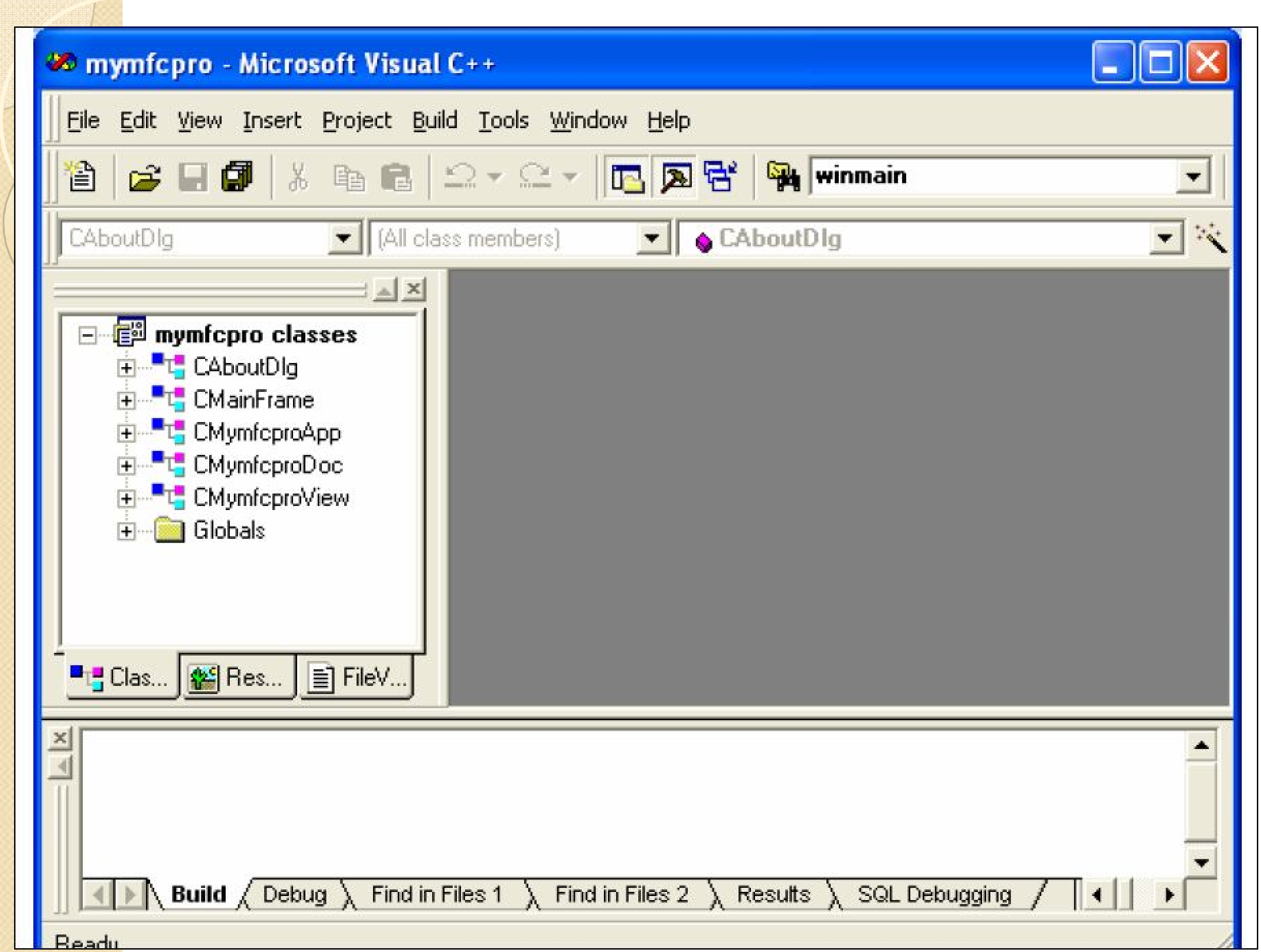
Rapid Application Development Menus


mymfcpro - Microsoft Visual C++
File
Edit
View
Insert
Project
Build
Tools
Window
Help
winmain
CAboutDlg
(All class members)
CAboutDlg
mymfcpro classes
CAboutDlg
CMainFrame
CMymfcproApp
CMymfcproDoc
CMymfcproView
Globals
Clas...
Res...
FileV...
Build
Debug
Find in Files 1
Find in Files 2
Results
SQL Debugging
Ready

Menus

- The popdown windows that displays a set of options or menu items that the user can select from .we will learn
- How to place menu items into menus
- How to create whole new menus .
- How to graying out or checking out menus using shortcut keys, acceralator keys and status bar prompts.
- How to create and use submenus
- How to place a new button in the toolbar.

