Table Of Contents

Introduction

oTab(T) is a TopClass version of oTab(), a tabbed dialog class for Clip4Win. It allows you to create tabbed dialogs like those used in many Windows applications.

oTab currently works only with resource based dialogs. Each tabbed dialog consists of a "host" dialog and tab "pages" which are individual dialogs containing the controls for each "page".

When designing your tabbed dialogs it is recommended that you first create your "host" dialog then your page dialogs. Makeing the page dialogs the same size as the "host" will make placement of the controls on each page much easier. When placing your controls keep in mind that you must allow space for the tabs to be drawn.

Tabbed dialogs are limited in theory to a maximum of 256 controls including static ones. A more practical limit would be 100 or less. If you encounter a situation where all controls are not properly displayed or do not work correctly then you need to reduce the total number of controls in the dialog or try increasing the heap setting in your .def file.

This version of oTab only supports a single row of tabs and only allows tabs to be positioned along the top edge.

Tabbed dialogs are best used in situations such setting up/configuring an application. As shown in the demo program, you can also use them to do data entry if you keep the number of controls within reason.

oTab() Class.

Properties:

- **bottom**
- **ctl3D**
- **dialog**
- **dlgProc**
- **initProc**
- **left**
- **parent**
- **right**
- **tabStyle**
- **tabFont**
- **top**

Methods:

- **addPage()**
- **doModal()**
- **doModeless()**
- **close()**
- **disablePage()**
- **enablePage()**
- **getActivePage()**
- **init()**
- **setActivePage()**

oTab:bottom (Access/Assign)

This property defines the bottom edge of the tab area within the "host" dialog. The value is specified in dialog units.

If not set then it defaults to the height of the "host" dialog's client area - 1.

Example:

```
oTab        := oTab{}
oTab:parent := hMainWnd
oTab:dialog := ID_SETUP
oTab:left   := 1          // Left edge of tab area
oTab:top    := 1          // Top edge of tab area
oTab:right  := 200        // Right edge of tab area
oTab:bottom := 140        // bottom edge of tab area
```

oTab:ctl3D (Access/Assign)

This property controls how the tabs are drawn. When set to .T. oTab assumes that CTL3D.DLL is being used and draws the tabs to match the 3d look provided by CTL3D. If set to .F. then oTab uses the current system colors to draw the tabs.

The default is .T.

Example:

```
oTab         := oTab{}
oTab:parent  := hMainWnd
oTab:dialog  := ID_SETUP
oTab:left    := 1
oTab:top     := 1
oTab:right   := 200
oTab:bottom  := 140
oTab:ctl3D   := .F.         // No 3D please!
```

oTab:dialog (Access/Assign)

This property specifies the name/id of the dialog that will serve as the "host" dialog for the tabs. oTab only works with dialogs that are contained within the applications resource file.

Example:

```
oTab         := oTab{}
oTab:parent  := hMainWnd
oTab:dialog  := ID_SETUP   // Id of "host" dialog
oTab:left    := 1          // Left edge of tab area
oTab:top     := 1          // Top edge of tab area
oTab:right   := 200        // Right edge of tab area
oTab:bottom  := 140        // bottom edge of tab area
```

oTab:dlgProc (Access/Assign)

This is a code block that is evaluated by the internal oTab dialog procedure each time a message is received. The following parameters are passed:

Self, hDlgWnd, nMsg, nWparam, nLparam.

You should structure your dialog procedure the same way you do for a normal dialog. The only exception is that you **MUST** not use EndDialog() or DestroyWindow() when you wish to close the tabbed dialog. You **MUST** use the close() method.

Example:

```
STATIC FUNCTION DlgProc( oTab, hDlgWnd, nMsg, nWparam, nLparam )

      DO CASE
         CASE nMsg == WM_INITDIALOG
              LoadListBox( GetDlgItem( hDlgWnd, 1002 ) )

         CASE nMsg == WM_COMMAND
              DO CASE
                 CASE nWparam == IDOK
                      oTab:close()

                 CASE nWparam == IDCANCEL
                      oTab:close()

              ENDCASE
      ENDCASE
RETURN( 0 )
```

oTab:initProc (Access/Assign)

This is a code block that oTab will evaluate when it processes the WM_INITDIALOG message internally. This code block is evaluated **BEFORE** oTab automatically configures itself.

