

Table Of Contents

Introduction.....	1
Why custom dialog boxes?.....	2
Why use the Dialog Editor?.....	3
Using the Dialog Editor.....	5
Invoking the Dialog Editor - Where is it?.....	5
Creating a Custom Dialog Box.....	5
Building the Dialog Box.....	5
Selecting items in a dialog box: 5	
Selecting the dialog box: 6	
Removing items in a dialog box: 6	
Moving items in a dialog box: 6	
Moving the dialog box: 6	
Resizing an item or the dialog box: 6	
Canceling moving or sizing an item: 7	
Item Properties.....	7
Changing a item or a dialog property 7	
Description of Properties 8	
Working with Grouped Items:.....	9
Adding an item to a group box: 10	
Changing a group of items: 10	
Copying the Dialog Box Into a Word Macro:.....	10
Editing An Existing Dialog Definition in the Dialog Editor:.....	12
Limitations of the Dialog Editor.....	12
Defining the Tab Stop Direction Between Items:.....	12
Setting the Default Button in a Dialog Box:.....	13
Using Different Fonts in a Dialog Box.....	14
Changing the Windows System Font: 14	
APPENDIX A Suggested WordBasic Readings.....	15
APPENDIX B: Dialog Box Statistics.....	16
APPENDIX C: Error Messages.....	16

Introduction

There are three fundamental WordBASIC commands which are commonly used to prompt for or display information for the user. These commands are: Input, InputBox\$, and MsgBox(). Each command has advantages and limitations. Below is a brief summary of these advantages and disadvantages.

Input: Prompts the user for a string using the status bar as the input device

Example:

```
Input "What is your name?", Name$
```

Advantages	Disadvantages
Quick and easy	Limited input from user
Doesn't clutter the screen	Not easily visible
	The user must KNOW to look in the status bar for input

MsgBox: Presents the user with a message and receives input via predefined buttons.

Example:

```
choice = MsgBox("Would you like to continue?", "Title", 35)
```

Advantages	Disadvantages
Colorful and easy to use	No text input
Great for quick Yes/No responses	
Great for displaying messages	
Very visible	

InputBox\$: Prompts the user for a string using a simple text box as the input device

Example:

```
a$ = InputBox$("Enter you name", "Get Name")
```

Advantages	Disadvantages
Colorful and easy to use	Limited to one text string selection
Very visible	

Why custom dialog boxes?

Custom dialog boxes are important in that they allow you to prompt the user for more specific information using many of the same elements found in Word's own dialog boxes. These elements include push buttons, check boxes, option buttons, option groups, list boxes, combo boxes, and text boxes.

The following table summarizes the items and their possible uses in custom dialog boxes.

Item	Example	Use
Text	Directories:	Label items in a dialog box
Text Box		Prompt the user for text or numbers

OK Button		Default button for accepting information in a dialog
Cancel Button		Button to elect to cancel all selections in the dialog box
Push Button		To control the execution of a macro by choosing a different path.. say create a new file or open and existing one
Check Box		To choose one or more from a series of choices, such as Bold, Italic, Underline.
Group Box		Used to group other items together. Used primarily with option buttons.
Option Button		Similar to check boxes, but when used with other option buttons in a group box, only one can be chosen. Used to make a single selection from multiple choices.
List Box		Used to display a list of items to choose from.
Combo Box		Similar to a text box only the selected item will be copied into the attached text box. Selection can also be made by typing in the attached text box.

Why use the Dialog Editor?

Before the Dialog Editor with Word for Windows version 2.0, macro developers manually typed dialog definitions by hand. The Dialog Editor makes it easy to quickly design a dialog box without typing tedious dialog statements.

The following is a sample that contains many elements available to use in a custom dialog:

```
Begin Dialog UserDialog 440, 243, "Complex Dialog Box"  
  Text 16, 12, 370, 18, "Enter Text"  
  TextBox 16, 25, 190, 18, .Text  
  GroupBox 240, 10, 180, 90, "Group Box"  
  OptionGroup .OptionChoice  
  OptionButton 250, 30, 140, 18, "Option One"  
  OptionButton 250, 50, 140, 18, "Option Two"  
  OptionButton 250, 70, 140, 18, "Option Three"  
  CheckBox 16, 50, 160, 18, "Check Box One", .CheckOne  
  CheckBox 16, 70, 160, 18, "Check Box Two", .CheckTwo  
  CheckBox 16, 90, 160, 18, "Check Box Three", .CheckThree  
  ListBox 16, 115, 208, 84, ArrA$, .ListChoice  
  ComboBox 242, 114, 175, 120, ArrB$, .ComboChoice  
  OKButton 16, 210, 64, 21  
  CancelButton 88, 210, 64, 21  
  PushButton 160, 210, 64, 21, "Apply"  
End Dialog
```

