DuiVision

蓝蚂蚁工作室

http://www.blueantstudio.net

[DUIVISION 开发手册]

DuiVision 界面库文档[更新日期：2014-10-15]

目录

DuiVision 开发手册

1. 整体说明

DuiVision 是参考了仿 PC 管家程序、金山界面库、DuiEngine、DuiLib 等多个基于 DirectUI 的界面库开发的。

DirectUI 技术一般是指将所有的界面控件都绘制在一个窗口上，这些控件的逻辑和绘图方式都必须自己进行编写和封装，而不是使用 Windows 控件，所以这些控件都是无句柄的。

DirectUI 技术需要解决的主要问题如下：

1、窗口的子类化，截获窗口的消息。

2、封装自己的控件，并将自己的控件绘制到该窗口上。

3、封装窗口的消息，并分发到自己的控件上，让自己的控件根据消息进行响应和绘制。

4、根据不同的行为发送自定义消息给窗口，以便程序进行调用。

5、一般窗口上控件的组织使用 XML 来描述。

通常 DirectUI 的界面库都采用 XML 配置文件+图片+控制脚本（Lua、Javascript 等）的开发方式，非常类似于 Web 程序的开发方式，当然这里面控制脚本也可以直接使用 C++代码来实现。这种开发方式可以大大提高开发效率，将程序员从繁琐的界面工作中解脱出来，并且通过美工的设计，可以使界面更美观。

2. 开发说明

2.1. XML 资源文件定义

基于 DuiVision 界面库的程序，需要有一个默认的资源定义 XML 文件，此文件默认的位置是 exe 文件所在路径下的 xml\resource.xml 文件 如果使用了 zip 压缩文件来保存所有资源文件，则此文件的位置是在压缩包中的 xml\resource.xml 文件。此文件中可以定义程序的全局配置、XML 文件、字体、图片、文字等资源，示例如下：

```
<?xml version="1.0" encoding="utf-8"?>
<root>

<!--系统配置-->
<res type="cfg" name="defaultStyle" value="" />
<res type="cfg" name="logfile" value="demoui.log" />
<res type="cfg" name="loglevel" value="1" />
<res type="cfg" name="appMutex" value="MUTEX_DUIVISION_DEMO" />
<res type="cfg" name="enableDragFile" value="1" />
<res type="cfg" name="trayDbClickMsg" value="0" />
```

```
<!--风格设置-->
<res type="style" name="default" value="" />
<res type="style" name="qq" value="qq" />

<!—嵌套的XML资源定义文件-->
<res type="res" lang="zh-cn" file="xml\def_string_zh-cn.xml" />
<res type="res" lang="en-us" file="xml\def_string_en-us.xml" />

<!--XML资源-->
<res type="xml" name="dlg_main" file="xml\dlg_main.xml" />
<res type="xml" name="dlg_skin" file="xml\dlg_skin.xml" />
<res type="xml" name="dlg_about" file="xml\dlg_about.xml" />
<res type="xml" name="dlg_login" file="xml\dlg_login.xml" />
<res type="xml" name="dlg_msgbox" file="xml\dlg_msgbox.xml" />

<!--字体资源-->
<res type="font" lang="zh-cn" name="default" font="微软雅黑" size="12"
bold="false" />
<res type="font" lang="zh-cn" name="big" font="微软雅黑" size="14"
bold="true" italic="false" underline="false" strikeout="false" />

<!--图片资源-->
<res type="img" name="IDB_MAIN_FRAME" file="skins\WindowsBack.png" />
<res type="img" name="IDB_BT_CLOSE" file="skins\BT_CLOSE.png" />
<res type="img" name="IDB_BT_MIN" file="skins\BT_MIN.png" />
<res type="img" name="IDB_BT_MENU" file="skins\BT_MENU.png" />
<res type="img" name="IDB_BT_SKIN" file="skins\BT_SKIN.png" />

<res type="img" name="IDB_TAB_1" file="skins\Tab1.png" />
<res type="img" name="IDB_TAB_2" file="skins\Tab2.png" />

<res type="img" name="IDB_ICON_INFO" file="skins\info.png" />
<res type="img" name="IDB_ICON_WARN" file="skins\warning.png" />
<res type="img" name="IDB_ICON_ERROR" file="skins\error.png" />

<res type="img" name="IDB_MENU_UPDATE" file="skins\MENU_UPDATE.png" />

<!--字符串资源-->
<res type="str" lang="zh-cn" name="APP_NAME" value="DUI测试程序" />
<res type="str" lang="zh-cn" name="APP_VER" value="1.0.0.1" />
<res type="str" lang="zh-cn" name="OK" value="确定" />
```

```
<res type="str" lang="zh-cn" name="CANCEL" value="放弃" />
<res type="str" lang="zh-cn" name="LOGIN" value="登录" />

</root>
```

这些定义的说明如下：

1、全局配置定义

Xml 的 type 是 cfg，目前支持的配置如下：

logfile – 日志文件名，是相对 exe 的路径的文件名，如果未定义，则不会生成日志文件

loglevel – 日志级别，1 表示调试级别，2 表示信息级别，4 表示错误级别，8 表示致命级别

defaultStyle – 默认的风格，resource.xml 中的每个资源定义都可以加一个 style 属性，通过 style 属性指定这条资源定义是针对哪种风格的，只有和当前的风格相同的资源或者默认风格的资源才会被加载，资源定义中指定风格的例子如下：

```
<res type="xml" style="qq" name="dlg_main" file="xml\dlg_wnd.xml" />
```

appMutex – 应用程序互斥量的名字，如果指定了此变量，则应用程序只能创建一个运行的进程，第二个进程运行时候判断如果存在此名字的互斥量，则退出

enableDragFile – 是否允许拖拽一个图片文件到程序窗口来指定当前使用的背景图片

trayDbClickMsg – 在托盘图标双击是否发送消息，用于重新定义托盘图标双击的行为，托盘图标双击的默认行为是打开程序的主窗口，如果此变量设置为 1，则双击托盘图标会给应用程序发送一个 DUI 消息，消息类型为 MSG_TRAY_DBCLICK，消息的发送方 ID 和名字分别是 TRAY_ICON 和 NAME_TRAY_ICON，可以再 DUI 消息处理类中增加对此消息的处理，来实现自定义的双击动作

2、风格定义

type 是 style，name 是风格的名字，仅用于说明，value 是风格的值，每个资源定义中的 style 对应的是风格的 value 部分。

资源 XML 中所有的定义都可以加一个 style 属性来指定当前定义是针对特定风格的，如果和当前的风格一致，则使用此定义覆盖之前的未指定风格的同名的定义。

例如下面的两个定义是定义的名为 IDB_SCROLL_V 的图片资源，一个是缺省定义，一个是针对 qq 风格的定义，如果当前不是 qq 风格，就会用前一个定义，如果当前是 qq 风格，就会用针对 qq 风格的定义，如果当前是 qq 风格，但没有第二行针对 qq 风格的定义，则也会使用第一行的缺省定义。

```
<res type="img" name="IDB_SCROLL_V" file="skins\default\SCROLL_V.png"
```

```
/>
<res type="img" style="qq" name="IDB_SCROLL_V"
file="skins\qq\SCROLL_V.png" />
```

3、资源文件定义

Type 是 res，用于加载嵌套的资源定义文件，file 是对应的文件路径名，是相对 exe 所在的路径，通过 lang 属性可以指定此资源文件仅针对哪种语言。

4、xml 文件定义

Type 是 xml，name 是其他地方引用时候的名字，file 是对应的文件路径名，是相对 exe 所在的路径。

5、字体定义

Type 是 font，lang 表示是针对哪种语言的字体。其他属性说明：

name – 字体定义名，在其他地方用名字进行引用

font – 字体名

size – 文字大小

bold – 是否粗体（true|false）

italic– 是否斜体（true|false）

underline – 显示下划线（true|false）

strikeout – 显示删除线（true|false）

os – 表示此字体定义适用于哪些操作系统，例如 os="winxp,win7"表示此定义仅针对 xp 和 win7 操作系统，可用的操作系统名字符串包括 win98、winme、winnt、win2000、winxp、win2003、vista、win7、win8。

6、图片资源定义

Type 是 img，name 是其他地方引用时候的名字，file 是对应的文件路径名，是相对 exe 所在的路径。

7、字符串资源定义

Type 是 str，lang 表示是针对哪种语言的字体，name 是其他地方引用时候的名字，value 是字符串内容。

定义的字符串资源可以在各个空间的 title 属性中引用，引用时候要用[]包围，例如

title=”[APP_NAME]”就可以在 title 中引用 APP_NAME 对应的字符串。

8、资源中的多语言定义

资源定义 xml 中每一项都可以指定语言，如果定义中有 lang 属性，则表示这条定义是针对哪种语言的，如果和当前语言不相符，则不会被加载。

2.2. XML 对话框文件定义

程序中所有界面都是基于对话框或菜单等窗口的 ,每个对话框都需要有一个XML定义文件，用于描述对话框中的内容，对话框中主要是组成对话框的各个控件的定义，对话框的 XML 定义示例如下：

```
<?xml version="1.0" encoding="utf-8"?>
<dlg name="dlg_about" title="MsgBox" width="450" height="230" appwin="1"
resize="1" translucent="245" frame="" bkimg="skin:SKIN_PIC_7"
crbk="000000" >
    <base>
        <imgbtn name="button.close" pos="-45,0,-0,29"
skin="IDB_BT_CLOSE" show="1"/>
        <text name="title" crtext="FFFFFF" crmark="800000" font="big"
              pos="10,5,200,25" title="关于[APP_NAME]"
mask="[APP_NAME]" response="0" show="1" />
    </base>
    <body>
        <area name="area-1" pos="0,0,-0,40" begin-transparent="100"
end-transparent="30" />
        <area name="area-2" pos="0,40,-0,-0" begin-transparent="30"
end-transparent="30" />
        <area name="area-3" pos="0,-37,-0,-36" begin-transparent="70"
end-transparent="70" />
        <area name="area-4" pos="0,-36,-0,-0" begin-transparent="88"
end-transparent="88" />
        <img name="icon" pos="25,45" width="128" height="128"
image="skins\scriptnet.jpg" mode="normal" framesize="0" response="0"
show="1" />
        <text crtext="000000" pos="170,45,-25,65" title="[APP_NAME]
[APP_VER]" />
        <text crtext="000000" pos="170,65,-25,85" title="2013-2014" />
        <linkbtn name="linkbtn1" crtext="800000"
            pos="170,100,-25,130" show="1"
title="http://www.blueantstudio.net"
```

```
href="http://www.blueantstudio.net" />
        <button name="button.ok" skin="IDB_BT_DEFAULT" title="[OK]"
    pos="-100,-30,-20,-6" show="1" />
    </body>
</dlg>
```

其中由几部分组成，dlg 标签是对话框自身一些属性的描述，可以设置对话框的大小、背景图片、蒙版图片、透明度、应用程序窗口属性、改变大小属性等；

base 标签下面的内容都是属于对话框的基础控件，一般可以把对话框的标题和关闭按钮等放在基础控件部分定义；

body 标签下面的内容是属于对话框的普通控件，除了基础控件之外，其他内容都放在 body 标签下面定义。

base 和 body 下面都是具体控件的定义描述，可以参考后面关于每个控件的属性说明。

以上对话框 XML 定义的界面效果如下：

2.3. XML 菜单文件定义

菜单也是通过 XML 文件来定义，菜单 XML 的定义示例如下：

```
<?xml version="1.0" encoding="utf-8"?>
<menu title="TrayMenu" width="250" item-height="25" left="30"
    frame-width="0" top-height="72" bottom-height="30"
    bkmode="mid" width-lt="5" height-lt="70" width-rb="5" height-rb="30"
    bkimg="skins\menu\360TrayMenu_218.png" >

    <text pos="10,5,-10,25" crtext="FFFFFF" font="bold" title="DUI托盘
菜单" />
    <img pos="-60,5" width="48" height="48"
image="skins\icon\Movies.png"
        mode="normal" framesize="1" tip="图片" />
```

```
    <text pos="10,-25,-10,-5" crtext="808080" font="bold" title="蓝蚂蚁
工作室" />

    <menuitem name="restore_mainwnd" skin="" image="" title="恢复窗口"
font="bold" action="show-window:dlg_main" />
    <menuitem separator="1" skin="IDB_MENU_SEP" />
    <menuitem name="menu_main" menu="menu_main" />
    <menuitem name="menuitem.sub" title="子菜单" width="150"
skin="IDB_MENU_ARROW"
        bkmode="frame" bkimg="skin:IDB_MENU_BACK" frame-width="3"
top-height="0" bottom-height="0" >
        <menuitem name="360safe" title="360安全卫士"
action="dlg:dlg_login" />
        <menuitem name="360sd" title="360杀毒" action="dlg:dlg_login" />
        <menuitem name="menuitem.360.sub1" title="360工具" width="150"
skin="IDB_MENU_ARROW"
            bkmode="frame" bkimg="skin:IDB_MENU_BACK" frame-width="3"
top-height="0" bottom-height="0" >
            <menuitem name="360driver" title="360驱动修复"
action="dlg:dlg_login" />
            <menuitem name="360soft" title="360软件管家"
action="dlg:dlg_login" />
        </menuitem>
    </menuitem>
    <menuitem name="menuitem.runtest" title="执行进程"
action="run:{platpath}DuiVisionDemo.2008d.exe|testcmd" />
    <menuitem separator="1" skin="IDB_MENU_SEP" />
    <menuitem name="item_help" skin="IDB_MENU_HELP" img-count="3"
title="帮助" action="link:http://www.blueantstudio.net" />
    <menuitem separator="1" skin="IDB_MENU_SEP" />
    <menuitem name="close_app" skin="" image="" title="退出"
action="close-window:dlg_main" />

</menu>
```

