Overview.

CBiSplitterWnd is a MFC compatible class for creating binary splitter window which looks like Microsoft Outlook window. This class allows splitting window only on two parts, named panes. But this is not restrict the functionality because of panes subsplitting is allowable. This means that any pane can be split onto two subpanes and so on. Special bar, named splitter or gap, separates one pane from another. Splitter can be oriented vertically or horizontally. Pane contains view which has to be derived from CView or CWnd classes or from CBiSplitterWnd class in order to split pane onto subpanes. Pane can contain no view also. In this case a background color fills it. Multiple views are allowed for every pane. Besides view, pane consists of border and caption bar with close button and caption text which is drawn on it. All these items are optional. BiSplitter window supports pane gripping as well. The grip buttons for both panes are positioned on the gape. Pane gripping is not the same that pane closing. When pane is closed, for example by clicking on the close button, there is not simple way to open it for example by clicking on some control in BiSplitter window. To open closed pane application should call special function of the CBiSplitter class. Unlike the close button the grip button acts in both directions – to grip pane and to expand the gripped pane. But disadvantage of the gripping is that gripped pane takes some place in BiSplitter window which is needed to draw the grip button. Unlike this the close pane is invisible entirely. The BiSplitter window panes and buttons can be drawn in various styles both 3D and flat.

You create BiSplittter window by using Create function specified in CBiSplitterWnd class. Set BiSplitter initial styles when you create one. You can retrieve and change the styles after creating the BiSplitter window by using the AcquireStyles and ChangeStyles functions. Styles are combined with bitwise OR operator. After creating BiSplitter window you have to create views. You can create any quantity of views, but only two views can be displayed at the same time. There are 3 kinds of view. First kind is the view derived from CView class. For application which support doc/view architecture most of views will be derived from this class. Second kind is the view derived from CWnd class. Such views most appropriate for not doc/view application. Third kind is the views derived from CBiSplitterWnd class. These special-purpose views are destined to split pane onto two subpanes. To create view of the any kind you have to call function CreateView. There are three functions CreateView which have different set of the parameters. These functions are responsible for creating the different kinds of view. After views have been created you have to create two panes by using function CreatePane. Strictly speaking function CreatePane does not create pane it only initializes the specified pane assigning styles, view and caption title. These two panes exist always and you can’t delete any of them or create a new pane. To receive or modify the set of styles you have to call functions AcquirePaneStyles and ChangePaneStyles correspondingly. Function GetPaneView returns a pointer to the view of the specified pane. Call the function GetPaneViewID if only view identifier is needed. To receive or change pane caption text use functions GetCaptionText and SetCaptionText correspondingly. After pane initialization you can change pane view by calling function AssignViewToPane. The following example creates BiSplitter window with many views.

```
BOOL CMainFrame::OnCreateClient(LPCREATESTRUCT /*lpcs*/,
        CCreateContext* pContext)
{
        //      create splitter window
```

```
	if (!m_wndBiSplitter.Create(this, VSPLITTER | FIXED0 | VIEW3D ))
		return FALSE;

	m_wndBiSplitter.CreateView(RUNTIME_CLASS(CPaneTreeView), 0, pContext);
	m_wndBiSplitter.CreateView(RUNTIME_CLASS(CPaneListView), 1, pContext);
	m_wndBiSplitter.CreateView(RUNTIME_CLASS(CPaneEditView), 2, pContext);
	m_wndBiSplitter.CreateView(RUNTIME_CLASS(CPaneFormView), 3, pContext);

	m_wndBiSplitter.CreateView(RUNTIME_CLASS(CPaneTreeView), 10, pContext);
	m_wndBiSplitter.CreateView(RUNTIME_CLASS(CPaneListView), 11, pContext);
	m_wndBiSplitter.CreateView(RUNTIME_CLASS(CPaneEditView), 12, pContext);
	m_wndBiSplitter.CreateView(RUNTIME_CLASS(CPaneFormView), 13,
pContext);

	if (
	  !m_wndBiSplitter.CreatePane(0, THICK_BORDER | SMCAPTION |
THICK_CAPTIONBORDER | CLOSEBTN | FLAT_BUTTON | GRIPBTN
			, "First pane", m_wndBiSplitter.GetView(0) )

		||
		!m_wndBiSplitter.CreatePane(1, THICK_BORDER | SMCAPTION |
THICK_CAPTIONBORDER | FLAT_BUTTON
			, "Second pane", m_wndBiSplitter.GetView(11))
		)
	{
		m_wndBiSplitter.DestroyWindow();
		return FALSE;
	}

	return TRUE;
}
```

After panes have been created, you can set the gap position and width by using functions SetSplitterPos and SetSplitterGap. Functions HidePane and ShowPane allow to hide or to show either of panes.

