TabFrame V1.4

Properties Events Methods

The increasing complexity of the dialog boxes needed for todays Windows applications has lead to the development of the tabbed dialog box which originally appeared in Microsoft Word for Windows V6.0. The TabFrame custom control allows you to include tabbed dialog boxes in a Visual Basic application in a simple and straightforward manner.

The TabFrame control includes a number of features designed to simplify its use and to enhance the appearance of applications which use it.

- Control over the appearance of the tabs.
- Stacked (or multiu-line) tabs.
- Tab selection at design time using the right mouse button.
- Property synchronisation between tabs.
- Automatic 3D appearance for child controls.
- Focus tracking when tabs selected/deselected.
- Keyboard interface for tab selection.

To create a tabbed dialog you create the first TabFrame control in the normal way setting the size and position as required. Additional TabFrame controls are then created for each tab option by double clicking the TabFrame icon in the Visual Basic toolbox. Tabs are selected in design mode by clicking with the right mouse button and controls can then be drawn on each tab as required.

The TabFrame controls within any form or container make up a TabSet and some of the properties apply to the TabSet as a whole. Since the TabFrame control is itself a container it is easy to create nested TabSets.

The TabFrame control is fully integrated with the ToolTip VBX which can be used to provide MS style tool tips and a context sensitive status bar for your Visual Basic applications.

Application Notes

History

Registration

TabSet

A TabSet is a set of TabFrame controls within a form or container. Several of the TabFrame properties are synchronised across all TabFrame controls within a TabSet so that changing the property value for one of the TabFrame controls will automatically change the value for all other TabFrames in the TabSet.

Within a TabSet there is always one active tab and all controls on this tab will be visible. The other, inactive tabs, remain hidden until made active by clicking the tab with the mouse.

TabGroup

A TabGroup is a set of TabFrame controls which are displayed together as a single layer. If a TabSet contains multiple groups then the group containing the active tab is displayed at the front of a stack with other groups shown behind. Selection of a new tab will bring the group containing that tab to the front of the stack.

ToolTip

The ToolTip control allows you to add MS style tool tips and a context sensitive status bar to your Visual Basic applications. In addition to providing tool tips the control supports the tracking of mouse movements, menu selections and focus changes.

A tool tip is a small popup window which appears when the mouse remains over a control (e.g. a tool bar button) for longer than a preset delay. The window is used to provide a short text description of the controls function. Tool tips were made popular by MS Word V6 and Excel V5 and are becoming a standard feature of many leading Windows applications.

The ToolTip control is fully compatible with the TabFrame control and provides tool tips and mouse tracking for each individual tab.

A demonstration version of the ToolTip control is available for download from the MSBASIC forum on Compuserve - see file TOOLTIP.ZIP in library 17 (3rd Party Products) or search on the keyword TOOLTIP.

Active Tab

The active tab within a TabSet is the tab which is currently displayed. Only one tab may be active within a TabSet.

Appearance

Several properties are provided to control the appearance of the tabbed dialog.

TabWidth
TabHeight
TabIndent
Position
Style
Chamfer
Stagger
ActiveColor
InactiveColor
TextOrientation
PicturePosition

Stacked Tabs

If you need more tabs than can conveniently be displayed in a single row then the tabs can be split into several TabGroups to give a stacked display similar to that used in MS Word V6 etc. This capability allows even the most complex of dialog boxes to be implemented in a compact and easy to use manner.

Design Time Selection

Use of the TabFrame control in the Visual Basic design environment is simplified by allowing the user to select a tab by clicking on it with the right mouse button.

Note: Users of VBAssist will need to switch off the property pop-up function in order to select a new tab in this manner. Alternatively a new tab mey be selected from the control dropdown list in the Visual Basic (or VBAssist) property window.

Property Synchronisation

Many of the TabFrame properties, including some of the standard properties, are synchronised so that changing the value for one TabFrame control will automatically change the value for all other TabFrame controls on the same form. The properties which are synchronised in this way are listed below.

ActiveColor
AutoColor
AutoFont
AutoWidth
BackColor
Chamfer
FocusTab
FontBold
FontItalic
FontName
FontSize
FontStrikethru
FontUnderline
ForeColor
Height
InactiveColor
Left
PicturePosition
Position
Stagger
Style
TabHeight
TabIndent
TabStop
TabWidth
Top
Width

Notes:

1. The FontXxx and ForeColor properties are only synchronised if the AutoFont property is True.
2. The ActiveColor and InactiveColor properties are only synchronised if the AutoColor property is True.
3. The TabWidth property is only synchronised if the AutoWidth property is True.

