```
' McCub -- A Just BASIC / Liberty BASIC Case Conversion utility
' It allows you to --
' change the case of Reserved words, such as FOR, IF, SCAN
' change case of Functions, such as CHR$(, DATE$(, EOF(, SIN(
' case conversion for text inside strings.

 On Error Goto [ErrorTrapper]

' Just BASIC's RESERVED WORDS list

Data AND, APPEND, AS, BEEP, BMPBUTTON, _
 BMPSAVE, BOOLEAN, BUTTON, BYREF, CALL, _
 CASE, CHECKBOX, CLOSE, CLS, COMBOBOX, _
 CONFIRM, DATA, DIALOG, DIM, DO, _
 DUMP, ELSE, END, ERROR, EXIT, _
 FIELD, FILEDIALOG, FILES, FOR, FUNCTION, _
 GET, GLOBAL, GOSUB, GOTO, GRAPHICBOX, _
 GRAPHICS, GROUPBOX, IF, INPUT, KILL, _
 LET, LINE, LISTBOX, LOADBMP, LONG, _
 LOOP, LPRINT, MAINWIN, MAPHANDLE, MENU, _
 NAME, NEXT, NOMAINWIN, NONE, NOTICE, _
 ON, ONCOMERROR, OR, OPEN, OUTPUT, _
 PLAYMIDI, PLAYWAVE, PRINT, PROMPT, PUT, _
 RADIOBUTTON, RANDOM, RANDOMIZE, READ, READJOYSTICK, _
 REDIM, "REM", RESTORE, RETURN, RUN, _
 SCAN, SELECT, STATICTEXT, STOP, STOPMIDI ,_
 SUB, TEXT, TEXTBOX, TEXTEDITOR, THEN, _
 TIMER, UNLOADBMP, UNTIL, WAIT, WINDOW, _
 WEND, WHILE, WORD, XOR

' Missing Just BASIC RESERVED WORDS:
Data BINARY, BYVAL, CALLFN, COLORDIALOG, LOCATE, _
 MOD, PRINTERDIALOG, SEEK, STEP, TO

' Liberty BASIC-only RESERVED WORDS
Data CALLBACK, CALLDLL, CURSOR, DLL, DOUBLE, _
 DWORD, FONTDIALOG, GETTRIM, MAPHANDLE, OUT, _
 PASSWORD, POPUPMENU, PTR, RESUME, SHORT, _
 SORT, STRUCT, STYLEBITS, TITLEBAR, TRACE, _
 ULONG, USHORT, VOID

' End of Data marker
Data "ZZZ"

' Just BASIC FUNCTIONS:
' Note that the opening parenthesis is part of the function name:
```

```
Data "ABS(", "ACS(", "ASC(", "ASN(", "ATN(", _
 "CHR$(", "COS(", "DATE$(", "EOF(", "EXP(", _
 "INPUT$(", "INSTR(", "INT(", "LEFT$(", "LEN(", _
 "LOF(", "LOG(", "LOWER$(", "MIDIPOS(", "MID$(", _
 "MKDIR(", "NOT(", "RIGHT$(", "RMDIR(", "RND(", _
 "SIN(", "SPACE$(", "SQR(", "STR$(", "TAB(", _
 "TAN(", "TIME$(", "TRIM$(", "TXCOUNT(", "UPPER$(", _
 "USING(", "VAL(", "WORD$("

' Missing FUNCTIONS:
Data "LOC("

' Liberty BASIC-only FUNCTIONS
Data "DECHEX$(", "EVAL(", "EVAL$(", "HBMP(", "HEXDEC(", "HWND(", _
 "INP(", "INPUTTO$(", "MAX(", "MIN(", "WINSTRING("

' End of Data marker
Data "ZZZ"

 Global basfileOpen, mainOpen
 Dim allwords$(10), option(10)

' Read keywords$
 Gosub [LoadWords]
 nkeywords = n
 Print "Keywords read = "; n
 allwords$(1) = keys$

' Read functions$
 Gosub [LoadWords]
 nfunctions = n
 Print "Functions read = "; n
 allwords$(2) = keys$

    opt1str$ = "No change,First letter,lowercase,UPPERCASE"
    opt2str$ = "No change,First letter,lowercase,UPPERCASE"
    opt3str$ = "No change,First letter,lowercase,UPPERCASE"
    For a = 1 To 4
        opt1ray$(a) = Word$(opt1str$,a,",")
        opt2ray$(a) = Word$(opt2str$,a,",")
        opt3ray$(a) = Word$(opt3str$,a,",")
    Next

    WindowWidth  = 365
    WindowHeight = 390
```

```
    Statictext #main.st1, "Path: None Selected", 10, 20, 300, 25
    Statictext #main.st2, "File: None Selected", 10, 50, 300, 25
    Statictext #main.st3,
"How would you like your BASIC text?", 50, 90, 300, 25
    Statictext #main.opt1, "Keywords: ", 20, 130, 80, 25
    Statictext #main.opt2, "Functions:", 20, 160, 80, 25
    Statictext #main.opt3, "Quoted text:", 20, 250,120, 25
    Combobox #main.keywords,  opt1ray$(), setopt1, 140,130,170,25
    Combobox #main.functions, opt2ray$(), setopt2, 140,160,170,25
    Combobox #main.strings,   opt3ray$(), setopt3, 140,250,170,25

    Button #main.getfile, " Choose File ", ChooseFile, UL, 20, 300
    Button #main.go, " GO ", McCubit, UL, 260, 300

    Open "McCub: A Code Case Utility" For Window As #main
    #main "trapclose [mainquit]"
    mainOpen = 1
    #main "font Arial 12"
    #main.st1 "!font Arial_Narrow 12"
    #main.st2 "!font Arial_Narrow 12"
    #main.keywords  "selectindex 1"
    #main.functions "selectindex 1"
    #main.strings   "selectindex 1"
    #main.getfile "!setfocus"
Wait

[mainquit]
    If basfileOpen=1 Then Close #basfile
    If mainOpen=1 Then Close #main
End
' ----- End of Main Program -----

' Gosub to read DATA statements into string
[LoadWords]
    lastone$ = "ZZZ"
    n = 0 : Read k$ : keys$ = " "
    While k$<>lastone$
        n = n + 1
        keys$  = keys$ + k$ + " "
        Read k$
    Wend
Return

Sub setopt1 handle$
    #main.keywords, "selectionindex? x"
    option(1) = x
```

```
End Sub

Sub setopt2 handle$
    #main.functions, "selectionindex? x"
    option(2) = x
End Sub

Sub setopt3 handle$
    #main.strings, "selectionindex? x"
    option(3) = x
End Sub


' Choose File sub
Sub ChooseFile handle$
    Filedialog "Open BASIC source code", "*.bas;*.BAS", myfile$
    If myfile$<>"" Then
        allwords$(0) = myfile$  ' store in global array
        x = Len(myfile$)   ' separate path and filename
        While x>0 And Mid$(myfile$,x,1)<>"\"
          x = x - 1
        Wend
        mypath$ = Mid$(myfile$, 1, x)
        myfile$ = Mid$(myfile$, x+1, 256)
        #main.st1 "Path: "+mypath$
        #main.st2 "File: "+myfile$
        Print "BAS file chosen: "; mypath$, myfile$
    End If
End Sub  ' ChooseFile

' Open & read source file.  Apply user's selections to text.
