Dialog Bar Component: Overview

A dialog bar is a toolbar based on a dialog template. The Dialog Bar component adds a dialog bar to a frame window in your project. Your project must have at least one **CFrameWnd**-derived class to use this component.

Dialog Bar Component: Specifics

Dialog Bar Component: Results

Dialog Bar Component: Specifics

The dialog bar is implemented by MFC's **CDialogBar** class. The Dialog Bar component declares a member variable of type **CDialogBar** in your frame window class. The dialog bar is initialized during the creation of the frame window.

If you add a dialog bar to the main frame (usually named `CMainFrame`) of an ActiveX application, the dialog bar will not automatically appear when the application is used for in-place editing of an embedded object. If you want a dialog bar to appear during in-place editing, use the Dialog Bar component again and associate the new dialog bar with the application's in-place frame (usually named `CInPlaceFrame`).

For more information about dialog bars, see the Dialog Bars section in the Overview section of *Adding User Interface Features* in the *Visual C++ Programmer's Guide*.

Dialog Bar Component: Results

The Dialog Bar component makes the following changes to your project code:

- Adds a dialog template resource to your project's resource script. You can use the Visual C++ dialog editor to add controls to this dialog template.
- Adds a member variable of type **CDialogBar** to the declaration of the specified frame window class. You can change the default variable name (**m_wndMyDialogBar**) if you want; you can also specify initial properties of the Dialog Bar, such as whether it is visible by default, dockable, and so forth.
- Adds initialization code to the frame window's **OnCreate** member function. This code calls **CDialogBar::Create** and sets up the dialog bar's docking options. It also hides the dialog bar if you choose not to make it visible by default.
- Adds suppors for a menu item to display the dialog bar.

  Note that the component does not actually add a menu item to your menu resources for displaying the dialog bar. You must add the menu item yourself, typically to your View menu, using the Visual C++ menu editor. The component adds "TODO" comments that describe the steps you should perform to add the menu item. These comments can be found in the `OnCreate` function of the frame class where the dialog bar was added.
- Adds handlers for two functions to the message map of the frame window class:
  - The `OnUpdateXXX` function updates the state of the menu item's checkmark, as appropriate.
  - The `OnBarCheck` function then checks that the dialog bar is displayed or hidden, to conform to the state of the menu item's checkmark.
- Adds a prompt string for the menu item to the project's string table resource.

You can also use ClassWizard to implement command handlers for the controls on your dialog bar. Use the following procedure:

1. In ResourceView, open the dialog template that corresponds to the dialog bar.
2. Invoke ClassWizard (CTRL+W). ClassWizard displays the **Adding a Class** dialog.
3. Choose **Select an existing class** and click the **OK** button. ClassWizard asks you to select a class.
4. Choose the name of the frame window class that you specified when creating the dialog bar and click the **Select** button. ClassWizard warns you that the class is not a dialog class and asks whether you want to link the class to the dialog resource anyway.
5. Click the **Yes** button. The identifiers of the controls on your dialog bar appear in the Object IDs list on ClassWizard's Message Maps page.
6. Assign command handlers to the controls as usual.

Note If you add ActiveX controls to a dialog bar, you need to enable ActiveX control containment for your project. If you selected the **ActiveX controls** option when the project was originally created in AppWizard, then ActiveX control containment is already enabled. Otherwise, you should use the ActiveX Control Containment component (in the **Developer Studio Components** folder of the **Gallery**) to add the needed support to your project.

Dialog Bar Component

Dialog bar name

Provide any desired text string in this field. This string will be used in menu items and prompt strings. It will also be used for generating various identifiers that will be added to your code.

Member variable name

Provide a valid C++ identifier in this field. This identifier will be used as the name of the member variable that is added to your frame window class.

The identifier will be automatically generated, based on the string provided in the previous field, but you can modify it, if necessary.

Frame to contain dialog bar

If your project has more than one class derived from **CFrameWnd**, this field allows you to select which frame will contain the dialog bar. If your project has only one **CFrameWnd**-derived class, this field is disabled.

Default docking edge

Choose the edge of the frame window to which the dialog bar will be docked by default. If you choose **Top** or **Bottom**, the initial dialog bar template will have a horizontal orientation. If you choose **Left** or **Right**, it will have a vertical orientation.

Visible by default

This field indicates whether the default state of the dialog bar is visible or hidden.

Dockable

If you check this field, the end user of your application will be able to move or undock the dialog bar. Moving and undocking is accomplished by dragging and dropping the dialog bar with the mouse.