Function Reference

AcquirePaneStyles

Returns the styles on the specified pane

```
virtual UINT AcquirePaneStyles(UINT nPane);
```

Parameters

nPane – specifies a pane. Can be set to values BSW_FIRST_PANE or BSW_SECOND_PANE

Return Value

Return the set of the pane styles.

Remarks

Styles are combined with the OR operator

Example

```
CBiSplitterWnd wnd;
UINT styles;
styles = wnd.AcquirePaneStyles(BSW_FIRST_PANE);
```

See Also

AcquireStyles, ChangePaneStyles, ChangeStyles

AcquireStyles

Returns the styles on the splitter window

```
virtual UINT AcquireStyles();
```

Return Value

Return the set of the BiSplitter window styles

Remarks

Styles are combined with the OR operator

Example

```
CBiSplitterWnd wnd;
UINT styles;
styles = wnd.AcquireStyles();
```

See Also

AcquirePaneStyles, ChangePaneStyles, ChangeStyles

AssignViewToPane

To attach view to pane, call the AssignViewToPane member function

```
virtual void AssignViewToPane(UINT viewID, UINT nPane);
```

Parameters

viewID – specified a view.

nPane – specifies a pane. Can be set to values BSW_FIRST_PANE or BSW_SECOND_PANE

Remarks

Attached view should be previously created with CreateView function. The current pane view is automatically detached and hided.

Example

```
CBiSplitterWnd m_wndBiSplitter;
//create splitter window
if (!m_wndBiSplitter.Create(this, VSPLITTER | FIXED0 | VIEW3D ))
        return FALSE;
//create view which has identifier 0
m_wndBiSplitter.CreateView(RUNTIME_CLASS(CPaneTreeView), 0, pContext);

//create first pane and attach view 0 during creation
m_wndBiSplitter.CreatePane(BSW_FIRST_PANE, THICK_BORDER |
SMCAPTION | THICK_CAPTIONBORDER | CLOSEBTN | FLAT_BUTTON |
GRIPBTN
                        , "First pane", m_wndBiSplitter.GetView(0) )

//create view which has identifier 1
m_wndBiSplitter.CreateView(RUNTIME_CLASS(CPaneFormView), 1, pContext);

m_wndBiSplitter. AssignViewToPane(1, BSW_FIRST_PANE);
```

See Also

CreateView

ChangePaneStyles

To change pane styles, call ChangePaneStyles member function

virtual void ChangePaneStyles(UINT nPane, UINT excludeStyles, UINT includeStyles);

Parameters

nPane – specifies a pane. Can be set to BSW_FIRST_PANE or BSW_SECOND_PANE values

excludeStyles – set of styles which should be excluded from pane

includeStyles – set of styles that should be included in pane

Remarks

To change for example CAPTION to SMCAPTION you should remove style CAPTION and add style SMCAPTION

Example

```
CBiSplitterWnd m_wndBiSplitter;
//create splitter window
if (!m_wndBiSplitter.Create(this, VSPLITTER | FIXED0 | VIEW3D ))
        return FALSE;
//create view which has identifier 0
m_wndBiSplitter.CreateView(RUNTIME_CLASS(CPaneTreeView), 0, pContext);

//create first pane and attach view 0 during creation
m_wndBiSplitter.CreatePane(BSW_FIRST_PANE, THICK_BORDER |
SMCAPTION | THICK_CAPTIONBORDER | CLOSEBTN | FLAT_BUTTON |
GRIPBTN
                        , "First pane", m_wndBiSplitter.GetView(0) )

m_wndBiSplitter. ChangePaneStyles(BSW_FIRST_PANE, THICK_BORDER |
SMCAPTION, THIN_BORDER | CAPTION);
```

See Also

ChangeStyles

ChangeStyles

To change splitter window styles, call ChangeStyles member function

virtual void ChangeStyles(UINT excludeStyles, UINT includeStyles);

Parameters

excludeStyles – set of styles which should be excluded from pane

includeStyles – set of styles that should be included in pane

Remarks

To change for example FIXED0 to FIXED1 you should remove style FIXED0 and add style FIXED1

