

Re: fastest way to change case of string

Source:

<http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2006-05/msg02004.html>

- *From:* "Rick Rothstein" <rickNOSPAMnews@xxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 24 May 2006 11:42:55 -0400
-

Have tested this now and changing the case in the supplied string rather than concatenating a new string is indeed much faster.

This also means that as there usually are not that many quotes it is worth it to see where the quotes are and run UCase or LCase on larger strings. Haven't done this yet, but I suspect that will make it much faster still.

You tend to write large routines, I strive to only do what is required. I suspect (and this has been discussed before) that copying the text into and out of the byte array will take a significant amount of time when compared to the actual work being done. You can avoid that by simply working with the text data. Check out the addition I added to allow for different case selections... ;-)

HTH
LFS

(In a new form...)

```
Option Explicit
Private Enum ChangeCaseOption
ccUpperCase = 1
ccLowerCase = 2
ccProperCase = 3
End Enum
```

```
Private Sub Form_Load()
Const TestData = "ABcd 'EFgh' IJkl 'MNop' QRst"
Debug.Print ChangeCaseX(TestData, ccUpperCase)
Debug.Print ChangeCaseX(TestData, ccLowerCase)
Debug.Print ChangeCaseX(TestData, ccProperCase)
```

Re: fastest way to change case of string

End Sub

```
Private Function ChangeCaseX(ByRef Text As String, ByVal Action As
ChangeCaseOption) As String
Dim quoted As Collection
Dim posA As Long, posB As Long, vlu
Const QUOTE = ""

' Change case
ChangeCaseX = StrConv(Text, Action)
posA = InStr(1, Text, QUOTE, vbBinaryCompare)

If posA > 0 Then
' Find quotes
Set quoted = New Collection
Do While posA > 0
posB = InStr(posA + 1, Text, QUOTE, vbBinaryCompare)
quoted.Add Array(posA + 1, posB - posA - 1)
posA = InStr(posB + 1, Text, QUOTE, vbBinaryCompare)
Loop
' Restore quoted text
For Each vlu In quoted
Mid(ChangeCaseX, vlu(0), vlu(1)) = Mid$(Text, vlu(0), vlu(1))
Next
End If

End Function
```

Great minds think alike.<g> I was working up a similar approach to this (but it is slightly different in that it uses an array rather than a collection) when I got sidetracked helping a friend with a Windows problem. I finished with my friend and then finished up with my routine only to find you posted this similar solution. However, so my effort was not a total waste<g>, below is the solution I came up with (and, amazingly enough, I had called my function ChangeCaseX also).

Rick

```
Function ChangeCaseX(ByVal strString As String, _
bUpper As Boolean) As String
Dim X As Long
Dim Index As Long
Dim Position As Long
Dim Quotes() As Long
Position = InStr(strString, "")
If Position = 0 Then
If bUpper Then
strString = UCase$(strString)
Else
strString = LCase$(strString)
```

Re: fastest way to change case of string

Re: fastest way to change case of string

```
End If
Else
ReDim Quotes(Len(strString))
Quotes(0) = 1
Index = 1
Do While Position
Quotes(Index) = Position
Position = InStr(Position + 1, strString, "")
Index = Index + 1
Loop
Quotes(Index) = Len(strString)
For X = 0 To Index Step 2
If bUpper Then
Mid$(strString, Quotes(X) = UCase$(Mid$(strString, Quotes(X) _
, Quotes(X + 1) - Quotes(X) + 1))
Else
Mid$(strString, Quotes(X) = LCase$(Mid$(strString, Quotes(X) _
, Quotes(X + 1) - Quotes(X) + 1))
End If
Next
End If
ChangeCaseX = strString
End Function
```

.