

Convert BASIC-Source-Code to HTML with Power Basic

ConvertSourceCodeToHTML.bas

```
' ConvertSourceCodeToHTML.bas

' Program in Power Basic 10 for Windows XP and 7

#include "Listbox.inc"
#include "UserOwnModules.inc"

' Arrays
GLOBAL ListKeywordsPBstm() AS STRING
GLOBAL ListKeywordsPBops() AS STRING
GLOBAL ListKeywordsVbsStm() AS STRING
GLOBAL ListKeywordsHtaAtt() AS STRING
GLOBAL ListText() AS STRING, DisplayText() AS STRING

' Strings
GLOBAL Language AS STRING, PrevChar AS STRING, TAG2 AS STRING
GLOBAL ColTxt AS STRING, ColKey AS STRING, ColRem AS STRING
GLOBAL ColQum AS STRING, ColOps AS STRING, ColFms AS STRING
GLOBAL ColAtt AS STRING, ColTag AS STRING, ColNrs AS STRING

' Numerics
GLOBAL LineNr AS INTEGER
GLOBAL ScriptFlag AS INTEGER, ScriptFlag1 AS INTEGER ' Boolean

FUNCTION PBMAIN()
    DIM sTmp AS STRING
    sTmp = "Power Basic,HTML-Application,Keyword-Table,InsertCodeInWebsite"
    REDIM ListBoxItems(1 TO PARSECOUNT(sTmp,","))
    PARSE sTmp, ListBoxItems()
    ListboxTitle = "Convert Source Code to HTML" ' "Dialog1"
    ListboxLabelText = "Select a Function" ' "Label1"
    ListboxButtonText = "Start" ' "Button1"
    ShowDIALOG1 %HWNDDESKTOP ' Select Language Dialog Box
END FUNCTION
```

```

SUB Program(AnyText$) ' AnyText$ = Listbox-Text
' Definitions
DIM FileSpecKeywords_PBstm AS STRING
DIM FileSpecKeywords_PBops AS STRING
DIM FileSpecKeywords_VbsStm AS STRING
DIM AllSelectedFilesSpec AS STRING, FileSpecTextIn AS STRING
DIM FileSpecTextOut AS STRING, FileSpecTextOutCopy AS STRING
DIM sTmp AS STRING, sTmp2 AS STRING, sTmp3 AS STRING, xColor AS STRING
DIM CurrentPath AS STRING, CodeLine AS STRING
DIM docPath AS STRING, pgmPath AS STRING, keySpec AS STRING
DIM CommandLine AS STRING, BrowserSpec AS STRING
DIM HTMLtext AS STRING, HTMLtext1 AS STRING
DIM TaskNr AS WORD, i AS LONG, j AS LONG, HeadFootFlag AS WORD
CALL DeclareModuleVariables ' UserOwnModules activated
IF WindowsVersion = "XP" THEN
    docPath = "c:\Dokumente und Einstellungen\Erich\Eigene Dateien\"
    pgmPath = "c:\Programme\"
ELSEIF WindowsVersion = "7" THEN
    docPath = "c:\Users\Erich\Documents\"
    pgmPath = "C:\Program Files\"
ELSE
    MSGBOX "program not forseen for this Windows Version": EXIT SUB
END IF
FileSpecTextOutCopy = docPath & "html\SourceCode.htm"
BrowserSpec = pgmPath & "Internet Explorer\IEXPLORE.EXE"
Language = "": TAG2 = "</font>"
FOR i = 1 TO UBOUND(ListBoxItems) ' Rename ListBoxItems
    IF AnyText$ = ListBoxItems(i) THEN Language = PARSE$("PB,hta,table,insert", i)
NEXT
' ----- Program -----

DO ' this Do-Loop only runs through once,
' this is made only to can easy exit from algorithm
SELECT CASE Language
CASE "hta"
    keySpec = CurrentPath & "Keywords-"
    LoadFromDisk ListKeywordsVbsStm(), keySpec & "vbs-stm.dat"
    IF NOT (ARRAYATTR(ListKeywordsVbsStm(), 0)) THEN EXIT DO

```

```

AllSelectedFilesSpec = FileOpenSelect(CurrentPath, 1, "*.HTA;*.VBS")
IF AllSelectedFilesSpec = "" THEN MSGBOX "no file selected": EXIT DO
CASE "PB"
    keySpec = CurrentPath & "Keywords-" & Language
    LoadFromDisk ListKeywordsPBstm(), keySpec & "-stm.dat"
    IF NOT (ARRAYATTR(ListKeywordsPBstm(), 0)) THEN EXIT DO
    LoadFromDisk ListKeywordsPBops(), keySpec & "-ops.dat"
    IF NOT (ARRAYATTR(ListKeywordsPBops(), 0)) THEN EXIT DO
    AllSelectedFilesSpec = FileOpenSelect(CurrentPath, 1, "*.BAS;*.INC")
    IF AllSelectedFilesSpec = "" THEN MSGBOX "no file selected": EXIT DO
CASE "table"
    AllSelectedFilesSpec = FileOpenSelect(CurrentPath, 1, "*.DAT")
    IF AllSelectedFilesSpec = "" THEN MSGBOX "no file selected": EXIT DO
    i = MSGBOX("With Header and Footer ?", %MB_YESNO, "HTML-Table")
    IF i = %IDYES THEN HeadFootFlag = 1 ELSE HeadFootFlag = 0
CASE "insert"
    InsertCodeInWebsite: MSGBOX "job done"
END SELECT ' One or more files selected
FOR TaskNr = 1 TO PARSECOUNT(AllSelectedFilesSpec, CHR$(0)) - 1
    sTmp = AllSelectedFilesSpec
    sTmp2 = PARSE$(sTmp, CHR$(0), 1): sTmp3 = PARSE$(sTmp, CHR$(0), TaskNr + 1)
    FileSpecTextIn = sTmp2 & sTmp3
    FileSpecTextOut = sTmp2 & PATHNAME$(NAME, sTmp3) & ".htm"
    DO ' this Do-Loop only runs through once,
        ' this is made only to can easy exit from algorithm
        IF NOT (ISFILE(FileSpecTextIn)) THEN MSG(FileSpecTextIn): EXIT DO
        ReadArrayFromFile ListText(), FileSpecTextIn
        IF NOT (ARRAYATTR(ListText(), 0)) THEN MSG(FileSpecTextIn): EXIT DO
        IF Language = "table" THEN
            sTmp = RIGHT$(PATHNAME$(NAME, sTmp3), 3): xColor = "000000"
            IF sTmp = "stm" THEN xColor = "0000FF"
            IF sTmp = "ops" THEN xColor = "8000FF"
            HTMLtext = MakeTable(ListText(), sTmp3, xColor, 4, HeadFootFlag)
            HTMLtext1 = HTMLtext
            IF NOT (HeadFootFlag) THEN
                HTMLtext1 = MakeTable(ListText(), sTmp3, xColor, 4, 1)
            END IF
            REDIM ListText(1 TO PARSECOUNT(HTMLtext, $CRLF))

```

```

        PARSE HTMLtext,ListText(),$CRLF
        REDIM DisplayText(1 TO PARSECOUNT(HTMLtext1,$CRLF))
        PARSE HTMLtext1,DisplayText(),$CRLF
    ELSE
        CALL ConvertTextFromSourceCodeToHTML ' Conversion-Program-Call
    END IF
    ' Write Result on disk
    WriteArrayToFile ListText(), FileSpecTextOut
    sTmp = PATHNAME$(PATH, FileSpecTextIn)
    FileSpecTextOutCopy = sTmp & "\SourceCode.htm"
    WriteArrayToFile ListText(), FileSpecTextOutCopy
    IF TaskNr > 1 THEN EXIT DO
    ' Display Result in Browser
    IF Language = "table" THEN
        sTmp = sTmp2 & "tmp.htm"
        WriteArrayToFile DisplayText(), sTmp
    ELSE
        sTmp = FileSpecTextOut
    END IF: CommandLine = BrowserSpec & " " & sTmp
    SHELL(CommandLine): EXIT DO
LOOP
NEXT: EXIT DO
LOOP
END SUB

