

## **TextEdit Sample Help**

**Sample Description:** [TextEdit](#)

### **Points of Interest**

- [Construction](#)
- [Selecting Text](#)
- [Cut, Copy, Paste](#)
- [Word Wrap](#)
- [Scroll Bars](#)
- [Read/Write](#)
- [Printing](#)
- [Text Search](#)
- [Setting Fonts](#)

### **Tips & Techniques**

- [Checkmarks in Menus](#)
- [Enabling Menu Commands](#)
- [Yes/No Prompting](#)
- [Linking to an Embedded Object](#)
- [Creating SubMenus](#)
- [Interfacing to XferData Object](#)
- [Resizing the TextEdit Form](#)

### **Command Reference**

- [File Menu](#)
- [Edit Menu](#)
- [Search Menu](#)
- [Options Menu](#)

### **How To...**

- [Formatting Text](#)
- [Choosing a Font and Font Size](#)
- [Moving Around in the TextEditor](#)
- [Copying, Cutting and Pasting Text](#)
- [Selecting Text](#)
- [Finding Text](#)
- [Deleting Text](#)
- [Opening and Saving TextEdit Files](#)
- [Printing TextEdit Files](#)

### **Control**

- TextBox

For Help on Help, Press F1



**Sample Description**

The text editor developed in this example is similar to the Notepad utility that comes with Windows and demonstrates the concepts behind text box control.

## Construction

The text editor application is based on a reusable object "TextEditBox" which is derived from the standard TextBox object. This object contains the following specialized methods:

<b><u>Specialized Methods</u></b>	<b><u>Description</u></b>
Clear	Clears the contents of the TextEditBox.
Delete	Deletes the contents of the TextEditBox.
OpenFile	Posts an Open Dialog box for selecting a text file to edit.
PrintText	Creates a temporary text file and prints it using the shell print utility.
Save	Saves the current text by writing the contents to disk.
SaveAs	Saves the current text to another specified file.
SearchForward	Posts a Find Dialog to search for a text string.
SearchNext	Searches for the next occurrence of the currently specified text string.
SelectAll	Selects the entire text contents of the current textbox.
SetFont	Posts a Font Dialog for setting various font properties.
SetScrollBars	Sets horizontal and vertical scrollbar options as well as corresponding <u>WordWrap</u> properties.

In addition to the specialized methods listed above, the following standard methods are also used in the Text Edit application:

<b><u>Standard Methods</u></b>	<b><u>Description</u></b>
Cut	Cuts or removes the currently selected text and copies it to the Windows Clipboard.
Copy	Copies the currently selected text to the Windows Clipboard.
Paste	Pastes the contents (text only) of the Windows Clipboard that the current insertion point.
Undo	Reverses the effects of the last text editing operation.

## Selecting Text

You can select a block of characters by holding down the left mouse button as you drag the mouse over a block of characters. Then release the button. The SelText property replaces or returns the currently selected text within a textbox. For instance, the specialized Delete method simply sets the SelText property to (nothing) which removed currently selected text but does not copy it to the Windows Clipboard.

**Cut, Copy, Paste**

The Cut, Copy and Paste functions were implemented using standard methods available with the TextBox object. These methods are automatically linked to the XferData object which interfaces directly with the Windows Clipboard.

## **Word Wrap**

The WordWrap functionality is handled by the TextBox's WordWrap property. When set to True, text automatically wraps around to the next line when your typing reaches the right side of the TextBox. In addition, there is a MultiLine property that limits text input to a single line or allows it to be entered on multiple lines. If the MultiLine property is set to False, the WordWrap property is automatically set to False as well.

## Scroll Bars

ScrollBar functionality is provided to TextBoxes through a ScrollBar property. The ScrollBar property supports the following (4) types of ScrollBar behavior:

0 = None	No scrollbars are activated
1 = Horizontal	Only the horizontal scrollbar is activated
2 = Vertical	Only the vertical scrollbar is activated
3 = Both	Both the horizontal and vertical scrollbars are activated.

Scrollbar conditions are set by the Options/Scrollbar menu. In addition to setting the Scrollbar behavior, the menu options automatically configure the WordWrap properties to coincide with the intended functionality.

**Read/Write**

Reading and Writing the contents of the `TextBox` are done through the `File/TextFile` objects `ContentsAsString` and `SetContentsTo` methods. The name of the text file to be opened is obtained from the `OpenDialog`. This file name is used to set the `FileName` property of the `TextFile` object.



## **Printing**

There are several methods for printing text. The option used in this application example was chosen to demonstrate how an application can execute a DOS system command. A Windows API library function CW3215 is used to execute the DOS print command.