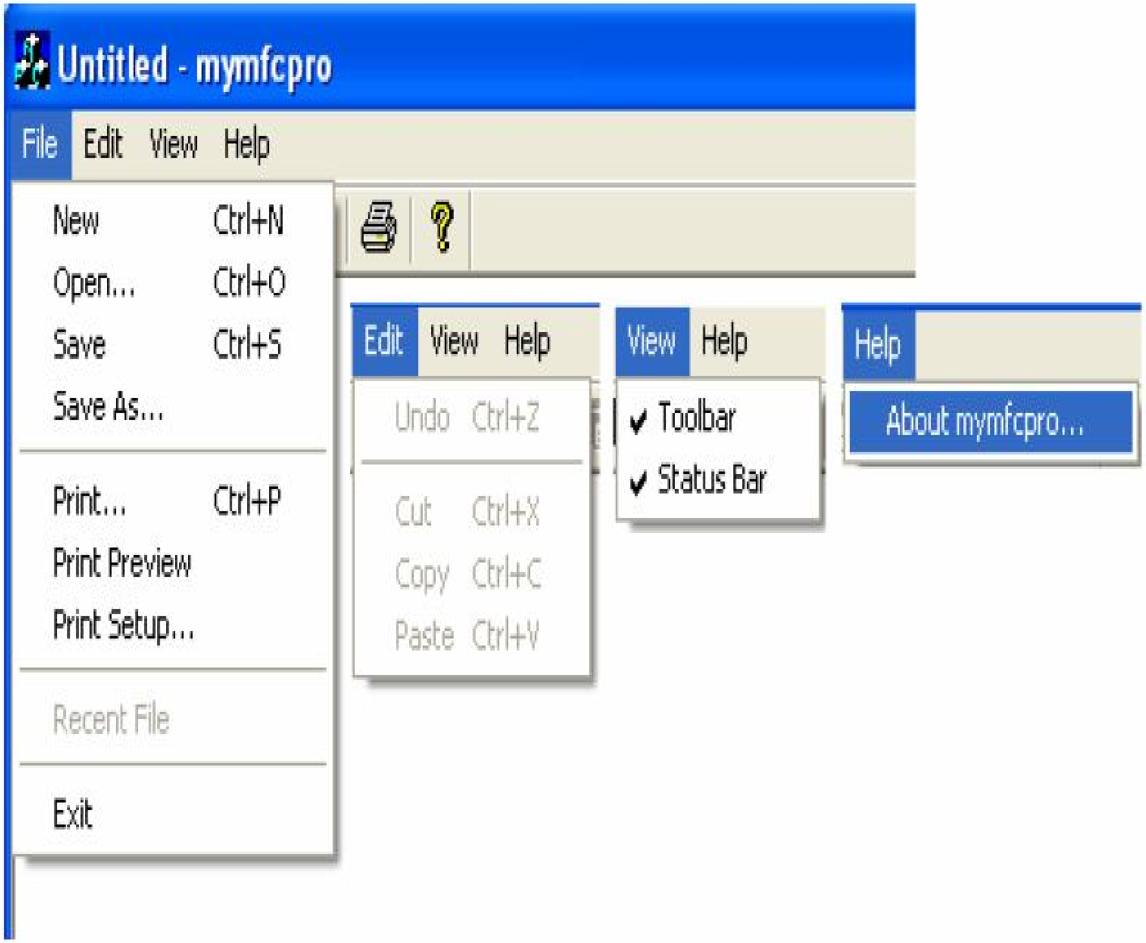
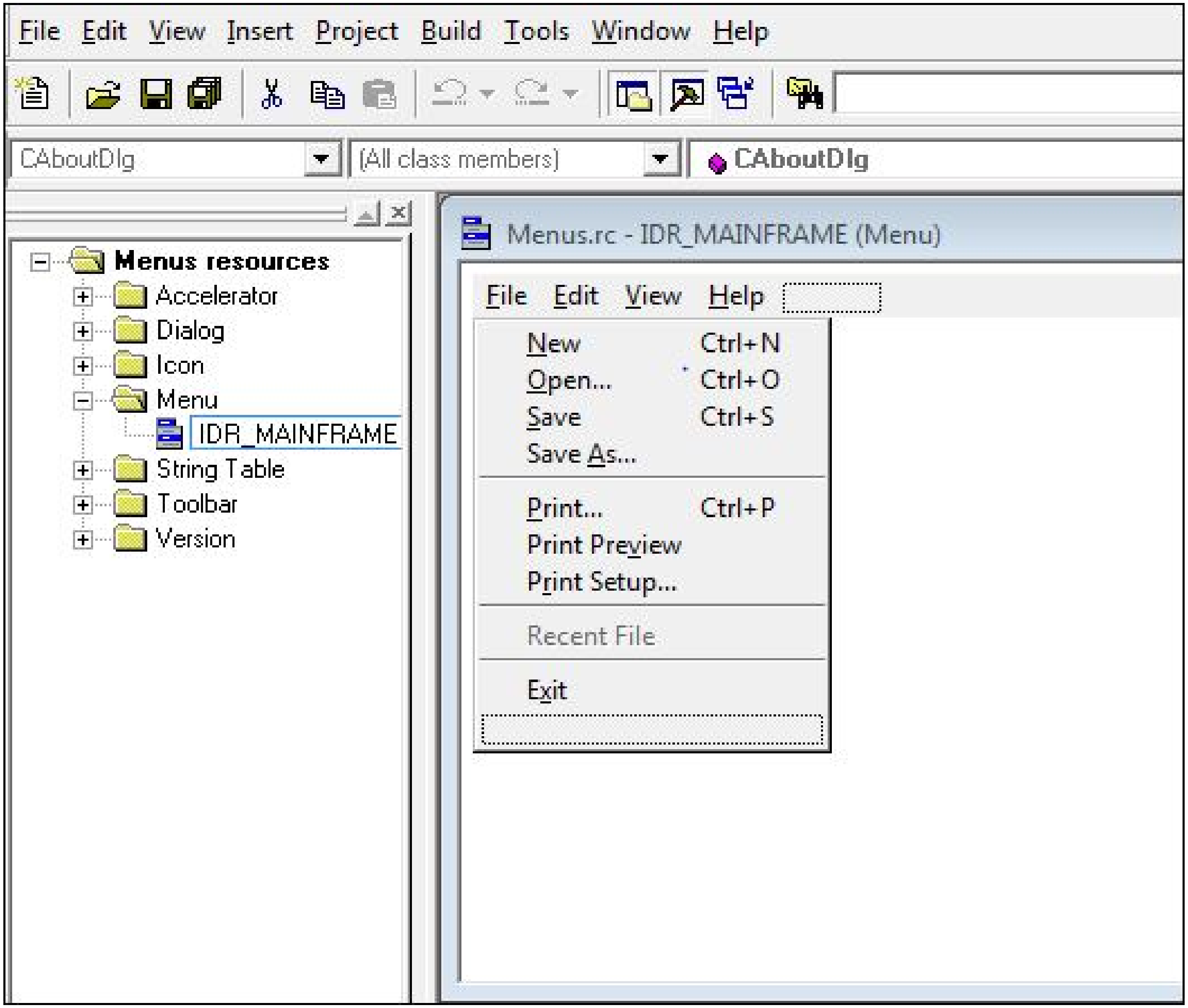