Although at first sight a dialog definition appears to be complex, it really is simple to create when using the Dialog Editor. The Dialog Editor will generate the correct macro code that describes all elements of the dialog. The same dialog above looks like the following in the Dialog Editor:

Using the Dialog Editor

Invoking the Dialog Editor - Where is it?

The Dialog Editor is located in the Word for Windows program directory and is called MACRODE.EXE. Start the editor by one of the following methods:

1. Run MACROODE.EXE from the File Run dialog from the Program Manager.
2. Create an icon in the Program Manager for MACRODE.EXE and invoke it by double clicking the icon.
3. Find and double click on the file MACRODE.EXE from the Windows File Manager.
4. Install and assign the LaunchDialogEditor macro to a menu in Word. The NEWMACRO.DOC file, located in the Word for Windows program directory, includes the LaunchDialogEditor macro as well as other sample macros.

When the Dialog Editor is started an empty dialog box is displayed in the middle of the screen.

Creating a Custom Dialog Box

The following three steps are used to create and use a custom dialog box:

1. The dialog is first constructed using the Dialog Editor.
2. The dialog is copied from the Dialog Editor to the clipboard and then pasted into the macro editing window.
3. Macro commands are written to utilize and take advantage of the new dialog box.

Building the Dialog Box

From the Insert menu choose the item that you want to use. If the item is a button or listbox, an additional dialog is displayed to further specify what type or variety of button or listbox is to be used.

The Dialog Editor will place the new item in a default location with a default size that can be manipulated.

Tip After an item is placed in the dialog, pressing ENTER will create a copy of the same item which will be placed just below the previous item. This is a great way to add multiple items to a dialog.

Selecting items in a dialog box:

In order to size, move, delete, or modify an item the item must first be selected. Items can be selected in two ways:

1. Click on the item with the mouse.
2. Press the TAB key until the item is selected. The TAB key will select through the

different items in the dialog.

Tip To select more than one item at a time for manipulation, hold down the SHIFT key and click on any other items to be included in the selection.

Selecting the dialog box:

In order to size, move, or modify the dialog box you must first select the dialog box. The dialog can be selected in two ways:

1. Click with the mouse on the caption bar of the dialog box.
2. Choose Select Dialog from the Edit menu.

Removing items in a dialog box:

Select the item or items to be removed from the dialog and then choose Clear from the Edit menu or press the DEL key on your keyboard.

Moving items in a dialog box:

Items can be moved in three ways:

1. Select the item with the mouse and drag it to the new location in the dialog.
2. Modify the X and Y coordinate properties for the item. See *Item Properties* for more information about editing and using properties.
3. Select the item and use the arrow keys on the keyboard to move the item. A gray outline box the same size as the item will appear when using the arrow keys. Position this to the desired location and press ENTER to accept the new position.

Tip Hold down the SHIFT key while dragging a item with the mouse and the item will only move vertically and horizontally, but not diagonally. This is ideal for moving items while ensuring that they remained aligned on the same horizontal or vertical axis.

Note If a dialog item refuses to move, you may have to uncheck the X or Y coordinate Auto property. The Auto property fixes the item on the X or Y axis to prevent it from being moved from that axis. To remove the Auto property select the item and choose Info... from the Edit menu or double click on the item to bring up the item information dialog.

Moving the dialog box:

The dialog box can be moved in three ways:

1. Click and drag the dialog caption bar to the new location.
2. Modify the X and Y coordinate properties for the dialog box.
3. Select the dialog box and use the arrow keys to move it.

Note If the dialog refuses to move, you may have to uncheck the X or Y coordinate Auto property. The Auto property fixes the dialog box in the middle of the screen.

Resizing an item or the dialog box:

After selecting the desired item or dialog box, it can be sized in two ways:

1. Using the mouse, drag one of the borders to stretch it to the desired size.
2. Modify the Width and Height properties in the item Information dialog (double click on the item or choose Info... from the Edit menu).

Tip If you double click on the border of the dialog box, the dialog will automatically resize to fit all the items in the dialog box. This is helpful in eliminating any extra "white space" around the circumference of the dialog box.

