MEDIATEK

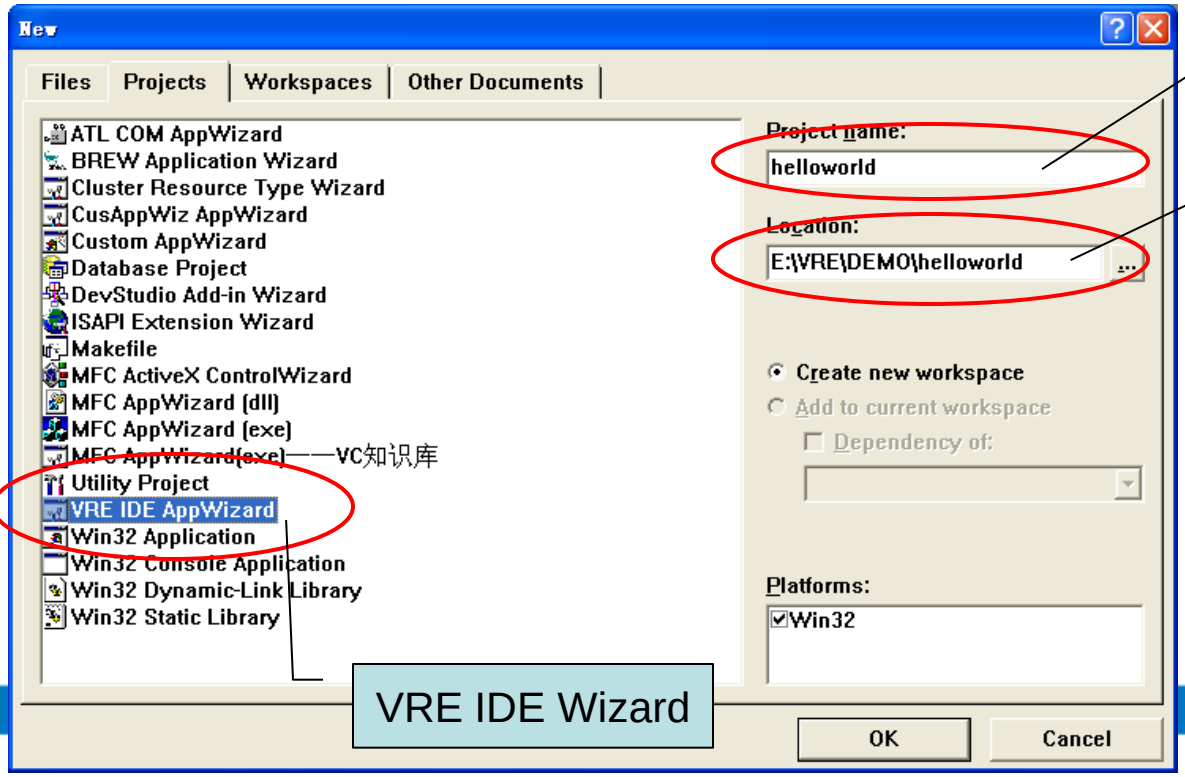
VRE SDK Guide





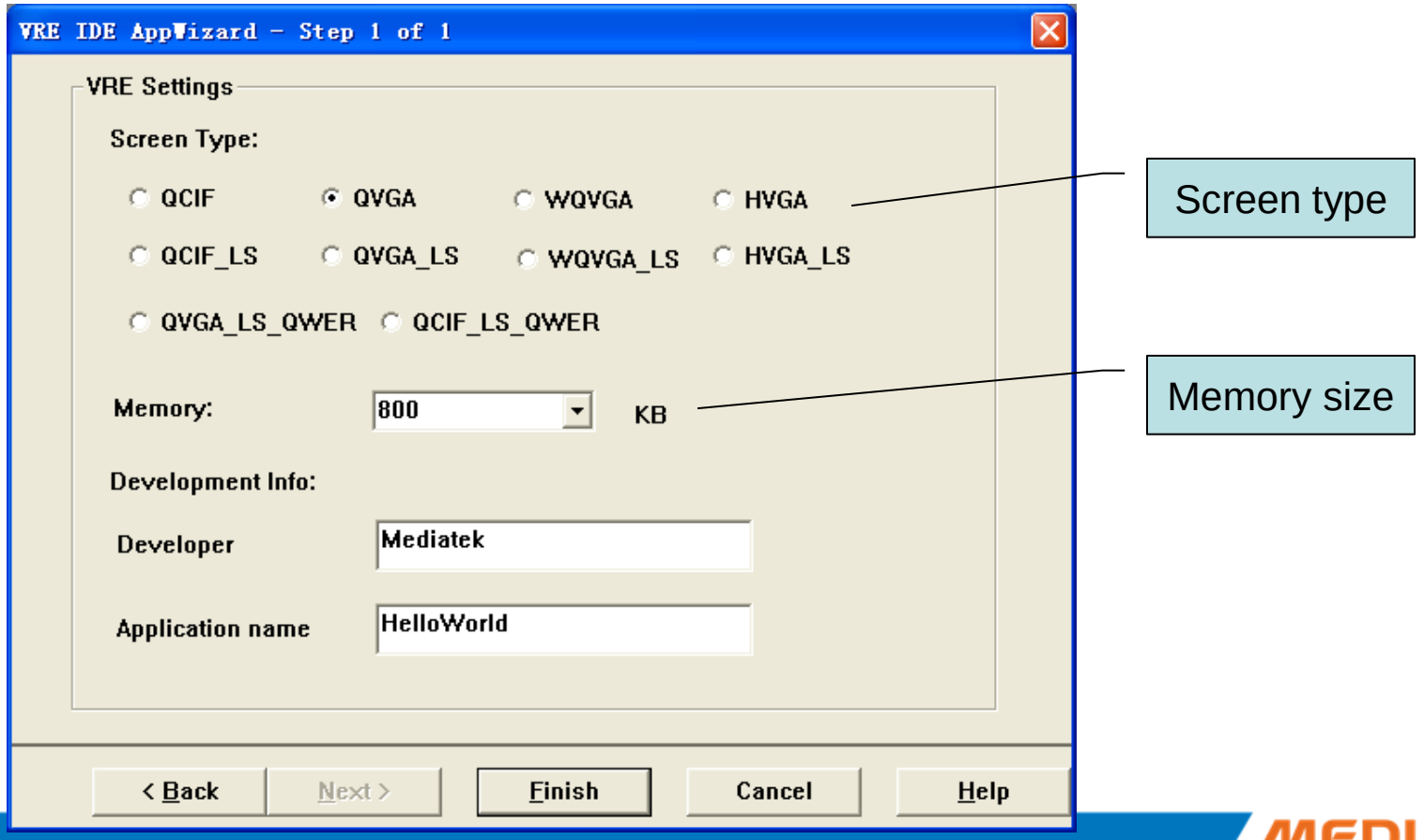
Create VRE App(1)

- Step1: Run the Microsoft Visual C++
- Step2: Create the VRE project
  - File->New->Projects->VRE IDE AppWizard



