMINT32 pos, IVpiFrame* cell, VMUINT32 flag) {
        cell->setRect(rect);
    }
};
// Now we can create the menu
MyMenu *menu;
VM_OBJ_CREATE(menu, MyMenu, this);
menu->setBounds(VMRect(0, 0, 320, 320));
```

MEDIATEK

VMButton



Copyright © MediaTek Inc. All rights reserved.

VMButton

- An instance of VMButton represents a button on the touch screen.
- A button would emit signal when user clicks on it.
- It provides a couple of methods for appearance customization.



Typical Use Case

```
VMcpButton* button;

// create button
VFX_OBJ_CREATE(button, VMcpButton, this);

// set content
button->setImage(VcpStateImage(IMG_GLOBAL_L1));
button->setText(VFX_WSTR("Top"));

// set properties
button->setPlacement(VCP_BUTTON_PLACEMENT_IMAGE_TOP);
button->setRect(VfxRect(10, 180, 60, 60));
button->setIsAutoResized(VFX_TRUE);

// connect click signal
button->m_signalClicked.connect(this, &VappMyScr::onButtonClick);
```

MEDIATEK

VMActivityIndicator



Copyright © MediaTek Inc. All rights reserved.

VMActivityIndicator

- An activity indicator is used to show an ongoing activity.
- App can start and stop the animation of the indicator according to the progress of the activity.



Typical Use Case

```
void MyApp::onInit (void)
{
    // Create indicator
    VM_OBJ_CREATE(m_actInd, VMCPActivityIndicator, this);

    // set property
    m_actInd->setPos(50, 100);
    m_actInd->setSize(50, 50);

    // start the animation
    m_actInd->startAnimating();
}

void MyApp::completeActivity (void)
{
    // stop the animation
    m_actInd->stopAnimating();
}
```

MEDIATEK

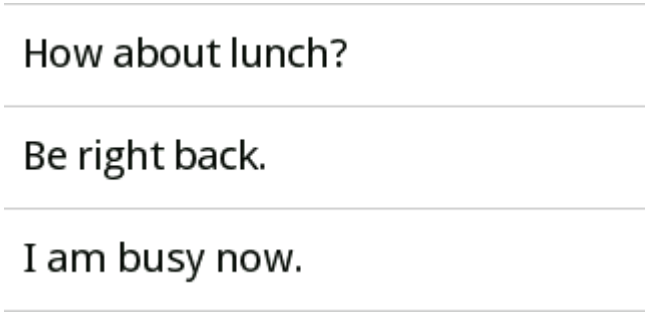
VMDynamicList



Copyright © MediaTek Inc. All rights reserved.

VMDynamicList

- VMDynamicList is used to create a dynamic list. That is all content of each cell items is created on demand.
  - List menu uses callback function to get data from APP
- Support 3 types of cell: one text, one text + one icon, and custom.
- Application can show any number of items in this list.
- Each item is with the same height.
- Handle action by connecting to m_signalItemTapped.



Typical Use Case

```
VMDynamicList *myList;

// create dynamic list
VM_OBJ_CREATE(myList, VMDynamicList, this);

// set number of items
myList->setItemCount(100);

// specify the updateCallback, dynamic list will send event and data to this callback
// for different cases, application may need to do some handling for each event
myList->m_UpdateItemCallback.connect(this, &Myapp::onUpdateCallback);

// specify the cell style
myList->setCellType(VM_LIST_CELL_TYPE_SINGLE_TEXT_ICON);

// set property
myList->setRect(VMRect(0, 0, 100, 100));
```

Typical Use Case

```
void onUpdateCallback(vm_list_menu_event_struct* data)
{
    switch(data->evt_id)
    {
        // for custom frame, we can create and set custom frame here
        case VM_LIST_EVT_ID_CLIENT_FRAME_CREATE:
            break;
        // set content for each cell
        case VM_LIST_EVT_ID_CLIENT_FRAME_UPDATE:
        {
            // cast the data back to cell instance
            VMListClientSingleTextCellView *clientFrame =
                dynamic_cast<VMListClientSingleTextCellView*>(data->cell);

