

BkWin 界面框架使用说明

张晓暄

2009/12/13

1、简单说明

1. 使用 XML 定义界面

窗口分为三部分：header、footer、body，至少应该包含一个部分，header 的空白部分是可以拖动的，窗口 resize 时 header 和 footer 高度不变，body 自适应



所有 xml 格式采用 **utf-8** 编码，或使用 ATL::CW2A 类由 Unicode 字符串转换

2. 资源统一管理

提供了 BkBmpPool、BkFontPool、BkPngPool、BkString 管理界面中用到的图片、字体、字符串资源，头文件放置在 bkres 目录下

BkBmpPool、BkPngPool 和 BkFontPool 一般情况下不需要调用

BkBmpPool 和 BkPngPool 会根据资源 ID 自动加载对应的位图资源，退出时一起释放

BkString 需要加载 xml 资源

2、XML 格式

1. skin 定义

i. Xml 实例

```
<skins>
  <imgframe name=panel src=110 top=2 left=2 crbg=F0F4FA/>
  <imgframe name=mainbgall src=104 top=14 left=3 crbg=DEEFFF/>
  <imgframe name=mainbghead src=104 top=14 left=3 crbg=DEEFFF part=top/>
  <imgframe name=mainbgbody src=104 top=14 left=3 crbg=DEEFFF part=middle/>
  <imgframe name=mainbgfoot src=104 top=14 left=3 crbg=DEEFFF part=bottom/>
  <button name=uglybtn/>
  <imghorzex name=normalbtn src=103 mode=mask subwidth=9 left=4/>
  <imghorzex name=tab src=109 mode=mask subwidth=7 left=3/>
  <imghorzex name=progressbg src=112 mode=mask left=3/>
  <imghorzex name=progresspos src=113 mode=mask left=3/>
</skins>
```

skin 用于定义一些固定的图片绘制格式

所有 skin 均包含 name 属性，在 style 和若干控件中指定，下面就不描述了

在应用程序开始时通过 BkSkin::LoadSkins 方法加载指定资源 ID 的 skin 定义 xml，资源分类必须为“XML”

skin 包含 Normal/Hover/Pushdown/Disable 四种状态，当 skin 使用了 ImageList 时，ImageList 中的图片必须按照此顺序水平排列

- ii. skins 标签
无属性，包含若干可能的皮肤格式
- iii. button 标签
使用简单渐变定义一个按钮样式，可以更改渐变颜色，具体见 demo
- iv. imgframe 标签
定义一个四角不变，中间一像素宽（高）四向拉伸的 frame 样式 skin

属性：

属性名：	说明
src	位图资源 ID
mode	none：不透明 mask：透明色 alpha：alpha 通道透明，需要 32 位 RGBA 位图
maskcolor	如果 mode=mask，则指定透明色，默认 FF00FF ■
subwidth	如果图片资源是 ImageList，用这个参数指定 ImageList 中单个图片的宽度
crbg	Frame 中间填充的背景颜色(十六进制 RRGGBB 格式)
top	Frame 上界高度(px)
left	Frame 左界宽度(px)
part	all：整个 frame（默认） top：不包含底部边框 middle：只有垂直中部边框 bottom：不包含头部边框 left：不包含右边框 center：只有水平中部边框 right：不包含左边框

- v. imghorzex 标签
定义一个左右不变，中间一像素宽水平拉伸的 frame 样式 skin

属性：

属性名：	说明
src	位图资源 ID
mode	none：不透明 mask：透明色 alpha：alpha 通道透明，需要 32 位 RGBA 位图（目前版本可能有 bug，未实验）
maskcolor	如果 mode=mask，则指定透明色，默认 FF00FF ■
subwidth	如果图片资源是 ImageList，用这个参数指定 ImageList 中单个图片的宽度

left	Frame 左界宽度(px)
-------------	----------------

vi. **imglist 标签**

定义一个图片列表样式 skin，一般用于 img 和 imgbtn 控件

属性:

属性名:	说明
src	位图资源 ID
mode	none: 不透明 mask: 透明色 alpha: alpha 通道透明, 需要 32 位 RGBA 位图
maskcolor	如果 mode=mask, 则指定透明色, 默认 FF00FF ■
subwidth	如果图片资源是 ImageList, 用这个参数指定 ImageList 中单个图片的宽度

2. style 定义

i. Xml 实例

```

<style>
<class name=mainall skin=mainbgall font=0000 crtext=000000 crbg=FFFFFF/>
<class name=mainhead skin=mainbghead font=0000 crtext=000000 crbg=FFFFFF/>
<class name=mainbody skin=mainbgbody font=0000 crtext=000000 crbg=FFFFFF
linespacing=10/>
<class name=mainfoot skin=mainbgfoot font=0000 crtext=000000 crbg=FFFFFF
linespacing=20/>
<class name=linktext font=0002 hoverfont=0002 crtext=3064A5 crhover=FF0000
/>
<class name=panel skin=panel margin=16 spacing=3 linespacing=5/>
<class name=uglybtn skin=uglybtn font=0000 crtext=000000 textmode=20
cursor=hand/>
<class name=normalbtn skin=normalbtn font=0000 crtext=FFFFFF crdisabled=A2A2A2
margin=5 cursor=hand/>
<class name=client font=0000 crtext=000000 margin=11 linespacing=20 />
<class name=linkimage cursor=hand/>
<class name=list spacing=5 linespacing=8 crtext=000000 />
<class name=toptext textmode=20 />
<class name=centertext textmode=25 />
<class name=tabpage margin=1/>
</style>

```

Style 用于定义控件的样式，包括使用的 skin、文字对齐方式、字体、文字颜色、背景色、鼠标指针形状等

流模式还可以指定边距、行间距、行内控件间距等等

在应用程序开始时通过 BkStyle::LoadStyles 方法加载指定资源 ID 的 Style 定义 xml，资源分类必须为“XML”

ii. class 标签

定义一个样式

属性:

属性名:	说明
name	Style 的名字, 用于控件的 class 属性
skin	指定使用的 skin, 如果指定了有效的 skinname, crbg 属性被忽略

textmode	文字对齐方式，即 DrawText 的最后一个参数，格式为十六进制不加“0x”
crbg	背景色，默认不填充背景
crtext	默认情况下文字颜色，默认为 000000 即黑色
crhover	鼠标移上去时的文字颜色，默认同 crtext
crdisabled	控件不可用时的文字颜色，默认 CCCCCC
font	字体，4 位 16 进制数，默认宋体 12 点字(0000)，前两位为字号，正值增大，负值缩小，大字体设置下字号不会改变 后两位 01 斜体、02 下划线、04 粗体，可叠加
hoverfont	鼠标移上去时的字体，格式同 font
x-margin	水平边距
y-margin	垂直边距
margin	边距，设置这个值会同时修改 x-margin 和 y-margin
cursor	鼠标移上去时的鼠标指针样式 arrow: 普通指针 (默认) hand: 手型

3. string 定义

i. Xml 实例

```
<string>
  <s id=1>应用程序标题</s>
  <s id=2>列 1</s>
  <s id=3>列 2</s>
  <s id=4>列 3</s>
</string>
```

定义应用程序中用到的字符串

在应用程序开始时通过 BkString::Load 方法加载指定资源 ID 的 Style 定义 xml，资源分类必须为“XML”

后加载的字符串表会覆盖先加载的表中相同 ID 的字符串，此特性可用于 OEM 订制

4. 窗口框架

i. 框架实例

```
<layer title="这是一个测试窗口" width=675 height=510 appwin=1>
  <header height=50>
    <header>
  <footer height=40>
    <footer>
  <body >
    <body>
</layer>
```

ii. layer 标签

定义一个窗口

属性:

属性名:	说明
title	窗口标题文字, 如果设置 appwin=1, 则显示在任务栏
height	窗口高度(px)
width	窗口宽度(px)
appwin	1 对应 WS_EX_APPWINDOW
resize	1 对应 WS_MAXIMIZEBOX WS_MINIMIZEBOX
noborder	1 对应 WS_POPUP WS_SYSMENU 以及 WS_EX_TOOLWINDOW WS_EX_TOPMOST 0 对应 WS_OVERLAPPED WS_SYSMENU

iii. header/footer/body 标签

定义窗口的一部分

属性:

属性名:	说明
height	高度(px), header 和 footer 可定义

5. 控件

i. 公用属性说明

属性名:	说明:
class	指定一个 Style Name
id	指定控件的 Command ID, 用于消息响应和显示控制, 在同一个窗口内必须唯一
href	指定一个链接或者应用程序, 用 shell 的 open 操作打开, 且不会再通知 click 消息
width	指定控件的宽度 0: 自适应文字宽度 (默认) full: 流模式或对话框模式但未指定宽度时会延伸到右边框, 对话框模式不建议使用
height	指定控件的高度 0: 自适应文字高度 (默认) full: 流模式或对话框模式但未指定高度时会延伸到底边框, 对话框模式不建议使用
pos	定义控件位置 格式为"left,top,right,bottom", 四个数值用半角逗号分隔, 正值为左(上)对齐, 负值为右(下)对齐 right 和 bottom 可省略, 此时控件大小由 width 和 height 属性控制 不建议流模式下使用
show	默认是否显示

ii. 控件的 InnerText 可包含预定义字符串, 用法为%str#strid#%

例:

定义字符串表

```
<string>
  <s id=1>AppName</s>
</string>
```

定义窗口

```
...  
<text>%str1% - 设置</text>  
...
```

则指定位置会显示“AppName - 设置”

iii. text 标签

定义一个文字控件
以指定字体、前景色和背景色在窗口绘制一段文本
可以用于制作超链接

iv. link 标签

就是 text，在 header 中，text 标签会被认为是空白位置可以拖动，使用 link 标签可以用来制作 header 上的超链接

v. button 标签

按钮，其实跟 text 一样，只是一般会指定一个带 skin 的 style

vi. img 标签

在窗口上显示图片

属性：

属性名：	说明
skin	指定 imglst
sub	指定当前显示的是 imglst 中的第几个图片，-1 则表示全部

vii. imgbtn 标签

图片按钮，使用 4 张图片的 ImageList，如果确定不会被 Disable，图片资源中可以只包含前三种状态，用来节省生成的程序大小

属性只有 skin，此控件会根据 Hover/Pushdown/Disable 状态切换 Sub Image

viii. progress 标签

进度条控件，用于显示一个进度条

属性：

属性名：	说明
bgskin	指定进度条背景 skin
posskin	指定进度条进度 skin
min	最小值
max	最大值
value	现在的值
showper	是否在正中间用文字显示 xx%，用 crtext 和 crbg 防止显示不清楚
cent	

ix. realwnd 标签

用于控制一个真实窗口的显示、隐藏

id 属性对应 DlgItemID 的控件

要绑定的真实窗口要指定 BkWin 窗口的 View 窗口为父窗口，比如在 OnInitDialog 时创建要绑定的窗口，父窗口参数必须为 GetViewHWND()，而不能直接用 m_hWnd

x. check 标签

复选钮

目前不能选择皮肤，自动套用 XP 以上的 Theme

xi. radio 标签

单选钮

目前不能选择皮肤，自动套用 XP 以上的 Theme

属性:

属性名:	说明
group	分组名，同一组的单选钮最多同时只能被选中一个，全局唯一标识，不能重复

xii. icon 标签

用于显示一个图标

属性:

属性名:	说明
src	指定图标资源 id
oem	是否 oem 图标 1: 是，用于加载 IDI_INFORMATION 等系统图标 2: 否，加载当前资源模块的图标
size	图标大小，16/32/48/64 等等 注意：由于 Windows API 一向的 bug，如果某 ICON 资源同时包含 16*16 和 32*32，则 size 指定 16 时，绘制的会是 32*32 缩小后的图标，所以 16*16 的图标需要单独放置一份

xiii. hr 标签

绘制直线，就不详细解释了

xiv. dlg 标签

对话框模式容器，对话框模式下的 header/footer/body 其实就是 dlg

也可以用于流模式下嵌套一个对话框模式排版的区域

属性名:	说明
onlydrawc hild	子控件只用于绘图，不响应鼠标消息

xv. tabctrl 标签

显示一个 tab 控件，标签页的内容用 tab 标签表示

属性:

属性名:	说明
------	----

curssel	当前选中的 tabpage, 从 0 开始
tabwidth	tab 标签的宽度
tabskin	tab 标签的 Skin Name
frameskin	tabpage 客户区的 Skin Name
n	
tableft	第一个 tab 标签和左边的间距
tabspacing	tab 和 tab 之间的间距, 负值则重叠
frameto	tabpage 客户区的位置, 负值则和标签区域有重叠
p	

xvi. tab 标签

属性:	
属性名:	说明
title	tab 标签标题
hide	是否隐藏

3、Classes

1. CBkDialogImpl

实现一个使用 BkWin 排版的窗口。

最简单的实例如下:

```
class CTestDlg
: public CBkDialogImpl<CTestDlg>
{
public:
    CTestDlg()
        : CBkDialogImpl<CTestDlg>(IDR_BK_TEST_DIALOG)
        , m_dwPos(0)
    {
    }

public:
    BK_NOTIFY_MAP(IDC_RICHVIEW_WIN)
        BK_NOTIFY_ID_COMMAND(IDBK_BTN_CLOSE, OnBkBtnClose)
        BK_NOTIFY_TAB_SELCHANGE(IDBK_TAB_MAIN,
OnBkTabMainSelChange)
    BK_NOTIFY_MAP_END()

    BEGIN_MSG_MAP_EX(CTestDlg)
        MSG_BK_NOTIFY(IDC_RICHVIEW_WIN)
        CHAIN_MSG_MAP(CBkDialogImpl<CTestDlg>)
    END_MSG_MAP()
};
```

在实例中, CBkDialogImpl 的第一个模板参数必须和派生类相同, 第二个模板参数使用默认表示使用对话框模式排版

构造函数可以指定窗口使用的模板资源 ID, 也可以在后面调用 Load 方法加载

BK_NOTIFY_MAP 和 BK_NOTIFY_MAP_END 宏用于处理消息

IDC_RICHVIEW_WIN 是 CBkDialogImpl 中 View 的 DlgCtrlID, 直接使用 CBkDialogImpl 的时候必须使用 IDC_RICHVIEW_WIN

目前可以处理的消息包括：更新内容参见 bkwndnotify.h

- i. **BKNM_COMMAND**
根据控件的 id 属性分发 click 消息
- ii. **BKNM_TAB_SELCHANGE**
tabctrl 切换页面时触发消息
根据 tabctrl 的 id 属性分发
- iii. **BKNM_MOUSEHOVER**
鼠标移到窗口时触发消息
和 **BKNM_MOUSELEAVE** 主要可以用于 pop 窗口消失时间的处理
注意：这两个消息没有根据 **id** 属性分发
- iv. **BKNM_MOUSELEAVE**
鼠标移出窗口触发消息

主要方法说明（具体参数见头文件定义）：

方法名：	说明
Load	加载指定资源 ID 的 xml 模板，资源必须为 BKRES 类型
GetViewHWND	获得 BkRichView 窗口句柄，用于创建 realwnd 标签控制的子窗口
SetPanelXml	重新设置某个 div 或 dlg 标签的内容，可以使用资源 ID 或 xml 字符串(utf-8 编码)
SetItemText	设置 button、text 的文字
FormatText	
SetItemAttribute	设置控件的某个属性，属性名用 utf-8 编码(其实跟 ansi 一样)
SetItemIntAttribute	值可以是 utf-8 字符串，也可以试 int、DWORD、COLORREF
SetItemDWordAttribute	
SetItemColorAttribute	
GetItemRect	获得某个控件的区域，可以用来控制 PopupMenu 的位置
GetItemCheck	返回/设置 check 或 radio 是否被选中
SetItemCheck	
IsItemVisible	返回/设置控件是否可见
SetItemVisible	
IsItemEnable	返回/设置控件是否可用
EnableItem	
SetTabCurSel	设置 tabctrl 当前显示的页面 id，从 0 开始
SetWindowCaption	设置窗口标题，在 Load 之后 Create 之前调用，可以动态改变窗口标题，以前用于合作版本的，可以忽略
DontShowWindow	在 OnInitDialog 时调用，可以不显示窗口、不抢焦点，为包含界面的流程提供后台模式非常实用
Create	创建非模态窗口
DoModal	模态创建窗口并调用