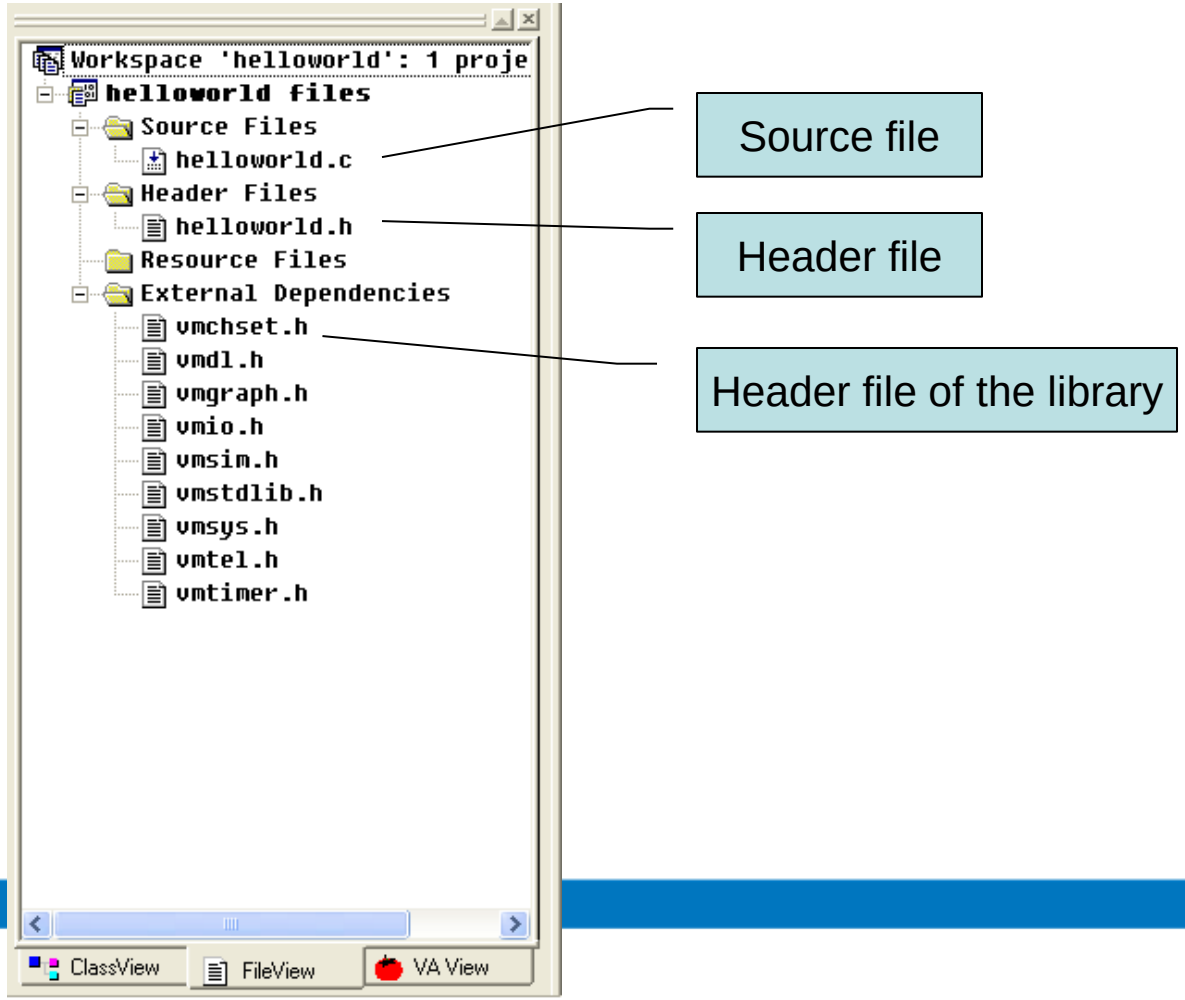
Create VRE App(2)