Tip An item such as an OK button, can be sized back to its default size by selecting the item and choosing Resize from the Edit menu.

Canceling moving or sizing an item:

Press ESC before releasing the mouse button or pressing ENTER to accept an item or dialog box property change.

Item Properties

Items in a dialog box, as well as the dialog box itself, have a number of characteristics or properties that can be altered. These include properties to change the position and size of the item, the text on or in the item, caption bar titles etc...

Changing a item or a dialog property

Properties can be altered within the Info dialog box. First, select the item whose properties are to be modified or viewed (if the dialog itself is required, choose Select Dialog from the Edit menu).

There are two ways of invoking the Info dialog box.

1. Double click on the item to be altered or on the caption bar if the dialog is to be modified.
2. Choose Info from the Edit menu.

When the info dialog is invoked, make the necessary changes and then choose the OK button. The following dialog is the Info dialog box for a Push Button item.

Different items may or may not use some of the properties found in the dialog. These properties will be disabled for editing, or grayed out as in .Field in the example above.

Description of Properties

Position Available for ALL items and the dialog box.

Consists of an X and Y coordinate position measure. For the dialog box, these indicate the offset from the upper left corner of the Word window. For items, these indicate the offset from the upper left corner of the dialog box.

The unit of measure used for this property are:

width = 1/8 the Windows system font

height = 1/12 the Windows system font

Auto Checkboxes

When Auto is selected for items, Word will automatically position the item to a default X or Y position. The X coordinate will be positioned identical to that of the last item created so they align horizontally. The Y coordinate will be positioned immediately below the last item created. For example, if a push button is created immediately following the creation of an OK button and the Auto checkboxes are marked for each item, the push button will appear immediately below the OK button aligned horizontally. Further, if the OK button is moved, the push button will move with it automatically.

When Auto is selected for a dialog box, Word will position the dialog centered horizontally and vertically in the middle of the document window. The dialog will display centered in the Word window when activated.

Size Available for ALL items and the dialog box.

Consists of a Width and Height measure with units of measure equivalent to the Position property.

The maximum size of a dialog box is 510 x 380 in Word versions 1.x and 2046 x 1535 for Word versions 2.x.

Auto Checkboxes

When checked for items, Word will automatically size the item according to a fixed default size. This size is the same size that new items take when created. There is no way to change the initial or default size.

When checked for a dialog box, Word will automatically size the dialog box down to fit the items that appear in it, essentially cropping any extraneous white space between the dialog borders and the items contained within them. This is a nice feature to use after all items have been added to the dialog box.

Tip Double clicking the borders of the dialog box will cause the borders to

resize to fit the items that appear on the form.

Tip The Position and Size properties are an excellent way to have more precision as far as placement of items in the dialog box. Sometimes it is very difficult to use the mouse to line things up perfectly.

Text\$ Available for push and option buttons, text labels, check and group boxes.

For push buttons, text labels, and group boxes, this is the text that will appear on the item. For option buttons and check boxes, this is the text that appears immediately to the right of the item. For the dialog box, this is the text that appears in the caption bar.

The maximum number of characters that can be entered and displayed is 255.

Auto Quote Checkbox

This checkbox is used to indicate whether the value entered into the Text\$ property is an actual string or if it is a WordBASIC string variable that will be set prior to displaying the macro.

By checking the box, the Dialog Editor will automatically put quotes around the text in the dialog definition.

```
PushButton 396, 23, 88, 21, "Enter you name"
```

By unchecking the box, the Dialog Editor assumes that the value is a variable and omits the quotes.

```
PushButton 396, 23, 88, 21, NameText$
```

.Field Available for text boxes, option buttons, check boxes, list and combo boxes.

Indicates the symbolic name to be assigned to the item for use by the macro to set and extract values from the custom dialog box. This field must be preceded by a period.

Although Word will assign a default name in this field, such as ".TextBox1" for a text box, it is a good idea to use descriptive names that correspond to the item's purpose. If a text box is used to type in a zip code number, the field might read ".ZipCode".

This field is very important for a macro to be able to retrieve data from an item after the dialog is presented to the user.

Comment Available for ALL items.

Adds a comment to the generated code to further clarify the use of the item. This is optional, but recommended for code readability.

Array\$ Available for list and combo boxes only.

Indicates the name of the WordBASIC array to be used to fill the list and combo boxes with items.