## **Text Search**

The text search function is implemented by first posting the Find Dialog to allow the user to enter a search string. If text is currently selected, it is automatically placed in the Find Dialog. As the Find button is clicked, a SearchPanel\_Find method is automatically executed. The actual search is conducted using the Instr function which locates the occurrences of a search string in the TextBox. The SelStart and SelLength properties are then used to highlight the text string that was found.

## Setting Fonts

Fonts are set in the TextBox by a SetFont method. The TextBox object contains an embedded Font object named TextFont. The SetFont method posts a Font Dialog to allow the user to set various Font object properties such as FaceName, Size and Color. Once the Font Dialog OK button is clicked, these properties are used to set the TextFont object. The TextBox has a font "reference" to the TextFont object, so changes to the TextFont objects properties appear immediately in the TextBox.

## **Checkmarks in Menus**

The PopupMenu object contains a method called CheckItem that when set to the value of "1", adds a checkmark to the popup menu item. A value of "0" removes the checkmark.

The PopupMenu object also contains a function called ItemsChecked which returns a True/False depending of if that item is currently checked.

In order to allow the ScrollBar menu to have only one item checked at a time involves using a ClearScrollBarCheckMarks method which first of all, clears all check marks in the menu. Each menu click method should call this clear method, then set the checkmark for its individual menu item.

## **Enabling Menu Commands**

There are times when menu commands are not applicable. Instead of having to write "protection" routines to prevent the user from running commands that will fail, it is much better to disable the commands by setting the Enabled property to "False" until they are able to be executed.

An example of this is the "Print" command. If there is no text in the Text Edit Box, then the print command is automatically disabled. As soon as text is entered, the print command is automatically enabled.

## **Yes/No Prompting**

When ever there is a chance that an end user may accidentally click a command that they do not wish to execute, it is a good idea to implement a prompt to ask the user to confirm their intentions.

An example in the Text Edit Application is the File menu Exit command. Even though a separator has been used to help separate this command from the other commands, a YesNoBox Dialog has been implemented to prompt the user if they really want to exit the application. Clicking the Yes button will allow the exit to continue while the No button will abort the exit operation.

### **Linking to an Embedded Object**

Developing reusable objects involves creating specialized objects by adding new properties, methods and events which gives them specialized functionality. An example of this is the Clear method in the TextEditBox. Clear is normally a standard method for TextBox, however in this application, an additional function was needed.

To add this function, the standard Clear method was enhanced to also clear the FileName property of the OpenFileDialog. This method is called from the parent form's New popup menu command (i.e., txtEditBox.Clear). The object txtEditBox is an instance of an embedded TextEditBox object. The Clear method, when invoked on the parent form, actually runs the enhanced Clear method on the TextEditBox object.

## **Creating SubMenus**

Creating a submenu is a matter of calling out the name of the submenu in the Name area of the Menu Editor. In the Text Edit Application, the name of the Options menu is `TextEditOptionsMenu`. This menu is defined as an entry in the `TextEditMenuBar`.

Within the `TextEditOptionsMenu`, there are two entries: `OptionsFonts` and `OptionsScrollBarMenu`. The `OptionsFont` entry calls a corresponding `OptionsFont_Click` method on the parent form. The `OptionsScrollBarMenu` entry calls another menu (i.e., submenu).



## **Interfacing to XferData Object**

The interfacing between the Windows Clipboard and Envelop applications is done through the XferData object. One of the properties in the XferData object is Text. This property displays the text currently stored in the Windows Clipboard.

The TextBox object's Cut, Copy, and Paste methods are automatically linked to the XferData object. This allows you to copy text from one Windows application and paste it into an Envelop TextBox. Should your application need to review and possibly modify the contents of the Clipboard before they are pasted, you may modify the contents of the XferData's Text property.

### **Resizing the TextEdit Form**

A Resize method was implemented on the parent form to automatically keep the size of the TextEditBox the same size as the form. This was done through the following code: `txtEditBox.Move 0, 0, ScaleWidth, ScaleHeight`

## **File Menu**

<b>Use This</b>	<b>To Do This</b>
New	Clears the current document and starts a new text file. When you choose New, you can save changes to the text file you have been working on.
Open	Opens a text file for editing.
Save	Saves changes to the text file you have been working on.
SaveAs	Saves the current text file to another text file. The original text file remains unchanged.
Print	Prints one copy of the current text file.
Exit	Closes the current text file you have been working on and exits the Text Edit application.

## **Edit Menu**

<b>Use This</b>	<b>To Do This</b>
Undo	Undoes your last editing or formatting action, including cut and paste actions. Some actions can not be undone, such as saving text to a file and printing.
Cut	Removes text that is currently selected and copies it onto the Windows Clipboard, replacing the previous Clipboard contents.
Copy	Copies text that is currently selected onto the Windows Clipboard, leaving the original selected text in place.
Paste	Pastes a copy of the Windows Clipboard contents at the current point, or replaces currently selected text.
Delete	Removes selected text, but does not copy the selection to the Windows Clipboard.
Select All	Selects all the text in the current file. You can copy the selected text onto the Windows Clipboard to paste into another application.

