



Main Menu	<input type="checkbox"/>
Main Menu	<input type="checkbox"/>

PROGRAMS

AXOLOTL

AXOLOTL IDE

BASIC CONTROLS

CUSTOM CONTROLS

BUGEE#@!

SERVER SIDE INSTALLATION

HOW TO USE THE SCRIPTS

HOW TO USE THE CLIENT

Controls Specification

The controls that the designer can use , have different properties for different functionality. But some properties are shared from all controls. In the following description the properties of the property editor will being discussed.

All the controls have in their properties the following :

Specs

A special property that shows a dialog with a description of the control and its functionality .

Name

All the controls have a unique name .

ID

The Unique ID identifier that the control will have inside the C/C++ code . Note that some of the controls may do not need an id identifier , like the ImageList , but they will have one nevertheless for making the parser of the Axolotl life easier!

Top , Left , Width , Height

All the controls have the above properties. Some controls have their position or dimensions locked.

Parent



The parent of the control . This is for the internal logic of the Axolotl designer. Most of the cases this is the parent that the control (window) will have in C/C++ code.

RealParent

The Real Parent of the control. This is the parent that the control will have inside the C/C++ code .

PopupMenu

This is the popmenu that the control will have when the user will right click it . This property appears in the Properties of all controls BUT not all controls support popup menus . For example you can set a popup menu for an ImageList , but in reality an ImageList is not an visual control , so the popup menu will never be functional . Of course you can assign the same popup menu in multiple controls .

Funcs

This is a special property that shows a dialog with all the internal functions that the control may publish . This functions the user can copy with double click or from the popup menu of the right click and then paste it in the code.

ToolTip

Another special property , but in contrast with the other ones it will not being there for every control . This property is defined inside the xml file of the control , so some control may have it , some will not . This control is just a text field. its string will appear inside the Application as a simple tooltip .

Notes about the Property Editor .

The property editor it's not perfect. That means that some times may will not shown the buttons that can enable the special dialogs of the properties . That happens when the user clicks in a field with a special dialog with out have first enable the editor . So if you have clicked the property and the button its not showing , just click another field and return to the property. The trick will be performend and with a single press of the arrow down or up of the keyboard .

The property editor does not do a type check in the values , except for the Top , Left , Width and Height . Thats is because some time the user may wants to set a property with a defined name and not with a number , for example MY_MAX_ICONS . Its your responsibility that the values are the correct type for the field. And dont worry , if you do not spot the error , your compiler will !

WARNING!!

Due to various reasons the property editor has a bug that for the moment i'm unable to correct. When are you trying to make a selection in a property with a dropdown list , if you leave the list mid air with out making a selection , and then you try to click in a control in some frame , the editor try to update it self and throw exceptions . After the 3 or 4 annoying messages the editor will have clear alot of the properties of the control and you will be in the position to must re enter all its properties.

so becarefull , make your selection in the dropdown list before clicking!

WARNING 2 !!

If you edit a property , dont try to **Update All Frames Code** with out changing the focus into another proeprty , because the change will not be reflected to the C/C++ code. That means that as while you have the input focus inside the property

the editor will not update the control properties because it will not know if you are done with the editing. So after editing a property click another one , or press the enter key or the up/down arrows in the keyboard. Do something that will change the focus !

Caption

A

A simple static text .

Text

The visible text .

FontSize

The font size. This value is valid only if the FontName is not DEFAULT.

FontName

The font name if this property is DEFAULT the default size and font are applied.

TextColor

The color of the Text. For this property to have any effect you must edit the WM_CTLCOLORSTATIC message and add the instance control function `void Caption$name$Color(LPARAM lParam , WPARAM wParam)`

example if the caption is named **Caption6** :

```

case WM_CTLCOLORSTATIC:
{
    // you must add only the below function all the other code is generated from the Axolotl

    CaptionCaption6Color(lParam , wParam);

    //-----
    SetBkMode((HDC)wParam,TRANSPARENT);
    if(trbrush != NULL) DeleteObject(trbrush);
    trbrush = CreateSolidBrush(GetPixel((GetDC(GetParent((HWND)lParam))),0,0));
    return (LRESULT) trbrush;
}
break;

```

The PopMenu is not being used .

Button



A Push Button .

Caption

The text of the Button.