```

' ----- Procedures -----

```

SUB ConvertTextFromSourceCodeToHTML
    DIM htaCode AS STRING, LTxt$, SCR1$, SCR2$
    SELECT CASE Language ' Color-Definition
    CASE "PB" ' ----- PB-Code -----
        AssignColors
        FOR LineNr = 1 TO UBOUND(ListText)
            ConvertCodeLine
        NEXT
    CASE "hta" ' ----- hta-Code -----
        ScriptFlag = ISFALSE 1: htaCode = ""
        SCR1$ = "<" & "SCRIPT Language=""VBScript"" & ">"
    
```

```

SCR2$ = "<" & "/SCRIPT" & ">"
FOR LineNr = 1 TO UBOUND(ListText)
    LTxt$ = ListText(LineNr)
    IF NOT(ScriptFlag) THEN
        AssignColors
        IF INSTR(LTxt$, SCR1$) <> 0 THEN ScriptFlag = ISTRUE 1
    ELSE
        AssignColors
        IF INSTR(LTxt$, SCR2$) <> 0 THEN ScriptFlag = ISFALSE 1
    END IF
    IF NOT(ScriptFlag) THEN
        IF htaCode <> "" THEN htaCode = htaCode & $CRLF
        htaCode = htaCode & LTxt$
    ELSE
        IF INSTR(LTxt$, SCR1$) <> 0 THEN
            IF htaCode <> "" THEN htaCode = htaCode & $CRLF
            htaCode = htaCode & LTxt$
            htaCode = ConvertHtaCode(htaCode)
            WriteBackHtaText LineNr, htaCode
        ELSE
            ConvertCodeLine
        END IF
    END IF
NEXT
htaCode = ConvertHtaCode(htaCode)
WriteBackHtaText UBOUND(ListText), htaCode
END SELECT
ListText(1) = "<font face=""Courier New"" SIZE=""2"">" & ListText(1)
ListText(UBOUND(ListText)) = ListText(UBOUND(ListText)) & TAG2
END SUB

SUB ConvertCodeLine
    DIM OldLine AS STRING, NewLine AS STRING, CurCol AS STRING
    DIM sTmp AS STRING, sTmp2 AS STRING, i AS INTEGER, j AS INTEGER
    DIM HiLite AS STRING, CurColor AS STRING
    OldLine = ListText(LineNr): NewLine = ""
    IF TRIM$(OldLine) = "" THEN
        ListText(LineNr) = ""
    
```

```

ELSE
  IF UCASE$(LEFT$(TRIM$(OldLine),4)) = "REM " THEN ' RemarkCodeLine
    NewLine = enTAG(OldLine, ColRem)
  ELSE
    ' Word-Recognition
    IF Language = "PB" THEN
      sTmp2 = "#PBFORMS,ShowDIALOG"
      FOR i = 1 TO PARSECOUNT(sTmp2, ",")
        sTmp = PARSE$(sTmp2, ",", i)
        IF LEFT$(OldLine, LEN(sTmp)) = sTmp THEN
          NewLine = enTAG(OldLine, ColFms)
          ListText(LineNr) = NewLine: EXIT SUB
        END IF
      NEXT
    END IF: PrevChar = ""
    DO WHILE LEN(OldLine)
      ' Char$- and Word-Recognition, Byte-wise detect
      SELECT CASE LEFT$(OldLine, 1)
      CASE ""
        ' 1234567i 1i
        ' "....." ""
        i = INSTR(2, OldLine, "")
        IF i = 0 THEN i = LEN(OldLine)
        sTmp = LEFT$(OldLine, i): sTmp2 = ""
        IF i > 1 THEN sTmp2 = MID$(sTmp, 2, i-2)
        sTmp2 = ReplaceSpecChars(sTmp2)
        sTmp = "" & sTmp2 & ""
        PrevChar = MID$(OldLine, i, 1)
        NewLine = NewLine & enTAG(sTmp, ColQum)
        OldLine = MID$(OldLine, i+1)
      CASE " "
        sTmp = ReplaceSpecChars(OldLine)
        NewLine = NewLine & enTAG(sTmp, ColRem)
        EXIT DO
      CASE ELSE
        sTmp2 = RecogniseHiLite(PrevChar, OldLine)
        IF sTmp2 <> "" THEN
          HiLite = PARSE$(sTmp2, ",", 1)

```

```

        CurColor = PARSE$(sTmp2, ",", 2)
        NewLine = NewLine & enTAG(HiLite, CurColor)
        OldLine = MID$(OldLine, LEN(HiLite)+1)
    ELSE
        sTmp = LEFT$(OldLine, 1)
        i = INSTR($TAB & " <>äÄöÖüÛß", sTmp)
        IF i > 0 THEN sTmp = ReplaceSpecChars(sTmp)
        PrevChar = LEFT$(OldLine, 1)
        NewLine = NewLine & sTmp
        OldLine = MID$(OldLine, 2)
    END IF
END SELECT
LOOP
END IF
END IF
IF LineNr < UBOUND(ListText) THEN NewLine = NewLine & "<BR>"
ListText(LineNr) = NewLine
END SUB

SUB AssignColors
' If Language = PB then AnyFlag irrelevant
ColTxt = "000000": ColRem = "007F00"
SELECT CASE Language
CASE "PB"
    ColKey = "0000C0": ColOps = "8000FF" ' Keywords and Operators
    ColFms = "C06400": ColQum = "C020C0" ' Forms and Quotation marks
CASE "hta"
    ColKey = "0000FF": ColNrs = "A52A2A" ' Keywords and Numbers
    ColTag = "A52A2A": ColAtt = "FF0000" ' Tagnames and Attributes
    IF NOT(ScriptFlag) THEN ColQum = "408080" ELSE ColQum = "808080"
END SELECT
END SUB

SUB LoadFromDisk(BYREF AnyList() AS STRING, AnyFileSpec AS STRING)
DIM AnyList() AS STRING
IF NOT(ISFILE(AnyFileSpec)) THEN MSG(AnyFileSpec): EXIT SUB
ReadArrayFromFile AnyList(), AnyFileSpec
IF NOT(ARRAYATTR(AnyList(), 0)) THEN MSG(AnyFileSpec): EXIT SUB

```

```

WriteArrayToFile AnyList(), AnyFileSpec & ".bak"
ARRAY SORT AnyList(), COLLATE UCASE, ASCEND
RemoveDoubleItems AnyList()
WriteArrayToFile AnyList(), AnyFileSpec
SortForWordLenght AnyList()
END SUB

SUB MSG(AnyFileSpec AS STRING)
DIM MsgTmp$: MsgTmp$ = AnyFileSpec
IF MID$(AnyFileSpec,2,2) = ":\ " THEN
    MsgTmp$ = "file "" & MID$(AnyFileSpec,_
        INSTR(-1, AnyFileSpec, "\")+1) & "" missing"
END IF: MSGBOX MsgTmp$, "Error"
END SUB

SUB SortForWordLenght(BYREF AnyArray() AS STRING)
DIM sTmp AS STRING, i AS LONG, Flag AS INTEGER: Flag = ISTRUE 1
WHILE Flag
    Flag = ISFALSE 1
    FOR i = 1 TO UBOUND(AnyArray)-1
        IF LEN(AnyArray(i)) < LEN(AnyArray(i+1)) THEN
            sTmp = AnyArray(i): AnyArray(i) = AnyArray(i+1)
            AnyArray(i+1) = sTmp: Flag = ISTRUE 1
        END IF
    NEXT
WEND
END SUB

FUNCTION ConvertHtaCode(AnyCode AS STRING) AS STRING
' Converts chain of multiple lines
DIM TAGdyed AS STRING, sTmp AS STRING, j AS INTEGER, k AS INTEGER
TAGdyed = ""
DO WHILE LEN(AnyCode): j = INSTR(AnyCode, "<"): k = INSTR(AnyCode, ">")
    IF j = 0 OR k = 0 OR j > k THEN
        sTmp = AnyCode
        REPLACE "<" WITH "&lt;" IN sTmp
        REPLACE ">" WITH "&gt;" IN sTmp
        TAGdyed = enTAG(sTmp, ColTxt): EXIT DO
    END IF
    AnyCode = MID$(AnyCode, j+1, k-j)
LOOP
END FUNCTION