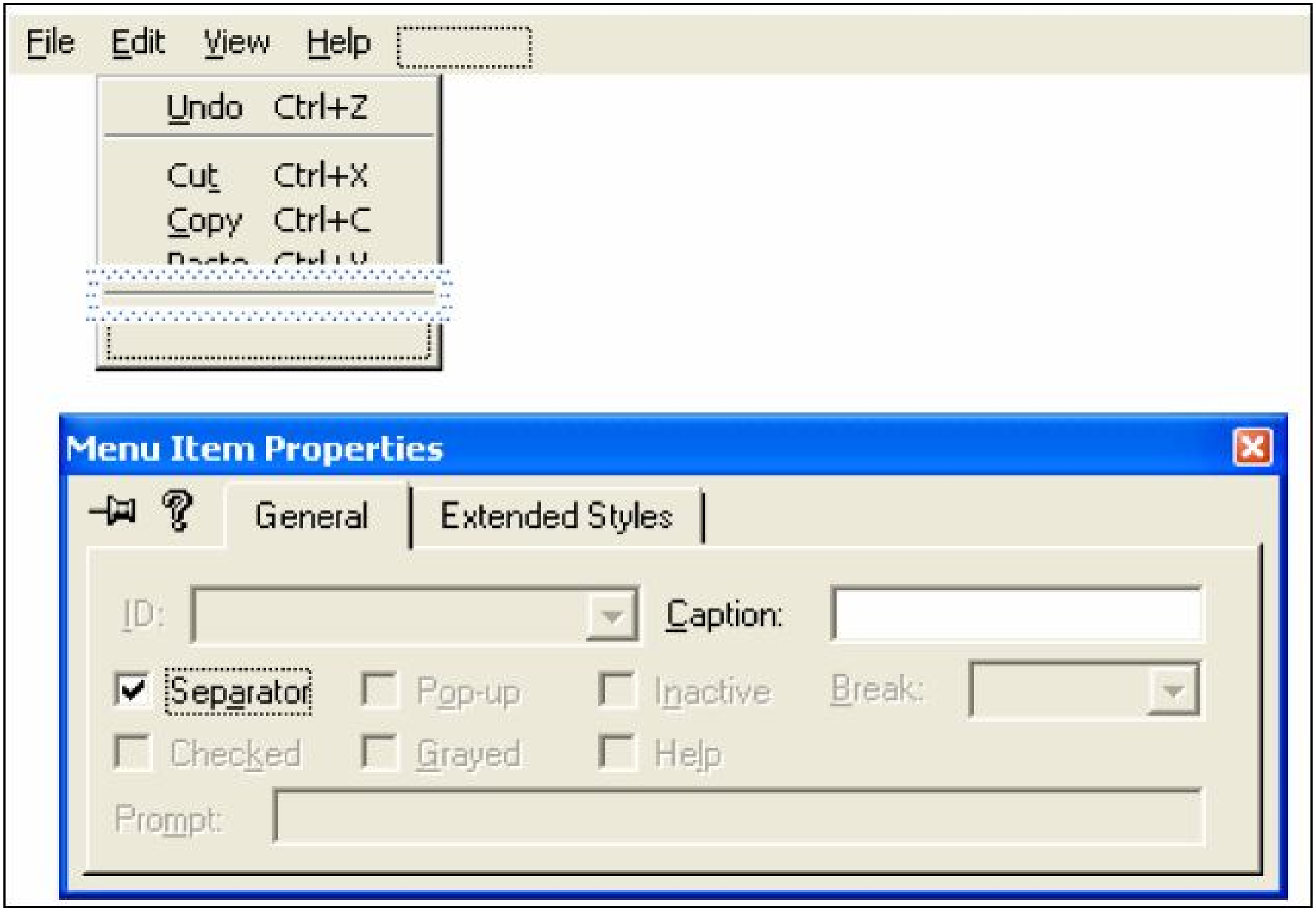