Text

If the Style has the BS_COMMANDLINK or the BS_DEFCOMMANDLINK flag set , then the Text is the long text below the caption .

ShowTextOnImage

If this is set as true and the button has a Bitmap or Icon set , then the button will display the Caption too along with the image . Try different style combination for achieving the desirable alignment .

Bitmap

This value is an ID that is already set inside the resources.rc. For example , if in the resources.rc you have set the following value :

```
#define ID_BIT1 202
ID_BIT1 BITMAP "C:\\progressbar.bmp"
```

And you have set the Bitmap property to 202 then the Button will have the progressbar.bmp bitmap as image .

Icon

This value is an ID that is already set inside the resources.rc. For example , if in the resources.rc you have set the following value :

```
#define ID_ICON 202
ID_ICON ICON "C:\\progressbar.ico"
```

And you have set the Icon property to 202 then the Button will have the progressbar.ico icon as image .

IWidth

The width of the image . If this property is 0 then the actual image width will be used .

IHeight

The height of the image . If this property is 0 then the actual image height will be used .

StyleEx

The extra style of the button .

Style

The style of the button .

Edit



A simple Edit .

Text

The Text of the Edit .

StyleEx

The extra style of the control .

Style

The Style of the control .

ComboBox



A simple ComboBox .

Items

A list of strings that will be added to the ComboBox .

ComboBoxEx



A ComboBoxEx that have the ability to display images along with the text .

ImageList

If not NULL then this ImageList will be associated with the ComboBox .

CheckBox



A typical checkbox .

For the messages and the functionality of the CheckBox see the BS_CHECKBOX style for button control at

<https://msdn.microsoft.com>

Caption

The Caption of the checkbox .

Checked

If this is true the checkbox will be checked .

StyleEx

The extra styles of the control .

Style

The style of the control .

Radio Button



A typical Radio Button .

For the messages and the functionality of the Radio Button see the BS_RADIOBUTTON style for button control at <https://msdn.microsoft.com>

Caption

The Caption of the radiobutton .

Grouped

If this property is true then the radiobutton when clicked will reset all the others radiobuttons that belong to the same group. To group radio buttons , add them in a container like the groupbox .

Selected

If this is true then the radio button is selected .

StyleEx

The extra styles of the control .

Style

The style of the control .

SysLink



A typical SysLink.

For the messages and the functionality of the SysLink see the SysLink control at <https://msdn.microsoft.com>

Link

The Command Link of the control .

Style

The style of the control .

IPAddress



A typical IP Address control .

Byte1

The first byte of the Ip Address that the control will have at the start of the program .

Byte2

The second byte of the Ip Address that the control will have at the start of the program .

Byte3

The third byte of the Ip Address that the control will have at the start of the program .

Byte4

The forth byte of the Ip Address that the control will have at the start of the program .

StyleEx

The extra style of the control .

Style

The Style of the control .

ListBox



A Typical List Box .

Items

A list of strings that the List box will have at creation .

Style

The style of the ListBox .

StyleEx

The extra style of the ListBox .

ListView



A typical List View .

Style

The style of the ListView .

ExtraStyle

Extra styles for the listview .

Columns

The string that will be added in this list , will become Columns in the listview .

ColWidth

The width of the Columns. For more control over the Columns see the macros for the ListView in the msdn.microsoft.com

ImageListNormal

The image List with the Normal Icons .

ImageListSmall

The image list with the small icons .

ImageListState

The images for the states of the items in the ListView .

ImageListGroupH

Image list for group header .

Image



A typical Static control displaying an Image .

Type

BITMAP or ICON .

Image

This is the ID of a image resource. The image must have being inside the exe as resource .The image will being stretch to fit the control .

ProgressBar



A typical Progress Bar .

Orientation

The orientation of the control .

Style

The style of the control .

UpDown



A typical Up-Down.

For the messages and the functionality of the Up-Down see the Up-Down control at <https://msdn.microsoft.com>

Style

The styles of the control .

Orientation

The orientation of the control .

Buddy

The buddy control of the UpDown . Usually an Edit control .

Max

The Max value for the increment .

Min

The Min Value for the increment .

DateTimePicker



A simple DateTimePicker

Format

The format of the Calendar .

TreeView



A typical Treeview.

For the messages and the functionality of the Treeview see the Treeview control at <https://msdn.microsoft.com>

ExStyle

The exstyles of the control .