```



```

ELSE
    TAGdyed = TAGdyed & DyeTAG(LEFT$(AnyCode,k))
    AnyCode = MID$(AnyCode,k+1)
END IF
LOOP: ConvertHtaCode = TAGdyed
END FUNCTION

FUNCTION DyeTAG(AnyString AS STRING) AS STRING ' One TAG
' ".....<..." ".....>..." ".....>.....<..." ".....<.....>..."
DIM TAG AS STRING, TAGtxt AS STRING, TAGname AS STRING, TAGattr AS STRING
DIM TAGdyed AS STRING, sTmp AS STRING, sTmp2 AS STRING
DIM i AS INTEGER, j AS INTEGER, k AS INTEGER
TAG = "": TAGtxt = "": TAGname = "": TAGattr = "": TAGdyed = ""
IF LEN(AnyString) THEN
    j = INSTR(AnyString,"<"): k = INSTR(AnyString,">")
    IF j > 1 THEN
        TAGdyed = LEFT$(AnyString, j-1)
        REPLACE " " WITH "&nbsp;" IN TAGdyed
        AnyString = MID$(AnyString,j): j = 1: k = INSTR(AnyString,">")
    END IF: TAG = LEFT$(AnyString,k)
    IF TAG <> "" THEN
        i = INSTR(TAG," "): IF i = 0 THEN i = LEN(TAG)
        TAGname = MID$(TAG,2,i-2): TAGattr = MID$(TAG, i, LEN(TAG) - i)
        IF MID$(TAG,2,3) = "!--" THEN
            TAGdyed = TAGdyed & enTAG("&lt;" & TAGname & _
            TAGattr & "&gt;", ColRem)
        ELSE
            TAGdyed = TAGdyed & enTAG("&lt;",ColKey) & enTAG(TAGname,ColTag)
            ' Dye Attributes for example size="3" name="mylistbox"
            IF LEN(TAGattr) THEN
                sTmp2 = "": i = 1
                DO
                    j = INSTR(i, TAGattr,"="): IF j = 0 THEN EXIT DO
                    sTmp = MID$(TAGattr,i,j-i)
                    REPLACE $TAB WITH " " IN sTmp
                    REPLACE " " WITH "&nbsp;" IN sTmp
                    sTmp2 = sTmp2 & enTAG(sTmp,ColAtt)
                    i = j: j = INSTR(j,TAGattr,""): IF j = 0 THEN EXIT DO
                DO
            END IF
        END IF
    END IF
END IF

```

```

                j = INSTR(j+1,TAGattr,"'"): IF j = 0 THEN EXIT DO
                sTmp2 = sTmp2 & enTAG(MID$(TAGattr,i,j-i+1),ColQum)
                i = j + 1
            LOOP
        END IF: TAGdyed = TAGdyed & sTmp2 & enTAG(">",ColKey)
    END IF
END IF
END IF: DyeTAG = TAGdyed
END FUNCTION

SUB WriteBackHtaText(AnyLineNr AS INTEGER, BYREF AnyCode AS STRING)
    DIM i AS INTEGER, j AS INTEGER, pcount AS INTEGER
    pcount = PARSECOUNT(AnyCode,$CRLF)
    FOR i = 1 TO pcount
        j = AnyLineNr + i - pcount
        ListText(j) = PARSE$(AnyCode,$CRLF,i)
        IF j < UBOUND(ListText) THEN ListText(j) = ListText(j) & "<BR>"
    NEXT: AnyCode = ""
END SUB

FUNCTION ReplaceSpecChars(AnyString AS STRING) AS STRING
    DIM SpecChars AS STRING, Char AS STRING, Chars AS STRING
    DIM i AS INTEGER, j AS INTEGER, sTmp AS STRING : sTmp = ""
    ' insert HTML-code for special characters
    ' & must be on first place
    ' at the end ";" must be missing
    REPLACE $TAB WITH " " IN AnyString
    Chars = "&äÄöÜß ""<>$"
    SpecChars = "&amp;&auml;&Auml;&ouml;&Ouml;&uuml;" & _
        "&Uuml;&szlig;&nbsp;&quot;&lt;&gt;&sect"
    FOR i = 1 TO LEN(AnyString)
        Char = MID$(AnyString,i,1): j = INSTR(Chars,Char)
        IF j > 0 THEN Char = PARSE$(SpecChars,";",j) & ";"
        sTmp = sTmp & Char
    NEXT: ReplaceSpecChars = sTmp
END FUNCTION

FUNCTION RecogniseHiLite(AnyPrevChar AS STRING, AnyString AS STRING) AS STRING

```

```

LOCAL KeyWord AS STRING, CurColor AS STRING
LOCAL j AS WORD, sTmp AS STRING: FUNCTION = ""
SELECT CASE Language
CASE "PB"
    KeyWord = RecogniseWord(AnyPrevChar, AnyString, ListKeywordsPBstm(), 3)
    IF KeyWord <> "" THEN
        ' Exceptions
        sTmp = "%IDC_,%IDD_"
        FOR j = 1 TO PARSECOUNT(sTmp)
            IF UCASE$(KeyWord) = UCASE$(PARSE$(sTmp, ",", j)) THEN
                CurColor = ColTxt: EXIT SELECT
            END IF
        NEXT: CurColor = ColKey: EXIT SELECT
    END IF
    KeyWord = RecogniseWord(AnyPrevChar, AnyString, ListKeywordsPBops(), 3)
    IF KeyWord <> "" THEN CurColor = ColOps
CASE "hta"
    KeyWord = IsNr(AnyPrevChar, AnyString)
    IF KeyWord <> "" THEN CurColor = ColNrs: EXIT SELECT
    KeyWord = RecogniseWord(AnyPrevChar, AnyString, ListKeywordsVbsStm(), 2)
    IF KeyWord <> "" THEN CurColor = ColOps
END SELECT: IF KeyWord <> "" THEN FUNCTION = KeyWord & "," & CurColor
END FUNCTION

FUNCTION IsNr(AnyPrevChar AS STRING, AnyString AS STRING) AS STRING
    DIM sTmp$, Chrs$, C0$, C$, C1$, i AS INTEGER, Number AS INTEGER
    C0$ = AnyPrevChar: Number = ISTRUE 1: IsNr = ""
    Chrs$ = $TAB & " .,;+-*\/()=<>"
    DO
        IF NOT(InStrAny(C0$, ",", Chrs$)) THEN EXIT DO
        FOR i = 1 TO LEN(AnyString)
            sTmp$ = MID$(AnyString,i,2): C$ = LEFT$(sTmp$,1)
            IF NOT(InStrAny(C$, "0-9", ".")) THEN Number = ISFALSE 1: EXIT FOR
        NEXT: i = i - 1: IF Number THEN i = LEN(AnyString)
        C1$ = MID$(AnyString,i+1,1)
        IF NOT(InStrAny(C1$, ",", Chrs$)) THEN EXIT DO
        IsNr = LEFT$(AnyString,i): EXIT DO
    LOOP

