A Property Sheet in a **Dock**ing Toolbar

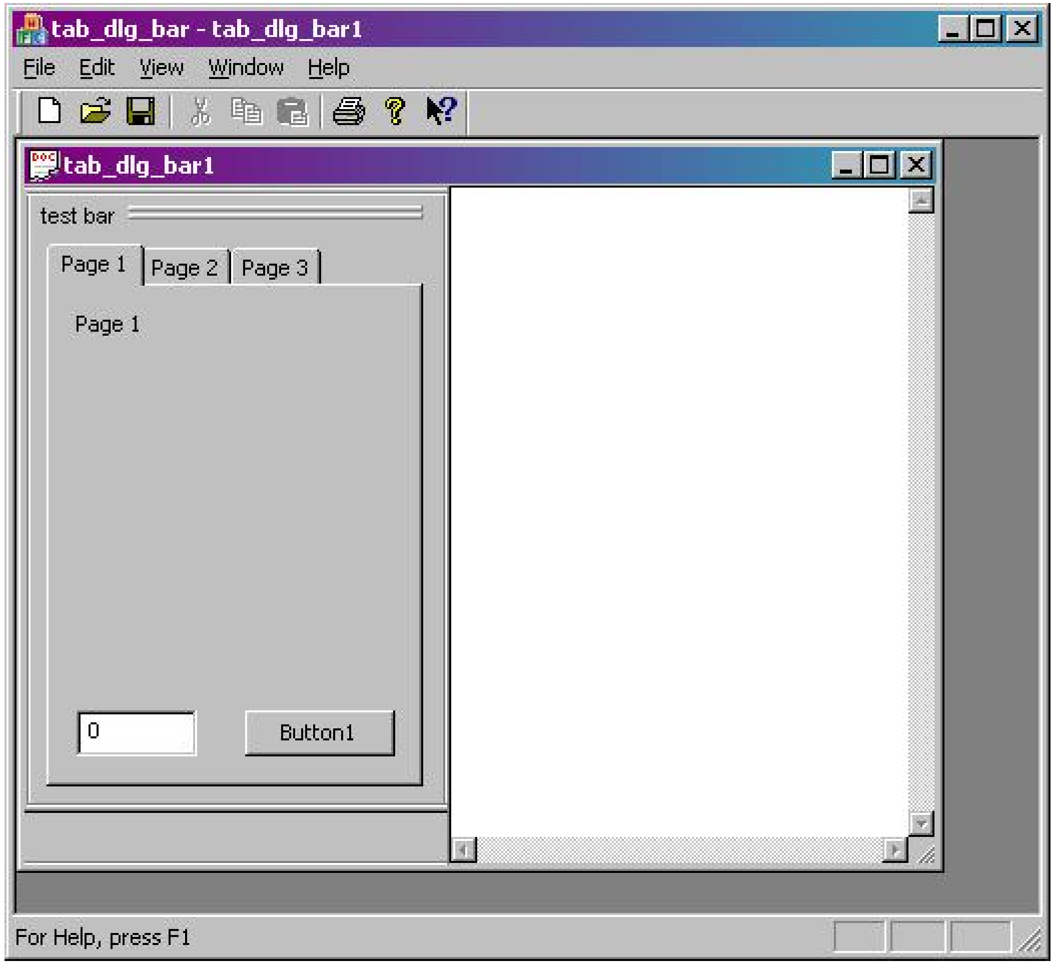
Bill Heitler, 27 Nov 2004 CPOL

Rate:

★★★★☆ 4.06 (7 votes)

Explains how to put a CPropertySheet into a CControlBar.

Download demo project - 105 Kb



Introduction

I wanted to display a PropertySheet within a **dock**able toolbar, I wanted the sheet to have a title in both the **dock**ed and un**dock**ed state so that the user knew what it was about, and I wanted it to **dock** to an MDI View rather than to the main frame. I doubt if these requirements would be much of a challenge to a professional **MFC** programmer, but since I'm not one, it took me a while to figure it out, and I thought it might save others some time if I shared the code.

The main problems were as follows. First, getting a property sheet to display in a raw `CControlBar`. Second, standard property sheets display with a certain (quite large) minimum width and height, but in general, one wants toolbars to be fairly compact. Luckily, this is a problem that has already been solved in an excellent article by Antonio Lacaci, and I use his code with just slight modification. Third, control bars have a caption bar and title in the un**dock**ed floating state, but just show the grippers in the **dock**ed state, so the title has to be hand coded when the bar is **dock**ed. Fourth, and trivially, to get a toolbar to **dock** to a `CView`-derived class, you have to remember that it actually **dock**s to the frame parent of the view, not to the view itself.