Working with Grouped Items:

The group box item allows us to organize other items together. It can be used for aesthetic purposes only, or it can be used to facilitate the exclusive selection of an option button from a set of option buttons. Further, once some items are grouped in a group box, they can be moved, resized, or deleted together. After a group box is created, other items can be added into or removed from a group box. Group boxes are created just like other items from the Insert menu.

Adding an item to a group box:

1. Choose the group box to contain the item.
2. From the Edit menu choose the item to add to the group box.

Tip If an item already exists in the dialog box and needs to be added to a group box it can be cut to clipboard, the group box selected and then the item can be pasted back into the group box.

Changing a group of items:

1. Select the group box.
2. Choose Select Group from the Edit menu.
3. Move, resize, or delete the group just like any other singular item.

The following dialog uses each of the Word for Windows dialog items in a single dialog.

Copying the Dialog Box Into a Word Macro:

After creating or modifying a dialog box you can copy and paste it into a macro.

To copy a dialog box to a macro:

1. From the Edit menu choose Select Dialog.
2. From the Edit menu choose Copy.
3. Start or switch to Word for Windows and open or create a macro.
4. Position the insertion point where the dialog definition is needed.
5. Choose Paste from the Edit menu.

The following steps outline the steps to create a new macro with a dialog box definition.

1. Copy the dialog box to the clipboard via the instructions above.
2. Start or switch to Word for Windows and choose Macro from the Tools menu.
3. In the Macro Name box, type "MyCustomDialog", then choose the Edit button.
4. Position the insertion point between the "Sub MAIN" and "End Sub" lines.
5. Choose Paste from the Edit menu.

6. Position the insertion point in front of the "End Sub" statement and press ENTER 2 times to create 2 blank lines at the bottom of the macro.
7. On the first of the 2 lines type "Dim dlg As UserDialog"
8. On the second of the 2 lines type "choice = Dialog(dlg)"

The macro should now look similar to what is found below:

```
Sub MAIN
Begin Dialog UserDialog 362, 275, "My First Custom Dialog"
  Text 17, 6, 81, 18, "Text Label"
  TextBox 110, 5, 230, 18, .Text
  GroupBox 190, 29, 152, 86, "Group Box" 'Group box around options
  OptionGroup .OptionChoice 'Define option group
  OptionButton 200, 47, 121, 18, "Option One"
  OptionButton 200, 67, 128, 18, "Option Two"
  OptionButton 200, 87, 127, 18, "Option Three"
  CheckBox 18, 33, 160, 18, "Check Me 1", .Check1
  CheckBox 18, 55, 160, 18, "Check Me 2", .Check2
  CheckBox 18, 77, 160, 18, "Check Me 3", .Check3
  CheckBox 18, 99, 120, 16, "Check Me 4", .CheckBox4
  ListBox 17, 126, 160, 107, ArrA$, .ListChoice
  ComboBox 190, 126, 160, 108, ArrB$, .ComboChoice
  OKButton 16, 246, 79, 21
  CancelButton 106, 246, 79, 21
  PushButton 192, 246, 161, 21, "Push Me!"
End Dialog 'End of dialog definition
Dim dlg As UserDialog
choice = Dialog(dlg)
End Sub
```

To Run The Macro: Choose the Start button on the macro toolbar. If the dialog box does not look the way you expect it can always be edited further in the Dialog Editor. The additional syntax needed to display the dialog box is as follows:

```
Dim dlg as UserDialog
Dialog dlg
```

The following example macro uses the Dialog function to determine which button in the dialog box was selected (OK, Cancel or Pushbutton)

```
Sub MAIN
Begin Dialog UserDialog 320, 120, "Microsoft Word"
  OKButton 210, 56, 88, 22
  CancelButton 211, 83, 88, 21
  PushButton 50, 15, 197, 21, "PushButton"
End Dialog
Dim dlg As UserDialog
n = Dialog(dlg)
If n = -1 Then MsgBox "OK Button was pushed"
If n = 0 Then MsgBox "Cancel Button was pushed"
If n = 1 Then MsgBox "Pushbutton was pushed"
End Sub
```

Editing An Existing Dialog Definition in the Dialog Editor:

One fantastic feature of the Dialog Editor is that it can be used to not only create new custom dialog boxes, but modify existing ones! We simply need to copy the dialog definition from the macro and paste it into the Dialog Editor.