菜单定义文件的根节点是 menu 节点，menu 节点可以定义一些菜单的属性，包括菜单的大小、图标区域的宽度、背景图片等。

menu 节点下面是每个菜单项和控件的定义，菜单项一般用 menuitem 标签进行定义，除了菜单项之外，还可以定义菜单分隔线，以及其他的控件，例如图片控件。

菜单的显示区域可以分为 3 段，中间是菜单项部分，上面和下面是可以显示其他控件的区域，如果要在底部区域显示图片、文字等控件，这些控件的位置定义中的 Y 坐标都应该使用负数

来定义。

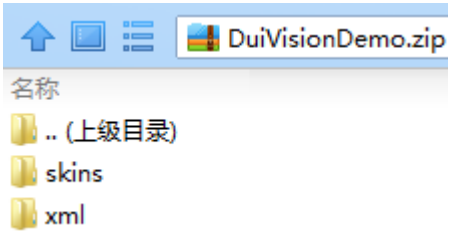
以上菜单定义文件的实际显示效果如下：

2.4. 从 ZIP 压缩文件中加载资源

DuiVision 支持将所有的图片和 XML 资源文件放在一个 zip 格式的压缩文件中，如果使用 zip 格式的资源文件，需要在主程序代码中初始化部分指定使用的压缩文件的文件名。

如果使用 zip 资源文件，则 resource.xml 文件的位置默认是放在 zip 文件中的 xml 子目录下。

建议 zip 文件按照 xml、skins 这样的子目录来压缩，见下面的压缩文件示例：



有 zip 资源文件的情况下，资源文件的加载并不一定是加载的 zip 文件中的内容，加载的优先级如下：

1)如果只有 zip 压缩文件，没有非压缩的 xml 和 skins 目录，则只会加载 zip 文件中的内容；

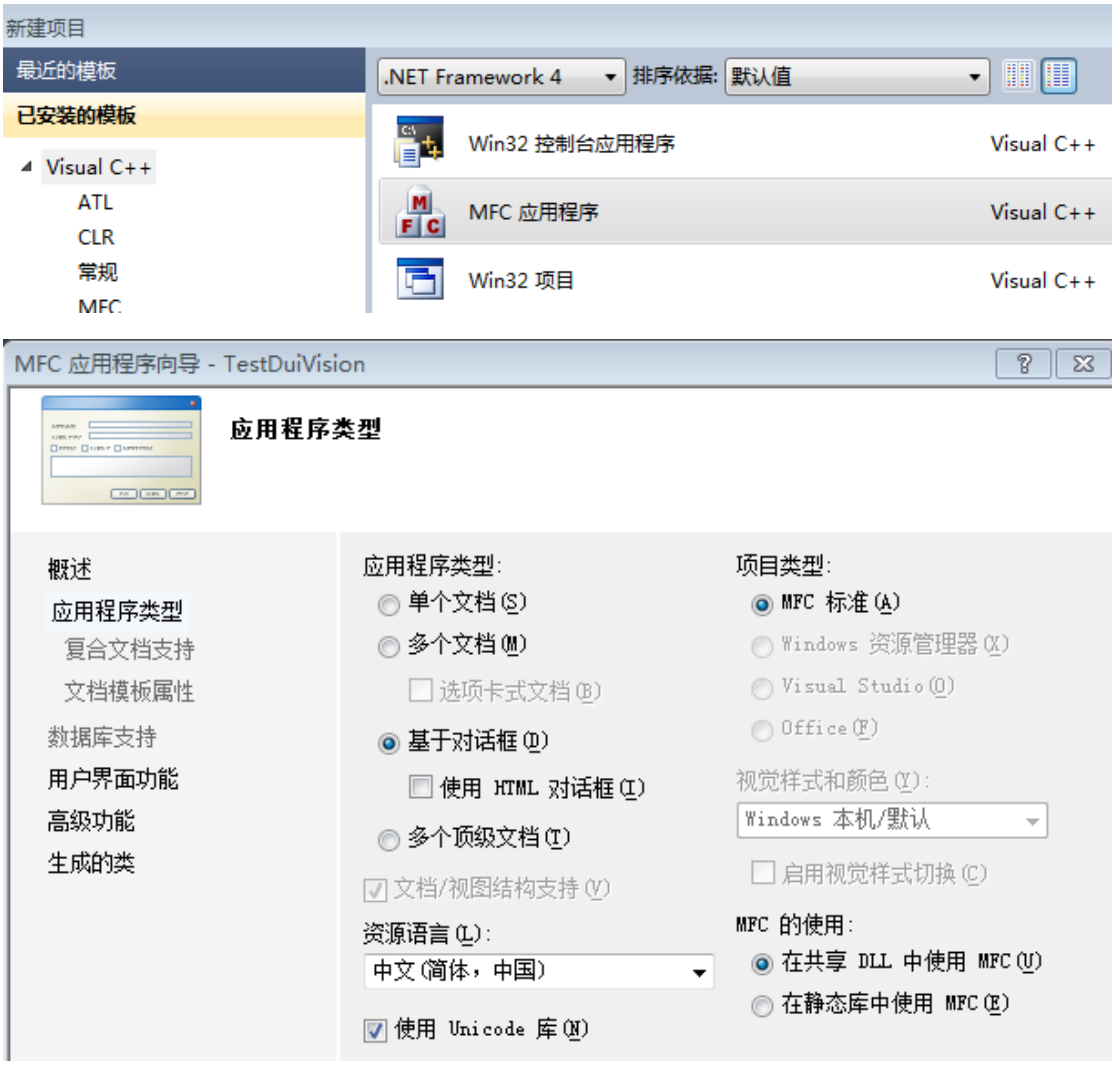
2 如果 zip 压缩文件和非压缩的 xml 和 skins 目录同时存在 则优先加载非压缩的 xml 和 skins 目录中的文件，对应的文件不存在的情况下才去 zip 文件中查找是否存在并加载。

之所以这样定义是方便通过非压缩的文件替换压缩文件中部分内容 ,以及方便调试和发布工

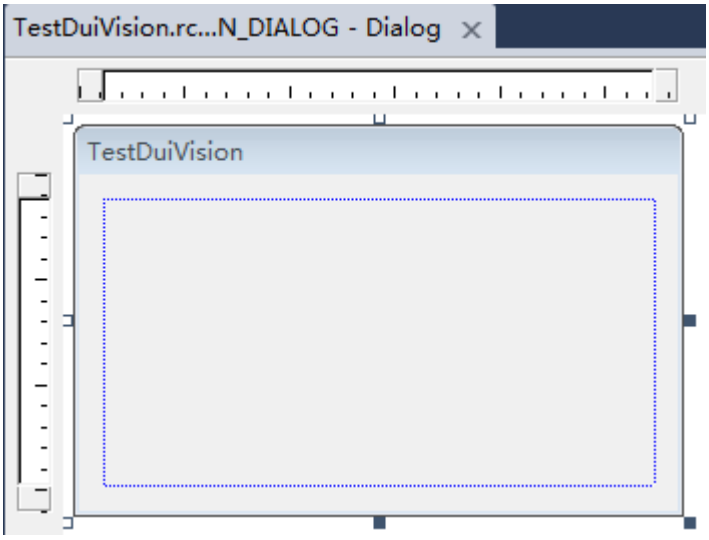
程，调试阶段可以直接修改非压缩的目录中文件，不用每次修改之后都要再打一次压缩包。

2.5. 创建主程序的方法（人工创建）

1、创建一个基于 DuiVision 的界面程序是比较简单的，在 VC 中创建一个 MFC 对话框工程，注意工程要使用 Unicode 库：



工程创建之后，需要将默认对话框资源中的几个按钮和文字都删除，变成一个干净的对话框资源：



2、设置 DuiVision 的头文件和 lib 文件目录

将 DuiVision 的头文件和 lib 文件放在某个位置，并在工程的头文件和 lib 文件路径定义部分添加相应的目录。

然后在 stdafx.h 文件中添加如下几行对 DuiVision 的头文件和 lib 文件的引用。

```
#include "DuiVision.h"
```

Lib 库采用在工程设置中添加引用库文件的方式，针对 VC2008 的 DuiVision 库文件分别是 DuiVision.2008d.lib 和 DuiVision.2008.lib。

添加 lib 引用之后应该就可以编译代码了。

3、DuiVision 库的初始化以及主窗口的定义

在主程序的 App 类 InitInstance()函数中添加 DuiVision 库的引用代码，示例代码如下：

```
    // 初始化DuiVision界面库,可以指定语言,dwLangID为表示自动判断当前语言
    // 1116是应用程序ID，每个DUI应用程序应该使用不同的ID
    // ID主要用于进程间通信传递命令行时候区分应用
    // IDD_DUIVISIONDEMO_DIALOG是工程中创建的那个对话框资源ID，所有窗口都是共用此ID的
    // 第三个参数可以指定资源文件名，如果不指定默认使用xml\resource.xml的定义进行资源的加载，如果指定了一个其他的文件名，则表示指定的是一个zip压缩格式的资源文件，需要从压缩文件中加载资源
    DWORD dwLangID = 0;
    new DuiSystem(m_hInstance, dwLangID, _T("DuiVisionDemo.ui"), 1116,
IDD_DUIVISIONDEMO_DIALOG, "");
```

```
// 创建主窗口
CDlgBase* pMainDlg = DuiSystem::CreateDuiDialog(_T("dlg_main"), NULL, _T(""), TRUE);
// 给主窗口注册事件处理对象
CDuiHandlerMain* pHandler = new CDuiHandlerMain();
pHandler->SetDialog(pMainDlg);
DuiSystem::RegisterHandler(pMainDlg, pHandler);

// 初始化提示信息窗口
DuiSystem::Instance()->CreateNotifyMsgBox(_T("dlg_notifymsg"));

// 按照非模式对话框创建主窗口,可以设置为默认隐藏
pMainDlg->Create(pMainDlg->GetIDTemplate(), NULL);
//pMainDlg->ShowWindow(SW_HIDE);
INT_PTR nResponse = pMainDlg->RunModalLoop();

// 如果是按照模式对话框运行主窗口,只要改为如下代码就可以
//INT_PTR nResponse = pMainDlg->DoModal();

// 释放DuiVision界面库的资源
DuiSystem::Release();
```