```

END FUNCTION

```
FUNCTION RecogniseWord(AnyPrevChar AS STRING, AnyString AS STRING, _
    AnyListOfKeyWords() AS STRING, AnyMode AS WORD) AS STRING
    ' Mode: 0 = charsize of AnyString, 1 = CharSize of AnyWord, 2 = CharSize UCase(AnyWord)
    DIM i AS WORD, L AS WORD, Ubd AS WORD: Ubd = UBOUND(AnyListOfKeyWords)
    DIM sTmp AS STRING, sTmp1 AS STRING, sTmp2 AS STRING
    DIM KeyWord AS STRING: FUNCTION = ""
    sTmp1 = "A-z,0-9" & $LF & "!""$%#_": sTmp2 = "A-z,0-9" & $LF & "$$.äöüÄÖÜß_"
    IF ISFALSE (AnyMode AND 3) THEN EXIT FUNCTION
    IF AnyString = "" THEN EXIT FUNCTION
    FOR i = 1 TO Ubd ' KeywordArray
        KeyWord = AnyListOfKeyWords(i): L = LEN(KeyWord)
        IF L <= LEN(AnyString) THEN
            IF UCASE$(LEFT$(AnyString,L)) = UCASE$(LEFT$(KeyWord,L)) THEN
                sTmp = LEFT$(KeyWord,1)
                IF INSTR("()",sTmp) = 0 THEN ' all brackets violet in PB
                    IF InStrAny(AnyPrevChar, PARSE$(sTmp1,$LF,1), _
                        PARSE$(sTmp1,$LF,2)) THEN EXIT FUNCTION
                END IF
                ' CharSize
                sTmp = LEFT$(AnyString,L) & "," & LEFT$(KeyWord,L) _
                    & "," & UCASE$(LEFT$(KeyWord,L))
                FUNCTION = PARSE$(sTmp, ",", AnyMode)
                IF L < LEN(AnyString) THEN
                    sTmp = LEFT$(KeyWord,1)
                    IF INSTR("()",sTmp) = 0 THEN ' all brackets violet in PB
                        IF InStrAny(MID$(AnyString,L+1,1), PARSE$(sTmp2,$LF,1), _
                            PARSE$(sTmp2,$LF,2)) THEN FUNCTION = ""
                    END IF
                END IF: EXIT FUNCTION
            END IF
        END IF
    NEXT
END FUNCTION
```

```
FUNCTION MakeTable(AnyList() AS STRING, AnyTitle AS STRING, AnyColor AS STRING, _
    NrCol AS WORD, HeadAndFoot AS INTEGER) AS STRING
```

```

LOCAL Head AS STRING, TableTxt AS STRING, Foot AS STRING
LOCAL LineTxt AS STRING, sTmp AS STRING, sTmp2 AS STRING
LOCAL sp AS STRING, sp1 AS STRING
LOCAL TM AS STRING ' Top Margin
LOCAL BM AS STRING ' Bottom Margin
LOCAL LM AS STRING ' Left Margin
LOCAL RM AS STRING ' Right Margin
LOCAL Ubd AS WORD, rw AS WORD, clm AS WORD, ListPos AS WORD
LOCAL HeaderAndFooter AS INTEGER ' used as flag

TM = "40": BM = "40": LM = "40": RM = "40"

HeaderAndFooter = ISTRUE HeadAndFoot
sp = TRIM$(STR$(2*NrCol)): sp1 = TRIM$(STR$(2*NrCol+1))
Head = "": Foot = ""

IF HeaderAndFooter THEN
    Head = "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" " " & _
        ""http://www.w3.org/TR/html4/loose.dtd"">" & $CRLF & _
        "<html><head><title>Tablemaking</title>" & $CRLF & _
        "<meta NAME=""author"" CONTENT=""Erich"">" & $CRLF & _
        "<meta NAME=""editor"" CONTENT=""html-editor phase 5"">" & _
        "</head>" & $CRLF & _
        "<body text=""#000000"" bgcolor=""#DDDDDD"" link=""#FF0000"" " & _
        "alink=""#FF0000"" vlink=""#FF0000"">" & $CRLF
    Foot = "</body></html> "
END IF
TableTxt = $CRLF & "<span style=""font-family:Courier New;font-size:10pt;color:#" & _
AnyColor & ";"">" & $CRLF & _
"<table border=""0"" cellpadding=""0"" cellspacing=""0"" " & _
"align=""center"" bgcolor=""#FFFFFF"">" & $CRLF & _
"<tr><td colspan="" & sp1 & "" height="" & TM & ""></tr>" & $CRLF & _
"<tr><td><td colspan="" & sp & ""><B>" & AnyTitle & _
"</B><BR><BR></td></tr>"

Ubd = UBOUND(AnyList)
FOR rw = 0 TO INT((Ubd)/NrCol)-1
    sTmp = "<td>": sTmp2 = "<td>"

```

```

IF rw = 0 THEN
    sTmp = "<td width=""120"">": sTmp2 = "<td width="" & RM & "">"
END IF: LineTxt = ""
FOR clm = 0 TO NrCol-1
    ListPos= clm*INT((Ubd-1)/NrCol) + rw + 1
    LineTxt = LineTxt & $CRLF & sTmp & AnyList(ListPos) & _
    "</td>" & sTmp2 & "</td>"
NEXT
sTmp = "<td>": IF rw = 0 THEN sTmp = "<td width="" & LM & "">"
LineTxt = sTmp & LineTxt & "</td>"
TableTxt = TableTxt & $CRLF & "<tr>" & LineTxt & "</tr>"
NEXT
TableTxt = TableTxt & "<tr><td height="" & BM & " colspan="" & sp1 & _
""></tr>" & $CRLF & "</table></span>" & $CRLF & $CRLF
FUNCTION = Head & TableTxt & Foot
END FUNCTION

SUB InsertCodeInWebsite
LOCAL FileSpec1 AS STRING, FileSpec2 AS STRING, FileSpecBak AS STRING
LOCAL HTMLText1 AS STRING, HTMLText2 AS STRING
LOCAL TxtMark1 AS STRING, TxtMark2 AS STRING
LOCAL StartPath AS STRING
LOCAL sTmp AS STRING, i AS LONG, j AS LONG
' Definitions
StartPath = "d:\WebsitePath\"
TxtMark1 = "<!-- Beginn des Tabelleninhaltes -->"
TxtMark2 = "<!-- Ende des Tabelleninhaltes -->"
' Program
IF NOT(ISFOLDER(StartPath)) THEN MSGBOX "Homepagefolder does not exist": EXIT SUB
sTmp = FileOpenSelect(CurrentPath, 0, "*.HTM;*.HTML")
IF sTmp = "" THEN MSGBOX "no file selected": EXIT SUB
FileSpec1 = PARSE$(sTmp,CHR$(0),1) & PARSE$(sTmp,CHR$(0),2)
sTmp = FileOpenSelect(StartPath, 0, "*.HTM;*.HTML")
IF sTmp = "" THEN MSGBOX "no file selected": EXIT SUB
FileSpec2 = PARSE$(sTmp,CHR$(0),1) & PARSE$(sTmp,CHR$(0),2)
ReadTextFromFile HTMLText1, FileSpec1
ReadTextFromFile HTMLText2, FileSpec2
' make a bak-file of website

```

```

FileSpecBak = PATHNAME$(PATH, FileSpec2) & PATHNAME$(NAME, FileSpec2) & _
"-bak" & PATHNAME$(EXTN, FileSpec2)
WriteTextToFile HTMLText2, FileSpecBak
' find markers and insert code
i = INSTR(HTMLText2, TxtMark1): sTmp = "anything wrong with textmark"
IF i = 0 THEN MSGBOX sTmp: EXIT SUB
i = i + LEN(TxtMark1) - 1: j = INSTR(i + 1, HTMLText2, TxtMark2)
IF j = 0 OR j <= i THEN MSGBOX sTmp: EXIT SUB
HTMLText2 = LEFT$(HTMLText2,i) & $CRLF & "<BR>" & _
HTMLText1 & "<BR>" & $CRLF & MID$(HTMLText2,j)
WriteTextToFile HTMLText2, FileSpec2
END SUB

