

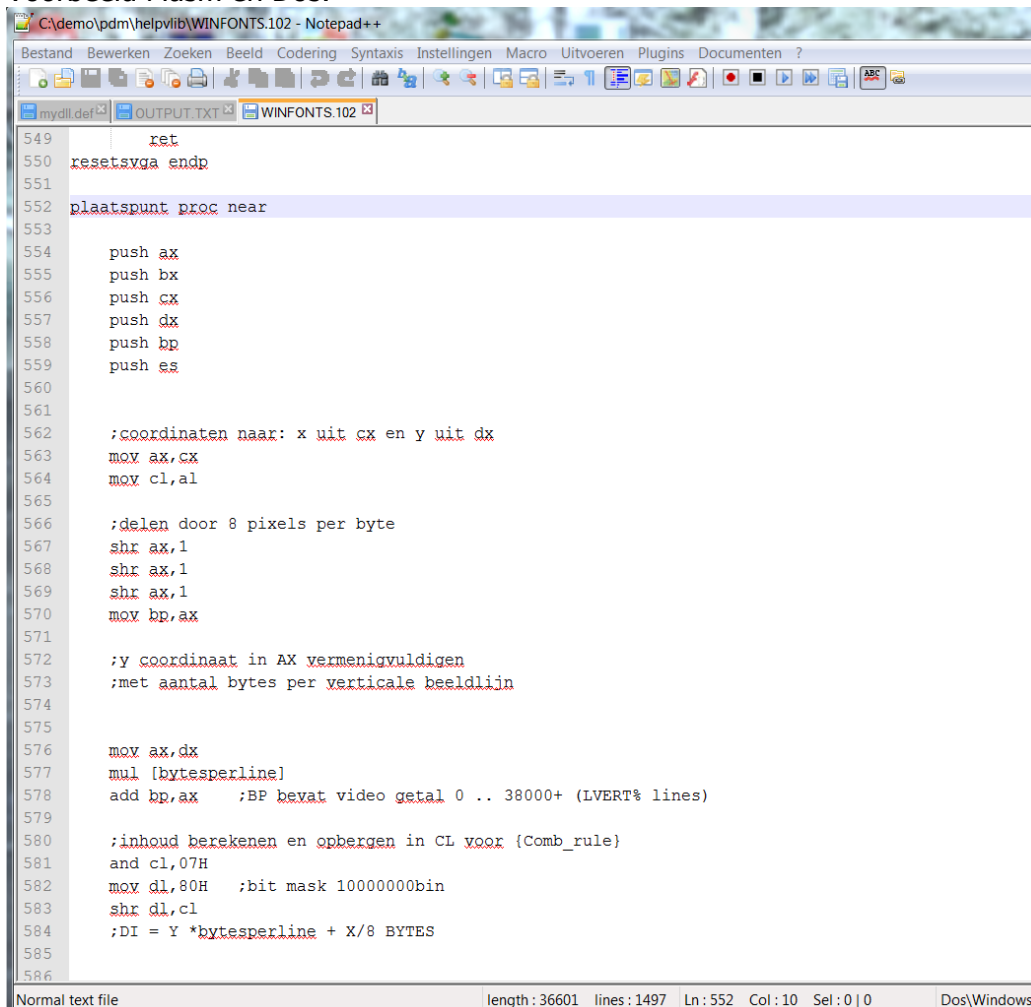
## Hoe werkt dat 'onder water': **Tekst op het scherm**

Tekst op het scherm plaatsen lijkt heel eenvoudig. Type 'print "Hello World"' en het verschijnt op het scherm. Lang geleden, zo rond 1985, waren we daar heel blij mee. Tegenwoordig werk dat anders.

Rond 1993 ontstonden (in PD&M) de eerste bruikbare grafische interfaces. Deze interfaces draaiden toen nog onder DOS doch hadden het uiterlijk van Windows 3.11 en later editie 95. Dit was een expliciete keuze om de stabiliteit van de software te kunnen garanderen vanuit de DOS omgeving met een Windows 'look alike' grafische gebruikers interface (GUI). En vergeet niet: de eerste Windows systemen waren ook een laag boven de DOS omgeving.

De eerste grafisch teksten (1993) werden geprogrammeerd in Macro-Assembler. Daarbij werd dankbaar gebruik gemaakt van Assembler code die werd meegeleverd met Visual Basic Professional (voor DOS omgeving) die royalty free gebruikt mogen worden. Deze Assembler code is door ondergetekende indertijd aangepast waardoor deze aanzienlijk sneller werd, een beveiliging is verwijderd en de zelf ontwikkelde Assembler pixel schrijf routine is geïntegreerd. Deze Pixel schrijf routine dateert uit de tijd dat bij PD&M een geïntegreerd 2D CADsysteem behoorde (tot 1992). Met deze fontbesturing in Assembler lag de weg open voor een onafhankelijk indelen van het scherm met – toen nog – gebruikt van Windows Font files onder ..... Dos.

