

# SetText Method Confusion

**Source:** <http://www.tech-archive.net/Archive/Word/microsoft.public.word.vba.beginners/2004-08/0167.html>

---

**From:** Jonathan E. (*anonymous\_at\_discussions.microsoft.com*)

**Date:** 08/12/04

Date: Thu, 12 Aug 2004 14:05:51 -0700

I'm a rank beginner with Word (2002) macros and have run into a problem I can't get past. The short question is: what is the correct syntax for the SetText Method? The syntax diagram is different from every example I've found and neither will compile when I use them in my macro.

By way of more explanation ... I am trying to write a macro that will go through one Word document (which itself has been ported over to Word from PageMaker) copying one word at a time [Ctrl-Shift- ->] then switching to a second, newly-created-with-each-macro-run document, and pasting the previously copied text at the end of the new document. The only exception the macro makes to this is if the copied text contains a fully CAPITALIZED word. In that case it converts it to all lower case and prepends "~~~" to the word (for later lookup, inspection and possible tweeking) before pasting the changed string to the new document. In this way a changed copy of the file is generated without affecting the original.

The problem I'm having is that I seem to be able to write compilable commands to copy text from the original document into the clipboard, from the clipboard into a DataObject, and from the DataObject into a string so the string can be examined and manipulated as necessary but I can't reverse the process. When I try to follow the syntax of the SetText method in my macro I get a syntax error saying I need an "=" sign. To make matters worse, every example I've found of the command, either in the Help files or online, show the method used with a different syntax. However, when I follow that format I get a compile-time error[-2147024809 (80070057)]. I've also tried using several other variations but nothing seems to let the macro compile or run.

Any assistance would be appreciated.

Jonathan

According to the Help file the following is the official syntax for SetText:

#### SetText Method

Copies a text string to the DataObject using a specified format.

#### Syntax

```
object.SetText( StoreData [, format])
```

The SetText method syntax has these parts:

#### Part Description

**object** Required. A valid object.

**StoreData** Required. Defines the data to store on the DataObject.

**format** Optional. An integer or string specifying the format of StoreData. When retrieving data from the DataObject, the format identifies the piece of data to retrieve.

#### Settings

The settings for format are:

#### Value Description

1 Text format.

X A string or integer value other than 1 is a user-defined DataObject format.

But the example shows the following line as an example of the using the SetText method.

```
MyDataObject.SetText TextBox1.Text
```

My as-of-yet uncompiled code is as follows:

```
Dim OrigDoc, NewDoc As Document
Dim OrigDocName, OrigDocPath, NewDocName, NewDocPath,
ClipString As String
Dim ClipData As DataObject
Set ClipData = New DataObject
Dim Select_str As String
```

```
' Check to see that only one document is open. Assumes it
is the correct one.
```

```
If Documents.Count = 1 Then
```

```
    OrigDocName = ActiveDocument.Name
```

```
    OrigDocPath = ActiveDocument.Path
```

```

Else
  If Documents.Count = 0 Then
    MsgBox "Please first open the document to be processed
before starting this macro"
  Else
    MsgBox "Please close all documents EXCEPT the one to be
processed for Upper Case occurrences"
  End If
End ' Quit macro
End If

' Open and activates a new blank document for processing
' the results of the first one.

NewDocPath = OrigDocPath + "\"
NewDocName = OrigDocName + ".out"
Documents.Add DocumentType:=wdNewBlankDocument
Documents(1).SaveAs (NewDocPath + NewDocName)

Select_str = ""
Do Until InStr(Select_str, "~***~") = 1
  Windows(2).Activate
  Selection.MoveRight Unit:=wdWord, Count:=1, Extend:=wdExtend
  ' copy selection to the clipboard
  Selection.Copy
  ' copy clipboard contents to Data Object
  ClipData.GetFromClipboard
  ' copy Data Object (text) to string
  Select_str = ClipData.GetText(1)
  '
  Call ProcessSelection(Select_str)

  ' Put string back in Data object – neither technique
  works
  ClipData.SetText(Select_str,1) ' <== SetText format
  according to Syntax
  diagram
  ClipData.SetText Select_str ' <== SetText format
  according to example

  ' ===== Don't know if anything past here compiles ...

  ' ClipData.SetText(Select_str,1)
  ' Put Data Object text into Clipboard
  ClipData.PutInClipboard
  ' paste clipboard into forming document
  Windows(1).Activate
  Selection.PasteAndFormat (wdPasteDefault)
  Windows(2).Activate
  ' set up for next iteration

```

## microsoft.public.word.vba.beginners: SetText Method Confusion

```
Selection.MoveRight Unit:=wdCharacter, Count:=1  
Selection.LtrPara
```

```
Loop  
Documents(2).SaveAs (NewDoc)  
Documents(1).Close  
Documents(2).Close  
End Sub
```

```
Public Sub ProcessSelection(Full_str)  
Dim AllUpper_b As Boolean  
Dim CountLen, Count As Integer  
  
AllUpper_b = True  
CountLen = Len(Full_str)  
Count = 1  
Do While (AllUpper_b) And (Count <= CountLen)  
  If InStr(UpperCase_str, Full_str(Count)) = 0 Then  
    AllUpper_b = False  
  End If  
  Count = Count + 1  
Loop  
  
If (AllUpper_b) And (Count > 1) Then  
  Full_str = "~~~" + LCase$(Full_str)  
  
End If  
  
End Sub
```