如果已经定义了主窗口的XML定义文件 ,添加上面的代码之后应该就可以创建出主窗口了。这段代码做的事情主要是：

1、DuiVision 库的初始化，并指定资源文件的位置（不指定则使用默认的位置）

2、根据 dlg_main 定义加载主窗口界面

3、创建主窗口的事件处理对象，并注册给主窗口（注册之后主窗口的事件都会自动发送给此事件处理对象的 OnDuiMessage 处理函数进行处理）

4、显示主窗口，使用非模态方式运行主窗口的消息循环

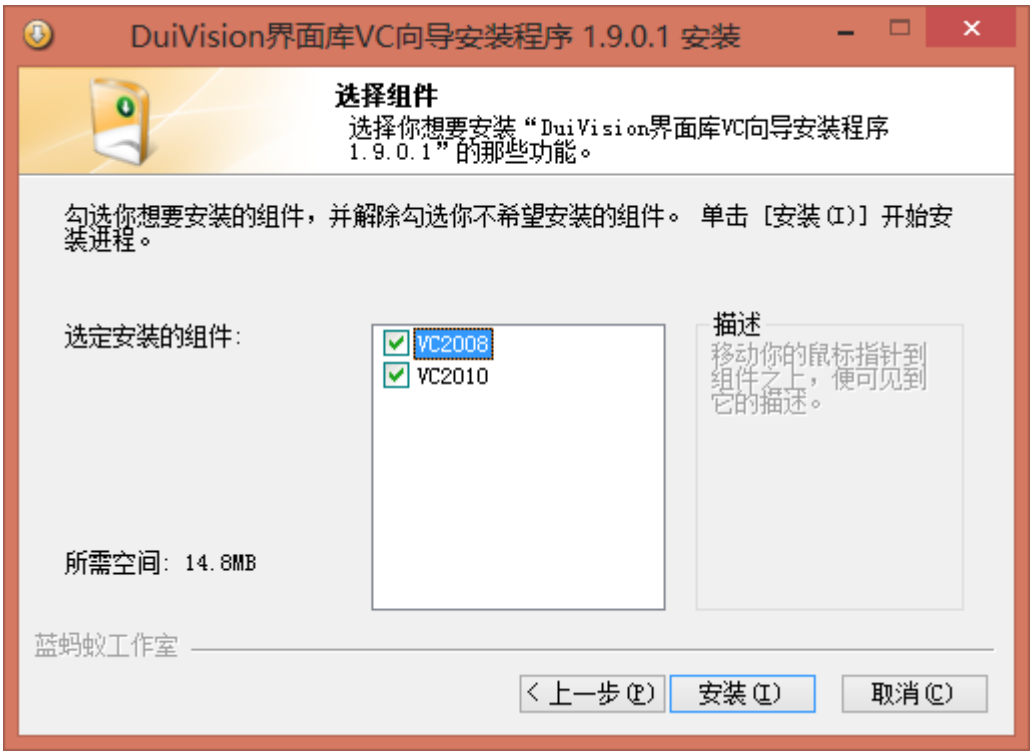
5、主界面关闭之后进行 DuiVision 库的释放

注意运行之前要把图片、xml 定义都放在对应的位置，默认图片都在 skins 目录，xml 定义文件都在 xml 目录。

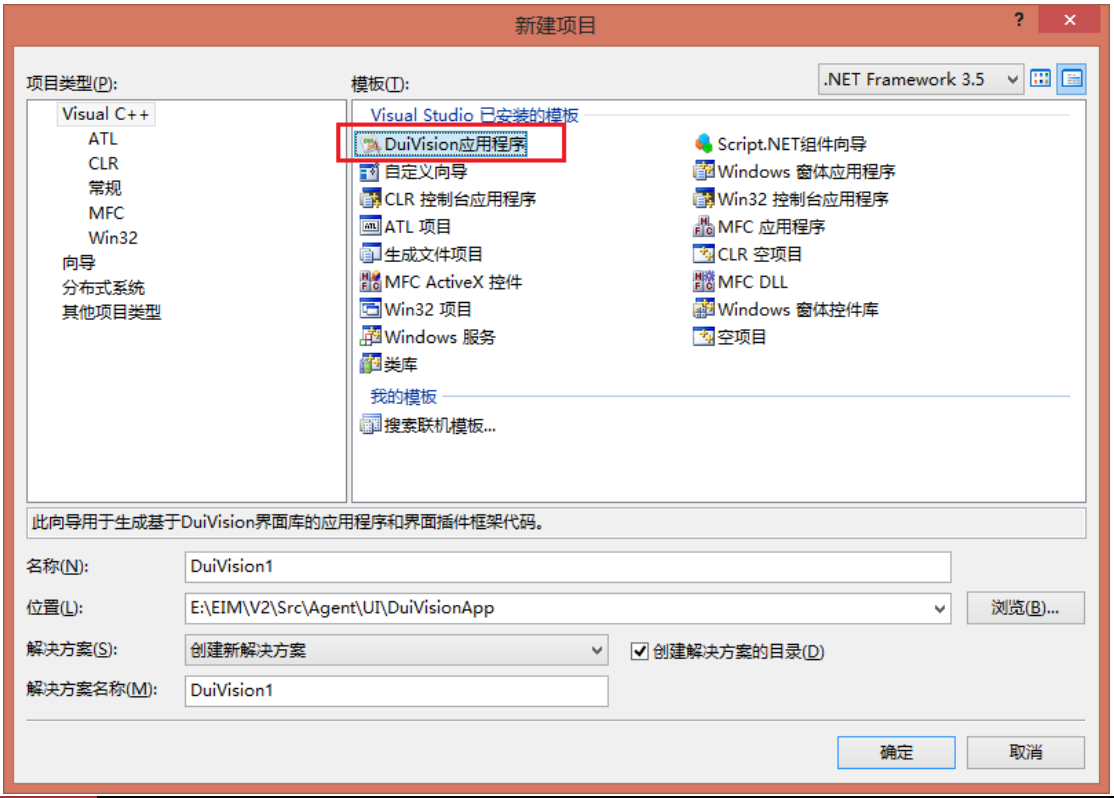
2.6. 创建主程序和插件的方法（通过向导创建）

DuiVision 提供了创建工程的 VC 向导，目前支持 VC2008 和 VC2010，也可以稍微修改之后用于其他的 VC 版本，请先从蓝蚂蚁工作室网站或 QQ 群下载安装 VC 向导，VC 向导发布了通

过安装包自动安装的版本和人工安装的压缩包版本。如果使用向导安装包进行安装，可以选择安装到哪些 VC 版本中，选择界面如下：



向导安装之后，在 VC 中新建工程，可以看到模板中多了一个 DuiVision 应用程序模板：

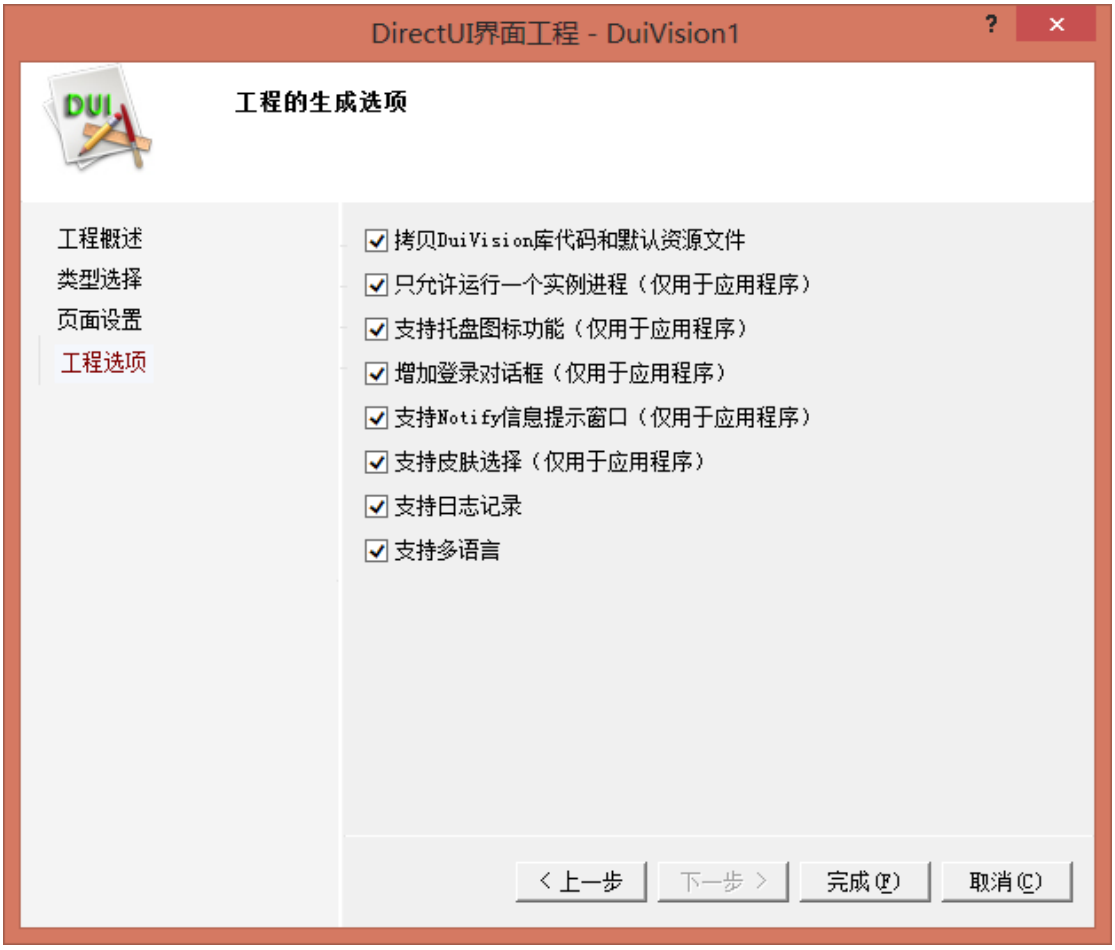


选择此模板就可以创建 DuiVision 工程，创建工程一共有四个步骤，第一步是设置应用程序标题和版权信息：

第二步是选择创建的工程类型，有两种类型，DuiVision 主程序和 DuiVision 插件：

第三步是设置界面中多个 Tab 页情况下每个 Tab 页的标题和事件处理类名字，如果不设置任何 Tab 页，也会自动创建一个默认 Tab 页：

第四步是工程的一些选项设置，如果是新建的一个解决方案工程，则默认会将 DuiVision 库的代码和资源文件拷贝到新建的解决方案目录下，如果是在现有解决方案中新建一个工程，则默认不拷贝，你也可以修改这个选项：



生成的工程中包含 DuiVision 库、生成的工程代码、bin 目录和 bin 目录下的资源文件，针对创建的每个 Tab 页 ,都会在 bin 目录下的 xml 目录下生成对应的 Tab 页 xml 文件框架 ,DuiVision 应用程序的 xml 文件在 xml\app 目录下，DuiVision 从插件程序的 xml 文件在 xml\plugin 目录下。

向导中包含的 DuiVision 库可能不是最新的库，可以在工程生成之后，将工程目录下的 DuiVision 目录和 bin 目录下的一些基础资源文件替换为最新的文件。

2.7. 事件处理类编写

除了界面的描述之外，最主要的工作就是业务逻辑的处理，为了将业务逻辑和界面展示能够更好的分离，DuiVision 中定义了事件处理基类，所有的业务逻辑都应该写在派生的事件处理类中，并把事件处理对象注册到相应的对话框或控件上，这样对应的子控件有事件需要处理的时候，就会自动调用注册的事件处理对象的相应函数。事件处理类只要在处理函数中根据控件的 ID 或名字决定该做什么事情，写相应的处理代码就可以。时间处理类中同时提供了一些函数方便根据 ID 或名字获取到对应的控件对象，并对控件进行操作，例如改变控件文字、获取控件的某个状态等。

事件处理基类是CDuiHandler，这个类的一些函数如下，提供了获取控件对象、设置控件的一些参数的函数，方便在事件处理类中对控件的操作：

```
void SetDuiObject(CDuiObject* pDuiObject);
CControlBase* GetControl(UINT uControlID);
CControlBase* GetControl(CString strControlName);
CDlgBase* GetControlDialog(UINT uControlID);

void SetVisible(CString strControlName, BOOL bIsVisible);
void SetDisable(CString strControlName, BOOL bIsDisable);
void SetTitle(CString strControlName, CString strTitle);
CString GetTitle(CString strControlName);

virtual void OnInit();
virtual LRESULT OnDuiMessage(UINT uID, CString strName, UINT Msg, WPARAM wParam, LPARAM lParam);
```

下面这段代码是在派生的事件处理类的 OnDuiMessage 函数中判断如果点击了某个按钮，就修改一个进度条进度的代码：

```
if(strName == _T("button_normal_3"))
{
    CDuiProgress* pControl = (CDuiProgress*)GetControl(_T("progress"));
    if(pControl && (Msg == BUTTOM_UP))
    {
        static int g_nProgress = 0;
        g_nProgress += 10;
        if(g_nProgress > 100)
        {
            g_nProgress = 0;
        }
        pControl->SetProgress(g_nProgress);
    }
}
```