- Step3: Setting the VRE Application



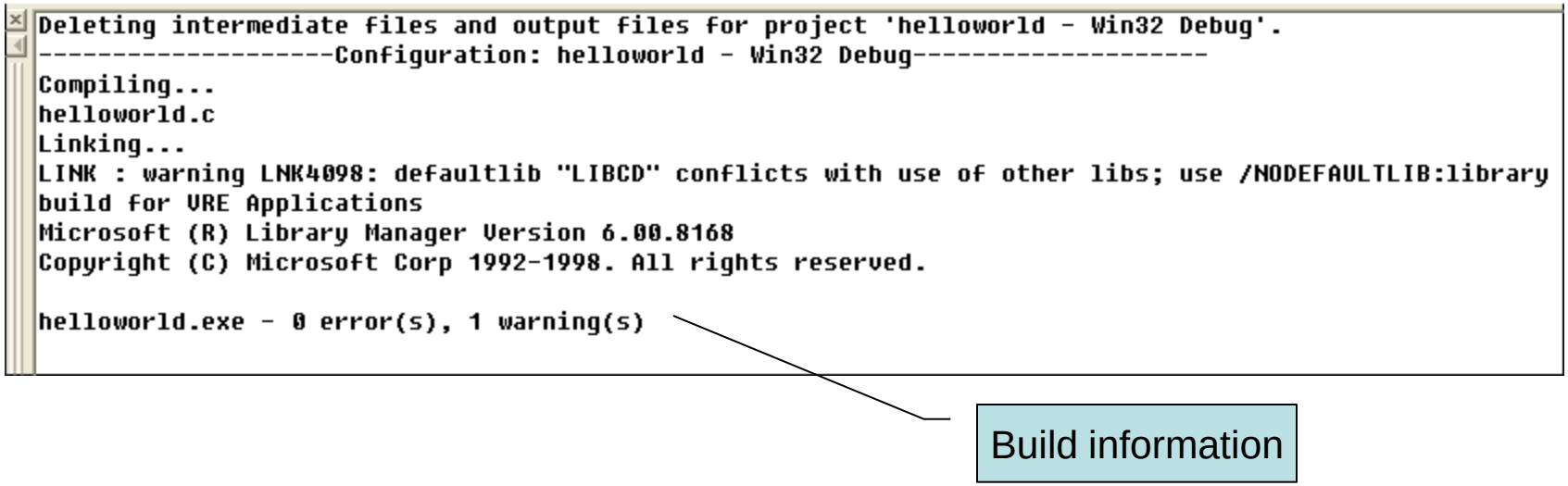
VRE App Source File

- VRE App Source File



Build the VRE App

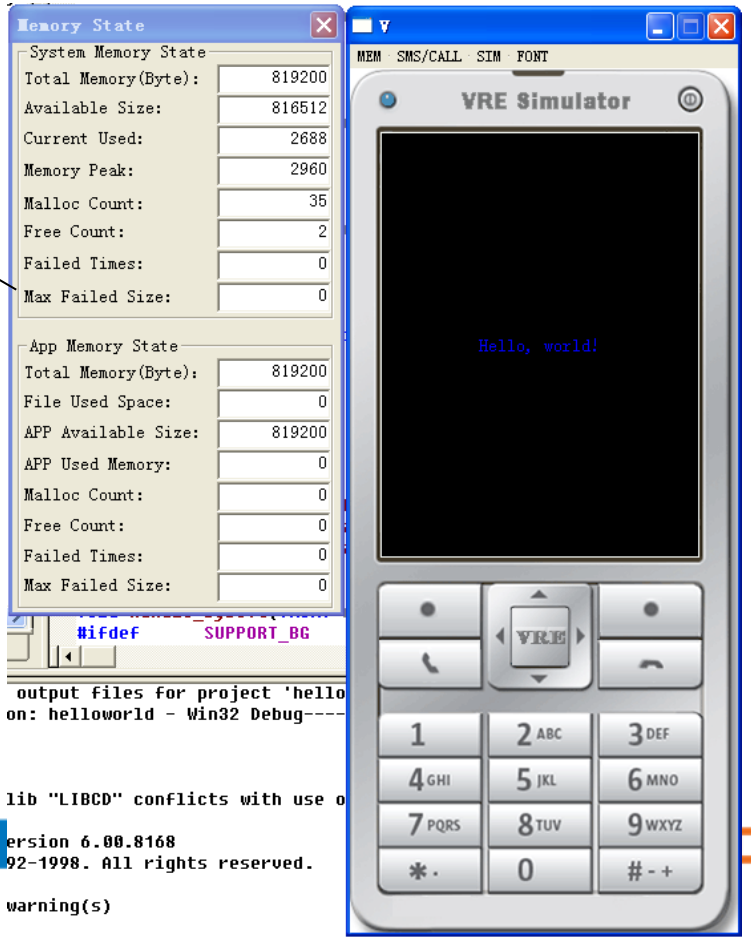
- Build the VRE App in VC
  - Menu: Build->Rebuild All
- The build information will show in the Build page.



Run the VRE App

- Run the VRE App in VC
  - Menu: Build->Execute(Ctrl+F5)
  - Toolbar:

VRE Application
HelloWorld



VRE App Main Function

- VRE app entry function
  - vm_main()
- Register event handler
  - vm_reg_sysevt_callback()
    - handle_sysevt() – System event handle function
  - vm_reg_keyboard_callback()
    - handle_keyevt() – Key event handle function
  - vm_reg_pen_callback()
    - handle_penevt() – Pen event handle function

```
/**
 * entry
 */
void vm_main(void) {
    layer_hdl[0] = -1;

    vm_reg_sysevt_callback(handle_sysevt);
    vm_reg_keyboard_callback(handle_keyevt);
    vm_reg_pen_callback(handle_penevt);
}
```

handle_sysevt()