When evaluated the following parameters are passed:

Self, hDlgWnd, nMsg, nWparam, nLparam.

Please note that nMsg will always be WM_INITDIALOG.

Example:

```
oTab          := oTab{ hMainWnd, ID_SETUP, 0, 0, 200, 100 }
oTab:initProc := {|oTab,hDlgWnd,nMsg,nWparam,nLparam|;
                  CenterWindow( hDlgWnd ) }
```

oTab:left (Access/Assign)

This property defines the left edge of the tab area within the "host" dialog. The value is specified in dialog units.

If not set then it defaults to 1.

Example:

```
oTab        := oTab{}
oTab:parent := hMainWnd
oTab:dialog := ID_SETUP
oTab:left   := 1          // Left edge of tab area
oTab:top    := 1          // Top edge of tab area
oTab:right  := 200        // Right edge of tab area
oTab:bottom := 140        // bottom edge of tab area
```

oTab:parent (Access/Assign)

This property specifies the handle to the window that is the parent of the "host" dialog that will contain the tabs.

Example:

```
oTab        := oTab{}
oTab:parent := hMainWnd   // Parent window handle
oTab:dialog := ID_SETUP   // Tab host dialog ID#
oTab:left   := 1          // Left edge of tab area
oTab:top    := 1          // Top edge of tab area
oTab:right  := 200        // Right edge of tab area
oTab:bottom := 140        // bottom edge of tab area
```

oTab:right (Access/Assign)

This property defines the right edge of the tab area within the "host" dialog. The value is specified in dialog units.

If not set then it defaults to the width of the "host" dialog's client area - 1.

Example:

```
oTab        := oTab{}
oTab:parent := hMainWnd
oTab:dialog := ID_SETUP
oTab:left   := 1          // Left edge of tab area
oTab:top    := 1          // Top edge of tab area
oTab:right  := 200        // Right edge of tab area
oTab:bottom := 140        // bottom edge of tab area
```

oTab:tabStyle (Access/Assign)

This property controls how the tabs will be sized. If you set this property to 1 then oTab will make each tab the same size based on the tab area available. If you set this to 2 then oTab will size the tabs based on the tab label width for each tab.

The default is 1.

Example:

```
oTab          := oTab{}
oTab:parent   := hMainWnd   // Parent window handle
oTab:dialog   := ID_SETUP   // Tab host dialog ID#
oTab:left     := 1          // Left edge of tab area
oTab:top      := 1          // Top edge of tab area
oTab:right    := 200        // Right edge of tab area
oTab:bottom   := 140        // bottom edge of tab area
oTab:tabStyle := 2          // Size tabs based on label width
```

oTab:tabFont (Access/Assign)

This is a handle to a valid font that will be used for the tab labels. Please note that oTab will not destroy this font for you. If no font is specified then oTab uses the font for the "host" dialog.

Example:

```
oTab          := oTab{}
oTab:parent   := hMainWnd   // Parent window handle
oTab:dialog   := ID_SETUP   // Tab host dialog ID#
oTab:left     := 1          // Left edge of tab area
oTab:top      := 1          // Top edge of tab area
oTab:right    := 200        // Right edge of tab area
oTab:bottom   := 140        // bottom edge of tab area
oTab:tabStyle := 2          // Size tabs based on label width
oTab:tabFont  := hTfont     // Use our font for the tab labels
```

oTab:top (Access/Assign)

This property defines the top edge of the tab area within the "host" dialog. The value is specified in dialog units.

If not set then it defaults to 1.

