

Form Fields Lose Text When Protected for Forms

Article ID: Q181109

Last Reviewed:
October 25, 1999Provided by [Microsoft Product Support Services](#).

 Did the information in this article help answer your question?

Yes

No

Did not apply

Please provide additional comments about this information. If you require a response or technical support, please click [Contact Us](#).
(255 character max)

 The information in this article applies to:

- Microsoft Word 98 Macintosh Edition

SUMMARY

This article describes methods that allow you to:

- Retain information you type into a form field when you protect a form.
- Unprotect a forms document, perform a spell check, and retain information in the form fields when you reprotect the form.

MORE INFORMATION

Microsoft provides programming examples for illustration only, without warranty either expressed or implied, including, but not limited to, the implied warranties of merchantability and/or fitness for a particular purpose. This article assumes that you are familiar with the programming language being demonstrated and the tools used to create and debug procedures. Microsoft support professionals can help explain the functionality of a particular procedure, but they will not modify these examples to provide added functionality or construct procedures to meet your specific needs. If you have limited programming experience, you may want to contact the Microsoft fee-based consulting line at (800) 936-5200. For more information about the support options available from Microsoft, please see the following page on the World Wide Web:

<http://www.microsoft.com/support/supportnet/refguide/>

Method 1: Alter the Protect/Unprotect Command Functionality

The following sample Microsoft Visual Basic for Applications macros (Sub procedures) protect your form without causing you to lose the text that you entered into a form field. The macros can be stored in the actual form template to allow you to manually unprotect and reprotect the form while preserving the form field contents.

The following three Visual Basic macros can be used to ensure that your form field values are not reset to their defaults when you reprotect the form.

- The first macro runs when you click the Protect Form button on the Forms toolbar.

- The second macro runs when you click either Protect Document or Unprotect Document on the Tools menu.
- The third macro allows you to specify which sections to protect while maintaining previous form field values.

NOTE: The name of this macro must be ProtectForm.

```
Sub ProtectForm( )

    ' *****
    ' ProtectForm Macro.
    ' Toggles protection for the active document
    ' when the Protect Form button on the forms
    ' toolbar is clicked.
    ' *****
    If ActiveDocument.ProtectionType = wdNoProtection Then
        ActiveDocument.Protect Type:=wdAllowOnlyFormFields, NoReset:=True
    Else
        ActiveDocument.Unprotect Password:=""
    End If

End Sub
```

The following sample Visual Basic macro protects the active document without displaying the Protect dialog box. When you run this macro, it will reprotect the active document while maintaining previous form field values.

NOTE: The name of this macro must be ToolsProtectUnprotectDocument.

```
Sub ToolsProtectUnprotectDocument( )

    ' *****
    ' ToolsProtectUnprotectDocument Macro
    ' Sets protection for the active document
    ' when Protect Document or Unprotect Document
    ' is clicked on Tools menu
    ' *****
    If ActiveDocument.ProtectionType = wdNoProtection Then
        ActiveDocument.Protect Type:=wdAllowOnlyFormFields, NoReset:=True
    Else
        ActiveDocument.Unprotect Password:=""
    End If

End Sub
```

The following sample Visual Basic macro allows you to specify which sections to protect while maintaining previous form field values. You can assign this macro to a toolbar button or menu.

```
Sub ProtectNoReset( )

    Dim pDoc As Dialog
    Dim x As Integer
    On Error GoTo ProtectNoResetErr

    ' If the document is protected...
    If ActiveDocument.ProtectionType <> wdNoProtection Then

        ' Unprotect the document.
        ActiveDocument.Unprotect

    End If

End Sub
```

```

' Display the Protect dialog box.
Set pDoc = Dialogs(wdDialogToolsProtectDocument)
x = pDoc.Display

' If Cancel was chosen, exit this procedure.
If x = 0 Then Exit Sub

' Protect the document.
ActiveDocument.Protect Password:=pDoc.DocumentPassword, _
    NoReset:=True, Type:=2

ProtectNoResetErr: 'NOTE: This line MUST be left aligned.
    If Err <> 0 Then MsgBox Err.Description

End Sub

```

Method 2: Create a Macro to Protect/Unprotect Your Document

The following examples protect the active document for forms without resetting the contents of the form fields. Create the macro and assign the macro to a key, menu, or toolbar button for easy access.

```

If ActiveDocument.ProtectionType = wdNoProtection Then
    ActiveDocument.Protect Type:=wdAllowOnlyFormFields, NoReset:=True
End If

```

Method 3: Unprotect, Check Spelling or Update a Field, Reprotect a Document

Because form field text is formatted for No Proofing, you can use the following macro to:

- Temporarily Unprotect the form.
- Change the language of the form fields.
- Perform a spell check or update a field.
- Reprotect the form while preserving the text you've typed into the form fields.

You can use this macro as an On Exit macro for the last form field so you can check the spelling or update a field before you save the form.

```

Sub FormsSpellCheck()

' If document is protected, Unprotect it.
If ActiveDocument.ProtectionType <> wdNoProtection Then
    ActiveDocument.Unprotect Password:=""
End If

' Set the language for the document.
Selection.WholeStory
Selection.LanguageID = wdEnglishUS

' Perform Spelling/Grammar check.
If Options.CheckGrammarWithSpelling = True Then

```

```
ActiveDocument.CheckGrammar
Else
ActiveDocument.CheckSpelling
End If

' ReProtect the document.
If ActiveDocument.ProtectionType = wdNoProtection Then
ActiveDocument.Protect Type:=wdAllowOnlyFormFields, NoReset:=True
End If

End Sub
```

REFERENCES

For more information about getting help with Visual Basic for Applications, please see the following article in the Microsoft Knowledge Base:

[Q163435](#) VBA: Programming Resources for Visual Basic for Applications

Additional query words: lost text preserve retain available checkspelling

Keywords : kbdt kbdta code OffVBA macword98

Version : MACINTOSH:98

Platform : MACINTOSH

Issue type : kbinfo

Last Reviewed: October 25, 1999

[© 2000 Microsoft Corporation. All rights reserved. Terms of Use.](#)