Lab 5.4: Adding a Shortcut Menu

Objectives

After completing this lab, you will be able to:

- Add a generic shortcut menu handler from the Component Gallery.
- Modify the menu resources associated with a shortcut menu.
- Process right-click messages to trigger a shortcut menu.
- Process messages sent from a shortcut menu.

Prerequisites

Familiarity with the topics covered in this chapter.

Lab Setup

To run the solution to this lab, click this icon.

To see a demonstration of the solution to this lab, click this icon.

Estimated time to complete this lab: **30 minutes**.

Exercises

The following exercise provides practice working with the concepts and techniques covered in this chapter.

Exercise 1: Adding a Shortcut Menu Component

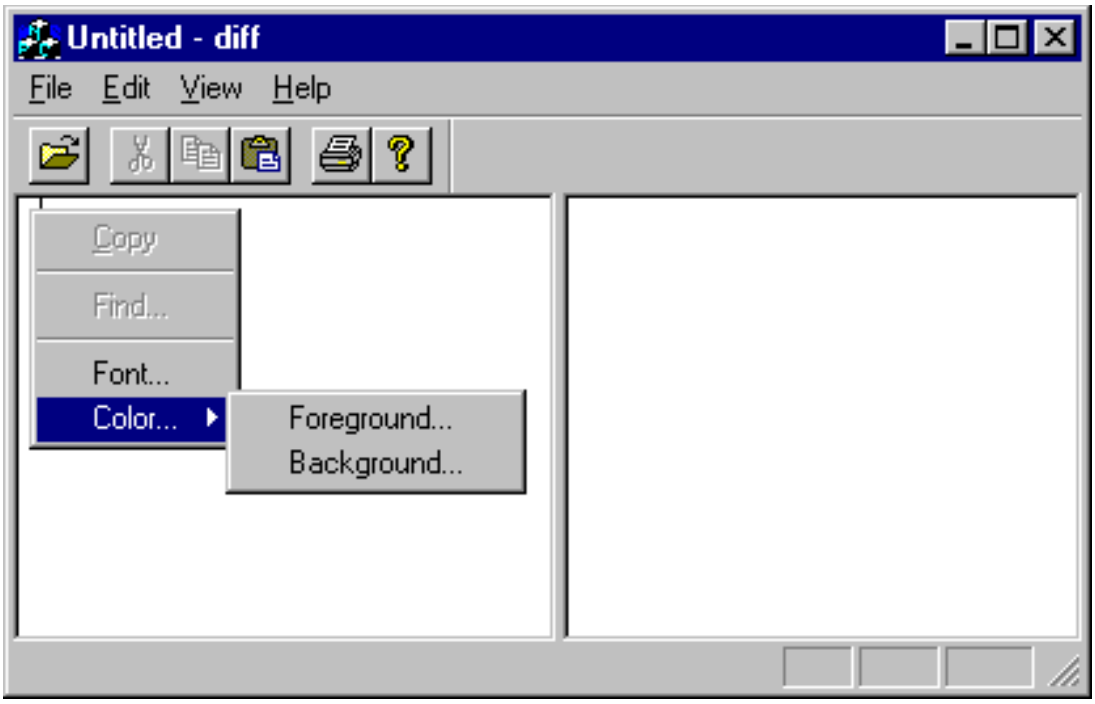
In this exercise, you will add a shortcut menu to an application using the Component Gallery.

Copy the starting point for this project from \Labs\C05\Lab04\Baseline. The completed code for these exercises is in \Labs\C05\Lab04\Xxx, where Xxx is the exercise number.

Exercise 1: Adding a Shortcut Menu Component

If you do not have a starting point for this exercise, the code that forms the basis for this exercise is in \Labs\C05\Lab04\Baseline.