Example:

```
oTab        := oTab{}
oTab:parent := hMainWnd
oTab:dialog := ID_SETUP
oTab:left   := 1          // Left edge of tab area
oTab:top    := 1          // Top edge of tab area
oTab:right  := 200        // Right edge of tab area
oTab:bottom := 140        // bottom edge of tab area
```

oTab:addPage()Method

Add a page to a tabbed dialog.

Syntax:

addPage(<nPage>, <xLabel>, [<nColor>], [<nFG>], [<nBG>]) -> **Self**

Arguments:

<nPage> is the id of the dialog containing the controls for this page.

<xLabel> is a character string/variable to be used for the tab label for this page. An array can be passed if you want more than one line of text on the tab label.

[<nColor>] is a RGB() value to be used for the tab color.

[<nFG>] is a RGB() value to be used for the foreground color of the tab label.

[<nBG>] is a RGB() value to be used for the background color of the tab label.

Returns:

Self.

Description:

This method adds a page to a tabbed dialog. You **MUST** add all pages before you call doModal() or doModeless(). Pages cannot be added once the tabbed dialog is active.

Example:

```
oTab          := oTab{}
oTab:parent   := hMainWnd   // Parent window handle
oTab:dialog   := ID_SETUP   // Tab host dialog ID#
oTab:left     := 1          // Left edge of tab area
oTab:top      := 1          // Top edge of tab area
oTab:right    := 200        // Right edge of tab area
oTab:bottom   := 140        // bottom edge of tab area
oTab:tabStyle := 2          // Size tabs based on label width
oTab:tabFont  := hTfont     // Use our font for the tab labels
oTab:ctl3D    := .F.        // No grays here please
oTab:dlgProc  := {|oTab,hWnd,nMsg,nWparam,nLparam|;
                  DlgProc( oTab, hWnd, nMsg, nWparam, nLparam ) }

// Add our pages...

oTab:addPage( ID_PAGE1, "Fonts", RGB( 128, 0, 0 ),;
             RGB( 255, 255, 255 ), RGB( 128, 0, 0 ) )
oTab:addPage( ID_PAGE2, "Directories", RGB( 128, 0, 0 ),;
             RGB( 255, 255, 255 ), RGB( 128, 0, 0 ) )
oTab:addPage( ID_PAGE3, "Misc", RGB( 128, 0, 0 ),;
             RGB( 255, 255, 255 ), RGB( 128, 0, 0 ) )
```

oTab:doModal()

Activate a modal tabbed dialog.

Syntax:

doModal() -> **Self**

Arguments:

None.

Returns:

Self.

Description:

This method is called once you have created and configured a tabbed dialog. The tabbed dialog is created as a modal dialog.

Example:

```
oTab          := oTab{}
oTab:parent   := hMainWnd   // Parent window handle
oTab:dialog   := ID_SETUP   // Tab host dialog ID#
oTab:left     := 1          // Left edge of tab area
oTab:top      := 1          // Top edge of tab area
oTab:right    := 200        // Right edge of tab area
oTab:bottom   := 140        // bottom edge of tab area
oTab:tabStyle := 2          // Size tabs based on label width
oTab:tabFont  := hTfont     // Use our font for the tab labels
oTab:ctl3D    := .F.        // No grays here please
oTab:dlgProc  := {|oTab,hWnd,nMsg,nWparam,nLparam|;
                  DlgProc( oTab, hWnd, nMsg, nWparam, nLparam ) }

// Add our pages...

oTab:addPage( ID_PAGE1, "Fonts", RGB( 128, 0, 0 ),;
             RGB( 255, 255, 255 ), RGB( 128, 0, 0 ) )
oTab:addPage( ID_PAGE2, "Directories", RGB( 128, 0, 0 ),;
             RGB( 255, 255, 255 ), RGB( 128, 0, 0 ) )
oTab:addPage( ID_PAGE3, "Misc", RGB( 128, 0, 0 ),;
             RGB( 255, 255, 255 ), RGB( 128, 0, 0 ) )

// Make it modal and active...

oTab:doModal()
```

oTab:doModeless() Method

Activate a modeless tabbed dialog.