            // set data for this cell
            clientFrame->setText(VM_WSTR("My String");
            clientFrame->setImage(MYIMage);
            break;
        }
    }
}
```

MEDIATEK

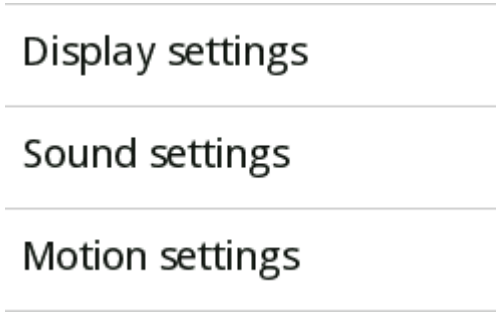
VMStaticList



Copyright © MediaTek Inc. All rights reserved.

VMStaticList

- VMStaticList is used to create a static list. That is, all data of each cell item is created beforehand.
- It is suitable for showing a list within a small and fixed number of items (16).
- Application can use this to show a simple and basic list.
- Support 3 types of cell: one text, one text + one icon, and custom.
- Handle action by connecting to m_signalItemTapped.



Typical Use Case

```
VMStaticList *myList;

// create static list menu
VM_OBJ_CREATE(myList, VMStaticList, this);

// set cell type
myList->setCellType(VM_LIST_CELL_TYPE_SINGLE_TEXT);
myList->setRect(VMRect(0, 0, size.width, size.height));

for(int i = 0; i < 30; i++)
{
     // create each data, this data will be freed after cell is closed
    VMSingleTextItemData *itemdata = (VMSingleTextItemData*) vm_malloc(sizeof(VMSingleTextItemData));

     // set data
    itemdata->m_text = VM_WSTR("MY Item");

     // add to list menu
    myList->addItem(itemdata);
}
```

MEDIATEK

VMTextView



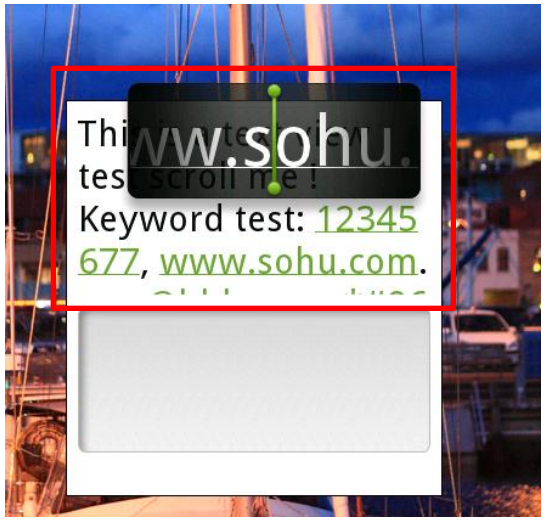




Copyright © MediaTek Inc. All rights reserved.

VMTextView

- VMTextView is a text display component which allows user to view a multi/single-line text.
- Support rich text with formatted text.



Typical Use Case

```
VMTextView *m_text;

// create text view
VFX_OBJ_CREATE(m_text, VMTextView, this);

// set pos and size
m_text->setPos(VfxPoint(TEXT_X_OFFSET, TEXT_Y_OFFSET));
m_text->setBounds(VfxRect(0, 0, TEXT_WIDTH, TEXT_HEIGHT));

// set text
m_text->setText(text);
```

MEDIATEK

VMTextEditor



Copyright © MediaTek Inc. All rights reserved.

VMTextEditor

- VMTextEditor is a text edit component for user input.



Typical Use Case

```
VMTextEditor *m_text;

// create text editor
VFX_OBJ_CREATE(m_text, VMTextEditor, this);

// set pos and size
m_text->setPos(VfxPoint(TEXT_X_OFFSET, TEXT_Y_OFFSET));
m_text->setBounds(VfxRect(0, 0, TEXT_WIDTH, TEXT_HEIGHT));

// set text
m_text->setText(text, NULL);

// set IME mode
m_text- >setIME(IMM_INPUT_TYPE_PHONE_NUMBER)