EndDialog

关闭窗口，和 MFC 和 WTL 不同的是，非模态窗口也可以调用

大部分方法都包含 `ulItemID` 参数，表示需要获取/设置信息的控件的 `id` 属性

注意：**Redraw** 时 **DC** 需要引用 **HBITMAP**，因为做了资源收集，所以如果多个线程同时 **Select** 同一个 **HBITMAP**，会导致绘图失败，所以涉及到界面的调用比如窗口创建、设置控件属性，最好都在界面线程，不要直接在回调中使用上表中的方法

另外 `CBkDialogImpl` 响应了回车和 ESC 键，重载 `OnOK` 和 `OnCancel` 可以响应这两个键，默认什么都不做

2. `CBkSimpleDlg`

构造传入一个 `Xml` 模板资源 ID，以此模板创建一个对话框，`DoModal` 后点击任何 `button` 或 `imgbtn` 均导致窗口消失并返回点击的按钮 ID 属性值，对于简单提示窗口很有效

`OnOK` 和 `OnCancel` 也会分别返回 `IDOK` 和 `IDCANCEL`

3. `CBkDialogViewImpl`

`BkWin` 的真正实现所在，可以在普通对话框中创建实例，也可以直接使用 `CBkDialogImpl` 来创建整个对话框

在普通对话框中使用时，在消息循环内加入一行

```
MSG_BK_NOTIFY(#BK_DIALOG_VIEW_ID#)
```

即可处理 `BkWin` 分发的消息

`#BK_DIALOG_VIEW_ID#` 为 `CBkDialogViewImpl::Create` 时传入的 `DlgCtrlID`，用于当对话框中包含多个 `BkDialogView` 时区分

主要方法和 `CBkDialogImpl` 类似

4、新的资源定义方式(091225 新增)

新提供了一种定义资源的方式，使用步骤如下：

1. 添加 `bkres\bkres.rc` 到工程
2. 在公用头文件引用 `bkres\bkres.h`
3. 在工程目录下新建 `bkwinres.rc2`，以下面格式定义资源即可，会比使用 VS 资源编辑器的方式简单

```

#pragma once

/////////////////////////////////////////////////////////////////
// XML

DEFINE_XML(IDR_BK_SKIN_DEF,          100, "res\\def_skin.xml")
DEFINE_XML(IDR_BK_STYLE_DEF,        101, "res\\def_style.xml")
DEFINE_XML(IDR_BK_STRING_DEF,       102, "res\\def_string.xml")
DEFINE_XML(IDR_BK_MAIN_DIALOG,      103, "res\\dlg_main.xml")

/////////////////////////////////////////////////////////////////
// ICON

DEFINE_ICO(IDI_BEIKESAFE,           100, "res\\beikesafe.ico")
DEFINE_ICO(IDI_SMALL,               101, "res\\small.ico")

/////////////////////////////////////////////////////////////////
// BITMAP

DEFINE_BMP(IDB_BTN_NORMAL_BG,       100, "res\\btn_normal_bg.bmp")
DEFINE_BMP(IDB_BTN_SYS_CLOSE,       101, "res\\btn_sys_close.bmp")

/////////////////////////////////////////////////////////////////
// CONTROL ID

#define IDC_BTN_SYS_CLOSE           60001
#define IDC_BTN_SYS_MAX             60002
#define IDC_BTN_SYS_MIN             60003

#define IDC_BTN_FAST_SCAN           100
#define IDC_BTN_FULL_SCAN           101
#define IDC_BTN_CUSTOM_SCAN         102

/////////////////////////////////////////////////////////////////
// STRING ID

#define IDS_APP_NAME                 1
#define IDS_EXAM_SCORE               2

```