## Search Menu

<u>Use This</u>	<u>To Do This</u>
Find...	Searches for characters or words in the current text file.
Find Next	Repeats the last search without opening the Find Dialog box.

## Options Menu

<b>Use This</b>	<b>To Do This</b>
Fonts...	Posts a Font Dialog for setting or changing font attributes such as FaceName, Size and Color.
ScrollBar	
None	Sets the TextBox to contain no scroll bars. When this option is set, the WordWrap property is automatically set to True.
Horizontal	Sets the TextBox to contain a Horizontal scrollbar. When this option is set, the WordWrap property is automatically set to False.
Vertical	Sets the TextBox to contain a Vertical scrollbar. When this option is set, the WordWrap property is automatically set to True.
Both	Sets the TextBox to contain both a Horizontal and a Vertical scrollbar. When this option is set, the WordWrap property is automatically set to False.

## Formatting Text

<b>To</b>	<b>Do this</b>
End a line	Press ENTER.
Indent a line	Press TAB.
Insert a tab	Press TAB.
Insert a blank line	Move the point to the beginning or end of a line, and then press ENTER.

## **Choosing a Font and Font Size**

You can select a font and font size to modify the appearance of your text. The availability of fonts depends on the number of fonts you have installed.

### **To choose a font**

1. From the Option menu, choose Font...
2. Select a font from the Font Dialog box.
3. Select a size from the Size box. The Sample box changes to display your selection.
4. Select the OK button when you are finished selecting font attributes.



## Moving Around in the Text Editor

<b>Press</b>	<b>To move to...</b>
HOME	The beginning of a line.
END	The end of a line.
CTRL+HOME	The beginning of the text file.
CTRL+END	The end of the text file.

## **Copying, Cutting and Pasting Text**

You can use the Cut, Copy, and Paste commands to transfer information to and from the Windows Clipboard.

### **To copy text**

1. Select the text you want to copy.
2. From the Edit menu, select Copy.
3. Move the insertion point to the place you want the copied text to appear. Or select the text you want to replace with the copied text.
4. From the Edit menu, choose Paste.

### **To move text**

1. Select the text you want to move.
2. From the Edit menu, choose Cut.
3. Move the insertion point to the place you want the cut text to appear. Or select the text you want to replace with the copied text.
4. From the Edit menu, choose Paste.

## **Finding Text**

You can start a search for specific text at any point in a TextEdit box.

### **To find specific characters or words**

1. Move the insertion point to where you want the search to begin.
2. From the Search menu, choose Find.
3. Type the characters or words you want to locate in the Find Dialog box.
4. Click the Find Next button.

If there are no occurrences of the text in the portion of the text file searched, the TextEdit box will display a message indicating the text could not be found.

5. Choose the OK button.
6. To find the next occurrence of the text, choose the Find Next button.

## **Deleting Text**

When you want to remove text from the current text file but you do not want to erase the current contents of the Windows Clipboard, use the BACKSPACE or DEL key, or the Delete command from the Edit menu.

### **To delete one character at a time**

1. Press BACKSPACE to delete the characters to the left of the insertion point
2. Press DEL to delete the character to the right of the insertion point

### **To delete more than one character**

1. Select the text you want to delete.
2. Press the BACKSPACE or DEL. Or choose Delete from the Edit menu. To place the text onto the Windows Clipboard, choose Cut from the Edit menu.

## **Opening and Saving Files**

### **To open a file**

1. From the File menu, choose Open. The Open Dialog box appears.
2. If the file you want to use is on a different drive, select the drive you want from the Drives box.
3. In the Directories box, choose the directory you want to open.
4. From the list of files, select the file you want to open.
5. Double-click the filename, or choose the OK button.

### **To save existing TextEdit contents**

- n From the File menu, choose Save.

### **To save a new file or an existing file under a new name**

1. From the File menu, choose Save As. The Save As Dialog box appears.
2. If you want to save the file on a different drive, select the drive you want from the Drives box.
3. In the Directories box, choose the directory in which you want to save the file.
4. In the File Name box, type in a name for the file.
5. Choose the OK button.

## **Printing TextEdit Files**

TextEdit automatically prints to the default printer.

### **To print the current text contents**

- n From the File menu, choose Print

## **Select Text**

You can use this feature to copy an entire TextEdit box onto the Windows Clipboard, and then merge it into another file.

### **To select all text in a TextEdit box**

n From the Edit menu, choose Select All.

To unselect the text, simply click anywhere in the text editing area.