为了简化事件处理类的编写，使事件处理的代码看起来更清晰一些，DuiHandler.h 中定义了一些事件处理函数的消息映射宏，如下表所示：

宏	参数	说明

DUI_DECLARE_MESSAGE_BEGIN	类名	事件处理类的消息映射宏开始
DUI_DECLARE_MESSAGE_END	无	事件处理类的消息映射宏结束
DUI_CONTROL_ID_MESSAGE	控件 ID、处理函数	根据控件 ID，执行相应的处理函数
DUI_CONTROL_IDMSG_MESSAGE	控件 ID、消息、处理函数	根据控件 ID 和消息，执行相应的处理函数
DUI_CONTROL_NAME_MESSAGE	控件名、处理函数	根据控件名，执行相应的处理函数
DUI_CONTROL_NAMEMSG_MESSAGE	控件名、消息、处理函数	根据控件名和消息，执行相应的处理函数

实际定义的样例如下：

```
// 消息处理定义
DUI_DECLARE_MESSAGE_BEGIN(CDuiHandlerMain)
    DUI_CONTROL_ID_MESSAGE(APP_IPC, OnDuiMsgInterprocess)
    DUI_CONTROL_NAME_MESSAGE(NAME_SKIN_WND, OnDuiMsgSkin)
    DUI_CONTROL_NAMEMSG_MESSAGE(NAME_TRAY_ICON, MSG_TRAY_DBCLICK,
OnDuiMsgTrayIconDClick)
    DUI_CONTROL_NAMEMSG_MESSAGE(L"notify_button_1", BUTTOM_UP, OnDuiMsgNotifyButton1)
    DUI_CONTROL_NAMEMSG_MESSAGE(L"notify_button_2", BUTTOM_UP, OnDuiMsgNotifyButton2)
    DUI_CONTROL_NAMEMSG_MESSAGE(L"notify_button_3", BUTTOM_UP, OnDuiMsgNotifyButton3)
    DUI_CONTROL_NAMEMSG_MESSAGE(L"listctrl_1", BUTTOM_DOWN, OnDuiMsgListCtrl1Click)
    DUI_CONTROL_NAMEMSG_MESSAGE(L"listctrl_2", BUTTOM_DOWN, OnDuiMsgListCtrl2Click)
DUI_DECLARE_MESSAGE_END()
```

在编写自己的 Handler 事件处理类时候可以参考样例，定义消息映射宏，每个需要处理的消息定义一行内容，定义出哪个控件的什么消息需要处理，由哪个函数处理，消息处理函数必须按照固定的参数格式来编写，消息处理函数的样例如下：

```
// 显示信息对话框消息处理
LRESULT CDuiHandlerMain::OnDuiMsgMsgBoxButton1(UINT uID, CString strName, UINT Msg, WPARAM
wParam, LPARAM lParam)
{
    DuiSystem::DuiMessageBox(NULL, _T("演示对话框！"));
    return TRUE;
}
```

每个消息处理函数必须按照以上样例中的参数格式，函数的返回值表示此消息是否不再需要

向下传递继续处理了，如果返回 TRUE，则消息处理结束，如果返回 FALSE，则此消息还可以继续被后面定义的其他函数处理。

使用以上的宏定义，会自动在进入消息处理函数时候记录日志，日志内容类似于下面这样：

DEBUG 2014-09-07 22:43:31[98544] : CDuiHandlerMain::OnDuiMessage:uID=1109, name=menu_1, msg=1, wParam=0, lParam=0

2.8. 控件的唯一标识和控件查找

DuiVision 库的一个特点就是可以通过 ID 和 name 两种方式灵活的进行控件的查找，每个控件对象创建的时候都会自动分配一个唯一 ID，同时也可以给控件命名，查找一个控件也可以通过 ID 和 name 两种方式进行查找，因为 ID 方式查找不够灵活，所以一般情况下都建议用 name 的方式进行查找，控件的 name 有两种方式可以设置，一种方式是在 xml 文件中定义控件的 name 属性，另一种方式是控件创建之后调用 SetName 函数进行设置。

控件查找可以通过 GetControl 函数进行查找，这个函数在事件处理基类、控件基类、对话框中都有定义。其中控件基类中 GetControl 函数的查找方式是递归查找当前控件的子控件，看是否有对应名字或 ID 的子控件；事件处理类中 GetControl 函数的查找方式是查找此事件处理对象关联的控件对象，然后查找此控件对象或子控件中是否有对应名字或 ID 的控件；对话框类中的 GetControl 函数的查找方式是递归查找当前对话框中的所有控件，看是否有对应名字或 ID 的控件。

2.9. 控件的快捷键和焦点的支持

每个控件可以设置快捷键，设置方法是在 xml 中设置 shortcut 属性，例如下面这个控件设置快捷键为 ESC 键：

```
<imgbtn name="button.close" pos="-45,0,-0,29" skin="IDB_BT_CLOSE" shortcut="ESC" />
```

快捷键的写法是 flag+char 的形式，flag 可以是 CTRL、ALT、SHIFT，分别表示几个控制键是否按下，char 是键的名字，可以用的包括字母、数字，以及下面这些键：

RETURN – 回车

ESC – 取消

BACK – 回退

TAB – TAB 键

SPACE – 空格键

PRIOR – 上翻页

NEXT – 下翻页

END – 到文件尾

HOME – 到文件头

LEFT、UP、RIGHT、DOWN – 几个方向键

PRINT – 打印

INSERT – 插入

DELETE – 删除

F1-F12 – 功能键

如果没有 flag，也可以直接写键的名字。

某个控件如果设置了快捷键，当按下快捷键时候，相当于在控件上点击了一次鼠标，实际动作就是自动触发针对这个控件的一次鼠标按下消息和鼠标放开消息。

DuiVision 的控件支持焦点状态，如果一个控件要支持焦点的话，可以通过设置控件的 tabstop 属性来实现，tabstop 为 1 表示此控件可以处于焦点状态，tabstop 属性也可以通过 API 查询，就是控件的 IsTabStop 函数。一些控件默认是会设置为 tabstop 为 1 的状态的，这些控件包括按钮、检查框、RadioButton、编辑框。

焦点的切换有两种方式，一种是通过键盘操作，TAB 键和 SHIFT+TAB 键分别表示切换到下一个或上一个可以获取焦点的控件上面，另一种方式是鼠标点击了一个控件之后，这个控件就会成为当前的焦点控件。为了区分焦点控件，DuiVision 提供了默认的焦点控件显示方式，就是在控件的内部靠近控件边框位置画灰色的虚线框，有些控件不会采用这样的方式画焦点，例如编辑框处于焦点状态时候会显示编辑框的输入光标，而编辑框失去焦点时候会取消光标的显示，同时会在编辑框内显示出编辑框控件的 tip 信息。

2.10. 控件的 tip 的支持

控件可以设置 tip 属性，也可以通过 API 函数 SetTooltip 来设置。

设置了 tip 之后，当鼠标移动到此控件并停留一段时间，就会显示此控件的 tip 信息，例如下面这个图所示：

2.11. 日志文件定义

DuiVision 提供了日志文件的操作函数，在任何地方都可以通过如下代码调用日志函数来写日志：

```
DuiSystem::LogEvent(LOG_LEVEL_DEBUG, _T("CDuiHandlerTab3::OnDuiMessage:uID=%d, name=%s, msg=%d, wParam=%d, lParam=%d"),
                    uID, strName, Msg, wParam, lParam);
```

此函数的第一个参数是日志的级别，后面的参数类似于 C 语言的 printf 的写法。级别定义如下：

```
#define LOG_LEVEL_DEBUG   0x0001      //调试信息
#define LOG_LEVEL_INFO    0x0002      //一般信息
#define LOG_LEVEL_ERROR   0x0004      //错误信息
#define LOG_LEVEL_CRITICAL 0x0008      //致命信息
```

日志文件默认是在 exe 所在路径下面，日志文件名和日志级别可以在全局资源定义（resource.xml）中定义，定义如下：

```
<!--系统配置-->
<res type="cfg" name="logfile" value="demoui.log" />
<res type="cfg" name="loglevel" value="1" />
```

其中 logfile 表示日志文件的名字，是对应的 exe 文件的路径，loglevel 表示记录的日志的最低级别，日志级别有 4 种级别，分别是 1-调试级别、2-一般级别、4-错误级别、8-致命级别。如果这里的 loglevel 定义的是 1，就表示调试和以上级别的日志都会记录。

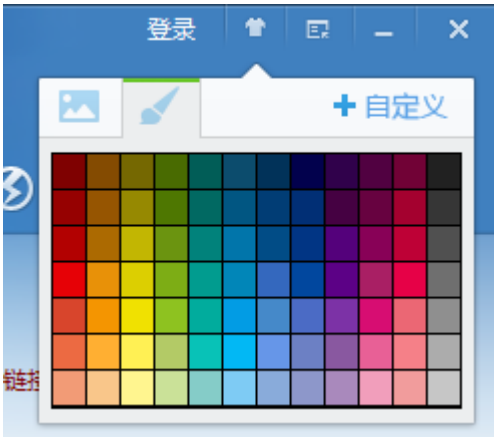
2.12. 任务类

基于 MFC 的界面程序中，如果存在多线程，一般情况下只有主线程（界面线程）可以调用 Windows 窗口相关的函数，否则如果在其他线程中调用了界面函数，很可能会造成异常。为此 DuiVision 库提供了一个任务队列和相应的调度机制，可以将各种任务对象放到任务队列中按顺序执行，通过任务队列，可以做到其他线程和界面线程之间的中转调用，方法是创建任务对象时候指定是需要界面线程处理的任务，则任务调用过程中会通过跨进程的消息将此任务通知界面线程来执行。

DuiVision 中定义了任务的基类和几种派生类，基类是 IBaseTask 类，派生类目前有两种，分别是 DuiSystem 中定义的 CDuiActionTask 类（用于执行菜单等动作），另一种是 DuiSystem 中定义的 CDuiNotifyMsgTask 类，用于执行桌面右下角的弹出提示窗口，这两个任务类的核心执行代码都在 DoAction 函数中，如果需要派生其他的任务类，可以参考者两个类的实现方式。

2.13. 界面皮肤定义

DuiVision 界面库支持定义窗口的背景皮肤，如果某个窗口未指定特殊的背景皮肤，则会使用全局的背景皮肤，通过界面库提供的皮肤窗口可以更改界面皮肤，皮肤窗口如下：



通过皮肤窗口可以将背景皮肤更换为默认的 9 张图片之一，或者是选择的某种颜色，或者选择一个自定义的图片背景。

默认的 9 张背景图片是放在 exe 所在路径的 bkimg 子目录下，图片文件名分别是 SKIN_PIC_0.png-SKIN_PIC_8.png，如果要更改默认的图片，只要在制作安装包时候替换这个目录中相应名字的图片就可以。

皮肤选择窗口是通过皮肤按钮来打开的，只要窗口的某个控件定义的名字是皮肤按钮的名字，就自动具有皮肤按钮的功能，皮肤按钮的名字是 button.skin，例如下面这段对话框 xml 中的定义，就自动会支持皮肤窗口的功能：

```
<imgbtn name="button.skin" pos="-140,0,-111,29" skin="IDB_BT_SKIN"
tip="皮肤" show="1"/>
```

通过皮肤窗口更改了背景皮肤之后，这个改动并不会保存下来，如果下次运行这个程序，就会又回到最初的状态（默认显示的是第一张背景图片），为了将选择的皮肤保存下来，需要

在主程序中写一些代码来实现，例如将选择的皮肤信息保存在注册表中，具体实现方式可以参考 DuiVision 界面库 demo 程序中的 DuiHandler_main.cpp 的 CDuiHandlerTest::ProcessSkinMsg 函数的实现。

2.14. 托盘图标

DuiVision 界面库封装了 Windows 托盘图标的相关操作，可以创建托盘图标，并设置图标文件、托盘的 tip 信息，也可以处理托盘的单击、双击、右键菜单的事件。

通过调用下面的函数可以进行托盘的初始化：

```
DuiSystem::Instance()->InitTray();
```

初始化一般放在主的事件处理类 OnInit 函数中，可以参考 demo 程序的代码。设置托盘的图标文件盒 tip 信息可以调用 DuiSystem 的 SetTrayIcon、SetTrayTip 函数。

托盘的右键操作是打开右键菜单，右键菜单在 resource.xml 中通过 menu_tray 名字的资源项定义具体的菜单 xml 文件。

托盘的左键双击默认动作时打开主窗口，也可以更改为自定义的处理方式，resource.xml 中下面的配置项用于定义托盘双击的动作，如果为 0 就表示执行默认的打开主窗口的动作，如果为 1，则会发送 MSG_TRAY_DBCLICK 消息，通过在事件处理类中响应这个消息，就可以处理双击事件。

```
<res type="cfg" name="trayDbClickMsg" value="0" />
```