To edit an existing dialog definition:

1. Select the dialog definition from the macro window to be modified. Be sure to include the lines starting with "Begin Dialog UserDialog" and "End Dialog"
2. Choose Copy from the Edit menu.
3. Start or switch to the Dialog Editor.
4. Choose Paste from the Edit menu.

Note Any dialog that was in the Dialog Editor will be removed to make way for the new dialog box. Be sure to copy any old dialog out of the editor if it is to be preserved.

Limitations of the Dialog Editor

Defining the Tab Stop Direction Between Items:

There is currently no facility in the Dialog Editor for indicating the order in which items are visited via the TAB key. The actual order is determined by the position of the item descriptions within the actual dialog definition text inserted into the macro. Currently this order is determined by the order in which items were created in the dialog box. The only mechanism for changing this order is to physically reorder the item definitions in the dialog definition AFTER it has been copied into the macro.

Consider the following example:

```
Begin Dialog UserDialog 362, 275, "My First Custom Dialog"  
  CancelButton 106, 246, 79, 21  
  OKButton 16, 246, 79, 21  
  TextBox 110, 5, 230, 18, .Text  
  CheckBox 18, 33, 160, 18, "Check Me 1", .Check1  
End Dialog      'End of dialog definition
```

The order visited would be:

1. Cancel button
2. OK button
3. Text box
4. Check box

Normally we would want to have the OK button receive the initial focus. In order for this to occur we simply need to copy the OKButton line in front of the CancelButton line to read:

```
Begin Dialog UserDialog 362, 275, "My First Custom Dialog"  
  OKButton 16, 246, 79, 21  
  CancelButton 106, 246, 79, 21
```

```
TextBox 110, 5, 230, 18, .Text
CheckBox 18, 33, 160, 18, "Check Me 1", .Check1
End Dialog          'End of dialog definition
```

Setting the Default Button in a Dialog Box:

It is possible to control which push button will automatically be selected as the default button in a dialog box. The default button is the button in the dialog box that is selected by default when you press ENTER. This button is distinguished by a bold border around the circumference of the button.

Unfortunately this property cannot be determined by the Dialog Editor. The problem is related to the order in which the push buttons appear in the dialog description and an undocumented parameter for the Dialog command which displays the custom dialog box. The WordBASIC Dialog command has an undocumented second argument, called DefaultButton, which you can use to specifically identify this default button.

The correct, complete syntax for this command is as follows:

```
Dialog DialogRecord [, Defaultbutton]
n = Dialog(DialogRecord [, Defaultbutton])
```

In the above syntax, the optional Defaultbutton argument indicates the dialog button that is selected by default when you press ENTER. The first button found in the dialog definition will be designated automatically as the default button under the following circumstances:

1. The Defaultbutton argument is omitted
2. The first item in the dialog definition is a button.

The following values are valid settings for the Defaultbutton argument:

- 1 The OK button
- 0 The Cancel button
- >0 A PushButton. Defaultbutton chooses the PushButton in the dialog definition based on its declared position: 1 for the first declared PushButton, 2 for the second, and so on.

If the Defaultbutton value is less than -1 or greater than the number of PushButton declarations, no button is selected as the default button. The following macro demonstrates the use of the Defaultbutton argument:

```
Sub MAIN
Defaultbutton = 0 ' The Cancel button is set as the default
Begin Dialog UserDialog 190, 130, "Buttons"
  TextBox 110, 21, 80, 18, .TextBox1
  OKButton 10, 78, 88, 21
  CancelButton 10, 102, 88, 21
  PushButton 10, 6, 88, 21, "One"
  PushButton 10, 30, 88, 21, "Two"
  PushButton 10, 54, 88, 21, "Three"
End Dialog
Dim dlg As UserDialog
On Error Goto CancelPressed
Choice = Dialog(dlg, Defaultbutton)
Select Case Choice
```

```
Case - 1
    MsgBox "OK pressed!"
Case 1
    MsgBox "ONE pressed!"
Case 2
    MsgBox "TWO pressed!"
Case 3
    MsgBox "THREE pressed!"
End Select
Goto Bye
CancelPressed:
    MsgBox "CANCEL pressed!"
Bye:
End Sub
```

To further demonstrate the behavior of the `DefaultButton` argument, change the value of the `Defaultbutton` argument to an integer between -1 and 3. Run the macro and press ENTER when the dialog box is displayed. If you set the value of the `Defaultbutton` argument to less than -1 or greater than 3, no button is selected when you press ENTER.

