

**Find Dialog Sample Help**

**Sample Description:** [Find Dialog](#)

**Points of Interest**

[Adding a Find Dialog to your Application](#)

[Posting the Find Dialog](#)

[Selecting Search Options](#)

[Finding a Search String in a TextBox](#)

**Control**

FindDialog

For Help on Help, Press F1

## **Find Dialog**

The Find Dialog is used to specify a text or search string to locate somewhere in your application form or document. The user can specify several search options in the Find Dialog. For instance, the user can specify the direction of the search, whether the application should match the case of the specified string, and whether the string to match is an entire word. The Find Dialog is created through a Windows Application Program Interface (API) and is available for use with Envelop applications.

### **To run the Find Dialog sample:**

1. Enter a text or search string in the box labeled **Search For:**.
2. Then, click the **Find** button. The Find Dialog will be displayed on the screen and should display your search string in a dialog box.
3. Click a desired option, then click the **Find Next** button. Repeated clicking of the Find Next button should result in subsequent occurrences of the search string being found in the large textbox.

As you examine this sample, notice that the Find Dialog does not actually perform any searches, it merely presents a list of options that a user may use to select various search parameters. Actual search program code must be written to implement the various search functions.

### Adding a Find Dialog to your Application

A Find dialog can easily be added to your application for searching textboxes for text strings. This is accomplished by embedding a FindDialog object as an object in your form's properties list. In the example shown, the FindDialog was embedded and given a name "FindPanel."

By expanding the FindDialog in the Property Editor, you will first notice several True/False parameters, such as **DisableWholeWord**, **HideMatchCase**, and **SearchDown**. These are some of the parameters that configure the search capabilities of the FindDialog. For instance, setting the **HideWholeWord** option to True, will cause this option to be hidden when the FindDialog is posted. For each of these functions to work, you must provide corresponding application code.

### Posting the Find Dialog

In our test sample, we have implemented a search capability for a single textbox. The textbox `txtSearchString` is used to capture the text string we intend to search for. The **Find** button runs a method called `btnFind_Click`, which checks to make sure we have specified a search string. If no search string is specified, a warning message is displayed. In our example, several of the advanced search options are hidden by setting their corresponding property values to `True`.

Several of the advanced search options are turned off in our example. The `FindDialog` is displayed on the screen by executing its **Show** method. For example: `FindPanel.Show`. When the dialog is displayed, the text string you typed into the Search For: textbox should appear in a dialog box.

The Find dialog box contains controls that make it possible for a user to specify the following:

- The string that the application should find
- Whether the string specifies a complete word or part of a word
- Whether the application should match the case of the specified string
- The direction in which the application should search (preceding or following the current cursor location)
- Whether the application should resume the search and continue searching for the next occurrence of the string

### Selecting Search Options

Selecting search options is only a matter of using the mouse to click the various options. Searching forward or backward is specified by clicking the Direction **Up** or **Down** option button. Matching Case and Whole Word searches are specified by clicking the corresponding checkbox.

The text string that you specified in the Search For: box, is set to the **FindString** property of the FindDialog. If you should decide to change the search string in the FindDialog, this property will automatically be updated, however the contents of the Search For: box will remain the same.

The **Find Next** button will execute the search, but will not remove the Find Dialog from the screen. The **Cancel** button will cancel the search operation.

### Finding a Search String in a TextBox

When the Find Next button is clicked, a corresponding "Find" method is automatically executed. To program the find function, you must have a Find method defined on the FindPanel object. This method is shown below.

```
Sub FindPanel_Find()  
    Dim loc As Integer  
  
    ' The FindPanel invokes a Find method when the FindNext button is clicked  
    loc = Instr(txtSearchBox.SelStart + 1 + txtSearchBox.SelLength, txtSearchBox.Text, FindPanel.FindString)  
    If loc Then  
        txtSearchBox.SelStart = loc - 1  
        txtSearchBox.SelLength = Len(FindPanel.FindString)  
        txtSearchBox.SetFocus  
    Else  
        InfoBox.Msg("Cannot find " & FindPanel.FindString & "")  
    End If  
End Sub
```

This routine performs the actual search for the specified text string. The search in a TextBox is performed by the **Instr** command. It works by searching for the next occurrence of the specified text string immediately following the current selection or insertion point. If no search is found, a message is displayed, otherwise the **SelStart** and **SelLength** methods of the TextBox object will carry out the search instruction and select the found text string.