Example

```
CBiSplitterWnd m_wndBiSplitter;
//create splitter window
if (!m_wndBiSplitter.Create(this, VSPLITTER | FIXED0 | VIEW3D ))
        return FALSE;
m_wndBiSplitter. ChangeStyles(VSPLITTER, HSPLITTER);
```

See Also

ChangePaneStyles

Create

To create a splitter window, call Create member function

virtual BOOL Create(CWnd *pWnd, UINT bswStyles, UINT nID = AFX_IDW_PANE_FIRST);

Parameters

pWnd – pointer to parent window

bswStyles – specifies the styles of the splitter window.

nID – unique splitter window identifier. It is identical the child-window identifier.

Return Value

If the function succeeds, the return value is TRUE. If the function fails the return value is FALSE.

Remarks

By default the ID value is AFX_IDW_PANE_FIRST. If parent window is derived from CFrameWnd the ID should be set to its default value.

Example

```
//        create splitter window
if (!m_wndBiSplitter.Create(this, VSPLITTER | FIXED0 | VIEW3D ))
        return FALSE;
```

See Also

CreatePane, CreateView

CreatePane

To create a pane, call CreatePane member function

virtual BOOL CreatePane(UINT nPane, UINT bspStyles, LPCTSTR lpszPaneCaptionText = NULL, CWnd *pWnd = NULL);

Parameters

nPane – specifies a pane which is created. It can be set to BSW_FIRST_PANE or to BSW_SECOND_PANE values.

bspStyles – specifies the pane styles.

lpszPaneCaptionText – pointer to null-terminated string that specifies the pane name. It is displayed in the caption bar of the pane.

pWnd – pointer to view window. View window can be derived from CWnd, CView or CBiSplitterWnd classes. If pWnd is NULL the pane will have empty view.

Return Value

If function succeeds the return value is TRUE. Otherwise it is FALSE.

Example

```
//        create splitter window
if (!m_wndBiSplitter.Create(this, VSPLITTER | FIXED0 | VIEW3D ))
        return FALSE;

m_wndBiSplitter.CreateView(RUNTIME_CLASS(CPaneTreeView), 0, pContext);
m_wndBiSplitter.CreateView(RUNTIME_CLASS(CPaneListView), 1, pContext);

if (
  !m_wndBiSplitter.CreatePane(0, THICK_BORDER | SMCAPTION | THICK_CAPTIONBORDER | CLOSEBTN | FLAT_BUTTON | GRIPBTN
                , "First pane", m_wndBiSplitter.GetView(0) )

        ||
        !m_wndBiSplitter.CreatePane(1, THICK_BORDER | SMCAPTION | THICK_CAPTIONBORDER | FLAT_BUTTON
                , "Second pane", m_wndBiSplitter.GetView(1))
        )
{
```

```
            m_wndBiSplitter.DestroyWindow();
            return FALSE;
        }
```

See Also

Create, CreateView

CreateView for CBisplitter derived view

To create view derived from BiSplitterWnd class, call CreateView member function

virtual CWnd* CreateView(CRuntimeClass* pBiSplitterWndClass, UINT nID, UINT bswStyles);

Parameters

pBiSplitterWndClass – specifies the CRuntimeClass of the view. It should be derived from CBiSplitterWnd class

nID – the view identifier. It is identical the child-window identifier.

bswStyles – specifies the splitter-window styles

Return Value

If function succeeds the return value is pointer to the new view. Otherwise the return value is NULL.