Using the Property Sheet Bar

The functionality is implemented in the class `CPropSheetBar`, which is instantiated as a member variable within the View. To use the class, you first have to construct and add pages derived from `CPropertyPage`, just as you would with a normal property sheet. These pages must live as long as the property sheet bar, and are best constructed on the heap within the `OnCreate` function of the view, before creating the bar itself:

So, in *View.h*, we have:

Hide   Copy Code

```
CPropSheetBar m_PropSheetBar;
CPage1 *page1;
CPage2 *page2;
CPage3 *page3;
```

and in `View::OnCreate`, we have:

Hide   Copy Code

```
page1=new CPage1;
page2=new CPage2;
page3=new CPage3;
// for data exchange, I give the page a view pointer
page1->m_pView=this;
page2->m_pView=this;
page3->m_pView=this;

m_PropSheetBar.AddPage(page1);
m_PropSheetBar.AddPage(page2);
m_PropSheetBar.AddPage(page3);

CFrameWnd *pFrameWnd=(CFrameWnd *)GetParent();
if (!m_PropSheetBar.Create("test bar", pFrameWnd,
                    AFX_IDW_CONTROLBAR_FIRST+40))
  return -1; // fail to create
m_PropSheetBar.EnableDocking(CBRS_ALIGN_ANY);
pFrameWnd->EnableDocking(CBRS_ALIGN_ANY);
pFrameWnd->DockControlBar(&m_PropSheetBar);
```

You must add the property pages before calling `CPropSheetBar::Create`, because `Create` needs to know the size of the dialog resources that it will display. You delete the pages in the View destructor. You pass in the title of the property sheet bar in the `Create` function.

Implementing CPropSheetBar

`CPropSheetBar` derives from `CControlBar`, and has a member variable `m_cPropSheet` of class `CShrinkingPropSheet`. This latter is derived from a standard `CPropertySheet`, and implements Antonio's fix for the size problem. `CPropSheetBar` has an `AddPage` utility function that just adds pages to the embedded property sheet:

Hide Copy Code

```
void AddPage(CPropertyPage * pPage) {m_cPropSheet.AddPage(pPage);}
```

`CPropSheetBar` has in it the following overrides:

Hide Copy Code

```
virtual void DrawGripper(CDC *pDC,const CRect &rect);
virtual CSize CalcDynamicLayout( int nLength, DWORD dwMode );
virtual BOOL Create(LPCTSTR lpszWindowName, CWnd* pParentWnd, UINT nID);
```

and several of the concepts used in these have been picked up from Alger Pike's article "A DevStudio-like CControlBar".

`CPropSheetBar::Create` first creates the control bar itself, and then creates the property sheet within it. It measures the size of the property sheet (which is a shrinking sheet, no thanks to Microsoft) and stores it.

Hide Copy Code

```
BOOL CPropSheetBar::Create(LPCTSTR lpszWindowName, CWnd* pParentWnd, UINT nID)
{
  // create the base window
  CString lpszClassName = AfxRegisterWndClass(CS_DBLCLKS,
    LoadCursor(NULL, IDC_ARROW),m_brushBkgd, 0);
  DWORD style=WS_CHILD | WS_VISIBLE | CBRS_LEFT | CBRS_GRIPPER |
              CBRS_TOOLTIPS | CBRS_FLYBY;
  m_dwStyle = style & CBRS_ALL;
  if (!CControlBar::Create(lpszClassName, lpszWindowName,
                   style, CRect(0,0,0,0), pParentWnd,NULL))
    return FALSE;

  m_strTitle=lpszWindowName;

  // create the property sheet
  m_cPropSheet.Create(this,WS_CHILD|WS_VISIBLE);
  m_cPropSheet.SetTitle(lpszWindowName);
  CClientDC dc(this);
  m_sizeTitle=dc.GetTextExtent(m_strTitle);
  CRect rc;
  m_cPropSheet.GetWindowRect(rc); // screen coordinates
  m_sizePropSheet=rc.Size();

  return TRUE;
}
```

`CPropSheetBar::CalcDynamicLayout` is the key function that Windows calls when it displays a control bar. The override tells Windows what the size of the bar is now that it contains the property sheet, allowing suitable margins around the sheet. It also moves the property sheet to position it within the bar appropriately, depending on whether or not we have to hand-draw a title.