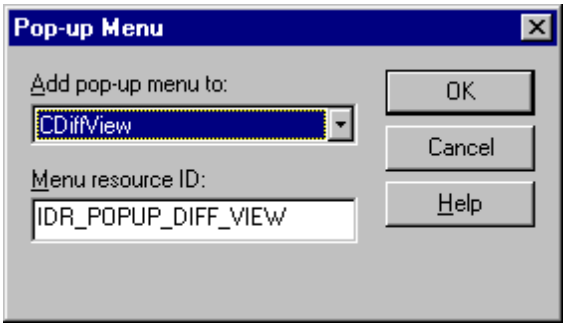
In this exercise, you will add a shortcut menu and its associated test handlers to an application. When a user clicks the right mouse button in either pane of the ShowDiff application, it displays a menu at the location of the mouse click. Usually, displaying a menu in this manner is a very simple task. However, since ShowDiff uses **CRichEditView** based on the Rich Edit Control, the **DefWindowProc** will not translate **WM_RBUTTONUP** to **WM_CONTEXTMENU** automatically; you must do that translation yourself.



➢ **Insert the Popup Menu Component into CDiffView**

1. From the Project menu, choose Add to Project, and then choose Components and Controls. In the Gallery dialog box, choose Developer Studio Components.

2. Select the Popup Menu component, choose Insert and click OK. You will then be asked whether you want to create a Popup Menu or a Wizard. Choose Popup Menu and click Next.
3. The Popup Menu dialog box will be displayed. Add the popup menu to the **CDiffView** class. Leave the menu resource ID at its default CG_IDR_POPUP_DIFF_VIEW, and click OK. This screen shot shows the completed Popup Menu dialog box.



4. The Component Gallery will insert an **OnContextMenu** handler into the implementation of **CDiffView**, DiffView.Cpp, as shown in this code:

```
void CDiffView::OnContextMenu(CWnd*, CPoint point)
{
    // CG: This block was added by the Popup Menu component
    {
```

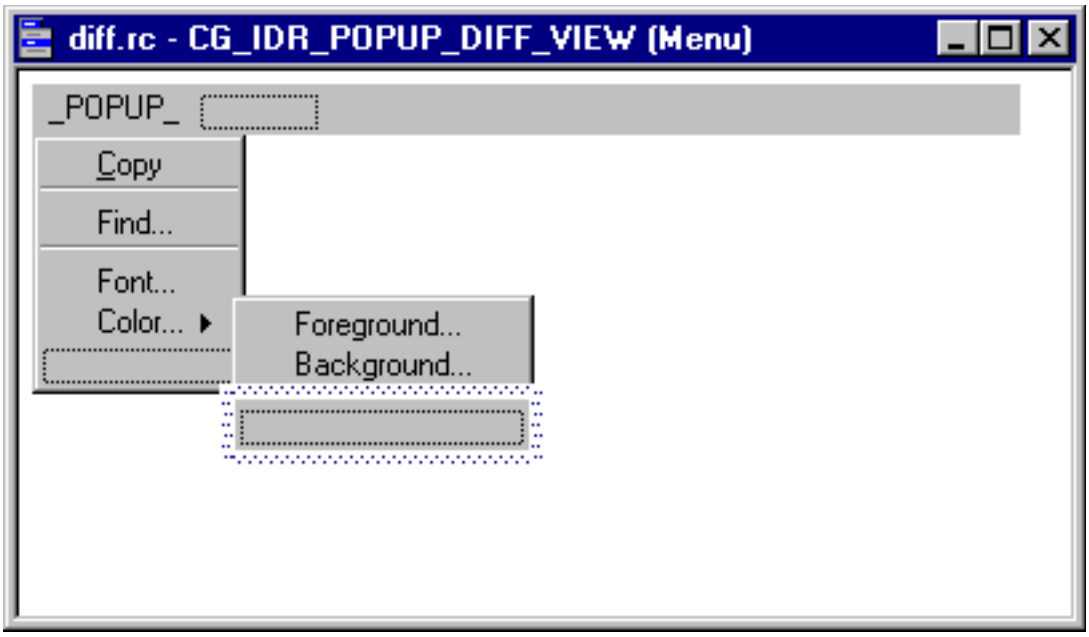
```
        if (point.x == -1 && point.y == -1){
            //keystroke invocation
            CRect rect;
            GetClientRect(rect);
            ClientToScreen(rect);

            point = rect.TopLeft();
            point.Offset(5, 5);
        }

        CMenu menu;
        VERIFY(menu.LoadMenu(CG_IDR_POPUP_DIFF_VIEW));

        CMenu* pPopup = menu.GetSubMenu(0);
        ASSERT(pPopup != NULL);
        CWnd* pWndPopupOwner = this;

        while (pWndPopupOwner->GetStyle() & WS_CHILD)
            pWndPopupOwner = pWndPopupOwner->GetParent();

        pPopup->TrackPopupMenu(TPM_LEFTALIGN | TPM_RIGHTBUTTON, point.x,
point.y,
            pWndPopupOwner);
    }
}
```