Figure 4: The standard SDI frame menus.


File Edit View Insert Project Build Tools Window Help
CAboutDlg
(All class members)
CAboutDlg
Menus resources
Accelerator
Dialog
Icon
Menu
IDR_MAINFRAME
String Table
Toolbar
Version
Menus.rc - IDR_MAINFRAME (Menu)
File Edit View Help
New Ctrl+N
Open... Ctrl+O
Save Ctrl+S
Save As...
Print... Ctrl+P
Print Preview
Print Setup...
Recent File
Exit

- **Adding a Menu Item to a Program’s File Menu.**
- When the user select the new menu item suppose “Print Welcome “ option it displays a msg on screen “welcome to menus”
- Create this program using AppWizard making it an SDI program.
- Give it the names “menus” in the visual c++ project name.
- Click the resource tab in the visual c++ viewer window.
- Find the folder marked Menus and open it .
- Double click on that folder **,IDR_MAINFRAME** , opening the menu editor.
- Click on the File menu ,this new item will go between the Print welcome and print setup menu item .
- Click on Print Setup Menu item and press insert key.
- A new menu box appears which is surrounded by a multi dotted lines.



ID:

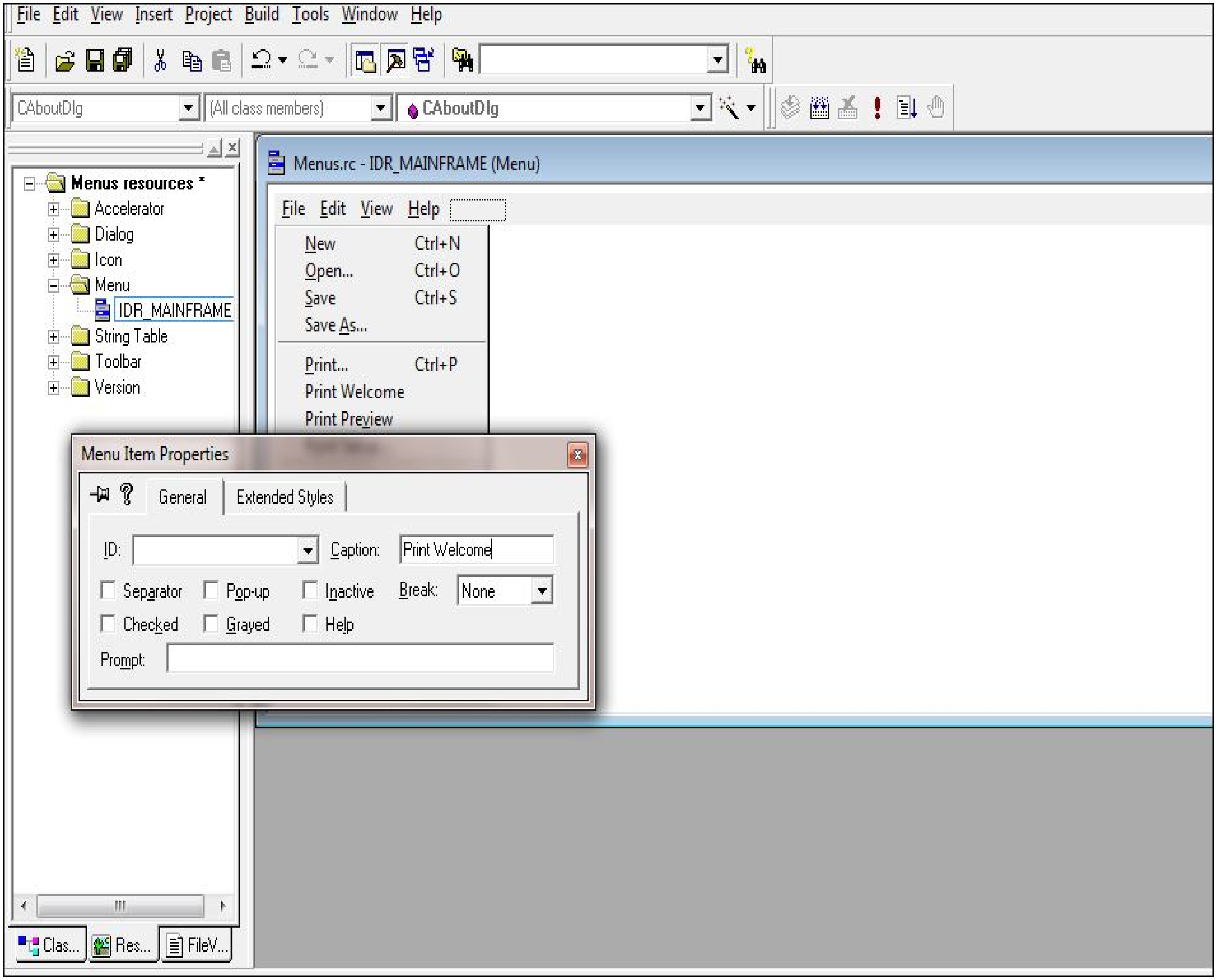
Caption:

☑ Separator □ Pop-up □ Inactive Break:

□ Checked □ Grayed □ Help

Prompt:

- **Adding properties to new menu named "Print Welcome "**
- Double click on new item box which opens the menu item properties box.
- Place the caption Print Welcome in the Caption box of that dialog box.
- Place the ID as ID_FILE_PRINTWELCOME to our new menu item and close the box.



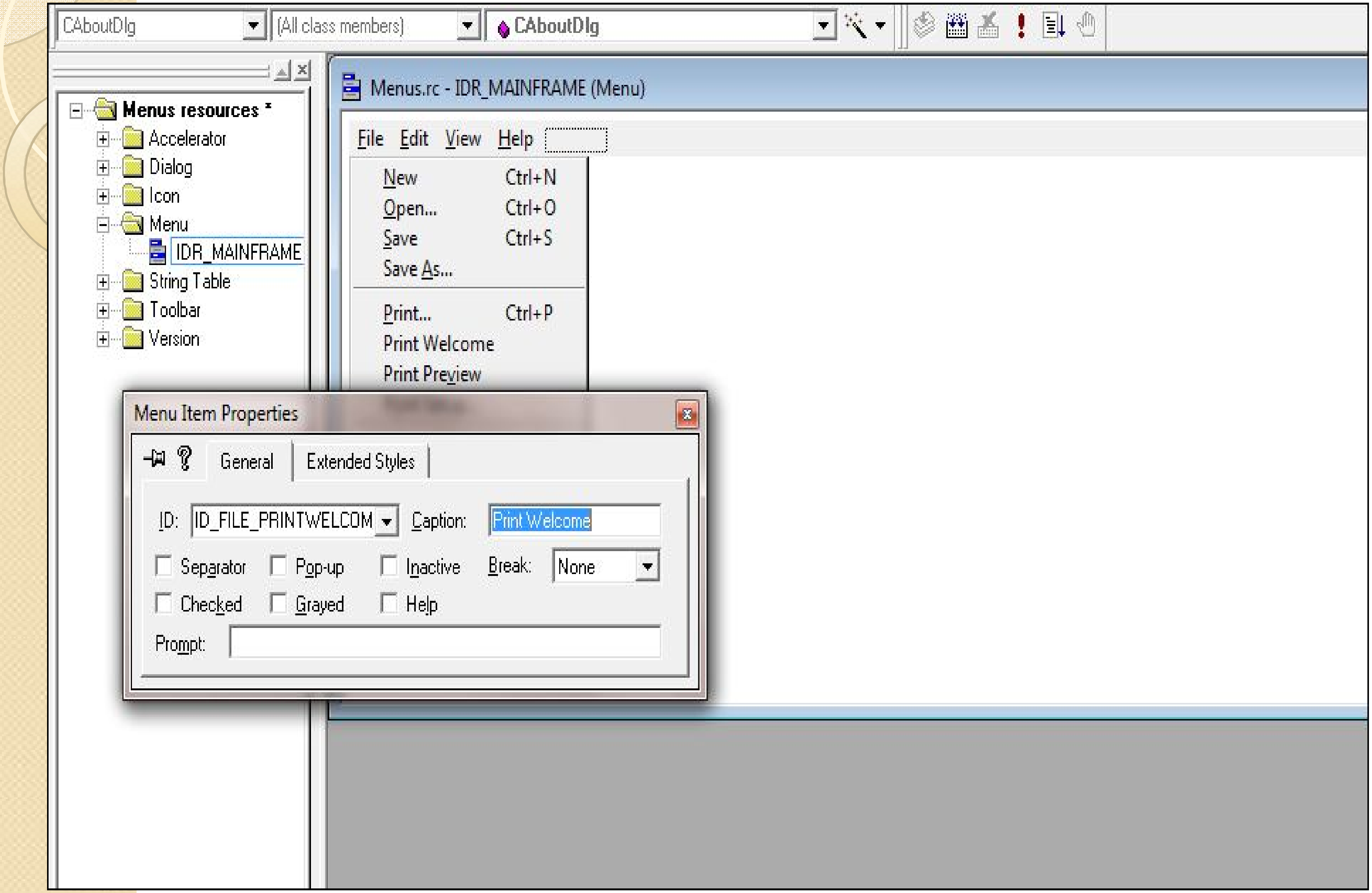
ID:

Caption: Print Welcome

□ Separator □ Pop-up □ Inactive Break: None

□ Checked □ Grayed □ Help

Prompt:



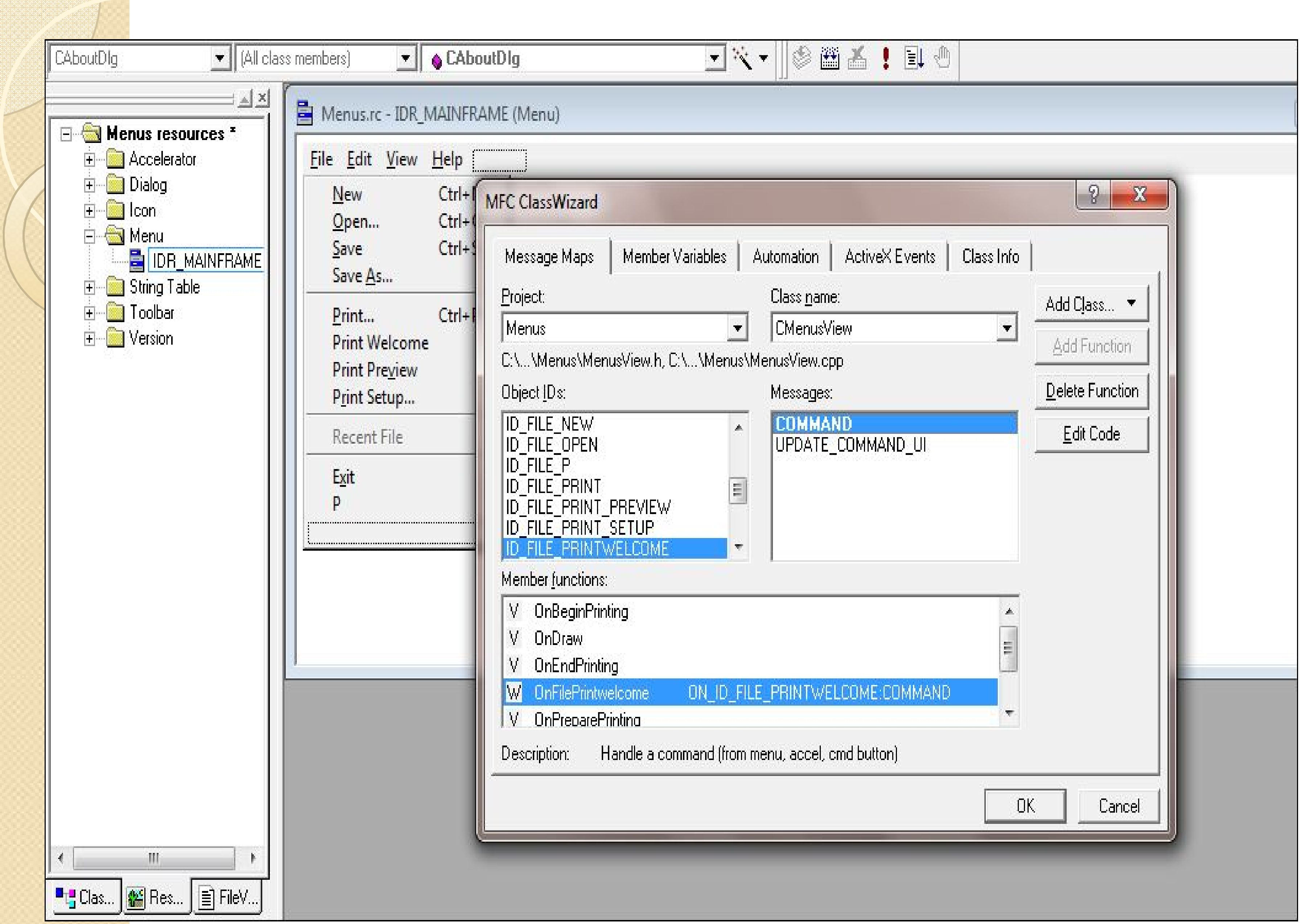
ID: ID_FILE_PRINTWELCOM

Caption: Print Welcome

□ Separator □ Pop-up □ Inactive Break: None

□ Checked □ Grayed □ Help

Prompt:



Project: Menus

Class name: CMenusView

C:\...\Menus\MenusView.h, C:\...\Menus\MenusView.cpp

Object IDs:

```
ID_FILE_NEW
ID_FILE_OPEN
ID_FILE_P
ID_FILE_PRINT
ID_FILE_PRINT_PREVIEW
ID_FILE_PRINT_SETUP
ID_FILE_PRINTWELCOME
```

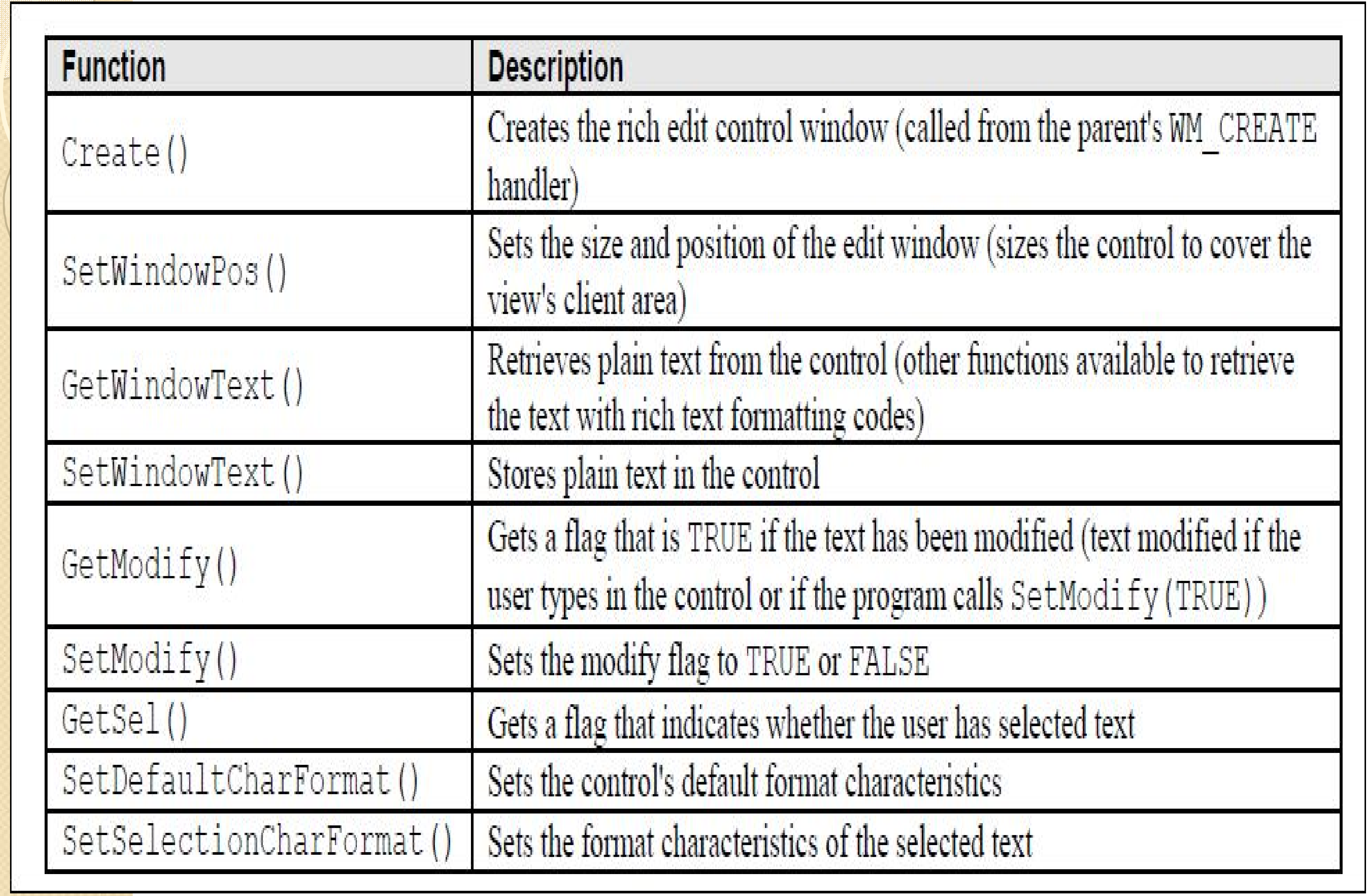
Messages:

```
COMMAND
UPDATE_COMMAND_UI
```

Member functions:

```
V  OnBeginPrinting
V  OnDraw
V  OnEndPrinting
W  OnFilePrintwelcome     ON_ID_FILE_PRINTWELCOME:COMMAND
V  OnPreparePrinting
```

Description: Handle a command (from menu, accel, cmd button)