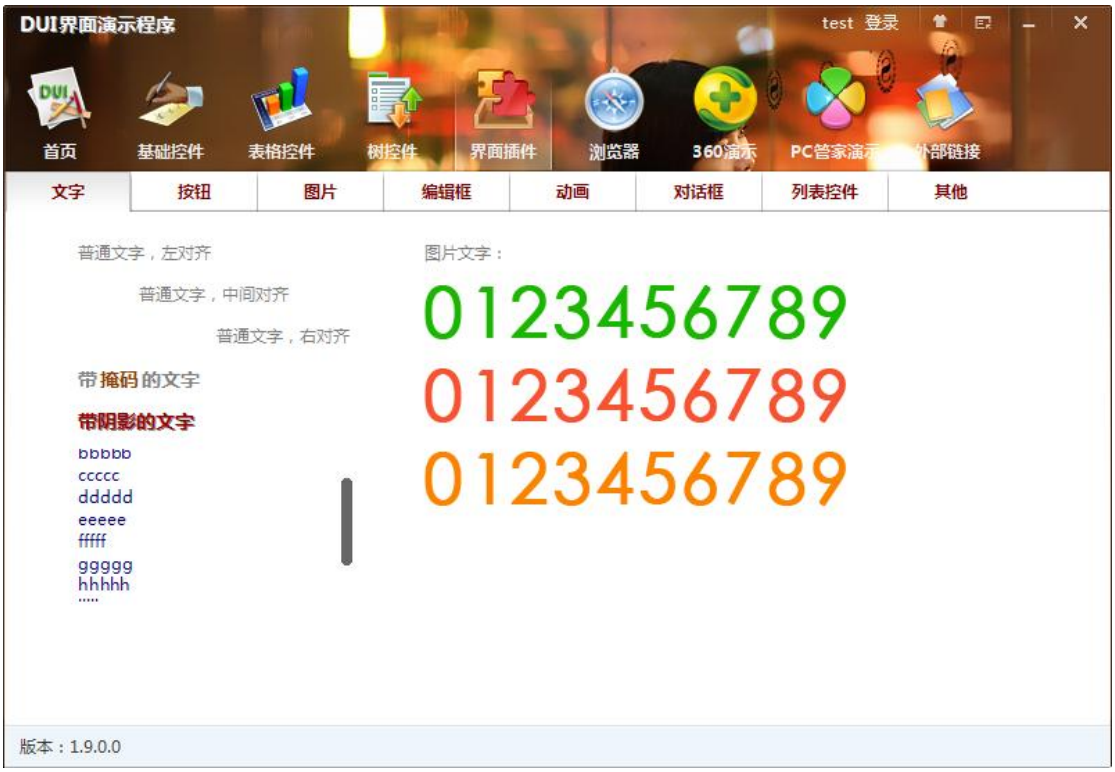
托盘左键的单击事件也会发送一个消息，消息 ID 为 MSG_TRAY_LBUTTONDOWN，通过在事件处理类中响应这个消息，就可以处理单击事件。可以参考 Demo 程序单击和双击事件响应函数。

2.15. 界面插件介绍

DuiVision 界面库支持界面插件，每个界面插件是一个独立的动态库，里面可以封装各种界面，界面插件会对外暴露标准的插件接口，主应用程序可以加载插件，将插件中界面集成到主程序指定的位置，插件中可以有自己的事件处理。通过插件可以将一个大的界面应用程序分离成多个小程序，DuiVision 界面插件的一个优点是基于接口进行集成，所以主程序和界面插件动态库可以使用不同版本的 DuiVision 库进行编译，减少了主程序和插件的版本依赖性，甚至基于其他界面库开发的界面，只要是符合界面插件规范也可以进行集成。

DuiVision 界面插件目前只支持通过 div 控件进行集成，因为 div 控件是一个容器控件，通过设置 div 控件的 plugin 和 plugin-debug 属性，指定插件的动态库文件，主程序在初始化这个 div 时候就会自动加载插件动态库，并将插件中的顶层 div 和主程序中这个 div 关联起来，插件显示的区域就是主程序这个 div 指定的区域。插件显示的效果如下图，下图中当前的这个

界面插件 tab 页中显示的内容都是通过插件显示出来的。



通过 DuiVision 工程向导可以创建出插件工程的框架代码，具体用法也可以参考 Demo 程序中的插件演示页面。

2.16. 界面演示

可以参考 demo 程序，demo 程序的一些界面如下：


DUI界面演示程序
test 登录
首页
基础控件
列表控件
表格控件
树控件
浏览器
360演示
PC管家演示
外部链接
DUI界面库介绍
DuiVision界面库是参考了仿PC管家程序、金山界面库、DuiEngine、DuiLib等多个基于DirectUI的界面库开发的。
DirectUI技术一般是指将所有的界面控件都绘制在一个窗口上，这些控件的逻辑和绘图方式都必须自己进行编写和封装，而不是使用Windows控件，所以这些控件都是无句柄的。
DirectUI技术需要解决的主要问题如下：
1、窗口的子类化，截获窗口的消息；
2、封装自己的控件，并将自己的控件绘制到该窗口上；
3、封装窗口的消息，并分发到自己的控件上，让自己的控件根据消息进行相应和绘制；
4、根据不同的行为发送自定义消息给窗口，以便程序进行调用；
5、一般窗口上控件的组织使用XML来描述。
通常DirectUI的界面库都采用XML配置文件+图片+控制脚本（Lua、Javascript等）的开发方式，非常类似于Web程序的开发方式，当然这里面控制脚本也可以直接使用C++代码来实现。这种开发方式可以大大提高开发效率，将程序员从繁琐的界面工作中解脱出来，并且通过美工的设计，可以使界面更美观。
DuiVision开源下载地址：https://github.com/blueantst/DuiVision
蓝蚂蚁工作室网站：http://www.blueantstudio.net
QQ群：325880743
微信公众号：blueantstudio 或搜索 蓝蚂蚁工作室 或扫描左侧二维码
版本：1.0.0.1


DUI界面演示程序-仿360安全卫士
登录
电脑体检
木马查杀
系统修复
电脑清理
优化加速
软件管家
您的电脑已经 5 天没有体检，建议立即体检
系统可能已经存在大量风险，安全度和性能都在急速下降
建议您每天坚持电脑体检，提高电脑的安全和性能
立即体检
版本：1.0.1.1


DUI界面演示程序
test 登录
首页
基础控件
表格控件
树控件
界面插件
浏览器
360演示
PC管家演示
外部链接
表格控件
Script.NET
蓝蚂蚁工作室开发的脚本语言集成开发环境，支持Tcl、...
已装：2.2.1.0
最新：4.0开源版
125.5M
点击下载
腾讯软件管理
更新内容
1.新的界面，丰富的内容；2.海量软件搜索；3.功能分类清...
已装：8.7.0.1
最新：10.0.0.1
16.5M
一键安装
Notepad++
1.修复功能列表崩溃问题;2.增强分隔符选择
已装：6.5.0.0
最新：6.6.0.0
7.5M
一键安装
迅雷看看
1.修复部分 Bug
已装：4.8.0.0
最新：4.9.14.2
43.5M
一键安装
Axialis IconWorkshop
1.易用性改进
已装：4.9.0.0
最新：6.8.1.2
13M
一键安装
为知笔记
1.改进了博客发布功能;2.改进了编写日记功能
已装：22.9.0.0
最新：24.0.1.1
29.5M
TortoiseGit 64位
暂无版本更新说明
已装：1.8.3.0
最新：1.8.9.0
7.5M
Notepad++
已装：6.5.0.0
版本：1.9.0.0


DUI界面演示程序
test 登录
首页
基础控件
列表控件
表格控件
树控件
浏览器
360演示
PC管家演示
外部链接
蓝蚂蚁工作室
http://www.blueantstudio.net
主页
下载中心
软件介绍
插件
关于
英文版
Search:
软件作品
Script.NET开发平台
Script.NET开源项目
FVision界面类库
金山界面库VC向导
下载中心
Script.NET
FVision
金山界面库VC向导
LuaEditor
blueant
Script.NET V2.2 版本发布
Tcl 脚本开发支持
Python 脚本开发支持
Perl 脚本开发支持
Ruby 脚本开发支持
Lua 脚本开发支持
.....
Script.NET 通用开发平台支持多种
脚本语言的开发、调试、运行 ...
立即下载
版本：1.0.0.1

以上界面中的整体布局和实现方式是，在主窗口 xml 中定义了一个 tabctrl 控件，控件中定义了几个 tab 页，每个 tab 页下面定义具体这个页面的控件，例如第三个页面中定义了一个 listctrl 列表控件和两个图片按钮控件。

用户点击了某个页面上的某个控件之后，需要做一些相应的处理，这些处理需要写在事件处理类中，demo 程序只定义了一个事件处理类，可以处理所有与页面的事件，也可以给某个 tab 页单独指定事件处理类，只要将处理某个页面的事件处理对象注册到这个页面控件就可以，对于其他没有注册事件处理对象的页面中的事件，会自动遍历每个注册的事件处理对象进行处理。具体实现请参考 demo 程序代码。

3. 控件说明

3.1. DUI 基类

类名：CDuiObject

控件名：无（基类，没有实体控件）

说明：所有 DUI 的基类（包括控件、对话框、菜单等）

属性：无

函数：

函数	是否虚函数	说明
IsClass	是	判断是否此类

GetObjectClass	是	获取类名
BaseObjectClassName	是	获取基类名
GetID	否	获取对象 ID
GetName	否	获取对象名字
IsThisObject	否	根据 ID 或名字判断是否此对象，ID 或名字任意一个匹配上就可以
RegisterHandler	否	注册事件处理对象
GetDuiHandler	否	获取事件处理对象指针
GetRect	否	获取控件的左上角、右下角坐标
ParseDuiString	否	解析字符串，替换其中的替换内容（用[]包围的定义内容），替换内容是在 resource.xml 或字符串定义文件中定义的字符串内容
ParseKeyCode	否	根据传入的字符串获取对应的键盘码，例如 CTRL+F1 会被转换为 0x11,0x70

3.2. DUI 控件基础类

类名：CControlBase

控件名：无（基类，没有实体控件）

说明：所有界面控件的基类

属性：无

属性名	类型	说明
show	1|0	控件是否可见
disable	1|0	控件是否被禁用
pos	位置	控件的位置坐标，可以是左上角坐标，例如 10,10，也可以是左上角+右下角坐标，例如 10,10,200,100。 支持正值和负值，正值表示从父控件左上角开始计算的值，负值表示从父控件右下角开始计算的值，例如-10,10 表示从右边往左 10 像素，从上往下 10 像素的位置。 也可以支持从父控件中间开始计算的坐标，用|表示中间位置 ,例如|10,|-10 表示横向中间向右 10 像素，

		纵向中间向上 10 像素的位置。
width	数字	控件宽度
height	数字	控件高度
action	字符串	控件的动作字符串 ,表示控件点击之后执行的动作，有几种类型： 1、dlg:xxxx，表示显示指定的对话框，显示的对话框可以是一个 xml 定义文件 ,也可以是一个定义(从 resource 资源定义文件中查找具体的 xml 定义文件) 2、menu:xxxx.xml，表示显示指定的菜单，菜单定义文件是指定的 xml 文件 3、link:url，表示使用浏览器或其他默认程序打开指定的链接或文件 4、run:xxx.exe|param，表示运行指定的程序，可以传递命令行，执行的程序后面用|分隔，|之后的表示传递的命令行参数
menupos	位置	菜单显示的位置坐标，必须是 x,y 的形式，表示菜单左上角坐标，例如 10,10。 支持正值和负值，正值表示从控件左上角开始计算的值，负值表示从父控件右下角开始计算的值，例如-10,10 表示从右边往左 10 像素，从上往下 10 像素的位置。 也可以支持从控件中间开始计算的坐标，用|表示中间位置，例如|10,|-10 表示横向中间向右 10 像素，纵向中间向上 10 像素的位置。
tip	字符串	定义鼠标移动到空间上过一段时间会出现的 tip 提示信息，tip 信息只有对话框的基础控件可以定义，其他控件即使定义了也没有效果
response	1|0	控件是否可以响应鼠标事件，如果不希望控件响应鼠标事件，可以设置此属性为 0
tabstop	1|0	控件是否可以响应焦点切换的事件，也就是按键盘 tab 键能否将焦点切换到此控件

taskmsg	1|0	调用事件响应函数时候是否采用任务方式，对于可能产生阻塞或耗时比较长的处理操作应该采用任务方式处理，避免造成界面不响应或界面异常
img-ecm	1|0	是否使用图片自身的颜色管理信息，默认为 0，表示加载图片文件时候使用系统的颜色管理信息而不使用图片自身的颜色管理信息，因为 XP SP1 以前的操作系统自带的 GDI+模块可能不支持图片自身的颜色管理信息，因此如果设置为 1 的话，在 XP SP1 以前的系统下运行图片可能无法正常显示
shortcut	字符串	定义控件的快捷键，快捷键字符串参考第二章的相应章节说明，当按下对应快捷键的时候，会自动触发此控件产生一个鼠标按下和鼠标放开的事件，模拟点击了此控件