Style

The styles of the control .

NormalImageList

The ImageList that the treeview will use for the items .

StateImageList

The image list that the treeview will use for the selected items .

RichEdit



A typical RichEdit .

For the messages and the functionality of the RitchEdit see the Ritch Edit control at <https://msdn.microsoft.com>

StyleEx

The extra style of the control .

Style

The style of the control .

Rebar



A typical Rebar .

Style

The style of the Rebar. for the use of the flags please see the rebar control at the <https://msdn.microsoft.com>
If you seek for a standard behavior then leave the flags as is .

ImageList

The ImageList that the bands will use for their images .

Remarks

This control can only accept the control Band as children .

Band



A Band for the Rebar .

ATTENTION

This control creates bands inside a Rebar . BUT the visual representation of the bands may not be in par with the actual visual effect that the application will have .

For example the rebar control can have in the designer any width and height. That is for convenience for when you are putting bands in it. But in reallity the Rebar dimensions are calculated with the internal logic of the Rebar control.

The bands MUST have only one control , and that control will fill all the width of the band. So Put a panel first inside the band , and inside it put your controls. To have the bands in diferent rows , set the RBBS_BREAK flag in the style property in each band that you want to have in another row .

If in the designer you have set your bands with a specific way , its propable that in the actual window the band will have another position.

Try the combination of the properties and styles and the designer positioning , to achieve the look that you desire.

If you feel that the designer does not meet you needs in formating the bands , you can always try to use in code the messages and

styles of the band control that you can find at <https://msdn.microsoft.com> .

ForeColor

Band foreground colors.

BackColor

Band background colors.

(The colors of the bands do not apply if the visual styles are enabled)

DisplayText

The display text for the band .

Image

Zero-based index of the image that will be displayed in the band. It must exists in the image list that is accosiated with the parent Rebar .

CustomSize

If this property is true then the `MinimumWidth` , `InitHeight` and `ResizeStep` are valid .

MinimumWidth

Minimum width in pixels. The band can't be sized smaller than this value.

(if `CustomSize` is true) .

MinimumHeight

Minimum height o in pixels. The band can't be sized smaller than this value.

BackGroundBit

The id (e.g 301) (you must have set this inside the resource file as a resource bitmap) of the bitmap that the band will load and use as a background .

Bitw

The `BackGroundBit` width.

Bith

The `BackGroundBit` Height.

InitHeight

Initial height of the band, in pixels.

(if `CustomSize` is true) .

ResizeStep

Step value by which the band can grow or shrink, in pixels. If the band is resized, it will be resized in steps specified by this value.

(if `CustomSize` is true) .

IdealWidth

Ideal width of the band, in pixels. If the band is maximized to the ideal width , the rebar control will attempt to make the band this width.

HeaderSize

Size of the band's header, in pixels. The band header is the area between the edge of the band and the edge of the child window.

This is the area where band text and images are displayed, if they are specified. If this value is specified, it will override the normal header dimensions that the control calculates for the band.

MonthCalendar



A simple Calendar .

StyleEx

The extra style of the control .

Style

The Style of the control .

TabControl



This control is a container for tabs. This is a special control for the program.

The control has a unique property named , clicking the button of this property the tabs manager dialog appears .

Tabs Manager Dialog

In this dialog you can manage the tabs of the tabcontrol . With a right click in the white area the three following options will appear .

Create Tab

Creates a new tab inside the TabControl .

The new tab will have a name and a caption. The name is the name that the tab will have inside the C/C++ code and the caption is the text that the tab will display .

If you change this values , do not forget to press the save button , or else the changes will be lost .

Delete Tab

Deletes the selected tab .

Swap Tabs

Swaps two tabs .

Notes

The TabControl and its children the tabs , are operational inside the C/C++ code. The code that must be called for the tabs to change and the code that hide and unhide the correct tab and its controls , is all being written inside the auto created frame callback. Inside the designer the tabs have the same behavior with the tabs inside the C/C++ code. The user can put other controls inside the tabs , even other tabcontrols and in general the behavior of the tab is as a container of controls .

Pager



A typical simple Pager .

How To Use

This control behave as a container , but dont put any regular controls in it. You must put only one control inside and must be a container; like the GroupBox. Resize your pager to a size that can accept the GroupBox, put the GroupBox inside it and populate the GroupBox with the controls that you want. After the GroupBox is ready, resize the pager to the desirable size and position it where you want.