Function	Description
`Create()`	Creates the rich edit control window (called from the parent's `WM_CREATE` handler)
`SetWindowPos()`	Sets the size and position of the edit window (sizes the control to cover the view's client area)
`GetWindowText()`	Retrieves plain text from the control (other functions available to retrieve the text with rich text formatting codes)
`SetWindowText()`	Stores plain text in the control
`GetModify()`	Gets a flag that is `TRUE` if the text has been modified (text modified if the user types in the control or if the program calls `SetModify(TRUE)`)
`SetModify()`	Sets the modify flag to `TRUE` or `FALSE`
`GetSel()`	Gets a flag that indicates whether the user has selected text
`SetDefaultCharFormat()`	Sets the control's default format characteristics
`SetSelectionCharFormat()`	Sets the format characteristics of the selected text

MenusDoc.h

```
class CMenusDoc : public CDocument
{
protected: // create from serialization only
CMenusDoc();
DECLARE_DYNCREATE(CMenusDoc)
public:
virtual ~CMenusDoc();
CString StringData;
}
```

MenusDoc.cpp

```
CMenusDoc::CMenusDoc()
{
// TODO: add one-time construction code here
StringData="";
}
```

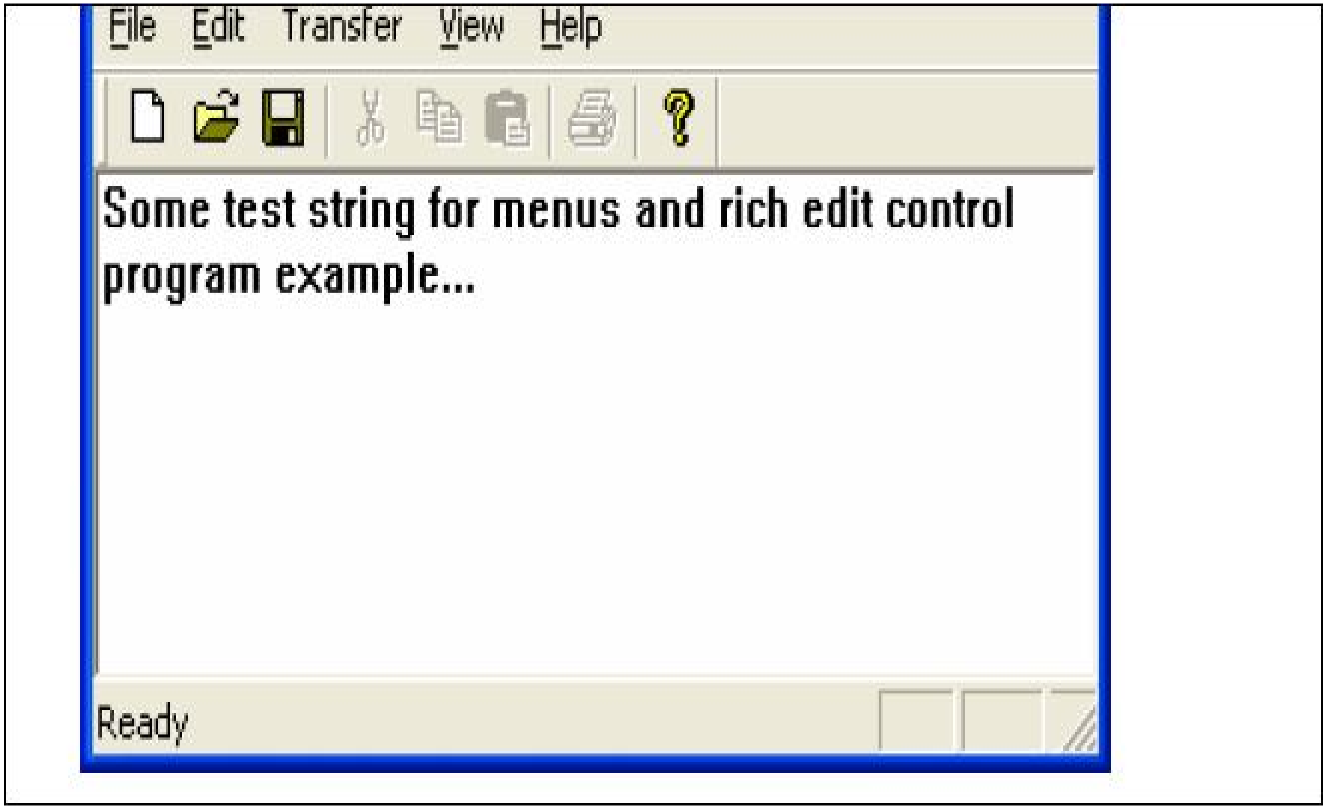
MenusView.cpp

```
void CMenusView::OnFilePrintwelcome()
{
  // TODO:Add your command handler code here
   CMenusDoc*pDoc=GetDocument();
   ASSERT_VALID(pDoc);
   pDoc->StringData = "welcome to new menus ";
  Invalidate();
}
```

MenusView.cpp

```
void CMenusView::OnDraw(CDC* pDC)
{
  CMenusDoc* pDoc = GetDocument();
  ASSERT_VALID(pDoc);
  pDC->TextOut(0,0,pDoc->StringData);
  // TODO: add draw code for native data here
}
```

Output Screen



Some test string for menus and rich edit control program example...

Ready