Remarks

Example

```
        if (!m_wndBiSplitter.Create(this, VSPLITTER | FIXED0 | VIEW3D ))
            return FALSE;


        if(!m_wndBiSplitter.CreatePane(0, FLAT_BUTTON | GRIPBTN , NULL
                    , m_wndBiSplitter.CreateView(RUNTIME_CLASS(CBiSplitterWnd),
1, HSPLITTER | FIXED0 | VIEW3D) )
       ||
       !m_wndBiSplitter.CreatePane(1, THICK_BORDER | SMCAPTION |
THICK_CAPTIONBORDER | FLAT_BUTTON, "Text view"
                    ,
m_wndBiSplitter.CreateView(RUNTIME_CLASS(CRichTextView/*CMultiSplitterView*/),
2, pContext)))
        {
            m_wndBiSplitter.DestroyWindow();
            return FALSE;
        }



        CBiSplitterWnd *pSplitterWnd2 = (CBiSplitterWnd
*)m_wndBiSplitter.GetPaneView(0);

        if (!pSplitterWnd2->CreatePane(0, THIN_BORDER | SMCAPTION |
THIN_CAPTIONBORDER | FLAT_BUTTON | CLOSEBTN
                    ,"Tree view 1", pSplitterWnd2-
>CreateView(RUNTIME_CLASS(CTreeView1), 11, pContext) )
            ||
```

```
		!pSplitterWnd2->CreatePane(1, THIN_BORDER | SMCAPTION |
THIN_CAPTIONBORDER | FLAT_BUTTON | CLOSEBTN, "Tree view 2"
			, pSplitterWnd2->CreateView(RUNTIME_CLASS(CTreeView2), 12,
pContext))
		)
	{
		m_wndBiSplitter.DestroyWindow();
		return FALSE;
	}
```

See Also

CreateView for CWnd deived view, CreateView for CView derived view

CreateView for CWnd derived View

To create a view derived from CWnd class, call CreateView member function
virtual CWnd* CreateView(CRuntimeClass* pWndClass, UINT nID);

Parameters

pWndClass - specifies the CRuntimeClass of the view. It should be derived from CWnd class

nID – the view identifier. It is identical the child-window identifier.

Return Value

If function succeeds the return value is pointer to the new view. Otherwise the return value is NULL.

Remarks

Example

```
		//	create splitter window
	if (!m_wndBiSplitter.Create(this, VSPLITTER | FIXED0 | VIEW3D ))
		return FALSE;

	m_wndBiSplitter.CreateView(RUNTIME_CLASS(CWnd), 0, pContext);
	m_wndBiSplitter.CreateView(RUNTIME_CLASS(CWnd), 1, pContext);

	if (
	  !m_wndBiSplitter.CreatePane(0, THICK_BORDER | SMCAPTION |
THICK_CAPTIONBORDER | CLOSEBTN | FLAT_BUTTON | GRIPBTN
			, "First pane", m_wndBiSplitter.GetView(0) )

		||
		!m_wndBiSplitter.CreatePane(1, THICK_BORDER | SMCAPTION |
THICK_CAPTIONBORDER | FLAT_BUTTON
			, "Second pane", m_wndBiSplitter.GetView(1))
		)
	{
		m_wndBiSplitter.DestroyWindow();
		return FALSE;
	}
```

See Also

CreateView for CBiSplitterWnd deived view, CreateView for CView derived view

CreateView

To create a view derived from CView class, call CreateView member function
virtual CWnd* CreateView(CRuntimeClass* pViewClass, UINT nID, CCreateContext* pContext);

Parameters

pViewClass - specifies the CRuntimeClass of the view. It should be derived from CWnd class

nID – the view identifier. It is identical the child-window identifier.

pContext – a pointer to the creation context used to create the view.

Return Value

If function succeeds the return value is pointer to the new view. Otherwise the return value is NULL.

Remarks

Example

```
BOOL CMainFrame::OnCreateClient(LPCREATESTRUCT /*lpcs*/,
CCreateContext* pContext)
{
	//	create splitter window
	if (!m_wndBiSplitter.Create(this, VSPLITTER | FIXED0 | VIEW3D ))
		return FALSE;

	m_wndBiSplitter.CreateView(RUNTIME_CLASS(CPaneTreeView), 0, pContext);
	m_wndBiSplitter.CreateView(RUNTIME_CLASS(CPaneListView), 1, pContext);

	if (
	  !m_wndBiSplitter.CreatePane(0, THICK_BORDER | SMCAPTION |
THICK_CAPTIONBORDER | CLOSEBTN | FLAT_BUTTON | GRIPBTN
			, "First pane", m_wndBiSplitter.GetView(0) )

		||
		!m_wndBiSplitter.CreatePane(1, THICK_BORDER | SMCAPTION |
THICK_CAPTIONBORDER | FLAT_BUTTON
			, "Second pane", m_wndBiSplitter.GetView(1))
		)
	{
		m_wndBiSplitter.DestroyWindow();
		return FALSE;
	}

	m_wndBiSplitter.SetSplitterPos(200);
	m_wndBiSplitter.SetSplitterGap(6);

	return TRUE;
}
```

See Also

CreateView for CBiSplitterWnd deived view, CreateView for CWnd derived view

GetActivePane

To determine which pane is active, call GetActivePane member function
virtual int GetActivePane();

Return Value

This function returns the pane identifier.