### Voorbeeld Masm en Dos:



```
549         ret
550     resetsvga endp
551
552     plaatspunt proc near
553
554         push ax
555         push bx
556         push cx
557         push dx
558         push bp
559         push es
560
561
562         ;coördinaten naar: x uit cx en y uit dx
563         mov ax,cx
564         mov cl,al
565
566         ;delen door 8 pixels per byte
567         shr ax,1
568         shr ax,1
569         shr ax,1
570         mov bp,ax
571
572         ;y coördinaat in AX vermenigvuldigen
573         ;met aantal bytes per verticale beeldlijn
574
575
576         mov ax,dx
577         mul [bytesperline]
578         add bp,ax    ;BP bevat video getal 0 .. 38000+ (LVERT% lines)
579
580         ;inhoud berekenen en opbergen in CL voor {Comb_rule}
581         and cl,07H
582         mov dl,80H    ;bit mask 10000000bin
583         shr dl,cl
584         ;DI = Y *bytesperline + X/8 BYTES
585
586
```

Normal text file | length: 36601 | lines: 1497 | Ln: 552 | Col: 10 | Sel: 0 | 0 | Dos\Windows

Al deze regels leiden tot onder andere de functie

```
;*****  
; fl_BltChar - Outputs a character's bitmap to the screen  
;  
; BASIC CALL:  
; fl.BltChar FontAddr(far), Char%, X%, Y%  
;  
;*****
```

In de Dos library wordt de Assembler functie voor grafische tekst (windows .fon bestand) aangeroepen in deze functie :

*STATIC FUNCTION OutGText% (Wform1, x AS SINGLE, Y AS SINGLE, Text\$)*

```
TextLen% = 0  
FOR Char% = 1 TO LEN(Text$)  
    ChVal% = ASC(MID$(Text$, Char%, 1))  
    BitMapPtr% = VARPTR(FontData(FGP.DSeg))  
    BitMapSeg% = VARSEG(FontData(FGP.DSeg))  
  
    CharLen% = flbltchar%(BitMapSeg%, BitMapPtr%, ChVal%, ix%, iy%)  
    ix% = ix% + FGP.XPixInc * CharLen%  
    iy% = iy% + FGP.YPixInc * CharLen%  
  
    TextLen% = TextLen% + CharLen%  
NEXT Char%
```

In de 3GL omgeving (VB DOS prof, PD&M release na 2001, source files met extensie \*.VDW) vindt de functieaanroep als volgt plaats voor .tlf font bestanden:

Textweergave via functie OutGtext in DOS-omgeving met Windows GUI

```
'* BACKPLANE:  
CAPTIONRESET% = 1  
CAPTIONSTATUS% = 1  
CALL FORM_FRAME(MYform1, LDISK$, lvert%, bx1%, by1%, bx2%, by2%, LX2%, CAPTIONTEXT$, CAPTIONSTATUS%)  
CALL MSWschaduwkader(MYform1, bx1% + 10, by1% + 37, bx2% - 70, by2% - 30, lvert%)  
CALL PASSIVATEBOX(MYform1, lvert%, bx1% + 180, by1% + 50, bx2% - 22, by2% - 29, 0, 15, 0)  
CALL MSWschaduwkader(MYform1, bx1% + 180, by1% + 50, bx2% - 20, by2% - 27, lvert%)  
  
SELECTFONT MYform1, REGISTERFONTFILE$, FONTSIZE$, 1  
  
GET #93, 30, record93a  
l% = OutGText(MYform1, bx1% + 15, by2% - 20, RTRIM$(record93a.lan1))  
  
setgtextcolor MYform1, 0  
Y_SNG! = by1% + 55  
  
'mb23032009 - voorgevoegd van t/m document nr  
FOR t% = 541 TO 542  
    GET #93, t%, record93a  
    l% = OutGText(MYform1, bx1% + 15, Y_SNG!, MID$(record93a.lan1, 1, 33))  
    Y_SNG! = Y_SNG! + 16  
NEXT t%
```