Child 3D Effects

The TabFrame control can automatically add 3D effects to child controls which are drawn on it. The following properties are provided to control these effects.

Child3D

ChildBevel

ChildBevelWidth

Focus Tracking

When a new tab is selected the TabFrame control can either gain the input focus itself or pass the focus to a control on the tab. If focus is passed to a control on the tab then it can either go to the first control in TabIndex order or to the control which had the focus when the tab was last selected. The following properties are provided to control the focus behaviour.

FocusTab
FocusTracking

The standard TabStop property is also provided to control whether or not the TabFrame control can be given the focus with the Tab key.

Keyboard Interface

When a TabFrame control has the input focus the arrow keys may be used to select a new tab. The function of arrow keys depends on the tab position in order to provide intuitive behaviour - e.g. pressing the Up arrow key will make the tab which is visually above the current tab active.

The focus can be given to a TabFrame control in several ways:

1. Selecting the tab with the mouse or a mnemonic key (only if the FocusTab property is True).
2. Using the Tab key (only if the TabStop property is True).
3. Clicking on the currently active Tab.

Selecting a new tab with an arrow key will always pass the focus to the new TabFrame control (even when the FocusTab property is False).

Properties

The following properties are supported by the TabFrame control.

Standard Properties

BackColor
BorderStyle
Caption
ClipControls
DragIcon
DragMode
Enabled
FontBold
FontItalic
FontName
FontSize
FontStrikethru
FontUnderline
ForeColor
Height
HelpContextID
Hwnd
Index
Left
MousePointer
Name
Parent
TabIndex
TabStop
Tag
Top
Visible
Width

Custom Properties

About
Active
ActiveColour
AutoColor
AutoFont
AutoWidth
BufferDisplay
Chamfer
Child3D
ChildBevel
ChildBevelWidth
ClientHeight
ClientLeft
ClientTop
ClientWidth
FocusTab
FocusTracking
Groups
InactiveColor
Picture
PicturePosition
Position
Stagger
Style
TabGroup
TabHeight
TabIndent
Tabs
TabNumber
TextOrientation
TotalTabs
TabWidth

Standard Properties

For complete documentation on the standard properties supported by the TabFrame control please refer to the Visual Basic Language Reference. The comments below indicate where the TabFrame control differs from other controls in the use of these properties.

Caption - An ampersand (&) may be included in the caption for a TabFrame control in order to define a mnemonic character which can be used to select that tab and make it active.

Enabled - If the enabled property is set to False for a TabFrame control then the caption will be grayed and the control will no longer respond to selection by the mouse, mnemonic key or keyboard interface.

Visible - The visible property of an inactive tab can be used to suppress the display of that tab - the other tabs in the group will be moved to fill the gap left. The visible property of the active tab can be used to make the entire TabSet invisible.

Left, Top, Height and Width - These properties are synchronised across all TabFrames in a TabSet so that moving or resizing one TabFrame will automatically update the other TabFrames in the TabSet.

BackColor - This property is synchronised across all TabFrames in a TabSet. The default value is determined by the BackColor of the container in which the first TabFrame control is created.

Font Properties and ForeColor - If the AutoFont property is True then these properties are synchronised across all TabFrames in a TabSet.

TabStop - This property is synchronised across all TabFrames in a TabSet.

About Property

Description

Use this property to display the About dialog box which shows the TabFrame version number. Available only at design time.

Active Property

Description

Determines if a TabFrame is the currently active tab within a TabSet. This property is not available at design time.

Usage

[form.]TabFrame.Active[[= True]

Remarks

This property can be used to make a TabFrame control the currently active tab in a TabSet. It cannot, however, be used to make the currently active tab inactive directly - i.e. the property cannot be set to False. This may only be done by making another TabFrame in the TabSet active.

Data Type

Integer (Boolean)

See Also

Activated Event, Deactivated Event

TabGroup Property

Description

Determines the TabGroup of which the TabFrame is a part.

Usage

[form.]TabFrame.TabGroup [= *numericexpression*]

Remarks

The valid range is any integer from 0 to *n* where *n* is the number of TabGroups in the TabSet (which can be determined from the Groups property).

