# TabDlg Version 2.0

A DLL for creating Microsoft Word 6.0 style tabs in a dialog box.

## Introduction

TabDlg will allow you to easily create tab-style option dialog boxes very similar to those found in Microsoft Word 6.0 and Visual C++ 1.5. TabDlg is distributed as freeware. This means that I retain ownership of the code but place only a few restrictions upon its use and redistribution. These are:

- If you are not distributing tabdlg2.dll with an application, please use the original archive file with the original contents and copyright notices.
- If you are using tabdlg2.dll with an application, you may use it free of charge, but I ask that you include this copyright line somewhere in the documentation:

  TabDlg.DLL is Copyright (c) 1994 Edward McCreary

Other than that I place no restrictions upon its use. I would like an email note if you use it in an application just so I can track how much its being used. This will act as an incentive for me to keep writing these little snippets of code.

Eddie McCreary
edm@twisto.compaq.com

## Disclaimer

Redistribution and use in source and binary forms are freely permitted provided that the above copyright notice and attribution and date of work and this paragraph are duplicated in all such forms. THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILILTY AND FITNESS FOR A PARTICULAR PURPOSE.

## Usage

Heres step-by-step instructions for using the tabdlg2.dll in an application.

1. Install tabdlg2.dll into the Windows system directory. Only install if the version information in the new dll is newer than the one in the system directory!

2. Copy tabldg2.lib and tabdlg.h into your source directory and include tabdlg2.lib in your project.

3. Design the master dialog. This is the template used with CreateDialog or however you create your dialog boxes. In it, place controls that are common across all tabs and a black frame where you want the boundaries of the tab to occur. Remember to leave space above the top edge of the frame for the tabs themselves.

4. Design a dialog box for each tab. The size of the template really doesnt matter but it should be as large as the master dialog box. If you want, you can place a black frame in the new template to act as a reference, but it must have the same resource id as the black frame in the master dialog box. Now, place the controls for that tab into the dialog box. Dont forget to give each control a unique resource id, even across tabs.

5. Setup an array of ids for the tabs. This is an array of TAGTEMPLATE structures which provide the dll with the resource ids of the templates and titles for the tabs. For example:

```
static TAGTEMPLATE temps[] = {
{ MAKEINTRESOURCE(IDD_TAB_ONE),"Tab One" },
{ MAKEINTRESOURCE(IDD_TAB_TWO), "Tab Two" }};
```

You can substitute MAKEINTRESOURCE with strings if thats how you identify your dialog templates.

6. Call BuildTabs in response to WM_INITDIALOG. Remember to call it after any function which may try to subclass the frame such as Ctl3dSubclassDlg. For example:

```
BuildTabs(hDlg,hInst, IDC_TABFRAME,(TAGTEMPLATE FAR *)temps,
   sizeof(temps)/sizeof(TAGTEMPLATE));
```

Thats all there is!

## Miscellaneous Stuff

Right now there is a problem using tabdlg2.dll with the all3d utility. All3d subclasses the frame after tabdlg2.dll does, so the tabs are not drawn properly. If you use ctl3d.dll with your application you wont have this problem.

I suggest using ctl3d.dll with your application if you can. I designed the tabs so that they would appear integrated with the 3-D controls. Remember to use AutoSubclassing if you do use ctl3d so that the controls in the tab get subclassed properly.

If you dont use ctl3d.dll, youll need fill the background of the dialog box with light gray to get the proper look. This is easily done by responding to WM_CTLCOLOR. See the C example on how to do this.

## Example Code

Three example programs are provided to demonstrate how to use tabdlg2.dll with your application. I wrote two small applications, one in C and one in C++ using MFC. Along with the MFC code is a small C++ class I wrote to make using ctl3d.dll easier. To use it, design your dialog boxes normal and create a class for it. Then, replace all of the references to CDialog with C3DDlg. Youll need to add these function calls to the InitInstance method of your CWinApp object:

```
Ctl3dRegister(AfxGetInstanceHandle());
Ctl3dAutoSubclass(AfxGetInstanceHandle());
```

and this function to your ExitInstance function:

```
Ctl3dUnregister(AfxGetInstanceHandle());
```

See the MFC example for details.

Thanks to Yiannis Antoniou (yianan@theseas.ntua.gr) for providing the Pascal example code.

## Function Reference

***BOOL WINAPI BuildTabs(HWND hDlg,HINSTANCE hInst, UINT wFrameID, const TAGTEMPLATE FAR *lpTemplates, int nNumTemplates);***

Call this in response to WM_INITDIALOG.

Parameters:

| | |
|---|---|
| hDlg | Handle to the master dialog box. |
| hInst | Instance handle of module where tab templates are stored. |
| wFrameID | Resource ID of the frame. |
| lpTemplates | Point to an array of TAGTEMPLATE structures which define the tabs. |
| nNumTemplates. | Number of tabs. |

### *BOOL WINAPI   SetTopTab(HWND hDlg, int nTabID);*

Set nTabID to be the top tab.   By default the first tab in the array is the initial tab.

Parameters:

hDlg Handle to the master dialog box.

nTabID zero based index into the array of TAGTEMPLATE structures used in *BuildTabs*.

Note: You must force a repaint after this to bring the new tab to the top if not being called in response to WM_INITDIALOG.   To do this, just call:

```
InvalidateRect(hDlg,NULL,TRUE);
UpdateWindow(hDlg);
```

### *void WINAPI   SetTabStyle(HWND hDlg, DWORD dwStyle);*

Set different tab styles.   Right now the only valid parameter is TAB_SHADOWTEXT which adds a shadow to the tab titles.   By default this is not turned on.

Parameters:

hDlg Handle to the master dialog box.

dwStyle Styles to set.

Note: You must force a repaint after you call this if its not being called in response to WM_INITDIALOG.