Deze herschreven DOS versie (2001-2011) is stapsgewijs voorbereid om via de uiteraard zelf gemaakte Xcompiler de broncode over te zetten naar GDI32/VB6frame. Dezelfde code anno 2019 omvat:

```

'* BACKPLANE:
CAPTIONRESET_int% = 1
CAPTIONSTATUS_int% = 1
Call FORM_FRAME(MYFORM1, ldisk$, lvert_INT%, BX1_INT%, BY1_INT%, BX2_INT%, BY2_INT%, LX2_INT%, CAPTIONTEXT$, lvert_INT%)

Call MSWSCHADUWKADER(MYFORM1, BX1_INT% + 10, BY1_INT% + 37, BX2_INT% - 70, BY2_INT% - 30, lvert_INT%)
Call PASSIVATEBOX(MYFORM1, lvert_INT%, BX1_INT% + 180, BY1_INT% + 50, BX2_INT% - 22, BY2_INT% - 29, lvert_INT%)
Call MSWSCHADUWKADER(MYFORM1, BX1_INT% + 180, BY1_INT% + 50, BX2_INT% - 20, BY2_INT% - 27, lvert_INT%)

SELECTFONT MYFORM1, REGISTERFONTFILE$, FONTSIZE$, 1

Get #93, 30, record93a
l_int% = OutGText(MYFORM1, BX1_INT% + 15, BY2_INT% - 20, RTrim$(record93a.lan1))

SetGTextColor MYFORM1, 0
Y_SNG_SNG! = BY1_INT% + 55

For t_int% = st_93a_INT% To sp_93a_INT%
    Get #93, t_int%, record93a
    l_int% = OutGText(MYFORM1, BX1_INT% + 15, Y_SNG_SNG!, Mid$(record93a.lan1, 1, 33))
    Y_SNG_SNG! = Y_SNG_SNG! + 16
Next t_int%

```

Zoek de verschillen ... in de parameter-extensies.

```

Public Static Function OutGText(flow1, X As Variant, Y As Variant, Gtext$)

    Call SetTextColor(Form1.hdc, LastSetGTextColor&)

    If InStr(Gtext$, Chr$(0)) > 0 Then
        For t% = 1 To Len(Gtext$)
            If Mid$(Gtext$, t%, 1) < " " Then
                Mid$(Gtext$, t%, 1) = " "
            End If
        Next t%
    End If

    tx& = X * Xsized
    ty& = Y * Ysized

    '* WINDOWS 2000 ERROR 3080 BIJ STRINGLENGTE VAN MEER DAN 32768...KARAKTERS !
    ,
    HLenGtext& = Len(Gtext)
    If HLenGtext& > 255 Then HLenGtext& = 255

    geprobeerd% = 0

Select Case PDM_CPUbits%

Case Is = 64
    REGISTERFONTFILE$ = PDM_CPU64fontname$
    Form1.Font.Name = REGISTERFONTFILE$

    'mb11112013 Lucida Console getrimd .. Form1.Font.Size = 9 * Xsized
    Form1.Font.Size = 7 * Xsized 'mb11112013 Lucida Console getrimd .. was 9

    'FontStepsize5! = 5 '* 1.2 'mb11102013
    'Form1.Font.Size = 9 * Xsized 'mb17102013 KW
    FontStepsize5! = 5 'mb17102013 KW
    FontStepsize7! = 7 'mb11102013
    FontStepsize5Xsized! = FontStepsize5! * Xsized 'mb11102013
    FontStepsize7Xsized! = FontStepsize7! * Xsized 'mb11102013

Case Else ' 32
    REGISTERFONTFILE$ = PDM_CPU64fontname$
    Form1.Font.Name = REGISTERFONTFILE$

    Form1.Font.Size = 7 * Xsized 'mb11112013 Lucida Console getrimd .. wa 8.5