函数：

函数	是否虚函数	说明
GetParent	否	获取父控件对象指针
GetControl	否	根据 ID 或 name 获取控件对象指针
AddControl	否	添加控件
RemoveControl	否	删除控件
GetControls	否	获取所有控件对象的列表
GetParentDialog	否	获取控件所在的对话框的指针
OnMessage	是	控件的消息处理函数
SendMessage	否	发送通知消息
SetRect	是	设置控件的位置
GetRect	是	获取控件的位置
OnPositionChange	是	刷新控件的位置信息
SetPosStr	否	设置控件的位置字符串
SetVisible	否	设置控件的可见性
GetVisible	否	获取控件的可见性
SetControlWndVisible	是	设置控件中的 Windows 原生控件是否可见的状

		态，由 Panel 对象中调用，对于 edit、activex 等使用了 Windows 原生控件的类需要重载此函数，并正确的进行原生控件的显示和隐藏
SetDisable	否	设置控件是否禁用
GetDisable	否	获取控件是否禁用
SetTabStop	否	设置控件是否能响应 tab 键
IsTabStop	否	获取控件是否能响应 tab 键
SetTooltip	否	设置控件的 tip 信息
GetTooltip	否	获取控件的 tip 信息
SetAction	否	设置控件的动作字符串
GetAction	否	获取控件的动作字符串
SetRresponse	否	获取控件是否可响应鼠标事件
GetRresponse	否	设置控件是否可响应鼠标事件
PtInRect	是	判断坐标是否在控件范围内
UpdateControl	否	刷新控件显示
SetWindowFocus	是	设置窗口焦点

3.3. DUI 文字控件基础类

类名：CControlBaseFont

控件名：无（基类，没有实体控件）

说明：所有界面控件中有显示文字信息的控件类的基类

属性：无

属性名	类型	说明
title	字符串	控件的显示标题
font	字体	控件的字体，可以引用资源定义中定义的某个字体，默认字体是 default
fontname	字符串	直接指定某种字体
fontwidth	数字	直接指定字体宽度
height	数字	控件高度
valign	枚举	文字的垂直对齐模式，top、middle、bottom
align	枚举	文字的水平对齐模式，left、center、right
skin	皮肤	控件的皮肤名，引用资源定义中的统一皮肤定义

image	图片	控件的图片，有 3 种定义方式： 1、图片文件：xxx.png，xxx.jpg 等，是相对 exe 的路径 2、图片资源：如果 image 不是文件格式，则认为是资源 ID，到程序的内嵌资源中去查找对应的图片资源 3、皮肤方式：skin:xxxx，如果是 skin:开始，则认为是皮肤格式，后面是皮肤名，到全局皮肤定义中查找具体图片
img-count	数字	定义图片的切片个数，如果一个图片文件中横向包含了多个等宽的小图片，根据这个定义，控件可以知道到底有几个小图片，并按照图片个数进行正确的切片

函数：

函数	是否虚函数	说明
SetTitle	否	设置标题文字
GetTitle	是	获取标题文字
SetAlignment	是	设置控件的水平对齐方式
SetVAlignment	是	设置控件的垂直对齐方式
SetImage	否	设置控件的图片
SetBitmapCount	否	设置控件图片的水平方向小图片个数

3.4. DUI 对话框

类名：CDlgBase

控件名：无

说明：对话框类，所有对话框都直接创建一个 CDlgBase 对象就可以。

代码中如果需要创建一个对话框，一般建议使用 DuiSystem 类中封装的若干对话框相关的函数来操作，包括创建对话框、删除对话框、根据对话框名获取对话框指针、显示通用对话框。

属性：

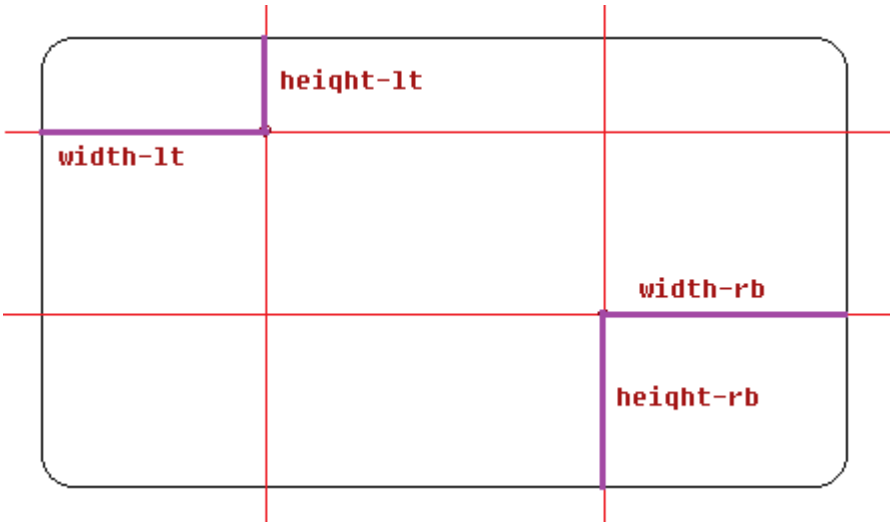
属性名	类型	说明
width	数字	窗口宽度

height	数字	窗口高度
resize	0|1	1 表示窗口可以改变大小
frame	字符串	窗口的 frame 层图片，frame 层是一个可选的半透明 Alpha 图片层，一般设置的这个图片是用于和背景图片进行 Alpha 混合，这一层的图片中每个像素都包含了自身颜色和透明度属性，通过透明度属性可以将背景图片进行半透明处理，默认只有主窗口设置了这个 frame 层图片 ,并且默认的 frame 图片是一个透明度渐变的 PNG 图片 ,从顶端的 100%透明到底端的完全不透明
framesize	数字	窗口的 frame 层图片的边框宽度，非九宫格方式有效
width-lt	数字	窗口的 frame 层图片的九宫格左上角位置距离边框的宽度
height-lt	数字	窗口的 frame 层图片的九宫格左上角位置距离边框的高度
width-rb	数字	窗口的 frame 层图片的九宫格右下角位置距离边框的宽度
height-rb	数字	窗口的 frame 层图片的九宫格右下角位置距离边框的高度
bkimg	字符串	窗口的背景图片，如果指定了就使用指定的背景图片，否则使用全局设置的背景图片
crbk	颜色	窗口的背景颜色，如果未指定背景图片，但指定了背景颜色，就使用指定的背景颜色，否则使用全局设置的背景图片
appwin	0|1	此窗口是否会显示在 Windows 任务栏中显示，见下面的截图说明
translucent	数字	窗口的整体透明度，取值范围是 1-255,1 表示全透明，255 表示不透明

说明：

1）九宫格方式 frame 层的说明：对于复杂的背景 frame 层图片，其所有边框宽度并不是固

定的，但一般都可以用九宫格方式来切分，就是把背景 frame 图片横向、纵向各用两条线切分，一共切分成九部分，应用时候四个角的图片大小是按照原始大小应用到窗口中的，其余几部分都会进行拉伸，对于这种方式，只要描述出九宫格的左上角和右下角坐标位置就可以，对应的就是 width-lt、height-lt、width-rb、height-rb 这 4 个属性。



2）appwin 属性的说明：下面截图中右边的任务栏窗口就是因为此窗口设置为 appwin 属性为 1 才会在任务栏中单独显示出来：

函数：

函数	是否虚函数	说明
SetXmlFile	否	设置对话框加载的 xml 文件

GetControl	否	根据 ID 或 name 获取对应的控件指针
DoOK	否	对话框的确定
DoCancel	否	对话框的取消
DoClose	否	对话框的关闭
SetControlVisible	是	设置指定控件的可见性
SetControlDisable	是	设置指定控件是否禁用
OpenDlgPopup	否	打开一个弹出框
CloseDlgPopup	否	关闭弹出框

界面示例：

3.5. DUI 菜单

类名：CMenuEx

控件名：无

说明：菜单有两种显示的位置，一种是在窗口顶部某个按钮点击后可以下拉一个菜单，另一种是托盘图标的右键菜单。

窗口中的菜单定义方式是 xml 文件中设置某个按钮的 action 属性，以 menu:开头，后面是菜单的 XML 文件名或 XML 定义名，例如下面这样定义：

```
<imgbtn name="button.menu" pos="-110,0,-77,29" skin="IDB_BT_MENU" tip="菜单" action="menu:mainmenu.xml"/>
```

托盘菜单默认是按照 resource.xml 中定义的 menu_tray 指向的 XML 文件来加载菜单。

两种方式加载的菜单定义 XML 文件格式都是相同的，参考前面 XML 说明章节的示例。

属性：

属性名	类型	说明
width	数字	菜单窗口宽度
item-height	数字	每个菜单项的高度
left	数字	菜单左侧图标区的宽度
sep-height	数字	菜单分隔线的高度

函数：

函数	是否虚函数	说明
LoadXmlFile	否	加载菜单 XML 文件
AddMenu	否	动态添加菜单项
AddSeparator	否	动态添加菜单分隔线

界面示例：

3.6. DUI 菜单项

类名：CMenuItem

控件名：menuitem

说明:菜单中加载的每个菜单项的控件，CControlBaseFont 中的所有属性和函数都可以使用。

属性：

属性名	类型	说明
seperator	0|1	是否分隔线
select	0|1	是否选择（如果是 checkbox 或 radiobutton 类型的菜单项，此属性必须设置为 1）
check	0|1	是否处于选中状态
group	字符串	广播按钮所属的组名，相同组名的广播按钮是属于一组的，可以联动，一组中只有一个会处于选中状态
value	字符串	广播按钮的值，一组广播按钮中的多个按钮值是不一样的，当获取这一组广播按钮的值时候，获取的就是选中的按钮的值
menu	字符串	引用其他的菜单的名字（通过资源定义可以找到的菜单的名字），设置了这个属性，则会将对应的菜单嵌入当前菜单中

函数：

函数	是否虚函数	说明
SetCheck	否	设置是否选择
GetCheck	否	获取是否选择的状态
IsSeparator	否	判断是否分隔线

SetGroupName	否	设置广播按钮组的名字
GetGroupName	否	获取广播按钮组的名字
GetValue	否	获取广播按钮的值
GetGroupValue	否	获取广播按钮组的值
ResetGroupCheck	否	刷新父控件下面所有同一个组的 RadioButton 控件的状态

3.7. DUI TabCtrl 控件

类名：CDuiTanCtrl

控件名：tabctrl

说明：多个 Tab 页的切换控件。

属性：

属性名	类型	说明
image	字符串	Tab 页的图标图片，可以试并排多个图片，通过 img-count 可以设置图片个数
img-sep	字符串	Tab 页之间的分割线图片
img-hover	字符串	鼠标移动到 tab 页时候的 tab 状态图片，并排的两张图片，分别是鼠标移动和鼠标点击状态的图片
item-width	数字	每个 tab 页的宽度，如果没有设置则按照 img-hover 的图片宽度
tab-height	数字	Tab 页图标部分高度，如果没有设置则按照 img-hover 的图片高度
animate	0|1	Tab 页切换时候是否显示动画效果
animate-count	数字	Tab 页切换动画的帧数，默认是 10 帧
crtext	颜色	Tab 页标题的文字颜色

函数：

函数	是否虚函数	说明
InsertItem	否	添加一个 tab 页
GetItemIndex	否	根据 tab 页名字获取索引
GetItemInfo	否	根据 tab 页索引获取 tab 页信息
RefreshItems	否	刷新所有 tab 页，重新计算位置信息

DeleteItem	否	根据索引或名字删除 tab 页
SetSelectItem	否	设置当前激活的 tab 页
SetItemVisible	否	设置 tab 页是否可见
LoadTabXml	否	从 XML 文件加载 tab 页