By default a new TabFrame control is added to the end of the TabGroup containing the currently active tab. If you change the TabGroup of a TabFrame control then it will be removed from its current TabGroup and added to the new one. If it was the only TabFrame in the old TabGroup then that group will be deleted and other TabGroups will be renumbered accordingly.

A new TabGroup is created whenever the TabGroup property of an existing TabFrame control is set to the number of TabGroups in the TabSet.

Data Type

Integer

See Also

TabNumber Property, Groups Property, Stagger Property

TabNumber Property

Description

Determines the order in which the tabs are displayed - left to right.

Usage

[form.]TabFrame.TabNumber [= *numericexpression*]

Remarks

The valid range is any integer from 0 to (*n*-1) where *n* is the number of TabFrame controls in the TabGroup of which the control is a part (which can be determined from the Tabs property array).

By default the TabNumber of a new TabFrame control is set so that the tab appears to the right of all other TabFrames in the TabGroup of which it is part. If you change the TabNumber of a TabFrame control then the TabNumbers of other controls in the TabGroup will be automatically updated.

Data Type

Integer

See Also

TabGroup Property, Tabs Property

TabWidth Property

Description

Determines the width, in Twips, of the tab. This property is synchronised across all TabFrames in a TabSet if the AutoWidth property is True.

Usage

[form.]TabFrame.TabWidth [= *numericexpression*]

Remarks

The default value of zero causes the tab width to be set automatically so that the TabFrame controls fill the width of the TabSet.

Data Type

Integer

See Also

TabHeight Property, AutoWidth Property

AutoWidth Property

Description

Deterimes whether or not the TabWidth property is synchronised across the TabFrames in a TabSet. This property is, itself, synchronised across all TabFrames in a TabSet.

Usage

[form.]TabFrame.AutoWidth [= {True|False}]

Remarks

If the AutoWidth property is True then the TabWidth property is synchronised across all TabFrames in the TabSet. If the AutoWidth property is False then thewidth of each tab may be set independently.

Data Type

Integer (Boolean)

See Also

TabWidth Property

TabIndent Property

Description

Determines the indent, in Twips, of the first tab from the edge of the TabSet. This property is synchronised across all TabFrames in a TabSet.

Usage

[form.]TabFrame.TabIndent [= *numericexpression*]

Remarks

If the default TabWidth is used for one or more tabs then the indent is also applied to the last tab in the TabSet.

Data Type

Integer

See Also

TabWidth Property

TabHeight Property

Description

Determines the height, in Twips, of the tab. This property is synchronised across all TabFrames in a TabSet.

Usage

[form.]TabFrame.TabHeight [= *numericexpression*]

Remarks

The default value of zero causes the tab height to be set automatically based on the size of the largest font or picture being used in the TabSet.

Data Type

Integer

See Also

TabWidth Property

Position Property

Description

Determines if the tab is shown at the top or bottom. This property is synchronised across all TabFrames in a TabSet.

Usage

[form.]TabFrame.Position [= *position*]

Remarks

The TabFrame Position property settings are :

Setting	Description
0	(Default) Tabs displayed at top.
1	Tabs displayed at bottom.
2	Tabs displayed at left.
3	Tabs displayed at right.
4	Tabs displayed at right with text facing out.
5	Tabs displayed at left with first tab at the top.

When left/right tabs are used the TabFrame control converts the selected font to the nearest equivalent TrueType font when displaying the caption text. Note also that due to a limitation of the Windows text output routines the underline normally used to denote a mnemonic access key is not supported for left/right tabs (although mnemonic access works as normal).

Data Type

Integer (Enumerated)

Style Property

Description

Determines how the TabFrame control is displayed. This property is synchronised across all TabFrames in a TabSet.

Usage

[form.]TabFrame.Style [= *style*]

Remarks

The TabFrame Style property settings are :

Setting	Description
0	2D
1	3D Raised (Active) - the active tab is shown raised.
2	(Default) 3D Raised (Both) - all tabs shown raised but with the active tab raised more than the inactive tabs.
3	3D Inset (Active) - the active tab is shown inset.
4	3D Inset (Both) - all tabs shown inset but with the active tab inset more than the inactive tabs.