```

```

'If PDM_CPU32fontname$ <> "PDMWfont" Then Form1.Font.Size = 7 * Xsized
FontStepsize5! = 5 'mb11102013
FontStepsize7! = 7 'mb11102013
FontStepsize5Xsized! = FontStepsize5! * Xsized 'mb11102013
FontStepsize7Xsized! = FontStepsize7! * Xsized 'mb11102013
End Select

Select Case fontspatiering%

Case Is = 5

Form1.Font.Bold = False

'* forceer fixed spacing
For t% = 1 To HLenGtext&
    ttx& = tx& + (t% - 1) * FontStepsize5Xsized! 'mb11112013 x 1.2 na trim
Lucida console 64 bit naar 8.5 x xsized
    THISCHAR$ = Mid$(Gtext$, t%, 1)
    If THISCHAR$ > Chr$(122) Then Call CHECK2ADAPT_CHARS(THISCHAR$) '>122
mb09102013
    ret = TextOut(Form1.hdc, ttx&, ty&, THISCHAR$, 1)
Next t%

Case Is = 7

Form1.Font.Bold = False

'* forceer fixed spacing
For t% = 1 To HLenGtext&
    ttx& = tx& + (t% - 1) * FontStepsize7Xsized!
    THISCHAR$ = Mid$(Gtext$, t%, 1)
    hTHISCHAR% = Asc(Mid$(Gtext$, t%, 1))
    If THISCHAR$ > Chr$(122) Then
        Call CHECK2ADAPT_CHARS(THISCHAR$) '>122 mb09102013*
    End If 'THISCHAR$ > Chr$(122) Then
    ret = TextOut(Form1.hdc, ttx&, ty&, THISCHAR$, 1)
    ret = TextOut(Form1.hdc, ttx& + 1, ty&, THISCHAR$, 1) 'MB19072012
Next t%

End Select

End Function

```

In tegenstelling tot DOS (.fon bestanden) zijn .ttf bestanden wispelturig in weergave van bepaalde karaktercodes. Bijvoorbeeld de alt-code voor het diameterteken. Via een tabel die als platte tekst bewerkt kan worden (bijvoorbeeld kladblok) kunnen conversie waarden aangegeven worden op de ldisk (language disk). Om reden van snelheid wordt deze tabel éénmalig in array CHAR2CHECK\_TABLE\_INT%() geladen.

```

Sub CHECK2ADAPT_CHARS(THISCHAR$)
'VERVANGEN ASCII CODES OP BASIS VAN FONTSETTINGS OP ldisk$ + "CHAR_CHANGE" +
Format$(PDM_CPUbits%, "000") For Input As #1

For t% = 1 To CHAR2CHECK%
    If Asc(THISCHAR$) = CHAR2CHECK_TABLE_INT%(t%, 1) Then
        THISCHAR$ = Chr$(CHAR2CHECK_TABLE_INT%(t%, 2))
        t% = CHAR2CHECK% + 99
    End If 'Asc(THISCHAR$) = CHAR2CHECK_TABLE_INT%(t%, 1)
Next t%

End Sub 'CHECK2ADAPT_CHARS(THISCHAR$)

```

Voorbeeld van een conversietabel op de ldisk. 01550248 betekent dat karakter code 0155 wordt vervangen door 0248. Het diameterteken scoort hoog; vier keer. De oorzaak is gelegen in het feit dat stuklijst gegevens uit verschillende CADsystemen worden geïmporteerd en elk CADsysteem zo zijn eigen ASCII code voor een diameterteken hanteert. Karaktercode en ASCIIcode zijn synoniem in dit verhaal. Voor printerbesturing is een afzonderlijke tabel beschikbaar.



```

char_change032.txt - Kladblok
Bestand  Bewerken  Opmaak  Beeld  Help
01550248  diameterteken
01300233  e met streep naar rechts
01370235  e met 2 punten
01710189  teken 1/2
01720188  teken 1/4
02050248  diameterteken
02210248  diameterteken
02370248  diameterteken
02410177  teken +-
02480176  gradenteken
02530178  tot de macht 2 teken
  
```

Laden table in array:

```

'MB03012013 - CHAR REPLACE FIXED FONT OUTGTEXT - LOAD IN INITFORM()
Public CHAR2CHECK%
Public CHAR2CHECK_TABLE_INT%(99, 2)
'mb12042013
Public printer_CHAR2CHECK%
Public printer_CHAR2CHECK_TABLE_INT%(99, 2)

'* scherm character conversie tabel
'* mb reset tbv. reload bij goto 1 in zelfde module (oa. P309)
CHAR2CHECK% = 0

Close #1
Open ldisk$ + "CHAR_CHANGE" + Format$(PDM_CPUBits%, "000") + ".TXT" For Random As #1
Close #1
Open ldisk$ + "CHAR_CHANGE" + Format$(PDM_CPUBits%, "000") + ".TXT" For Input As #1
While Not EOF(1)
  Line Input #1, record1$
  If Len(record1$) >= 8 Then
    CHAR2CHECK% = CHAR2CHECK% + 1
    CHAR2CHECK_TABLE_INT%(CHAR2CHECK%, 1) = Val(Mid$(record1$, 1, 4))
    CHAR2CHECK_TABLE_INT%(CHAR2CHECK%, 2) = Val(Mid$(record1$, 5, 4))
  End If 'LEN(RECORD1$)>=2
Wend
Close #1
  
```

\*\*\*\*