Example

```
int paneID = GetActivePane();
if(paneID == BSW_FIRST_PANE)
{
        ….
}
else
{
        ……
}
```

See Also

SetActivePane

GetPaneCaptionText

Returns the pane caption text

CString GetPaneCaptionText(UINT nPane);

Parameters

nPane – pane identifier

Return Value

This function returns CString value which contains the name of the pane

Example

```
CString text;
text = GetPaneCaptionText(BSW_FIRST_PANE);
```

See Also

SetPaneCaptionText

GetPaneView

Returns the current view of the pane

CWnd *GetPaneView(UINT nPane);

Parameters

nPane – specifies a pane. Can be set to values BSW_FIRST_PANE or BSW_SECOND_PANE

Return Value

Function returns pointer to the current view of the specified pane. If no view exists for pane the NULL is returned

Remarks

Example

```
CTreeView *pTree = m_wndBiSplitter.GetPaneView(BSW_SECOND_PANE);
if(pTree != NULL)
{
        …
}
```

See Also

AssignViewToPane

GetPaneViewID

Returns the identifier of the view attached to the pane

UINT GetPaneViewID(UINT nPane);

Parameters

nPane – specifies a pane. Can be set to values BSW_FIRST_PANE or BSW_SECOND_PANE

Return Value

Function returns identifier of the specified pane. If there is no view for pane debug exception is activated.

Remarks

Example

UINT viewID = m_wndBiSplitter.GetPaneViewID(BSW_SECOND_PANE);

See Also

GetPaneView

GetSplitterGap

Returns the thickness of the splitter gap

UINT GetSplitterGap();

Return Value

Function returns the width of the splitter gap for vertical splitter or height for horizontal splitter. The thickness is specified in pixels.

Remarks

Example

UINT size = m_wndBiSplitter.GetSplitterGap();

See Also

SetSplitterGap

GetSplitterPos

Returns current position of the splitter

int GetSplitterPos();

Return Value

Function returns the position in pixels of the left splitter side for vertical splitter and position of the top splitter side for horizontal splitter.

Remarks

Example

UINT pos = m_wndBiSplitter.GetSplitterPosition();

See Also

SetSplitterPosition

GetView

Retrieves a pointer to view for specified view identifier

virtual CWnd *GetView(UINT viewID);

Parameters

viewID – specifies the identifier of the view

Return Value

Function returns pointer to the view with specified identifier. If there is no view with such identifier the assertion exception will be raised.

Remarks

Example

CTreeView *pTreeView = (CTreeView *)m_wndBiSplitter.GetView(1);

See Also

CreateView

GripPane

Grips/expands the specified pane

virtual void GripPane(UINT nPane);

Parameters

nPane – specifies a pane. Can be set to values BSW_FIRST_PANE or BSW_SECOND_PANE

Remarks

Function grips specified pane if it is expanded state and expand pane if it is gripped

Example

m_wndBiSplitter.GripPane(BSW_FIRST_PANE);

See Also

IsPaneGripped

HidePane

Hides the specified pane

virtual void HidePane(UINT nPane);
virtual void HidePane(CWnd *pView);

Parameters

nPane – specifies a pane. Can be set to values BSW_FIRST_PANE or BSW_SECOND_PANE

pView – pointer to view which belongs to the specified pane

Remarks

In case of if the other pane is hidden than the splitter window became invisible.

Example

m_wndBiSplitter.HidePane(BSW_FIRST_PANE);

See Also

IsPaneVisible, ShowPane

IsPaneVisible

Indicates whether the specified pane is visible or not

virtual BOOL IsPaneVisible(UINT nPane);

Parameters

nPane – specifies a pane. Can be set to values BSW_FIRST_PANE or BSW_SECOND_PANE

Return Value

Function returns TRUE if specified pane is visible and FALSE otherwise.

Example

BOOL flag = m_wndBiSplitter.IsPaneVisible(BSW_FIRST_PANE);

See Also

HidePane, ShowPane

IsWndVisible

Indicates whether the splitter window is visible or not

virtual BOOL IsWndVisible()

Return Value

Function return TRUE if splitter window is visible and FALSE otherwise.