If one of the 3D styles is chosen then both the ActiveColor and InactiveColor properties should be set to light grey - RGB(192,192,192).

Data Type

Integer (Enumerated)

See Also

ActiveColor Property, InactiveColor Property

Chamfer Property

Description

Determines the size, in pixels, of the chamfer at the tab corners. This property is synchronised across all TabFrames in a TabSet.

Usage

[form.]TabFrame.Chamfer [= *numericexpression*]

Remarks

The default value of 4 gives a tab shape similar to that used in MS Word V6 etc. A value of 8 will give a display compatible with earlier versions (V1.0.x) of the TabFrame control.

Data Type

Integer

Stagger Property

Description

Determines the size, in pixelss, of the offest between TabGroups. This property is synchronised across all TabFrames in a TabSet.

Usage

[form.]TabFrame.Stagger [= *numericexpression*]

Remarks

The default value of 10 gives a display similar to that used in MS Word V6 etc.

Data Type

Integer

See Also

TabGroup Property

ActiveColor Property

Description

Determines the background colour of the active tab. This property is synchronised across all TabFrames in a TabSet if the AutoColor property is True.

Usage

[form.]TabFrame.ActiveColor [= *color*]

Remarks

The TabFrame control uses the Microsoft Windows environment RGB scheme for colours and allows the following ranges of settings :

Range of Settings	Description
Normal RGB Colours	Colors specified by using the Color palette, or by using the RGB or QBColor functions in code.
System Default Colors	Colors specified with system color constants from CONSTANT.TXT, a Visual Basic file that specifies system defaults. The Windows environment substitutes the users choices as specified in the users Control Panel settings.

The default is light grey - RGB(192,192,192).

Data Type

Long

See Also

InactiveColor Property, AutoColor Property

InactiveColor Property

Description

Determines the background colour of the inactive tab. This property is synchronised across all TabFrames in a TabSet if the AutoColor property is True.

Usage

[form.]TabFrame.InactiveColor [= *color*]

Remarks

The TabFrame control uses the Microsoft Windows environment RGB scheme for colours and allows the following ranges of settings :

Range of Settings	Description
Normal RGB Colours	Colors specified by using the Color palette, or by using the RGB or QBColor functions in code.
System Default Colors	Colors specified with system color constants from CONSTANT.TXT, a Visual Basic file that specifies system defaults. The Windows environment substitutes the users choices as specified in the users Control Panel settings.

The default is light grey - RGB(192,192,192).

Data Type

Long

See Also

ActiveColor Property, AutoColor Property

AutoColor Property

Description

Deterimes whether or not the active and inactive colour properties are synchronised across the TabFrames in a TabSet. This property is, itself, synchronised across all TabFrames in a TabSet.

Usage

[form.]TabFrame.AutoColor [= {True|False}]

Remarks

If the AutoColor property is True then the following properties are synchronised across all TabFrames in the TabSet :-

ActiveColor

InactiveColor

If the AutoColor property is False then these properties may be set independently for each TabFrame.

Data Type

Integer (Boolean)

See Also

ActiveColor Property, InactiveColor Property

TextOrientation Property

Description

Determines the orientation of the caption text. This property is synchronised across all TabFrames in a TabSet.

Usage

[form.]TabFrame.TextOrientation [= *orientation*]

Remarks

The TabFrame TextOrientation property settings are :

Setting	Description
0	(Default) Text orientation depends on tab position.
1	Horizontal.
2	Vertical to be read from the right.
3	Vertical to be read from the left.

When vertical text is used the TabFrame control converts the selected font to the nearest equivalent TrueType font when displaying the caption text. Note also that due to a limitation of the Windows text output routines the underline normally used to denote a mnemonic access key is not supported for vertical text (although mnemonic access works as normal).

Data Type

Integer (Enumerated)

See Also

Position Property

PicturePosition Property

Description

Determines the position of the picture relative to the caption text. This property is synchronised across all TabFrames in a TabSet.

Usage

[form.]TabFrame.PicturePosition [= *position*]

Remarks

The TabFrame PicturePosition property settings are :

Setting	Description
0	(Default) Picture position depends on tab position and text orientation.
1	Picture displayed to left of caption text.
2	Picture displayed to right of caption text.
3	Picture displayed above caption text.
4	Picture displayed below caption text.