界面示例：

3.8. DUI Tab 控件

类名：CDuiTanCtrl

控件名：tab

说明：Tab 页控件的下层控件，每个 tab 表示一个页面。

属性：

属性名	类型	说明
name	字符串	可以根据名字获取到 tab 页对应的 DuiPanle 控件对象
title	字符串	Tab 页标题
active	true|false	此 Tab 页是否激活
show	0|1 或 true|false	此 Tab 页是否显示
visible	0|1 或 true|false	和 show 属性相同
image	字符串	此 tab 页的页签图片，是一个 3 张横向切片图组成图片，3 张图分别表示正常、鼠标移动、鼠标点击 3 种状态下的 tab 页签
img-index	数字	如果是多张图片组成的大图，则表示图片的索引
img-count	数字	如果是多张图片组成的大图，则表示图片的个数
div	字符串	如果一个 tab 页的内容想写在一个单独的 xml 文件中，可以在此处定义 xml 文件名
pos	字符串	Tab 页的位置信息字符串

outlink	0|1 或 true|false	表示此 tab 页是否是外部链接，外部链接是指点击之后打开一个网页或独立的窗口或程序，对于外部链接，是不会显示鼠标点击到此 tab 页的状态的
action	字符串	点击此 tab 页时候执行的动作
scroll	0|1	此 tab 页是否支持滚动

3.9. DUI Panel 控件

类名：CDuiPanel

控件名：div

说明：Panel 控件是一种虚拟的容器控件，本身不会有任何显示界面，但可以在下层包含若干其他的控件，通过设置 Panel 的可见性可以控制下层所有控件的可见性。Panel 类是从 CControlBaseFont 派生的，所以基类控件具有的属性和函数也可以使用。

属性：

属性名	类型	说明
xml	字符串	Panel 控件可以通过加载一个 xml 文件来加载下层控件，此处是指 xml 文件名
img-scroll	字符串	滚动条图片
scroll-width	数字	滚动条宽度
scroll	0|1	是否允许滚动
plugin	0|1	DuiVision 界面插件的 release 版本动态库文件，具体插件说明请参考插件章节，主程序是 release 版本编译时候此属性才有效
Plugin-debug	0|1	DuiVision 界面插件的 debug 版本动态库文件，具体插件说明请参考插件章节，主程序是 debug 版本编译时候此属性才有效

函数：

函数	是否虚函数	说明
LoadXmlFile	否	加载 XML 文件
SetVirtualHeight	否	设置 div 的虚拟高度
CalcVirtualHeight	否	自动计算 div 的虚拟高度，按照 div 中所有子控件的位置进行计算，找到位置最大值作为 div 的虚拟

		高度
SetEnableScroll	否	设置 div 是否允许滚动
GetEnableScroll	否	获取 div 是否允许滚动

3.10. DUI 文字控件

类名：CDuiText

控件名：text

说明：显示指定的文字，支持文字中指定的字符串设置不同的颜色，是从 CControlBaseFont 派生的，所以基类控件具有的属性和函数也可以使用。

属性：

属性名	类型	说明
crtext	颜色	文字颜色
crhover	颜色	鼠标移动时候的颜色
crshadow	颜色	文字阴影颜色
crback	颜色	背景颜色
crmark	颜色	特殊显示的文字颜色
mask	字符串	特殊显示的字符串
img-scroll	字符串	滚动条图片
scroll-width	数字	滚动条宽度
bk-transparent	数字	背景透明度，未指定背景颜色时才有效，0 表示全透明，100 表示不透明，默认为 0

函数：

函数	是否虚函数	说明
SetTitleMarkText	否	设置整个字符串和特殊显示部分字符串

界面示例：

3.11. DUI 图片控件

类名：CDuiPicture

控件名：img

说明：显示指定的图片，是从 CControlBaseFont 派生的，所以基类控件具有的属性和函数也可以使用。

属性：

属性名	类型	说明
mode	normal| tile| extrude| frame|mid	图片的现实模式，枚举类型，分别表示正常显示图片、平铺、拉伸、显示图片边框、九宫格方式显示图片
framesize	数字	显示图片边框的模式下，表示边框的宽度
width-lt	数字	九宫格模式下的九宫格左上角位置距离边框的宽度
height-lt	数字	九宫格模式下的九宫格左上角位置距离边框的高度
width-rb	数字	九宫格模式下的九宫格右下角位置距离边框的宽度
height-rb	数字	九宫格模式下的九宫格右下角位置距离边框的高度

函数：

函数	是否虚函数	说明
SetShowMode	否	设置图片显示模式
SetShowModeMID	否	设置九宫格方式的图片显示模式

界面示例：

九宫格图片显示

九宫格图片是将一个图片横向和纵向各用两条线分割，这样可以将一个图片分割成9部分，分隔之后四个角大小不变，其他块进行拉伸，从而可以将原图拉伸成任意大小的背景图。
这种显示方式可用于窗口背景显示或图片的显示。
DuiVision中的九宫格图片显示时候，只需要指定左上角和右下角分割点相对于左上角和右下角的宽度、高度就可以，属性名分别是：
width-lt,height-lt,width-rb,height-rb

3.12. DUI 动画图片控件

类名：CDuiAnimateImage

控件名：animateimg

说明：显示动画图片，是从 CControlBaseFont 派生的，所以基类控件具有的属性和函数也可以使用。此控件设置的图片按照横向若干张小图片拼接的方式，小图片的宽度必须相同，控件可以设置小图片的个数，控件内可以自动启动一个定时器，按照顺序循环定时显示小图片，达到动画的效果。

属性：

属性名	类型	说明
index	数字	当前显示第几张图片
maxindex	数字	小图片的个数
timer-count	数字	图片切换的定时器周期数，每个周期的时间是 30 毫秒，也就是图片切换的时间是多少个 30 毫秒
run	0|1	表示是否启动动画，启动之后动画图片可以自己按照设置的定时周期进行变化

函数：

函数	是否虚函数	说明
SetRun	否	设置是否启动动画
SetTimerCount	否	设置动画的图片切换周期

界面示例（动画图片的原图）：

上面这个图片是一个包含 4 个小图片的动画图片。

3.13. DUI 按钮控件

类名：CDuiButton、CImageButton

控件名：button、imgbtn

说明：显示按钮，是从 CControlBaseFont 派生的，所以基类控件具有的属性和函数也可以使用。按钮的背景图片是由 4 张切图拼接成的大图片，分别是普通状态、鼠标移动状态、鼠标按下状态、禁用状态的图片，背景图片设置可以通过 skin 或 image 属性设置。

按钮控件有两种控件名，button 和 imgbtn，唯一的差别是 imgbtn 默认会启用渐变效果定时器，并且渐变效果的帧数默认设置为 10 帧。

属性：

属性名	类型	说明
crtext	颜色	按钮文字的颜色
animate	0|1	是否启用渐变效果定时器
maxindex	0|1	渐变效果的最大帧数
img-btn	字符串	图片按钮的图片文件
showfocus	0|1	控件处于焦点状态时是否显示焦点虚线框（默认是显示）

函数：

函数	是否虚函数	说明
SetMaxIndex	否	设置渐变效果的最大帧数

界面示例：

3.14. DUI 文字按钮控件

类名：CTextButton

控件名：textbtn

说明:文字按钮是从 CControlBaseFont 派生的 ,所以基类控件具有的属性和函数也可以使用。此控件显示效果是一段可点击的文字，文字可以设置普通状态、鼠标移动状态、鼠标按下状态、禁用状态的颜色。

属性：

属性名	类型	说明
crtext	颜色	按钮文字的颜色
crhover	颜色	鼠标移动时候的颜色
crpush	颜色	鼠标点击时候的颜色
crdisable	颜色	禁用状态的颜色

界面示例：

文字按钮

3.15. DUI 链接按钮控件

类名：CLinkButton

控件名：linkbtn

说明 链接按钮是从 CControlBaseFont 派生的 ,所以基类控件具有的属性和函数也可以使用。此控件显示效果是一段可点击的文字，文字可以设置普通状态、鼠标移动状态、鼠标按下状态、禁用状态的颜色，文字点击之后可以打开一个网页或文件。

属性：

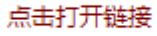
属性名	类型	说明
crtext	颜色	按钮文字的颜色

crhover	颜色	鼠标移动时候的颜色
crpush	颜色	鼠标点击时候的颜色
crdisable	颜色	禁用状态的颜色
href	字符串	链接 URL

函数：

函数	是否虚函数	说明
SetLink	否	设置链接 URL
GetLink	否	获取链接 URL

界面示例：



3.16. DUI 隐藏按钮控件

类名：CHideButton

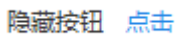
控件名：hidebtn

说明：文字按钮是从 CControlBaseFont 派生的，所以基类控件具有的属性和函数也可以使用。此控件显示效果是一段文字右边有一段默认隐藏的文字链接，正常情况下看不到右边的文字链接，鼠标移动到左边的文字时候才能看到右边的文字链接。文字可以设置普通状态、鼠标移动状态、鼠标按下状态、禁用状态的颜色。

属性：

属性名	类型	说明
crtext	颜色	按钮文字的颜色
crhover	颜色	鼠标移动时候的颜色
crpush	颜色	鼠标点击时候的颜色
crdisable	颜色	禁用状态的颜色
crtip	颜色	隐藏链接的文字颜色
text	字符串	隐藏链接的字符串显示内容

界面示例：



3.17. DUI 检查框控件

类名：CCheckButton

控件名：chkbtn

说明：显示检查框，是从 CControlBaseFont 派生的，所以基类控件具有的属性和函数也可以使用。

属性：

属性名	类型	说明
check	true|false	检查框是否处于选择状态
crtext	颜色	文字颜色
showfocus	0|1	控件处于焦点状态时是否显示焦点虚线框（默认是显示）

函数：

函数	是否虚函数	说明
SetCheck	否	设置检查框的值
GetCheck	否	获取检查框的值
SetTextColor	否	设置文字颜色

界面示例：

3.18. DUI 广播按钮控件

类名：CDuiRadioButton

控件名：radiobtn

说明：显示广播按钮，是从 CControlBaseFont 派生的，所以基类控件具有的属性和函数也可以使用。

属性：

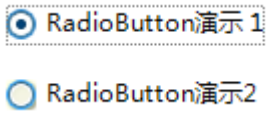
属性名	类型	说明
check	true|false	广播按钮是否处于选中状态
crtext	颜色	文字颜色
group	字符串	广播按钮所属的组名，相同组名的广播按钮是属于一组的，可以联动，一组中只有一个会处于选中状态
value	字符串	广播按钮的值，一组广播按钮中的多个按钮值是不

		一样的，当获取这一组广播按钮的值时候，获取的就是选中的按钮的值
showfocus	0|1	控件处于焦点状态时是否显示焦点虚线框（默认是显示）

函数：

函数	是否虚函数	说明
SetCheck	否	设置检查框的值
GetCheck	否	获取检查框的值
SetGroupName	否	设置广播按钮组的名字
GetGroupName	否	获取广播按钮组的名字
GetValue	否	获取广播按钮的值
GetGroupValue	否	获取广播按钮组的值
ResetGroupCheck	否	刷新父控件下面所有同一个组的 RadioButton 控件的状态
SetTextColor	否	设置文字颜色

界面示例：



3.19. DUI 进度条控件

类名：CDuiProgress

控件名：progress

说明：显示进度条，是从 CControlBaseFont 派生的，所以基类控件具有的属性和函数也可以使用。

属性：

属性名	类型	说明
value	数字	进度值，最大值不能超过 max-value 的值
max-value	数字	最大的进度值，默认为 100
timer-count	数字	定时器定时多少次，自动增长的进度值加 1
run	true|false	进度条是否自动运行（循环增长）

img-back	字符串	背景图片
img-fore	字符串	前景图片
head-len	数字	左右两侧圆角部分长度，如果此长度不为 0，表示头部是圆角的进度条，画图时候头部圆角按照实际图片画图，其余部分则拉伸显示
show-text	0|1	是否显示进度条的文字，如果显示文字，可以通过 align 和 valign 属性设置文字的对齐方式，显示的文字内容是前缀+当前进度值，如果进度最大值是 100，则自动在当前进度值后面加%
crtext	颜色	进度条文字的颜色
title	字符串	进度条文字的前缀，例如 title 设置为”进度：“ ，则显示出来的整体的进度条文字是”进度：xx%“

函数：

函数	是否虚函数	说明
SetProgress	否	设置进度条的值
GetProgress	否	获取进度条的值
SetMaxProgress	否	设置进度条的最大值
GetMaxProgress	否	获取进度条的最大值
SetRun	否	设置自动运行