➢ **Change the Default Menu to Support ShowDiff**

1. The Component Gallery will also add a menu resource to the project. Open the Menu Editor to edit CG_IDR_POPUP_DIFF_VIEW.
2. Delete all the items in the menu except for Copy.
3. Add the menu items shown in this table. The items and their IDs are included.

Menu Item	ID
Copy	ID_EDIT_COPY
Separator	
Find...	ID_EDIT_FIND
Separator	
Font...	ID_EDIT_FONT
Color	(popup)
Foreground	IDC_EDIT_COLOR_FOREGROUND
Deletions...	IDC_EDIT_COLOR_BACKGROUND

Your completed menu will be displayed in the menu editor, as shown in this illustration.



4. Save Diff.Rc.

➢ **Add a Handler for WM_RBUTTONUP**

1. Open ClassWizard from the View menu, or press CTRL+W.
2. Choose the **CDiffView** class and **CDiffView** object ID.
3. Select the WM_RBUTTONUP message and click the Add Function button.
4. Click the Edit Code button.
5. Insert this code into DiffView.Cpp:

```
void CDiffView::OnRButtonUp(UINT nFlags, CPoint point)
{
    CRichEditView::OnRButtonUp(nFlags, point);
}
```

6. You will not use the default WM_RBUTTONUP handler in **CRichEditView**. Comment it out.
7. **CView::OnRButtonup** is passed a **CPoint** relative to the client window. If you explore **OnContextMenu**, you will discover that **CMenu::TrackPopupMenu** requires a **CPoint** relative to the screen. Use **CWnd::ClientToScreen** to do this conversion, as shown in this code:

```
ClientToScreen (&point);
```

8. Call **CDiffView::OnContextMenu** with this point.

```
OnContextMenu ((CWnd *)NULL, point);
```

9. Save DiffView.Cpp. The complete function follows.

```
void CDiffView::OnRButtonUp(UINT nFlags, CPoint point)
{
    //CRichEditView::OnRButtonUp(nFlags, point);
    ClientToScreen (&point);
    OnContextMenu ((CWnd *)NULL, point);
}
```

➢ **Add Dummy Handlers for the Menu Messages**

1. Open ClassWizard from the View menu or press CTRL+W.
2. Choose the **CDiffView** class.

3. Choose the menu object IDs, adding a function for each and accepting their default function names, as detailed in this table.

Object ID	Function Name
ID_EDIT_FONT	OnEditFont
IDC_EDIT_COLOR_FOREGROUND	OnEditColorForeground
IDC_EDIT_COLOR_BACKGROUND	OnEditColorBackground

4. Choose one of these functions and press the Edit Code button to navigate to the handlers.
5. In each of the function bodies, add a message box to show that the handler has been called, as shown in this code:

```
void CDiffView::OnEditFont()
{
    AfxMessageBox ("In OnEditFont()");

}


void CDiffView::OnEditColorForeground()
{
    AfxMessageBox ("In OnEditColorForeground()");

}

void CDiffView::OnEditColorBackground()
{
    AfxMessageBox ("In OnEditColorBackground()");

}
```

6. Save DiffView.Cpp. Build ShowDiff and run it.

➢ **Compile and Run the ShowDiff Application**

The completed code for this exercise is in \Labs\C05\Lab04\Ex01.