Data Type

Integer (Enumerated)

See Also

TextOrientation Property, Position Property

AutoFont Property

Description

Deterimes whether or not font properties are synchronised across the Tabframes in a TabSet. This property is, itself, synchronised across all TabFrames in a TabSet.

Usage

[form.]TabFrame.AutoFont [= {True|False}]

Remarks

If the AutoFont property is True then the following standard properties are synchronised across all TabFrames in the TabSet :-

ForeColor

FontItalic

FontName

FontSize

FontStrike

FontUnder

The FontBold property setting is ignored and the font for the active tab is made bold whilst all other tabs are not bold.

If the AutoFont property is False then these properties (including FontBold) may be set independently for each TabFrame.

Data Type

Integer (Boolean)

FocusTab Property

Description

Determines if the TabFrame control will get the input focus when selected using the mouse or a mnemonic key.

Usage

[form.]TabFrame.FocusTab [= {True|False}]

Remarks

When a new tab becomes active as a result of being selected by the mouse or a mnemonic key the focus will be set to the TabFrame control if this property is True. Otherwise the focus will be given to a control on the tab.

Data Type

Integer (Boolean)

See Also

FocusTracking Property

FocusTracking Property

Description

Determines if a TabFrame remembers which control has the focus when it is made inactive and resets the focus to that control when it becomes active again. This property is ignored if the FocusTab property is True.

Usage

[form.]TabFrame.FocusTracking [= {True|False}]

Remarks

When a new tab becomes active the focus will be set to a control on the new active tab if, and only if, the focus was previously on a control on the old active tab.

Data Type

Integer (Boolean)

See Also

FocusTab Property

Child3D Property

Description

Determines if the TabFrame control gives child controls a 3D effect.

Usage

[form.]TabFrame.Child3D [= *child3D*]

Remarks

The TabFrame Child3D property settings are :

Setting	Description
0	(Default) None - no child controls are given a 3D effect.
1	All - all child controls are given a 3D effect.
2	Tagged - all child controls where the Tag property is not null are given a 3D effect.
3	Not Graphical - all non-graphical child controls are given a 3D effect.
4	Not Containers - all non-container child controls are given a 3D effect.
5	Not Graphical - all non-graphical and non-container child controls are given a 3D effect.

Data Type

Integer (Enumerated)

See Also

ChildBevel Property, ChildBevelWidth Property

ChildBevel Property

Description

Determines the type of 3D effect given to child controls on the TabFrame.

Usage

[form.]TabFrame.ChildBevel [= *childbevel*]

Remarks

The TabFrame ChildBevel property settings are :

Setting	Description
0	(Default) Inset
1	Raised
2	Shadow

Data Type

Integer (Enumerated)

See Also

Child3D Property, ChildBevelWidth Property

ChildBevelWidth Property

Description

Determines the width of the 3D effect given to child controls.

Usage

[form.]TabFrame.ChildBevelWidth [= *numericexpression*]

Remarks

The bevel width property is in pixels.

Data Type

Integer

See Also

Child3D Property, ChildBevel Property

Groups Property

Description

Specifies the number of TabGroups in the TabSet of which the TabFrame is a part. This property is not available at design time and is read only at runtime.

Usage

[form.]TabFrame.Groups

Remarks

This property is used to determine the number of TabGroups at runtime. A new tab group can be created by setting the TabGroup property of any TabFrame in the TabSet to the value returned by this property.

Data Type

Integer

See Also

TabGroup Property, TotalTabs Property

Tabs Property

Description

An array which specifies the number of TabFrame controls in a TabGroups. This property is not available at design time and is read only at runtime.

Usage

[form.]TabFrame.Tabs(TabGroup)

Remarks

This property is used to determine the number of TabFrames in a TabGroup at runtime.

Data Type

Integer Array

See Also

TabNumber Property, TotalTabs Property

TotalTabs Property

Description

Specifies the total number of TabFrame controls in the TabSet of which the TabFrame is a part. This property is not available at design time and is read only at runtime.

Usage

[form.]TabFrame.TotalTabs

Remarks

This property is used to determine the total number of TabFrame controls at runtime.

Data Type

Integer

See Also

Groups Property, Tabs Property

BufferDisplay Property

Description

Deterimes whether or not painting is done directly to the screen.