Example

BOOL flag = m_wndBiSplitter.IsWndVisible();

See Also

RecalcLayout

Call to redisplay a splitter window

virtual void RecalcLayout();

Remarks

Function recalculate panes sizes and position and redraw splitter window

Example

m_wndBiSplitter.RecalcLayout();

See Also

SetActivePane

Set a pane to be active one in the frame

virtual void SetActivePane(UINT nPane);

Parameters

nPane – specifies a pane. Can be set to values BSW_FIRST_PANE or BSW_SECOND_PANE

Example

m_wndBiSplitter.SetActivePane(BSW_FIRST_PANE);

See Also

GetActivePane

SetPaneCaptionText

Call this function to assign a title to specified pane

void SetPaneCaptionText(UINT nPane, const CString &sPaneCaptionText);

Parameters

nPane – specifies a pane. Can be set to values BSW_FIRST_PANE or BSW_SECOND_PANE

sPaneCaptionText – specifies string with caption text

Remarks

Example

m_wndBiSplitter.SetPaneCaptionText(BSW_FIRST_PANE, “First pane”);

See Also

GetPaneCaptionText

SetSplitterGap

Changes the thickness of the splitter gap

void SetSplitterGap(UINT nGap);

Parameters

nGap – specifies the new thickness of the splitter gap

Remarks

Example

m_wndBiSplitter.SetSplitterGap(6);

See Also

GetSplitterGap

SetSplitterPos

Sets the position of the splitter

void SetSplitterPos(int nSplitterPos);

Parameters

nSplitterPos – specifies new splitter position

Remarks

Example

m_wndBiSplitter.SetSplitterPos(100);

See Also

GetSplitterPos

ShowPane

Shows the specified pane

Shows the pane specified the view which is attached to this pane

virtual void ShowPane(CWnd *pView);
virtual void ShowPane(UINT nPane);

Parameters

nPane – specifies a pane. Can be set to values BSW_FIRST_PANE or BSW_SECOND_PANE

pView – pointer to view which is attached to the specified pane

Example

m_wndBiSplitter.ShowPane(BSW_FIRST_PANE);

See Also

HidePane, IsPaneVisible

BiSplitter Window Styles

The following styles can be specified for BiSplitter window. After the window has been created, these styles can be modified by means of calling ChangeStyles function.

VSPLITTER

A window has vertical splitter orientation.

HSPLITTER

A window has horizontal splitter orientation. This style can't be used with VSPLITTER style

FIXED0

When window size is changing the size of the second pane is changing first of all, the size of the first pane is fixed.

FIXED1

When window size is changing the size of the first pane is changing first of all, the size of the second pane is fixed.

PROPORTIONAL

When window size is changing the size of the both panes is changing proportional.

VIEW3D

Creates a window with 3D borders and caption bar.

SIZING_MODE_STYLES = FIXED0 | FIXED1 | PROPORTIONAL

SPLITTER_STYLES = VSPLITTER | HSPLITTER
ALL_BISPLITTER_STYLES = SIZING_MODE_STYLES | SPLITTER_STYLES | VIEW3D

Pane styles

The following styles can be specified for panes of the BiSplitter window

THIN_BORDER

Creates a pane with thin border

THICK_BORDER

Creates a pane with thick border

THIN_CAPTIONBORDER

Creates a pane with thin caption border

THICK_CAPTIONBORDER

Creates a pane with thick caption border

SMCAPTION

Creates a pane with a small caption height

CAPTION

Creates a pane with a normal caption height

THIN_BUTTON

Creates a pane buttons with thin border

THICK_BUTTON

Creates a pane buttons with thick border

FLAT_BUTTON

Creates a flat pane buttons

CLOSEBTN

Creates a pane with close button in the caption bar

GRIPBTN

Creates a pane with full size grip button in the splitter gap

SMGRIPBTN

Creates a pane with small grip button in the splitter gap

BORDER_STYLES = THIN_BORDER | THICK_BORDER
CAPTION_STYLES = SMCAPTION | CAPTION
CAPTIONBORDER_STYLES = THIN_CAPTIONBORDER | THICK_CAPTIONBORDER
BUTTON_STYLES = THIN_BUTTON | THICK_BUTTON | FLAT_BUTTON
GRIP_STYLES = GRIPBTN | SMGRIPBTN
ALL_PANE_STYLES = CLOSEBTN | BORDER_STYLES | CAPTION_STYLES | CAPTIONBORDER_STYLES | BUTTON_STYLES | GRIP_STYLES