Syntax:

doModeless() -> **Self**

Arguments:

None.

Returns:

Self.

Description:

This method is called once you have created and configured a tabbed dialog. The tabbed dialog is created as a modeless dialog.

Example:

```
oTab          := oTab{}
oTab:parent   := hMainWnd   // Parent window handle
oTab:dialog   := ID_SETUP   // Tab host dialog ID#
oTab:left     := 1          // Left edge of tab area
oTab:top      := 1          // Top edge of tab area
oTab:right    := 200        // Right edge of tab area
oTab:bottom   := 140        // bottom edge of tab area
oTab:tabStyle := 2          // Size tabs based on label width
oTab:tabFont  := hTfont     // Use our font for the tab labels
oTab:ctl3D    := .F.        // No grays here please
oTab:dlgProc  := {|oTab,hWnd,nMsg,nWparam,nLparam|;
                  DlgProc( oTab, hWnd, nMsg, nWparam, nLparam ) }

// Add our pages...

oTab:addPage( ID_PAGE1, "Fonts", RGB( 128, 0, 0 ),;
             RGB( 255, 255, 255 ), RGB( 128, 0, 0 ) )
oTab:addPage( ID_PAGE2, "Directories", RGB( 0,128, 0 ),;
             RGB( 255, 255, 255 ), RGB( 0 , 128, 0 ) )
oTab:addPage( ID_PAGE3, "Misc", RGB( 0, 0, 128 ),;
             RGB( 255, 255, 255 ), RGB( 0, 0, 128 ) )

// Make it modeless and active...

oTab:doModeless()
```

oTab:close() Method

Close a tabbed dialog.

Syntax:

close() -> **Self**

Arguments:

None.

Returns:

Self.

Description:

This is the **only** way to close a tabbed dialog. You **cannot** use EndDialog() or DestroyWindow() for a tabbed dialog.

Example:

```
STATIC FUNCTION DlgProc( oTab, hDlgWnd, nMsg, nWparam, nLparam )

      DO CASE
         CASE nMsg == WM_INITDIALOG
              LoadListBox( GetDlgItem( hDlgWnd, 1002 ) )

         CASE nMsg == WM_COMMAND
              DO CASE
                 CASE nWparam == IDOK
                      oTab:close()  // Don't even think about
                                    // using EndDialog()

                 CASE nWparam == IDCANCEL
                      oTab:close()  // Don't even think about
                                    // using DestroyWindow()

              ENDCASE
      ENDCASE
RETURN( 0 )
```

oTab:disablePage() Method

Disable a page in a tabbed dialog.

Syntax:

disablePage(<nPage>) -> **Self**

Arguments:

<nPage> Dialog page to disable.

Returns:

Self.

Description:

This method is used when you want to disable a tab page so the user cannot access it.

Example:

```
STATIC FUNCTION DlgProc( oTab, hDlgWnd, nMsg, nWparam, nLparam )

      DO CASE
         CASE nMsg == WM_INITDIALOG
              LoadListBox( GetDlgItem( hDlgWnd, 1002 ) )
              oTab:disablePage( 2 ) // No page 2 yet...

         CASE nMsg == WM_COMMAND
              DO CASE
                 CASE nWparam == ID_EDIT
                      oTab:enablePage( 2 ) // Make page 2 available
                 CASE nWparam == IDOK
                      oTab:close()

                 CASE nWparam == IDCANCEL
                      oTab:close()

              ENDCASE
      ENDCASE
RETURN( 0 )
```

oTab:enablePage() Method

Enable a previously disabled dialog page.

Syntax:

enablePage(<nPage>) -> **Self**

Arguments:

<nPage> is the dialog page to enable.

Returns:

Self.

Description:

This method is used to make a previously disabled tab page available.