Usage

[form.]TabFrame.BufferDisplay [= {True|False}]

Remarks

If this property is True then all display output will be buffered rather than being written directly to the screen. This will slightly increase the time taken to repaint the tabs but will reduce the display flicker particularly when using a larger number of tabs in a TabSet.

Data Type

Integer (Boolean)

Picture Property

Description

Determines the graphic to be displayed on a tab before the caption text.

Usage

[form.]TabFrame.Picture [= *picture*]

Remarks

The picture property settings are :

Setting	Description
(none)	(Default) No picture.
(bitmap, icon)	Specifies a graphic. You can load the graphic from the property window at design time. At runtime you can also set this property using the **LoadPicture** function on a bitmap or icon.

Note that metafiles are not supported.

Data Type

Integer

ClientLeft, ClientTop, ClientWidth, ClientHeight Properties

Description

Determines the size and position (in twips) of the active tab client area relative to the top left corner of the TabFrame control. These properties are not available at design time and are read-only at runtime.

Usage

[form.]TabFrame.ClientLeft

[form.]TabFrame.ClientTop

[form.]TabFrame.ClientWidth

[form.]TabFrame.ClientHeight

Remarks

These properties can be used to move and resize child controls when the active client area of a TabFrame changes. This could be caused by resizing of the TabFrame control or changing the value of one or more properties controlling the TabFrame appearance.

The active client area of a TabFrame is the area in which controls are normally placed. It excludes the tab area, the dead area used to represent stacked tabs and the area used to give the TabFrame a 3D effect.

Data Type

Long

Events

The following events are supported by the TabFrame control.

Standard Events

Click
DblClick
DragDrop
DragOver
GotFocus
LostFocus
MouseDown
MouseMove
MouseUp

Custom Properties

Activated
Deactivated

Standard Events

For complete documentation on the standard events supported by the TabFrame control please refer to the Visual Basic Language Reference. The comments below indicate where the TabFrame control differs from other controls in the use of these events.

Click and DblClick - These events will occur for the active tab whenever the mouse button is clicked (or double clicked) within the active tab area. Clicking the mouse button on an inactive tab will make that tab active while clicking in the tab area but not over a tab will be ignored.

MouseDown, MouseUp and MouseMove - These events will occur for the active tab whenever the mouse is moved or clicked within the area of the TabSet.

Activated Event

Description

Occurs whenever the TabFrame becomes active. This may be caused by the user clicking on the (inactive) tab with the left mouse button or when the TabFrame Active property is set True in code.

Syntax

Sub *ctlname*_Activated([Index as Integer])

Remarks

The argument Index identifies the control if it is part of a control array.

The activated event occurs after the new tab has become active.

See Also

Active Property, Deactivated Event

Deactivated Event

Description

Occurs whenever the TabFrame becomes inactive as a result of another TabFrame becoming active. This may be caused by the user clicking on another (inactive) tab with the left mouse button or when the TabFrame Active property for another tab is set True in code.

Syntax

Sub *ctlname*_Deactivated([Index as Integer,] Cancel as Integer)

Remarks

The argument index identifies the control if it is part of a control array.

The deactivated event occurs before the tab becomes inactive. Deactivation of the tab (and activation of a new tab) can be cancelled by setting the Cancel argument to True before returning.

See Also

Active Property, Activated Event

Methods

The following methods are supported by the TabFrame control.

- Drag
- Move
- Refresh
- ZOrder

Application Notes

The following topics describe techniques which are commonly used with the TabFrame control.

- Creating multiple TabSets on a form
- Creating a stacked tab dialog
- Using common controls
- Using control arrays
- Use with the ToolTip control

Creating Multiple TabSets

Each Visual Basic container can hold a single TabSet; usually the form is the container and new TabFrame controls can be added by selecting the form and double clicking the TabFrame icon in the toolbox. If, however, your application requires multiple TabSets on a form then the following procedure should be followed :-

1) Create the first TabSet in the usual way.

2) Add a container control (e.g. a picture box) for each additional TabSet required.

3) Add TabFrame controls to the additional TabSets by drawing each TabFrame control **inside** the appropriate container. This is made easier if you make the first TabFrame control in each additional TabSet smaller than the container and then resize the TabSet after all controls have been added.

Note that since the TabFrame control is itself a container you must **not** draw a new TabFrame control within an existing one unless you intend to created a nested TabSet (i.e. a TabSet within a TabFrame).