- System event handle function, deal with the system event message.
- Parameter
  - Message
    - VM_MSG_PAINT
      - Create layer, update the screen
    - VM_MSG_INACTIVE
      - Delete layer when popup
    - VM_MSG_HIDE
    - VM_MSG_QUIT
      - Delete layer and exit the app.
  - Param
    - Reserved

```
void handle_sysevt(VMINT message, VMINT param) {
#ifdef      SUPPORT_BG
    /* The application updates the screen when receiving the message VM_MSG_PAINT
    *  what is sent after the application is activated. The application can skip
    *  the process on screen when the VM_MSG_ACTIVE or VM_MSG_INACTIVE is received.
    */
    switch (message) {
        case VM_MSG_CREATE:
            /* the GDI operation is not recommended as the response of the message*/
            break;
        case VM_MSG_PAINT:
            /* cerate base layer that has same size as the screen*/
            layer_hdl[0] = vm_graphic_create_layer(0, 0,
                vm_graphic_get_screen_width(),
                vm_graphic_get_screen_height(), -1);

            /* set clip area */
            vm_graphic_set_clip(0, 0,
                vm_graphic_get_screen_width(),
                vm_graphic_get_screen_height());

            draw_hello();
            break;
        case VM_MSG_HIDE:
        case VM_MSG_QUIT:
            if( layer_hdl[0] != -1 )
            {
                vm_graphic_delete_layer(layer_hdl[0]);
                layer_hdl[0] = -1;
            }

            break;
    }
```

handle_keyevt()

- Key event handle function, deal with the KEY_DOWN, KEY_UP, etc.
- Parameter
  - Event
    - VM_KEY_EVENT_DOWN
    - VM_KEY_EVENT_UP
    - VM_KEY_EVENT_LONG_PRESS
    - VM_KEY_EVENT_REPEAT
  - Keycode
    - VM_KEY_OK
    - VM_KEY_LEFT_SOFTKEY
    - VM_KEY_RIGHT_SOFTKEY
    - VM_KEY_NUM0
    - VM_KEY_NUM1

```
void handle_keyevt(VMINT event, VMINT keycode)
{
    switch(event)
    {
        case    VM_KEY_EVENT_DOWN:
            break;
        case    VM_KEY_EVENT_UP:
            if(VM_KEY_RIGHT_SOFTKEY == keycode)
            {
                if( layer_hdl[0] != -1 )
                {
                    vm_graphic_delete_layer(layer_hdl[0]);
                    layer_hdl[0] = -1;
                }
                vm_exit_app();
            }
            break;
        case    VM_KEY_EVENT_LONG_PRESS:
            break;
        case    VM_KEY_EVENT_REPEAT:
            break;
        default:
            break;
    }
}
```

handle_penevt()

- Pen event handle function, deal with the PEN_DOWN, PEN_UP, etc
- Parameter
  - Event
    - VM_PEN_EVENT_TAP
    - VM_PEN_EVENT_RELEASE
    - VM_PEN_EVENT_MOVE
    - VM_PEN_EVENT_REPEAT
    - VM_PEN_EVENT_LONG_TAP
  - X
    - The x-coordinate of the pen touch
  - Y
    - The y-coordinate of the pen touch

```
void handle_penevt(VMINT event, VMINT x, VMINT y)
{
    switch(event)
    {
    case    VM_PEN_EVENT_TAP:
        break;
    case    VM_PEN_EVENT_RELEASE:
        if( x > (vm_graphic_get_screen_width() / 2) &&
            y > (vm_graphic_get_screen_height() * 9 / 10))
        {
            if( layer_hdl[0] != -1 )
            {
                vm_graphic_delete_layer(layer_hdl[0]);
                layer_hdl[0] = -1;
            }
            vm_exit_app();
        }
        break;
    case    VM_PEN_EVENT_MOVE:
        break;
    case    VM_PEN_EVENT_REPEAT:
        break;
    case    VM_PEN_EVENT_LONG_TAP:
        break;
    default:
        break;
    }
}
```

MEDIATEK

www.mediatek.com