After that you must call the CalcSizeOnNotify inside the WM_NOTIFY e.g

case WM_NOTIFY:

```
if (CalcSizeOnNotify( IParam , Pager2 , GROUPBOX6 ) == true ) break ;
```

In case of multiple pagers , you must call the CalcSizeOnNotify one time for every pager and do a break if the function returns true e.g

case WM_NOTIFY:

```
if (CalcSizeOnNotify( IParam , Pager2 , GROUPBOX6 ) == true ) break ;
```

```
if (CalcSizeOnNotify( IParam , Pager3 , GROUPBOX7 ) == true ) break ;
```

If you do not want to put your container inside the pager , or do you have already create the container with the cntrls and you do not want to copy paste it (changes all the controls names) or recreate it inside the pager , then you can call SetChild after you call the WINAPI function SetParent. And the groupbox will become the child of the pager .

For more control over the pager see the macros and events at the <https://msdn.microsoft.com>

Functions

```
void SetChild(HWND pager , HWND cntrl);
```

This function MUST be called for each pager so the cntrl that have as parent the pager , can notify the pager .

```
bool CalcSizeOnNotify(LPARAM IParam , HWND pager , HWND cntrl);
```

This function MUST be called for each pager inside the WM_NOTIFY message .

GroupBox



A GroupBox but with real grouping properties (its a real parent for the control.)

Caption

The text of the groupbox

Panel



A Panel with grouping properties (its a real parent for the control.)

The panel in the designer has a thin border for design convenience , in the application the panel its not visible , only its children are .

Status Bar



A typical Status Bar.

For the messages and the functionality of the StatusBar see the Status Bar control at <https://msdn.microsoft.com>

Simple

If this property is true then the SB_SIMPLE message is send upon the creation of the Status Bar .

If this property is false then the user has the responsibility to send the apropiate messages for the Status Bar type.

Text

If the Status Bar is Simple then this will be set as the Status Bar Text .

Icon

This is the ID for the Icon that the simple status bar will shows. The Icon must being set in the resource file . e.g #define MY_ICON 208 . (in this case the Icon property must be set as 208)

Style

The style of the control .

ToolBar



A typical toolbar.

For the messages and the functionality of the toolbar see the toolbar control at <https://msdn.microsoft.com>

ImageList

The image list that the toolbar will use for the buttons .

ImageListHot

The image list that the toolbar will use when a button has the mouse hover on it .

ImageListPressed

The image list that the toolbar will use when a button is being pressed .

ImageListDisabled

The image list that the toolbar will use when a button being disabled .

ExStyle

The extra style of the toolbar .

Style

The style of the toolbar .

Notes

The toolbar is a special case of a container , it can accept only toolbuttons . If a toolbar is added inside a frame , the toolbar will take the apropiate position based in the Style property. This is limiting the usefullness of the toolbar.

For best use , put the toolbar inside another container like a band of a rebar a pager or even inside a panel . That way you can positioning the panel or the rebar or the pager as you wish , even to easily make the parent of the panel or any container to a floating window .

ToolBarButton



A button for a toolbar.

This control supports the very basic of the buttons of a toolbar.

For the messages and the functionality of the toolbar see the toolbar control at <https://msdn.microsoft.com>

ImageIndex

The index that the toolbar will use to represent the button .The default is I_IMAGENONE .

(An imagelist must have been selected in the toolbar. Fill the imagelist with icon and not bitmaps if you want transparency, if you do not set an image list then set the text property or else the buttons will be not visible. If you want to use the system icons you must do it with your code.)

States

The states of the button .

Style

The style of the button .

Text

The string that the button wil have as a caption .

Order

This property is the Order of the buttons inside the toolbar. The Order of the buttons inside the designer is not the true Order of the buttons. Set this property for the correct Order .

Trackbar



A typical Trackbar.

For the messages and the functionality of the Trackbar see the Trackbar control at <https://msdn.microsoft.com>

Orientation

The orientation of the TrackBar .

Style

The style of the TrackBar .

Scrollbar



A typical Scrollbar.

For the messages and the functionality of the Scrollbars see the ScrollBar control at <https://msdn.microsoft.com>

MinPos

The minimum value e.g 0 .

MaxPos

The maximum value e.g 100 .

Orientation

The orientation of the control .