// set single-line or multi-line
m_text- >setLineMode(VM_TEXT_LINE_MODE_SINGLE);
```

MEDIATEK

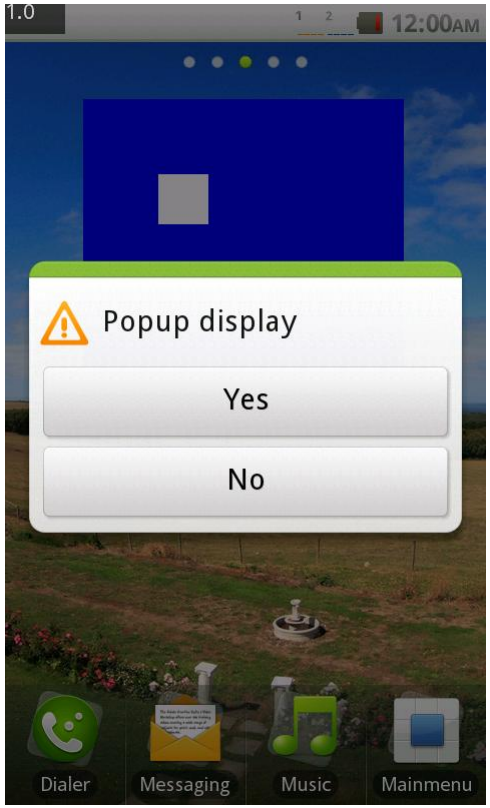
VMConfirmPopup



Copyright © MediaTek Inc. All rights reserved.

VMConfirmPopup

- VMConfirmPopup is used to display a popup with a title and an icon, and at most 2 buttons.
- Can support different info type (the same as native)
- Can support different button set (the same as ntaive)



Typical Use Case

```
// create a confirmpopup
VM_OBJ_CREATE(m_cnf_popup, VMConfirmPopup, this);

// set title
m_cnf_popup->setText(VM_WSTR("Popup display"));

// set type
m_cnf_popup->setInfoType(VM_POPUP_TYPE_WARNING);

// set button set
m_cnf_popup->setButtonSet(VM_CONFIRM_BUTTON_SET_YES_NO);

// connect signal
m_cnf_popup->m_signalButtonClicked.connect(this, &MyWidgetContent::popupButtonSignal);

// display the popup
m_cnf_popup->show(VM_TRUE);
```

MEDIATEK

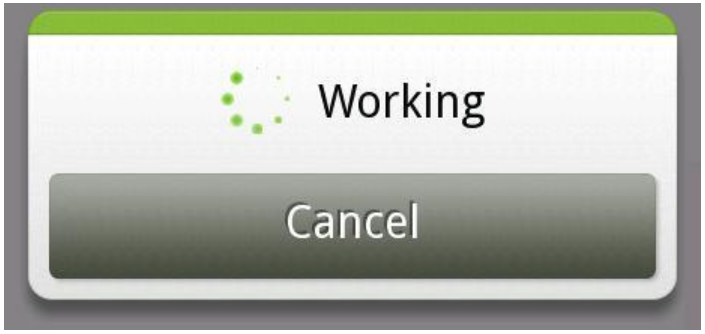
VMIndicatorPopup



Copyright © MediaTek Inc. All rights reserved.

VMIndicatorPopup

- VMIndicatorPopup is used to display an indicator inside a popup.
- Can support different types. (the same as native)



Typical Use Case

```
// create a confirmpopup
VFX_OBJ_CREATE(m_indPopup, VMIndicatorPopup, this);

// set title
m_indPopup->setText(VM_WSTR("Updating ..."));

// set type
m_indPopup->setInfoType(VM_INDICATOR_POPUP_STYLE_ACTIVITY);

// display the popup
m_indPopup->show(VM_TRUE);
```

MEDIATEK

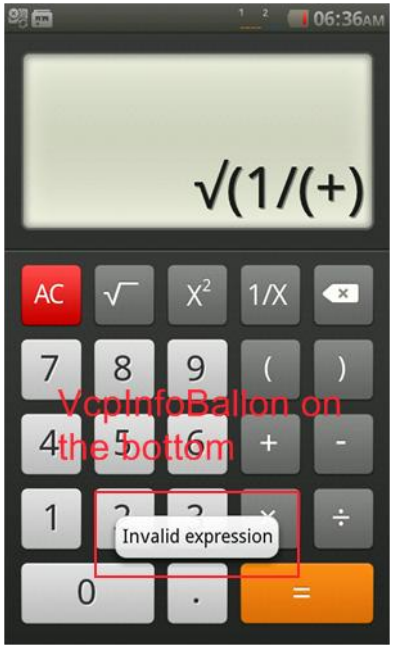
VMNotification



Copyright © MediaTek Inc. All rights reserved.

VMNotification

- Notification is used to display a message (text only) for user.
- Notification is a singleton object.



Typical Use Case

```
// get instance of notification and trigger a message display
VM_OBJ_GET_INSTANCE(VMNotification)->notify(VM_NOTIF_TYPE_ERROR, VM_WSTR("My first error!!"));
```

MEDIATEK

Thank you







Copyright © MediaTek Inc. All rights reserved.