Using Different Fonts in a Dialog Box

Custom dialog boxes use the system fixed font as a default for displaying text in a dialog box. There is currently no functionality for specifying the use of different fonts and formatting of text within the Dialog Editor. However, it is possible to select a different system fixed font for use in the system. In extreme cases it can be changed, but it is not recommended since this change will be made system wide to all applications in Windows. This font is used to display menu and caption bar text in windows.

Changing the Windows System Font:

1. Open the SYSTEM.INI file using a convenient text editor.
2. In the [BOOT] section find the line beginning with "Fonts.Fon="
3. Change the name of the font file following the equal sign. 8514SYS.FON is the largest font in height, followed by VGASYS.FON, and then EGASYS.FON.

Note These additional font files may not be available on your system. Please consult your Windows documentation for information on expanding and installing files from you Windows disks. *Please* make a backup copy of your SYSTEM.INI file prior to making any changes.

APPENDIX A Suggested WordBasic Readings

Microsoft Word for Windows User's Guide

Chapter 37 (Document Templates) and Chapter 42 (Using Macros) are very good resources.

Using WordBASIC

"Using WordBasic" (part number 059-050-574) by WexTech Systems, Inc. is available by calling Microsoft Consumer Sales at (800) 426-9400. Your first copy is free; additional copies cost \$15.00 each, plus tax and shipping charges.

The Macro Developer's Kit (MDK)

This kit, developed by Word for Windows Marketing, contains information for consultants, in-house developers, and others who need to develop macros with Word for Windows. The Microsoft Word for Windows Macro Developer's Kit includes the following items:

- "Using WordBasic": An updated version of the easy-to-read macro language tutorial (WordBasic command reference) written by WexTech Systems and Microsoft.
- Developer's Handbook: A handbook that includes materials on optimizing dynamic data exchange (DDE) with Word, tips and tricks to get the most out of WordBasic, examples of third-party products that can be integrated with Word, and information on how to create interactive help files using WordBasic. (The Developer's Handbook was compiled from the 1991 Microsoft Developer's Forum.)
- Printed Fields Documentation:
- RTF Specifications: Rich Text File format specifications used for add-on products that need to read Word for Windows' files.
- Microsoft Consultant Relations Program Information and Microsoft Certified Professional Program Information
- WordBasic Update Document: Documentation on changes from WordBasic 1.x to 2.x.

The cost of the MDK is \$19.95 plus shipping and handling. To order the Microsoft Word for Windows Macro Developer's Kit, call (800) 323-3577. If you are outside the United States, contact the Microsoft subsidiary for you are. To locate your subsidiary, call Microsoft International Customer Service at (206) 936-8661.

Reference Books

"Microsoft Word for Windows 2.0 Macros" (part number 1-55615-486-0) by Russell Borland is available at larger bookstores, or by calling (800) MS-PRESS, that is, (800) 677-7377. The suggested retail price is \$34.95. This book includes technical reference and macro information for versions 2.0, 2.0a, and 2.0a-CD of Word for Windows.

"Windows Programming for Mere Mortals," by Woody Leonard, is available at larger bookstores (Addison-Wesley Publishing Company ISBN 0-201-60832-4).

APPENDIX B: Dialog Box Statistics

Statistic	Word 1.x	Word 2.0, 2.0a
Max size of a dialog box	510 x 380	2046 x 1535
Max number of character in a Text\$ field	255	255
Max number of items in a dialog box	18	32
Width unit of measure	1/8 system font	same
Height unit of measure	1/12 system font	same

APPENDIX C: Error Messages

WordBASIC Err=5: Illegal function call

1. The maximum size of a dialog box may have been exceeded. The maximum sizes are 510x380 in Word versions 1.x and 2046x1535 for Word versions 2.x.
2. The dialog box may not contain a button. Make sure that an OK button or a push button appears in the dialog box.
3. The location of the dialog box may be positioned off the screen
4. The dialog box contains more that one OK or Cancel button.

WordBASIC Err=513: Dialog box too complex

The maximum number of items that can be displayed was exceeded. Word versions 1.x can have a maximum of 18 items and Word versions 2.0, 2.0a a maximum of 32 items. Word will highlight the last item that was trying to be displayed when this error occurs. If your dialog contains long text items, this error may appear before 32 items are added to the dialog. For long blocks of text, consider loading your text into an array and using a listbox.

WordBASIC Err=513: String too long

The maximum number of characters was exceeded for a text value property. The maximum number of characters is 255.