An alternative method for adding TabFrame controls to TabSets created within a container other than the form is to cut (or copy) a TabFrame control onto the clipboard, select the container control and then paste the TabFrame into it.

Creating Stacked Tab Dialogs

A stacked tab display is created by using mutliple TabGroups within a TabSet. The procedure for creating the required TabGroups is :-

1) Create the TabSet in the normal way and add all the TabFrames in the first group together with the first TabFrame in the second group.

2) Set the TabGroup property of the first TabFrame in the second group to 1.

3) Add the remaining TabFrames in the second group together with the first TabFrame in the third TabGroup.

4) Set The TabGroup property of the first TabFrame in the third group to 2.

5) Continue adding TabGroups and TabFrames until all the groups have been created.

Tabs can be moved between groups and within a group using the TabGroup and TabNumber properties. Note that changing the value of one of these properties for a TabFrame may cause the property values for other TabFrames in the TabSet to be updated in order to ensure that tab numbering remains contiguous.

Using Common Controls

Common controls are controls which appear on all TabFrames within a TabSet such as, for example, the Ok and Cancel buttons used on many dialogs. Such controls are easily implemented by drawing the control directly on the form (i.e. not within the TabFrame) and then dragging the control over the TabSet. It may also be necessary to force the control to the front of the display Zorder by selecting it and using the Edit menu Bring to Front command.

If controls are not required on all TabFrames within the TabSet then the Activated and/or Deactivated events can be used to set the visible property of the control as appropriate.

Using Control Arrays

It is highly recommended that all the TabFrame controls in a TabSet be made part of a control array. This not only reduces the system resources used by the TabSet but also simplifies (in most cases) the implementation of the Activated and Deactivated event handlers.

One method of creating a TabSet using a control array is :-

1) Create the first TabFrame in the normal way and set the Index property to 0.

2) Copy the TabFrame control created in step (1) to the clipboard.

3) For each additional TabFrame control required select the form (or container control if using multiple TabSets) and paste the TabFrame from the clipboard into the TabSet.

Each new TabFrame control will be added to the end of the currently active group.

Creating a new TabFrame control at runtime is done by adding to the control array as for any other control. The new TabFrame control will be added to the end of the currently active group.

Use with the ToolTip Control

When used in conjunction with the ToolTip VBX the TabFrame control can provide individual tool tip and mouse tracking for each tab. The ToolTip control GetToolTip, MouseEnter and MouseExit events are generated as the mouse moves over each tab (the client area of the active tab is ignored).

Change History

Date	Description
2-Feb-1994	V1.0.0 - original version.
21-Feb-1994	V1.0.1 - fixed bug which required LIC file for use with compiled programs.
2-Mar-1994	V1.1.0 - functional enhancements :-
	a) Support for stacked (multi-line) tabs.
	b) Support for left/right tabs.
	c) New Chamfer property to customise tab appearance.
	d) New properties Groups, Tabs and TotalTabs.
16-Mar-1994	V1.2.0 - functional enhancements :-
	a) Added BufferDisplay property.
	b) Caption on disabled tabs shown in gray text.
	c) Added AutoColor property.
	d) Improved painting to reduce display flicker.
18-Apr-94	V1.3.0
	a) Added support for pictures on tabs.
	b) Visible property can be used to hide inactive tabs.
	c) Support for outword facing text on right sided tabs.
	d) Fixed bug which allowed selection of disabled tab by mnemonic key.
	e) Fixed bugs concerning use of non-synchronised colours.
	f) Added support for ToolTip control.

Registration

This version of the TabFrame control is fully functional but may only be used in the development environment. Any attempt to use this version in conjunction with an EXE file will display a dialog box identifying this as a demonstration version and the control will fail to load.

If you find the TabFrame control useful then you can receive a full version by registering as follows :

1) In the SWREG forum on Compuserve. The fee is be $39.95 and the registration ID is 2018.

2) By sending a cheque or money order for £25 to :

GC Consulting Services Ltd
Fellsgarth House
Hognaston
Ashbourne
Derbyshire DE6 1PR
ENGLAND

In return for your registration you will receive the latest version of the TabFrame control and be eligible for product support (via the MSBASIC forum on Compuserve) and free product upgrades for a period of 12 months.

Any questions should be sent to Graham Cockell (Compuserve ID 100113,2774) via e-mail or the MSBASIC forum.