```

Listbox.inc

```
#PBFORMS CREATED V1.52 '-----
' The first line in this file is a PB/Forms metastatement.
' It should ALWAYS be the first line of the file. Other
' PB/Forms metastatements are placed at the beginning and
' end of "Named Blocks" of code that should be edited
' with PBForms only. Do not manually edit or delete these
' metastatements or PB/Forms will not be able to reread
' the file correctly. See the PB/Forms documentation for
' more information.
' Named blocks begin like this:      #PBFORMS BEGIN ...
' Named blocks end like this:        #PBFORMS END ...
' Other PB/Forms metastatements such as:
'      #PBFORMS DECLARATIONS
' are used by PB/Forms to insert additional code.
' Feel free to make changes anywhere else in the file.
'-----

#COMPILE EXE
#DIM ALL

GLOBAL ListBoxItems() AS STRING
GLOBAL ListboxTitle AS STRING
GLOBAL ListboxLabelText AS STRING
GLOBAL ListboxButtonText AS STRING

'-----
'      ** Includes **
'-----

#PBFORMS BEGIN INCLUDES #IF NOT %DEF(%WINAPI)
      #INCLUDE "WIN32API.INC"
#ENDIF
#PBFORMS END INCLUDES '-----

'-----
'      ** Constants **
'-----
```



```
#PBFORMS BEGIN CONSTANTS %IDD_DIALOG1 = 101
%IDC_LABEL1 = 1001
%IDC_LISTBOX1 = 1002
%IDC_BUTTON1 = 1003
#PBFORMS END CONSTANTS '-----
```

```
'-----
' ** Declarations **
'-----
```

```
DECLARE CALLBACK FUNCTION ShowDIALOG1Proc()
DECLARE FUNCTION SampleListBox(BYVAL HDLG AS DWORD, BYVAL IID AS LONG, BYVAL _
    lCount AS LONG) AS LONG
DECLARE FUNCTION ShowDIALOG1(BYVAL hParent AS DWORD) AS LONG
#PBFORMS DECLARATIONS '-----
```

```
'-----
' ** Main Application Entry Point **
'-----
'FUNCTION PBMAIN()
'    ShowDIALOG1 %HWND_DESKTOP
'END FUNCTION
'-----
```

```
'-----
' ** Callbacks **
'-----
```

```
CALLBACK FUNCTION ShowDIALOG1Proc()
    DIM TXT$
    SELECT CASE AS LONG CBMSG
        CASE %WM_INITDIALOG
            ' Initialization handler
        CASE %WM_NCACTIVATE
            STATIC hWndSaveFocus AS DWORD
            IF ISFALSE CBWPARAM THEN
                ' Save control focus
                hWndSaveFocus = GetFocus()
            ELSEIF hWndSaveFocus THEN
                ' Restore control focus
```

```

        SetFocus(hWndSaveFocus)
        hWndSaveFocus = 0
    END IF
CASE %WM_COMMAND
    ' Process control notifications
    SELECT CASE AS LONG CBCTL
        CASE %IDC_LABEL1
        CASE %IDC_LISTBOX1
            IF CBCTLMMSG = %LBN_DBLCLK THEN
                LISTBOX GET TEXT CBHNDL, %IDC_LISTBOX1 TO TXT$
                CALL Program(TXT$)
                DIALOG END CBHNDL
            END IF
        CASE %IDC_BUTTON1
            IF CBCTLMMSG = %BN_CLICKED OR CBCTLMMSG = 1 THEN
                LISTBOX GET TEXT CBHNDL, %IDC_LISTBOX1 TO TXT$
                IF TXT$ = "" THEN
                    MSGBOX "no item selected"
                ELSE
                    CALL Program(TXT$)
                    DIALOG END CBHNDL
                END IF
            END IF
        END SELECT
    END SELECT
END FUNCTION

' -----

' ** Sample Code **
' -----

FUNCTION SampleListBox(BYVAL HDLG AS DWORD, BYVAL IID AS LONG, BYVAL lCount _
    AS LONG) AS LONG
    LOCAL i AS LONG

    FOR i = 1 TO lCount
        LISTBOX ADD HDLG, IID, ListBoxItems(i)
    NEXT i

```

```

END FUNCTION
'-----

'-----
'  ** Dialogs **
'-----

FUNCTION ShowDIALOG1(BYVAL hParent AS DWORD) AS LONG
    LOCAL lRslt AS LONG

#PBFORMS BEGIN DIALOG %IDD_DIALOG1-->
    LOCAL HDLG AS DWORD

    DIALOG NEW hParent, ListboxTitle, 303, 164, 154, 140, %WS_POPUP OR _
        %WS_BORDER OR %WS_DLGFRAME OR %WS_SYSMENU OR %WS_CLIPSIBLINGS OR _
        %WS_VISIBLE OR %DS_MODALFRAME OR %DS_3DLOOK OR %DS_NOFAILCREATE OR _
        %DS_SETFONT, %WS_EX_CONTROLPARENT OR %WS_EX_LEFT OR _
        %WS_EX_LTRREADING OR %WS_EX_RIGHTSCROLLBAR, TO HDLG
    CONTROL ADD LABEL, HDLG, %IDC_LABEL1, ListboxLabelText, 30, 10, 100, 10, _
        %WS_CHILD OR %WS_VISIBLE OR %WS_BORDER OR %SS_CENTER OR %SS_SUNKEN, _
        %WS_EX_LEFT OR %WS_EX_LTRREADING
    CONTROL SET COLOR HDLG, %IDC_LABEL1, -1, %WHITE
    CONTROL ADD LISTBOX, HDLG, %IDC_LISTBOX1, , 30, 25, 100, 80, %WS_CHILD OR _
        %WS_VISIBLE OR %WS_TABSTOP OR %WS_VSCROLL OR %LBS_NOTIFY, _
        %WS_EX_CLIENTEDGE OR %WS_EX_LEFT OR %WS_EX_LTRREADING OR _
        %WS_EX_RIGHTSCROLLBAR
    CONTROL ADD BUTTON, HDLG, %IDC_BUTTON1, ListboxButtonText, 55, 110, 50, 15
#PBFORMS END DIALOG
    SampleListBox HDLG, %IDC_LISTBOX1, UBOUND(ListBoxItems)

    DIALOG SHOW MODAL HDLG, CALL ShowDIALOG1Proc TO lRslt

#PBFORMS BEGIN CLEANUP %IDD_DIALOG1 #PBFORMS END CLEANUP
    FUNCTION = lRslt
END FUNCTION
'-----

```

UserOwnModules.inc

```
' UserOwnModules.inc

#include ONCE "WIN32API.INC"
#include ONCE "Operating_System.inc"

DECLARE FUNCTION GetDriveLetterFromDriveName (STRING) AS STRING
DECLARE FUNCTION FileExists (STRING) AS INTEGER
DECLARE FUNCTION FolderExists (STRING) AS INTEGER
DECLARE FUNCTION FolderEmpty (STRING) AS INTEGER
GLOBAL Up, Down AS INTEGER
GLOBAL TmpText_ReadIn AS STRING
GLOBAL CurrentPath AS STRING, CurrentFileName AS STRING
GLOBAL DriveList () AS STRING
GLOBAL WindowsVersion AS STRING

SUB DeclareModuleVariables
    Up = 1: Down = -1
    CurrentPath = EXE.PATH$
    CurrentFileName = EXE.NAME$
    Operating_System_Os Os 'Get Operating System
    WindowsVersion = PARSE$(Os.@Version_Desc, " ", 2)
END SUB

' ----- Dialog Boxes -----

FUNCTION FolderSelect (StartFolder AS STRING) AS STRING
    ' DISPLAY BROWSE [hParent], [xpos&], [ypos&], title$, start$, flags& TO folder$
    ' DESKTOP GET SIZE TO ncWidth&, ncHeight&
    DIM folder$
    DISPLAY BROWSE %HWND_DESKTOP, 300, 200, _
    "FolderSelect", _ ' Title
    StartFolder, _ ' starting path
    %BIF_EDITBOX TO folder$
    FUNCTION = folder$
END FUNCTION
```

```

FUNCTION FileOpenSelect(StartFolder AS STRING, _
    MultiSelect AS WORD, _
    Filter AS STRING) AS STRING
    ' DISPLAY OPENFILE [hParent], [xpos&], [ypos&], title$, folder$, filter$, _
    ' start$, defextn$, flags& TO filevar$ [,countvar&]
    ' DESKTOP GET SIZE TO ncWidth&, ncHeight&
    ' Filter = "*.BAS;*.INC;*.DAT;*.HTA;*.VBS"
    DIM countvar&, filevar$, start$, sTmp AS STRING, i AS WORD
    IF MultiSelect <> 0 THEN MultiSelect = %OFN_ALLOWMULTISELECT
    DISPLAY OPENFILE %HWND_DESKTOP, 300, 200, _
    "FileOpenSelect", _           ' Title
    StartFolder , _              ' starting folder
    CHR$("BASIC", 0, Filter, 0), _ ' filter
    start$, _                   ' starting filename
    "", MultiSelect + %OFN_ENABLESIZING + _
    %OFN_FILEMUSTEXIST + %OFN_NOVALIDATE + _
    %OFN_PATHMUSTEXIST TO filevar$, countvar&
    IF filevar$ = "" THEN FUNCTION = "": EXIT FUNCTION
    IF RIGHT$(filevar$,1) = CHR$(0) THEN _
        filevar$ = LEFT$(filevar$,LEN(filevar$)-1)           ' Remove terminator chr$(0)
    IF countvar& = 1 THEN
        sTmp = PATHNAME$(PATH, filevar$)                     ' Split Path and Filename
        IF RIGHT$(sTmp,1) = "\" THEN sTmp = LEFT$(sTmp, LEN(sTmp)-1) ' Remove \
        filevar$ = sTmp & CHR$(0) & PATHNAME$(NAMEX, filevar$) ' Modified filevar$
    END IF: sTmp = PARSE$(filevar$,CHR$(0),1) & "\"
    FOR i = 1 TO countvar&
        sTmp = sTmp & CHR$(0) & PARSE$(filevar$,CHR$(0),i+1)
    NEXT : FUNCTION = sTmp
END FUNCTION

FUNCTION FileSaveSelect(StartFolder AS STRING) AS STRING
    ' DISPLAY SAVEFILE [hParent], [xpos&], [ypos&], title$, folder$, filter$, _
    ' start$, defextn$, flags& TO filevar$ [,countvar&]
    ' DESKTOP GET SIZE TO ncWidth&, ncHeight&
    DIM countvar&, filevar$, start$
    DISPLAY SAVEFILE %HWND_DESKTOP, 300, 200, _
    "FileSaveSelect", _           ' Title
    StartFolder , _              ' starting folder