Hide Copy Code

```
CSize CPropSheetBar::CalcDynamicLayout(int nLength, DWORD dwMode)
{
  if (IsFloating())
  {
    CRect rc(CPoint(6,6),m_sizePropSheet);
    m_cPropSheet.MoveWindow(rc);
    return m_sizePropSheet+CSize(16,16);
  }

  CRect rc(CPoint(4,m_sizeTitle.cy+sizeTitleOffset.cy), m_sizePropSheet);
  m_cPropSheet.MoveWindow(rc);
  return m_sizePropSheet+CSize(16, m_sizeTitle.cy+sizeTitleOffset.cy+10);
}
```

`CPropSheetBar::DrawGripper` calls the default if the bar is floating because Windows supplies a title bar, otherwise it draws grippers and writes the title. I have not made the gripper location dependent on the **dock**ing position, but this could be done by testing.

Hide Copy Code

```
if( m_dwStyle & CBRS_ORIENT_HORZ )
```

and coding appropriately.

Hide Shrink ▲ Copy Code

```
void CPropSheetBar::DrawGripper(CDC *pDC,const CRect &rect)
{
  if (IsFloating())
  {
    CControlBar::DrawGripper(pDC,rect);
    return;
  }

  CFont font;
  VERIFY(font.CreateFont(
    14, // nHeight
    0, // nWidth
    0, // nEscapement
    0, // nOrientation
    FW_NORMAL, // nWeight
    FALSE, // bItalic
    FALSE, // bUnderline
    0, // cStrikeOut
    ANSI_CHARSET, // nCharSet
    OUT_DEFAULT_PRECIS, // nOutPrecision
    CLIP_DEFAULT_PRECIS, // nClipPrecision
    DEFAULT_QUALITY, // nQuality
    DEFAULT_PITCH | FF_SWISS, // nPitchAndFamily
    "Arial")); // lpszFacename

  CFont *oF=pDC->SelectObject(&font);
  pDC->SetBkColor(GetSysColor(COLOR_BTNFACE));
  pDC->TextOut(rect.left+sizeTitleOffset.cx,
                rect.top+sizeTitleOffset.cy, m_strTitle);
  CSize sz=pDC->GetTextExtent(m_strTitle);

  // Draw the docking grippers
  CRect rectGrip1(rect.left+sz.cx+2*sizeTitleOffset.cx,7,rect.right-10,7+3);
  pDC->Draw3dRect(rectGrip1, RGB(255,255,255),
                   RGB(128,128,128));
  CRect rectGrip2(rect.left+sz.cx+2*sizeTitleOffset.cx,11,rect.right-10, 11+3);
  pDC->Draw3dRect(rectGrip2, RGB(255,255,255), RGB(128,128,128));
}
```

There is one small problem with the bar: if you repeatedly double click the bar to **dock** and un**dock** it, it gradually slides down the margin of the View! It's easily moved back up by hand, but it's irritating, and if anyone knows how to fix it, please do let me know.

That's it! It's not rocket science, but it may be useful.

License

Share

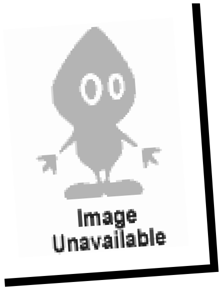
EMAIL TWITTER

About the Author

Bill Heitler

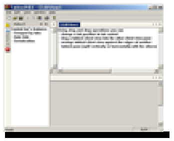
Software Developer The University of St. Andrews

United Kingdom



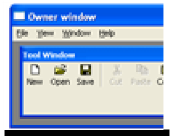
I am a university academic specialising in Neuroscience. I write simulation and analysis software.

You may also be interested in...

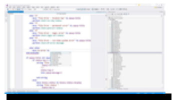


MFC Docking Framework

Mobile Messaging with Twilio

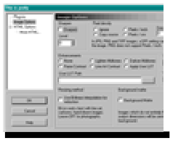


Docking Toolbars in Plain C



10 Ways to Boost COBOL Application Development

Visual COBOL New Release: Small point. Big deal



SAPrefs - Netscape-like Preferences Dialog

Comments and Discussions

Add a Comment or Question ?

Search Comments Go

Spacing Relaxed ▼ Layout Open All ▼ Per page 25 ▼ Update

First Prev Next

My vote of 5

Yonko Kondarev 8-Mar-11 2:34

This Article resolved my problems

Reply · Email · View Thread · Permalink · Bookmark

Alternative version?

Mark F. 8-Jan-06 9:19

This is what I have been trying to do in one of my apps. Have you a demo that will work in VS 6? It is a real pain to even checkout examples that have been created with dot net.

Thanks!

Mark

Reply · Email · View Thread · Permalink · Bookmark 1.80/5 (5 votes)

Last Visit: 18-Dec-16 17:03 Last Update: 20-Dec-16 6:43 Refresh 1

General News Suggestion Question Bug Answer Joke Praise Rant Admin