Example:

```
STATIC FUNCTION DlgProc( oTab, hDlgWnd, nMsg, nWparam, nLparam )

      DO CASE
         CASE nMsg == WM_INITDIALOG
              LoadListBox( GetDlgItem( hDlgWnd, 1002 ) )
              oTab:disablePage( 2 ) // No page 2 yet...

         CASE nMsg == WM_COMMAND
              DO CASE
                 CASE nWparam == ID_EDIT
                      oTab:enablePage( 2 ) // Make page 2 available

                 CASE nWparam == IDOK
                      oTab:close()

                 CASE nWparam == IDCANCEL
                      oTab:close()

              ENDCASE
      ENDCASE
RETURN( 0 )
```

oTab:getActivePage() Method

Get the currently active dialog page.

Syntax:

getActivePage() -> **nPage**

Arguments:

None.

Returns:

The number of the currently active dialog page.

Description:

This method is used to obtain the currently active tab page.

Example:

```
STATIC FUNCTION DlgProc( oTab, hDlgWnd, nMsg, nWparam, nLparam )

      DO CASE
         CASE nMsg == WM_INITDIALOG
              LoadListBox( GetDlgItem( hDlgWnd, 1002 ) )

         CASE nMsg == WM_COMMAND
              DO CASE
                 CASE nWparam == ID_EDIT
                      IF oTab:getActivePage() == 2
                         DoEdit( oTab )
                      ENDIF

                 CASE nWparam == IDOK
                      oTab:close()

                 CASE nWparam == IDCANCEL
                      oTab:close()

              ENDCASE
      ENDCASE
RETURN( 0 )
```

oTab:init() Method

Creates new oTab object.

Syntax:

oTab{ <hParent>, <xHost>, [<left>], [<top>]
[<right>], [<bottom>] } -> **Self**

Arguments:

<hParent> handle of the parent window for the "host" dialog.
<xHost> name or id# of the "host" dialog.
[<left>] left edge of the tab area.
[<top>] top edge of the tab area.
[<right>] right edge of the tab area.
[<bottom>] bottom edge of the tab area.

NOTE: If left, top, right, or bottom are not set then oTab will use the entire "host" dialog client area for the tab area.

Returns:

Self.

Description:

Creates a new oTab object. Please note that the values for the tab area are expressed in dialog units not in pixels.

Example:

```
// Create a new tabbed dialog object

oTab := oTab{ hMainWnd, ID_SETUP, 1, 1, 200, 140 }
```

oTab:setActivePage() Method

Set active dialog page.

Syntax:

setActivePage(<nPage>) -> **Self**

Arguments:

None.

Returns:

Self.

Description:

Sets the active tabbed dialog page.

Example:

```
STATIC FUNCTION DlgProc( oTab, hDlgWnd, nMsg, nWparam, nLparam )

      DO CASE
         CASE nMsg == WM_INITDIALOG
              LoadListBox( GetDlgItem( hDlgWnd, 1002 ) )

         CASE nMsg == WM_COMMAND
              DO CASE
                 CASE nWparam == ID_EDIT
                      oTab:setActivePage( 2 )  // Switch to page 2

                 CASE nWparam == IDOK
                      oTab:close()

                 CASE nWparam == IDCANCEL
                      oTab:close()

              ENDCASE
      ENDCASE
RETURN( 0 )
```

Technical Support

oTab users may obtain technical support via phone, fax or CIS Mail. Technical support is free of charge except for the cost of a stamp, phone call, or fax transmission! Phone support is available between 08:00 and 17:00 EST - Monday thru Friday except for national holidays. Technical support contact numbers:

Logical Systems
2635 Portsmouth Place
Hephzibah, GA 30815 USA

Phone: (706) 790-7303
FAX: (706) 790-7084
CIS: 73257,2066

For those of you who prefer to use Class(y) or HighClass instead of TopClass you can obtain a copy of oTab() which supports these products from the following:

S & A PC Solutions, Inc.
24 Lena Road
Forestburg, New York 12777

Phone: (914) 794-4647
FAX: (914) 794-6447
CIS: 71650,2251

Please contact them for current pricing and ordering information.