' Print to mainwin (or .new file)
Sub McCubit handle$
    quote$ = Chr$(34)
    Print "Keywords option  = "; opt1ray$(option(1))
    Print "Functions option = "; opt2ray$(option(2))
    Print "Quoted text option= "; opt3ray$(option(3))
    stopper$ = " ,:;()=<>+-*/"+Chr$(34)+Chr$(39)+Chr$(13)+Chr$(10)

    If allwords$(0)<>"" Then
        Open allwords$(0) For Input As #basfile
        Print allwords$(0); " opened."
        Print
"=================================================================
="
        While Eof(#basfile)=0
```

```
            Line Input #basfile, code$
            While Right$(code$,1)=" "    ' remove trailing whitespace
                code$ = Left$(code$,Len(code$)-1)
            Wend
            codePos = 0
' print code$
            newCode$ = ""
            While codePos<Len(code$)
                wrd$ = GetNextWord$(code$, codePos, stopper$)
                If Instr(allwords$(1), Upper$(" "+wrd$+" "), 1)>0 Then
                    Select Case option(1)
'if so convert it to users' choice
                        Case 2 : wrd$ = Capitalize$(wrd$)
' First letter cap
                        Case 3 : wrd$ = Lower$(wrd$)  ' lowercase
                        Case 4 : wrd$ = Upper$(wrd$)  ' UPPERCASE
                    End Select
                End If
                If Instr(allwords$(2), Upper$(" "+wrd$+"( "), 1)>0
Then
                    Select Case option(2)
'if so convert it to users' choice
                        Case 2 : wrd$ = Capitalize$(wrd$)
' First letter cap
                        Case 3 : wrd$ = Lower$(wrd$)  ' lowercase
                        Case 4 : wrd$ = Upper$(wrd$)  ' UPPERCASE
                  End Select
                End If
                If Right$(wrd$,1)=quote$ Then
                    Select Case option(3)
'if so convert it to users' choice
                        Case 2 : wrd$ = Capitalize$(wrd$)
' First letter cap
                        Case 3 : wrd$ = Lower$(wrd$)  ' lowercase
                        Case 4 : wrd$ = Upper$(wrd$)  ' UPPERCASE
                    End Select
                End If
' print wrd$
                newCode$ = newCode$ + wrd$ + Mid$(code$, codePos, 1)
            Wend
            Print newCode$
'           print #fout, newCode$
        Wend
        Close #basfile
'       close #fout
    End If
```

```
End Sub  ' McCubit

' code$ = line of source code to be examined.
' cpos  = current position in code$
' stopper$ = characters marking end of word
Function GetNextWord$(code$, Byref cpos, stopper$)

' if at start of comment ('), then skip to end of line
    If Mid$(code$,cpos,1)=Chr$(39) Then stopper$ = ""

' if at start of quote ("), then only stopper is closing "
    If Mid$(code$,cpos,1)=Chr$(34) Then stopper$ = Chr$(34)

    oldpos = cpos+1
    npos = Len(code$)  ' temporary setting of next stopper
    For a = 1 To Len(stopper$)
        s$ = Mid$(stopper$,a,1)
        p = Instr(code$, s$, oldpos)
        If p>0 And p<npos Then npos = p
    Next a
' print "next stopper = "; npos
    If npos=Len(code$) Then
        npos = npos+1
    Else
        If s$=Chr$(34) Then npos = npos+1
    End If

    cpos = npos
    GetNextWord$ = Mid$(code$, oldpos, npos-oldpos)
End Function  ' GetNextWord$

Function Capitalize$(w$)
    c1$ = Upper$(Left$(w$,1))
    c2$ = Lower$(Mid$(w$,2,Len(w$)))
    Capitalize$ = c1$ + c2$
End Function


Function FileExists(path$,file$)
    Dim info$(10,10)
    Files path$,file$, info$()
    FileExists = Val(info$(0, 0))  ' non zero is true
End Function


[ErrorTrapper]
```

```
    Print "Err    = "; Err
    Print "Err$   = "; Err$
Wait
```