```

```

    CHR$("BASIC", 0, "*.BAS;*.INC;*.BAK;*.htm", 0), _ ' filter
    start$, _ ' starting filename
    "", %OFN_OVERWRITEPROMPT + _
    %OFN_ENABLESIZING TO filevar$, countvar&
    FUNCTION = filevar$
END FUNCTION

' ----- Storage -----

SUB WriteTextToFile(AnyText AS STRING, BYVAL AnyFileSpec AS STRING)
    OPEN AnyFileSpec FOR OUTPUT AS #1
        PRINT #1, AnyText;
    CLOSE #1
END SUB

SUB ReadTextFromFile(BYREF AnyText AS STRING, BYVAL AnyFileSpec AS STRING)
    DIM sTmp$: AnyText = ""
    OPEN AnyFileSpec FOR INPUT AS #1
        WHILE ISFALSE EOF(1)
            LINE INPUT #1, sTmp$
            IF AnyText <> "" THEN AnyText = AnyText & $CRLF
            AnyText = AnyText & sTmp$
        WEND
    CLOSE #1
END SUB

SUB WriteArrayToFile(AnyArray() AS STRING, BYVAL AnyFileSpec AS STRING)
    DIM LastLine$, Ubd&, Lbd&
    IF NOT (ARRAYATTR(AnyArray(), 0)) THEN
        ' array() is not allocated, so file will be killed
        IF ISFILE(AnyFileSpec) THEN KILL AnyFileSpec
        EXIT SUB
    ELSE
        Ubd& = UBOUND(AnyArray)
        Lbd& = LBOUND(AnyArray)
        OPEN AnyFileSpec FOR OUTPUT AS #1
            LastLine$ = AnyArray(Ubd&)
            IF Ubd& > 0 THEN

```

```

        REDIM PRESERVE AnyArray(Lbd& TO Ubd&-1)
        PRINT #1, AnyArray()
        REDIM PRESERVE AnyArray(Lbd& TO Ubd&)
        AnyArray(Ubd&) = LastLine$
    END IF: PRINT #1, LastLine$;
CLOSE #1
END IF
END SUB

SUB ReadArrayFromFile(BYREF AnyArray() AS STRING, BYVAL AnyFilSpec AS STRING)
    DIM COUNT&: ERASE AnyArray
    IF ISFILE(AnyFilSpec) THEN
        OPEN AnyFilSpec FOR INPUT AS #1
        FILESCAN #1, RECORDS TO COUNT&
        IF COUNT& > 0 THEN
            DIM AnyArray(1 TO COUNT&)
            LINE INPUT #1, AnyArray() TO COUNT&
        END IF
        CLOSE #1
    END IF
END SUB

' ----- Arrays -----

SUB RemoveEmptyItemsFromArray(BYREF AnyArray() AS STRING)
    DIM i&, UBnd&: i& = 0: UBnd& = UBOUND(AnyArray)
    WHILE i& <= UBnd&
        IF TRIM$(AnyArray(i&)) = "" THEN
            ARRAY DELETE AnyArray(i&): DECR UBnd&: DECR i&
        END IF: INCR i&
    WEND: REDIM PRESERVE AnyArray(UBnd&)
END SUB

SUB RemoveDoubleItems(BYREF AnyArray() AS STRING)
    DIM uBnd AS LONG, i AS LONG, j AS LONG: i = 1
    IF NOT (ARRAYATTR(AnyArray(), 0)) THEN EXIT SUB
    uBnd = UBOUND(AnyArray)
    DO WHILE i < uBnd

```

```

    IF UCASE$(AnyArray(i)) = UCASE$(AnyArray(i+1)) THEN
        FOR j = i+1 TO uBnd-1
            AnyArray(j) = AnyArray(j+1)
        NEXT: DECR i: DECR uBnd
        IF uBnd < 1 THEN EXIT DO
    END IF: INCR i
LOOP: IF UBnd < 1 THEN ERASE AnyArray: EXIT SUB
IF uBnd < UBOUND(AnyArray) THEN REDIM PRESERVE AnyArray(1 TO UBnd)
END SUB

' ----- Strings -----

FUNCTION InStrAny(AnyString AS STRING, Subset AS STRING, AnyChars AS STRING) AS INTEGER
' InStrAny(AnyString, "a-z,A-Z,A-z,0-9","_[]ßäöüÄÖÜ€|")
' InStrAny(AnyString, "CHR$(32-127)","ßäöüÄÖÜ€")
DIM St$, Su$, Ac$, C$, C1$, C2$, U1$, U2$, L1$, L2$, X1%, X2%, OK%, strOK%
DIM Element AS STRING: DIM i AS INTEGER, j AS INTEGER, n AS WORD
St$ = AnyString: Su$ = TRIM$(Subset): Ac$ = AnyChars
OK% = ISFALSE 1: strOK% = ISTRUE 1: n = 0
IF Su$ <> "" THEN n = PARSECOUNT(Su$, ",")
IF St$ = "" THEN
    IF Su$ = "" THEN
        FUNCTION = ISFALSE 1: EXIT FUNCTION
    ELSE
        FOR i = 1 TO n
            IF TRIM$(PARSE$(Su$, ",", i)) = "" THEN
                FUNCTION = ISTRUE 1: EXIT FUNCTION
            END IF
        NEXT: FUNCTION = ISFALSE 1: EXIT FUNCTION
    END IF
ELSE
    IF Su$ = "" AND C$ = "" THEN FUNCTION = ISFALSE 1: EXIT FUNCTION
    FOR i = 1 TO LEN(St$)
        C$ = MID$(St$, i, 1): OK% = ISFALSE 1
        FOR j = 1 TO n: Element = TRIM$(PARSE$(Su$, ",", j))
            IF Element = "CHR$(32-127)" THEN
                IF C$ >= CHR$(32) AND C$ <= CHR$(127) THEN OK% = ISTRUE 1
            ELSEIF LEN(Element) = 3 AND MID$(Element, 2, 1) = "-" THEN

```



```

' UCASE$ = ISTRUE 1, LCASE$ = ISFALSE 1, Nrs treated as UCASE$
C1$ = LEFT$(Element,1): C2$ = RIGHT$(Element,1)
U1$ = UCASE$(C1$): U2$ = UCASE$(C2$)
L1$ = LCASE$(C1$): L2$ = LCASE$(C2$)
X1% = (U1$ = C1$): X2% = (U2$ = C2$)
IF X1% AND X2% THEN ' A-Z
    IF C$ >= U1$ AND C$ <= U2$ THEN OK% = ISTRUE 1
    IF C$ >= "0" AND C$ <= "9" THEN OK% = ISTRUE 1
ELSEIF NOT(X1%) AND NOT(X2%) THEN ' a-z
    IF C$ >= "a" AND C$ <= "z" THEN OK% = ISTRUE 1
ELSEIF (X1% AND NOT(X2%)) OR _
    (NOT(X1%) AND X2%) THEN ' A-z
    IF C$ >= U1$ AND C$ <= U2$ THEN OK% = ISTRUE 1
    IF C$ >= L1$ AND C$ <= L2$ THEN OK% = ISTRUE 1
END IF
END IF
IF Ac$ <> "" THEN IF INSTR(Ac$, C$) > 0 THEN OK% = ISTRUE 1
NEXT: IF NOT(OK%) THEN strOK% = ISFALSE 1: EXIT FOR
NEXT: IF NOT(strOK%) THEN FUNCTION = ISFALSE 1
END IF: FUNCTION = strOK%
END FUNCTION

FUNCTION enTAG(AnyStrg AS STRING, AnyCol AS STRING) AS STRING
' AnyCol = "007F00"
' FUNCTION = "<font color=""#007F00">AnyStrg</font>"
LOCAL s$, c$: s$ = AnyStrg: c$ = AnyCol: FUNCTION = s$
IF s$ = "" OR LEN(c$) <> 6 THEN EXIT FUNCTION
IF NOT(InStrAny(c$, "0-9,A-f", "")) THEN EXIT FUNCTION
FUNCTION = "<font color=""#" & c$ & """>" & s$ & "</font>"
END FUNCTION