Style

The style of the control .

ExToolTip



This is a ToolTip with extra properties.

Text

The Text of the ToolTip .

Style

The style of the ToolTip .

Flags

Flags for the creation of the ToolTip .

MaxTipWidth

This is the Width of the ToolTip. If -1 then the ToolTip extend with no limit. If any other value then the tooltip will warp the lines if they are bigger of the Width .

BackgroundColor

The Background Color of the ToolTip. If a Visual Style is enabled then this value is not used .

TextColor

The Text Color of the ToolTip. If a Visual Style is enabled then this value is not used .

Icon

A list with the Icons that the tooltip can display (one at the time) .

Extralcon

A list with extra Icons . This has priority over the "Icon" property .

UserIcon

This is a Icon that the user can set. It must be either a number that is bound with an Icon inside the resource file a.k.a 500 if #define Icon_1 500 Icon_1 Icon "C:\\Icon.ico" or the name a.k.a Icon_1 if the value is defined inside the code too. The UserIcon has priority over the other icons .

Title

The Title of the tooltip .

Control

This is the Control that the tooltip will bound to.

Note

The ExToolTip is a special type of control that needs to be in the right creation order to work. That means that if you put first the ExToolTip and AFTER that the control that it will display this tooltip , the tooltip will have not any way to know that its control will not be created at the time that the ExToolTip will try to bind it self in it. If you are in a situation like this , open the FrameList and put the ExToolTip at the bottom of the creation list .

ImageList

This Control represents the basic functionality of an image list.

For the messages and the functionality of the ImageList see the ImageList at <https://msdn.microsoft.com>

Files

A list of Files or a list of resource id's .

ImCX , ImCY

The dimensions of the images that the ImageList will store .

CreateMask

If true the ImageList will create masks for the images

ImageType

What type of Image the ImageList will store

LoadType

if NULL the image list will load the images from the disk , else the imagelist will load resource images from the module .

Masked

if Masked is true then the Imagelist sets as mask color the MaskColor

MaskColor

The color that the ImageList will use for mask . The Masked property must be true.

MaxIcons

The Max number of icons that the list will store. If the Files property has more entries than the MaxIcons value , then the MaxIcons value will be set inside the code to be as the Files entry count .

Flags

The Flags for the ImageList creation .

Bevel



A typical Bevel that creates rects and lines .

InnerValue

The flag of the inner line of the bevel.

OuterValue

The flag of the outer line of the bevel.

TypeFlags

The flags that change the drawing of the shape of the bevel.

This control is not a real winapi control. For this control to be operational , you must edit the WM_PAINT message of the window that you want the bevel to beign drawn

and inside this message you must call the instance control function DrawBevel\$name\$(HDC dc).

Example if the bevel is named BEVEL0 :

...

```

case WM_PAINT:
    {

        PAINTSTRUCT pstr ;
        HDC dc = BeginPaint( hwnd , &#38;pstr ) ;

        DrawBevelBevel0(dc);

        EndPaint( hwnd , &#38;pstr ) ;

    }

```

...

Procs

prc

This is not a visible control. When it is added in a frame the follow routines will be added to the c++ code. CAUTION insert this control only ONCE in a project. More than one of this control in a project does not create any benefits but in contrast creates more uselles creation routines.

Routines

ShowMessage(LPWSTR mess);

Creates a MessageBox with title "Message" and text the parameter .

```
void CreateFont( HFONT &hFont , LOGFONT lg );
```

Creates the font that is passed as parameter. The Font must be not created .

```
void SetFontInControl( HWND cntrl , HFONT hFont );
```

Set the font for the control. The font must be created first .

```
void ChangeFontValues( HWND cntrl , HFONT hFont , LOGFONT lg);
```

Change the values of a font , the "lg" parameter must have the desired values .

```
void PrepereLogFont( PLOGFONT lg , int fntsize , LPWSTR fontname );
```

Fills the "lg" parameter with default values except the size and the name of the font .

```
int GenStrLen( LPWSTR str );
```

This routine return the length of the "str" parameter .

```
WCHAR *GenIntToStr(long value);
```

Return the number as string . The programmer has the responsibility of delete the returned WCHAR * .

CONTACT US

@ELSE-RETURN0

@NIKOS MOURGIS

Donate

A small donation could help me invest more time sleepin... errr in my free projects!