界面示例：

3.20. DUI 输入框控件

类名：CDuiEdit

控件名：edit

说明：输入框控件，是从 CControlBaseFont 派生的，所以基类控件具有的属性和函数也可以使用。

属性：

属性名	类型	说明
password	0|1	是否是密码输入框
multiline	0|1	是否多行输入框
autohscroll	0|1	水平宽度超出是否可以滚动显示
autovscroll	0|1	垂直高度超出是否可以滚动显示
number	0|1	是否只能输入数字
readonly	0|1	是否只读
maxchar	数字	最多可以输入多少个字符
small-image	字符串	输入框右边显示的小图片

函数：

函数	是否虚函数	说明
GetEditText	否	获取编辑框文字
SetSmallBitmap	否	设置小图片

界面示例：

3.21. DUI 下拉列表控件

类名：CDuiComboBox

控件名：combobox

说明：下拉列表控件，是从 CDuiEdit 派生的，所以基类控件具有的属性和函数也可以使用。

属性：

属性名	类型	说明

value	字符串	下拉列表当前选择项的值
head-image	字符串	下拉列表项中左边部分图片的掩码图片
del-image	字符串	下拉列表项中每一行的删除按钮，如果没有设置则不能由用户删除列表项
xml	字符串	指定下拉列表内容定义的 xml 文件，xml 文件中的下拉列表项描述的格式和非独立 xml 文件方式定义的格式完全相同，如果不指定 xml 文件，则使用当前控件属性的下级节点定义的 xml 内容
crtext	颜色	列表项的颜色，如果是名字和描述两行内容，则表示名字的颜色
crdesc	颜色	名字和描述两行内容的情况下，表示描述部分的颜色
crhover	颜色	当前选中的行（鼠标移动的行）的颜色

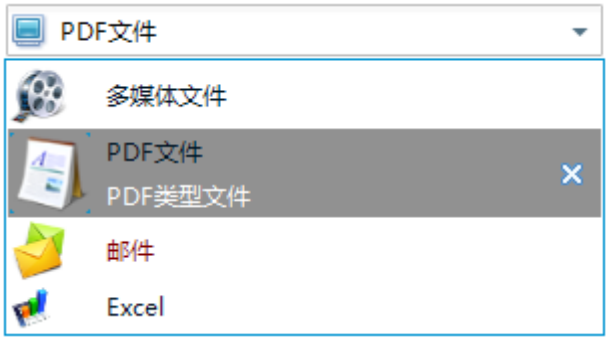
Combobox 的下来列表内容有三种定义的方式，分别是在控件 xml 的下级节点定义、通过独立 xml 文件定义、调用控件的 AddItem 函数通过程序添加，xml 定义方式中下拉项的 xml 节点名是 item，节点属性如下表：

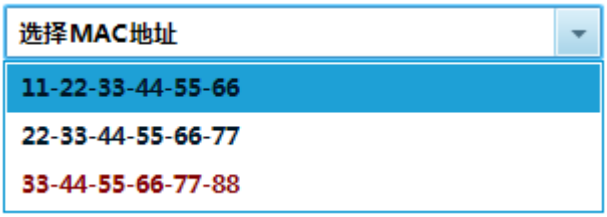
属性名	类型	说明
name	字符串	列表项名字
desc	字符串	列表项描述。 如果所有列表项的描述字符串都为空，则下拉列表自动调整为单行显示； 否则当鼠标移动到某一项时候此列表项显示为两行（名字和描述各一行），其他列表项则只显示一行（名字）
value	字符串	列表项的值
image	字符串	列表项中左边部分图片
crtext	颜色	列表项的颜色，如果是名字和描述两行内容，则表示名字的颜色
crdesc	颜色	名字和描述两行内容的情况下，表示描述部分的颜色

函数：

函数	是否虚函数	说明
SetComboValue	否	设置下拉列表的值
GetComboValue	否	获取下拉列表的值
GetItemCount	否	获取下拉列表项个数
AddItem	否	添加列表项
ClearItems	否	清空列表项

界面示例：





3.22. DUI 列表控件

类名：CDuiListCtrl

控件名：listctrl

说明：列表控件，是从 CDuiPanel 派生的，所以基类控件具有的属性和函数也可以使用。

属性：

属性名	类型	说明
img-scroll	字符串	滚动条图片
img-sep	字符串	行分隔线图片
font-title	字符串	标题行字体
crtext	颜色	正文文字颜色

crhover	颜色	鼠标移动时候的文字颜色
crpush	颜色	鼠标点击的行的文字颜色
crtitle	颜色	标题颜色
crsep	颜色	分割线颜色
crrowhover	颜色	鼠标移动时候行的背景颜色，可以设置透明度
img-width	数字	图片宽度
row-height	数字	行高度
bk-transparent	数字	背景透明度，0-100，0 表示全透明，100 表示不透明

函数：

函数	是否虚函数	说明
InsertItem	否	添加行
GetItemInfo	否	根据行索引号获取行的数据结构
ClearItems	否	清空所有行

界面示例：

3.23. DUI 表格控件

类名：CDuiGridCtrl

控件名：gridctrl

说明：表格控件，是从 CDuiPanel 派生的，所以基类控件具有的属性和函数也可以使用。

属性：

属性名	类型	说明
img-scroll	字符串	滚动条图片
img-sep	字符串	行分隔线图片
font-title	字符串	标题行字体
crtext	颜色	正文文字颜色
crhover	颜色	鼠标移动时候的文字颜色
crpush	颜色	鼠标点击的行的文字颜色
crtitle	颜色	标题颜色
crsep	颜色	分割线颜色
crrowhover	颜色	鼠标移动时候行的背景颜色，可以设置透明度
img-width	数字	图片宽度
row-height	数字	行高度
bk-transparent	数字	背景透明度，0-100，0 表示全透明，100 表示不透明
row	子节点	表示表格的行节点，具体属性参见行节点表

行属性：

属性名	类型	说明
id	字符串	行 ID，可用于定位
check	0|1	是否显示行左侧的检查框
image	字符串	行左侧的图片

right-img	字符串	行右侧的图片
crtext	颜色	正文文字颜色
item	子节点	表示此行的单元格节点，具体属性参见单元格节点表

单元格属性：

属性名	类型	说明
title	字符串	单元格标题
content	字符串	单元格内容
image	字符串	单元格图片
link	字符串	单元格链接文字
linkaction	字符串	单元格链接动作
crtext	颜色	正文文字颜色
font-title	0|1	是否使用标题字体显示单元格文字

单元格可以添加子控件，XML 中只要在单元格的 item 节点下面添加控件子节点就可以。

函数：

函数	是否虚函数	说明
InsertItem	否	添加行
GetItemInfo	否	根据行索引号获取行的数据结构
ClearItems	否	清空所有行
AddSubItemControl	否	添加单元格子控件
DeleteSubItemControl	否	删除单元格子控件

界面示例：

3.24. DUI 树控件

类名：CDuiTreeCtrl

控件名：treectrl

说明：树控件，是从 CDuiPanel 派生的，所以基类控件具有的属性和函数也可以使用。

属性：

属性名	类型	说明
img-scroll	字符串	滚动条图片
img-sep	字符串	行分隔线图片
img-check	字符串	节点左侧的选择框图片
img-collapse	字符串	行缩放图片
img-toggle	字符串	树节点收缩图片
font-title	字符串	标题行字体
crtext	颜色	正文文字颜色
crhover	颜色	鼠标移动时候的文字颜色
crpush	颜色	鼠标点击的行的文字颜色
crtitle	颜色	标题颜色
crsep	颜色	分割线颜色
img-width	数字	图片宽度
row-height	数字	行高度
bk-transparent	数字	背景透明度，0-100，0 表示全透明，100 表示不透明

node	子节点	树节点

树节点属性：

属性名	类型	说明
id	字符串	行 ID，可用于定位
check	0|1	是否显示行左侧的检查框
image	字符串	行左侧的图片
right-img	字符串	行右侧的图片
crtext	颜色	正文文字颜色
collapse	0|1	当前是否处于收缩状态
item	子节点	表示此行的单元格节点，具体属性参见单元格节点表

单元格属性：

属性名	类型	说明
title	字符串	单元格标题
content	字符串	单元格内容
image	字符串	单元格图片
link	字符串	单元格链接文字
linkaction	字符串	单元格链接动作
crtext	颜色	正文文字颜色
font-title	0|1	是否使用标题字体显示单元格文字
collapse	0|1	此单元格是否可以用于收缩和展开子节点

单元格可以添加子控件，XML 中只要在单元格的 item 节点下面添加控件子节点就可以。

函数：

函数	是否虚函数	说明
InsertNode	否	添加树节点
GetNodeInfo	否	根据节点句柄获取节点的数据结构
ClearNodes	否	清空所有节点
AddSubItemControl	否	添加单元格子控件
DeleteSubItemControl	否	删除单元格子控件

界面示例：

3.25. DUI 原生 Windows 控件

类名：CDuiNativeWnd

控件名：nativewnd

说明：Windows 原生控件，用于 DuiVision 库中没有的控件，但标准 windows 控件有，或者有自己扩展的 Windows 控件(从 CWnd 类派生的或者其他的 Windows 控件都可以),则可以通过原生控件的方式来使用，使用方法是在 xml 中定义原生控件的名字、位置信息，然后再代码中创建原生控件的对象 ,创建之后把控件对象或者窗口句柄传递给 CDuiNativeWnd 对象就可以。

属性：

属性名	类型	说明
delaycreate	0|1	是否延迟创建控件

函数：

函数	是否虚函数	说明
GetNativeHWnd	否	获取原生控件的 Windows 窗口句柄
GetNativeWnd	否	获取原生控件的 CWnd 对象指针
GetPaintHWnd	否	获取父对话框的 Windows 窗口句柄
GetPaintWnd	否	获取父对话框的 CWnd 对象指针
SetNativeHWnd	否	设置原生控件的 Windows 窗口句柄
SetNativeWnd	否	设置原生控件的 CWnd 对象指针(设置之后在控件

		析构时候可以自动删除此 CWnd 对象）

示例：

参考 Demo 程序中输入框控件演示页面中的原生输入框控件演示。

3.26. DUI ActiveX 控件

类名：CDuiActiveX

控件名：activex

说明：ActiveX 控件，是从 CControlBase 派生的，所以基类控件具有的属性和函数也可以使用。

属性：

属性名	类型	说明
clsid	字符串	ActiveX 控件的 CLSID
url	字符串	导航 URL
delaycreate	0|1	是否延迟创建 ActiveX 控件，直到控件需要显示的时候才创建
show-contextmenu	0|1	是否显示控件自己的右键菜单，默认是显示
show-scroll	0|1	是否显示控件自己的滚动条，默认是显示

函数：

函数	是否虚函数	说明
CreateControl	否	创建 ActiveX 控件
ParseFilePath	否	解析传入的文件路径

3.27. Web 浏览器控件

类名：CDuiWebBrowser

控件名：webbrowser

说明：浏览器控件，是从 CDuiActiveX 派生的，所以基类控件具有的属性和函数也可以使用。

属性：

属性名	类型	说明
url	字符串	导航 URL

delaycreate	0|1	是否延迟创建 ActiveX 控件，直到控件需要显示的时候才创建
show-contextmenu	0|1	是否显示控件自己的右键菜单

函数：

函数	是否虚函数	说明
Navigate	否	浏览器导航

界面示例：

3.28. Flash 控件

类名：CDuiFlash

控件名：flash

说明：Flash 控件，是从 CDuiActiveX 派生的，所以基类控件具有的属性和函数也可以使用。

属性：

属性名	类型	说明
url	字符串	导航 URL
delaycreate	0|1	是否延迟创建 ActiveX 控件，直到控件需要显示的时候才创建
show-contextmenu	0|1	是否显示控件自己的右键菜单

transparent	数字	透明度，0-100 的数字

函数：

函数	是否虚函数	说明
Navigate	否	浏览器导航

界面示例：

3.29. 媒体播放器控件

类名：CDuiMediaPlayer

控件名：mediaplayer

说明：媒体播放器控件，是从 CDuiActiveX 派生的，所以基类控件具有的属性和函数也可以使用。

属性：

属性名	类型	说明
url	字符串	导航 URL
delaycreate	0|1	是否延迟创建 ActiveX 控件，直到控件需要显示的时候才创建
show-contextmenu	0|1	是否显示控件自己的右键菜单
transparent	数字	透明度，0-100 的数字

函数：

函数	是否虚函数	说明
Navigate	否	浏览器导航

界面示例：