```

Operating_System.inc

```
*****
'*** Project:
'***     OpSysInfo
'*** Description:
'***     Wrapper for Operating System Information (Including Windows8) and report back the OS and if 64bit or 32bit OS
'*** Programmer:
'***     Cliff Nichols
'*** Based from:
'***     Prior postings of OS info
'*** Post Date:
'***     06-04-2012
'***
'*** Changes, Updates, Patches, etc....
'***     Started out as gathering all info for OS/Program/Compiler but now broken down into separate *.inc files for each catagory
'***
'*** OS Versions:
'***     Should work on 95/98/ME, NT/2000/XP/VISTA/WINDOWS7/Windows8
*****
'Windows Version Numbers
'-----
'           Win      Win      Win      Win      Win      Win      Win      Win      Win      Win      Win      Win      Win
'           3.1      95       98       Me       NT       NT       NT       2000    2003    XP       VISTA    7       8
'-----
'PlatformID  0       1       1       1       2       2       2       2       2       2       2       2       2
'Major Version 4       4       4       4       3       3       4       5       5       5       6       6       6
'Minor Version 0       0       10      90      0       1       0       0       2       1       0       1       2
'Build       950*    1111    1998    1381    2195    3790    2600    TBD     TBD     8400
'-----
#INCLUDE THIS ONCE
%OpSys_INC = 1                                     'Define Includes is declared
*****
'*** Support Includes Files
#INCLUDE ONCE "Win32Api.inc"                         'Windows Api
*****
'*** Declares
ENUM Operating_Branch
    Win_32s = 0
    Win_95 = 1
```

```

    Win_Nt = 2
END ENUM

TYPE Operating_System
    Platform_Id      AS DWORD
    Major_Version    AS DWORD
    Minor_Version     AS DWORD
    Build            AS DWORD
    Service_Pack      AS ASCIIZ * 128
    Version_Desc      AS STRING PTR
    Service_Pack_Desc AS STRING PTR
    BITS             AS LONG
END TYPE

GLOBAL OS AS Operating_System

FUNCTION Operating_System_Os ALIAS "Operating_System_Os" (Os AS Operating_System) AS LONG
    LOCAL OpSys AS OSVERSIONINFOA
    LOCAL Version_Desc AS STRING
    LOCAL Service_Pack_Desc AS STRING
    OpSys.dwOSVersionInfoSize = SIZEOF(OSVERSIONINFOEX)
    GetVersionEx OpSys
    PREFIX "Os."
        Platform_Id = OpSys.dwPlatformId
        Major_Version = OpSys.dwMajorVersion
        Minor_Version = OpSys.dwMinorVersion
        Build = OpSys.dwBuildNumber
        Service_Pack = OpSys.szCSDVersion
        Version_Desc = VARPTR(Version_Desc)
        Service_Pack_Desc = VARPTR(Service_Pack_Desc)
    END PREFIX
    SELECT CASE Os.Platform_Id
        CASE %Operating_Branch.Win_32s
            Os.@Version_Desc = "Windows 3.1 or DOS"
        CASE %Operating_Branch.Win_95
            IF (Os.Major_Version = 4) AND (Os.Minor_Version = 0) THEN
                Os.@Version_Desc = "Windows 95"
            ELSEIF (Os.Major_Version = 4) AND (Os.Minor_Version = 10) THEN
                Os.@Version_Desc = "Windows 98"
            ELSEIF (Os.Major_Version = 4) AND (Os.Minor_Version = 90) THEN
                Os.@Version_Desc = "Windows ME"
            END IF
    END SELECT
    'Local Declaration for Operating System Info
    'Create a seed for a valid pointer
    'Create a seed for a valid pointer
    'Should be SIZEOF(OsInfoEx) not hardcoded (128) like other examples
    'Get OS Information (Branch, Major, Minor, Build, etc)
    'Xfer Information to my easy-to-read UDT
    'Platform ID
    'Determine OS Version (ID = Family, Major and Minor = Version)

```

```

CASE %Operating_Branch.Win_Nt
  IF (Os.Major_Version = 3) AND (Os.Minor_Version = 0) THEN
    Os.@Version_Desc = "Windows NT 3.0"
  ELSEIF (Os.Major_Version = 3) AND (Os.Minor_Version = 1) THEN
    Os.@Version_Desc = "Windows NT 3.1"
  ELSEIF (Os.Major_Version = 4) AND (Os.Minor_Version = 0) THEN
    Os.@Version_Desc = "Windows NT 4"
  ELSEIF (Os.Major_Version = 5) AND (Os.Minor_Version = 0) THEN
    Os.@Version_Desc = "Windows 2000"
  ELSEIF (Os.Major_Version = 5) AND (Os.Minor_Version = 1) THEN
    Os.@Version_Desc = "Windows XP"
  ELSEIF (Os.Major_Version = 5) AND (Os.Minor_Version = 2) THEN
    Os.@Version_Desc = "Windows 2003"
  ELSEIF (Os.Major_Version = 6) AND (Os.Minor_Version = 0) THEN
    Os.@Version_Desc = "Windows Vista"
  ELSEIF (Os.Major_Version = 6) AND (Os.Minor_Version = 1) THEN
    Os.@Version_Desc = "Windows 7"
  ELSEIF (Os.Major_Version = 6) AND (Os.Minor_Version = 2) THEN
    Os.@Version_Desc = "Windows 8"
  END IF
END SELECT
Os.@Service_Pack_Desc = Os.Service_Pack
SELECT CASE GetProcAddress(GetModuleHandle("KERNEL32.DLL"), "Wow64DisableWow64FsRedirection")
  'Determine if 64bit? (Function does not exist in 32bit Kernel)
  CASE 9
    Os.BITS = 32
  CASE ELSE
    Os.BITS = 64
END SELECT
IF Os.BITS <> 0 THEN Os.@Version_Desc = Os.@Version_Desc + $SPC + "-" + $SPC + TRIM$(Os.BITS) + $SPC + "Bit"
IF Os.@Service_Pack_Desc <> "" THEN Os.@Version_Desc = Os.@Version_Desc + $SPC + "-" + $SPC + Os.@Service_Pack_Desc
FUNCTION = ERR 'Return Error Code (If Any)
END FUNCTION

```

Keywords-PB-stm.dat

keyword-lists are incomplete, but missing items can be easy added, the program sorts them in and removes double items and cares, that the output is either in same UCase or LCase character size as in the list or in case of Power Basic all in UCase. Because these files are changed by the program, the program generates bak-files.

#Align	%DS_MODALFRAME	%WS_EX_CONTROLPARENT	Bits
#Bloat	%DS_NOFAILCREATE	%WS_EX_LEFT	Bitse
#Com	%DS_SETFONT	%WS_EX_LTRREADING	Block
#Compile	%HIDDEN	%WS_EX_RIGHTSCROLLBAR	Bold
#COMPILE EXE	%HWND_DESKTOP	%WS_POPUP	Box
#Compiler	%IDC_BUTTON1	%WS_SYSMENU	Browse
#Debug	%IDC_LABEL1	%WS_TABSTOP	Build\$
#Dim	%IDC_LISTBOX1	%WS_VISIBLE	Button
#DIM ALL	%IDD_DIALOG1	%WS_VSCROLL	BYREF
#Else	%LBN_DBLCLK	ABS	BYVAL
#ElseIf	%LBS_NOTIFY	Accel	Calc
#EndIf	%NORMAL	Accept	Call
#If	%OFN_ALLOWMULTISELECT	Acode\$	CallBack
#Include	%OFN_ENABLESIZING	Add	CallSTK
#Messages	%OFN_FILEMUSTEXIST	Addr	CallSTK\$
#Optimize	%OFN_NOVALIDATE	Alloc	CallSTKcount
#Option	%OFN_OVERWRITEPROMPT	AND	Cancel
#PBForms	%OFN_PATHMUSTEXIST	Arc	Case
#Register	%READONLY	Array	CB
#Resource	%SS_CENTER	Arrayattr	CBCTL
#Stack	%SS_SUNKEN	AS	Cbctlmsg
#TOOLS	%SUBDIR	Asc	CBHNDL
#Utility	%SYSTEM	Asm	Cblparam
\$CR	%WHITE	Assign	CBMSG
\$CRLF	%WM_COMMAND	Atn	CBWPARAM
\$LF	%WM_INITDIALOG	Attach	CByt
\$TAB	%WM_NCACTIVATE	Bar	CCur
%ARCHIVE	%WS_BORDER	Beep	CCux
%BIF_EDITBOX	%WS_CHILD	Bgr	CDbl
%BN_CLICKED	%WS_CLIPSIBLINGS	Bin\$	Cdwd
%Def	%WS_DLGFRAME	Bit	Ceil
%DS_3DLOOK	%WS_EX_CLIENTEDGE	Bitmap	Cext

ChDir	Ctl	Disable	Extract\$
ChDrive	Ctlmsg	DiskFree	Field
Check	CurDir\$	DiskSize	FileAttr
Check3State	CVByt	Display	FileCopy
Checkbox	CVCur	DLLmain	FileName\$
Child	CVCux	Do	FILESCAN
Choose	CVD	Doevents	FILL
Chr	CVDwd	Draw	Find
CHR\$	CVE	Duplex	Fit
Cint	CVI	Dword	Fix
Class	CVL	Ellipse	Flush
Clear	CVQ	Else	Focus
Click	CVS	ElseIf	Font
Client	CVWrd	Enable	FOR
Clipboard	CWrd	End	FORMAT\$
Clng	Data	Environ	FormFeed
Close	Datacount	Environ\$	Frac
ClsID\$	Date\$	Eof	Frame
Code	DECLARE	Eqv	Free
CodePTR	Decr	ERASE	FreeFile
Collate	DefByt	Erl	FuncName\$
Color	DefCur	Erl\$	Function
ColorMode	DefCux	Err	Get
Column	DefDb1	ErrClear	Get\$
ComboBox	DefDwd	ERROR	GetAttr
Comm	DefExt	Error\$	GLOBAL
Command\$	DefInt	Events	GlobalMem
Content	DefLng	Exact	GOSUB
CONTROL	DefQud	Exe	GoTo
Copies	DefSng	EXE.NAME\$	Graphic
Copy	DefStr	EXE.PATH\$	Guid\$
Cos	DefWrd	Exit	GuidTxt\$
Count	DELETE	EXP	HANDLE
Cqud	Desktop	Exp10	hdlg
CREATE	Detach	Exp2	Header
Cset	Dialog	Expanded	Hex\$
CSet\$	DIM	Explicit	Hi
CSng	Dir\$	EXTENDED	HiByt

HiInt	ITERATE	MakPtr	Nmhdr
HiWrd	JOIN\$	MakWrd	Nmhdr\$
Hndl	KILL	Margin	Nmhwnd
Host	Label	Masked	NmID
IAmPowerBASIC	LBOUND	Mat	Not
Icon	LCASE\$	Max	NOTHING
Id	LEFT	MCASE\$	Notify
IDispInfo	Left\$	Me	Nul\$
If	LEN	MEMBER	NULL
IIF	Let	Menu	ObjActive
Image	LibMain	Method	OBJECT
Image2	LINE	Mid\$	ObjPTR
ImageList	Lines	Min	ObjResult
ImageX	ListBox	Mix	ObjResult\$
ImgButton	ListView	MkByt\$	Oct\$
ImgButtonX	Lo	MkCur\$	OEMtext
IMP	Load	MkCux\$	OF
In	LoByt	MkD\$	OFF
Incr	Loc	MkDir	On
Inkey\$	Local	MkDwd\$	Open
INPUT	Lock	Mke\$	OpenFile
Input#	Lof	Mki\$	Option
InputBox\$	Log	Mkl\$	Or
INSERT	Log10	Mkq\$	Orientation
Instance	Log2	Mks\$	OUTPUT
Instr	LoInt	Mkwrd\$	Overlay
Int	LONG	Mod	Page
INTEGER	Loop	Modal	PageSize
Interface	LoWrd	Mode	Paint
IsFalse	lParam	Modeless	Paper
IsFile	LPrint	MousePTR	Papers
IsInterface	LPrint\$	Msg	Parent
IsMissing	LSet	MSGBOX	Parse
IsNothing	LTRIM\$	MyBase	Parse\$
IsObject	Macro	Name	ParseCount
IsTrue	Mak	NEW	Parts
IsWin	MakDwd	Next	PathName\$
Item	MakInt	Nmcode	PathScan\$

PBLibMain
PBMAIN
Peek
Peek\$
Pie
Pixel
Pixels
Poke
Poke\$
Polygon
Polyline
Popup
Pos
Post
PPI
PRESERVE
Previous
PRINT
Print#
Printer\$
PrinterCount
Profile
ProgID\$
ProgressBar
Property
Put
Put\$
Quality
RaiseEvent
Randomize
Range
Read\$
RECORDS
Recv
ReDim
Redraw
RegExpr
Register

RegRepl
REM
Remain\$
Remove\$
Render
Repeat\$
Replace
Reset
RESUME
Retain\$
Return
Rgb
RIGHT
RIGHT\$
Rmdir
Rnd
Root
ROTATE
Round
RSet
RSet\$
RTRIM\$
Save
SAVEFILE
Scale
SCAN
ScrollBar
Seek
SelCount
SELECT
Send
Separator
Set
SetAttr
Seteof
Sgn
Shell
SHIFT

Show
Sin
Size
SizeOf
Sleep
SORT
Space\$
Sqr
State
STATIC
STATUS
StatusBar
Step
Str\$
StrDelete\$
Stretch
String
String\$
StrInsert\$
StrPTR
STRREVERSE\$
Style
StyleXX
SUB
SUSPEND
Swap
Switch
Tab
Tab\$
Tally
TAN
TANH
Tcp
Text
TextBox
THEN
THREAD
ThreadCount

Threaded
ThreadID
Time\$
TIMEOUT
Timer
Tix
TO
Toolbar
Trace
TrackPOS
Tray
Trays
TreeView
Trim\$
TRN
Try
TXT
TYPE
UBOUND
UCASE\$
UCode\$
UDP
Unicode
Union
Units
UnLock
UnSelect
Until
User
USING\$
Val
VARIANT
Variant#
Variant\$
VariantVT
VarPTR
VERIFY
Visible

WaitKey\$
Keywords-PB-stm.dat

Wend
WHILE
Width
With
WINDOW
WinMain
WORD
wParam
WRITE
WRITE#
XOR
XPRINT
XPrint\$

Keywords-PB-ops.dat

(
)
*
-
/
<
<=
<>
=
>
>=
AND
EQV
IMP
ISFALSE
ISTRUE
MOD
NOT
OR
XOR
\

Keywords-vbs-stm.dat

Abs
And
Case
Const
CreateObject
Dim
Do
Each
ElseIf
End
Exit
Explicit
False
For
ForReading
ForWriting
Function
If
In
InStr
InStrRev
Join
LBound
LCase
Left
Len
Loop
Mid
MsgBox
Next
Not
Option
Or
Preserve
Public
ReDim

Right
Select
Set
Split
Sub
Then
True
UBound
UCase
Until
vbCr
vbCrLf
vbLf
Wend
While

Engr. Erich Foltyn
Vienna, Austria
www.erich-foltyn.eu

Sept 29th 2012