Creating a PDF from SBASIC

Malcolm Lear has kindly given me permission to include a listing based on routines within his PCB Design program, showing how a QL BASIC program can write out a simple PDF file.

The term "PDF" stands for Portable Document Format, a file format created by Adobe many years ago. As the name implies, it's meant to be a system allowing a largely universal means of distributing documents which should be viewable on most systems. To this end, Adobe allows free use of the Adobe Reader program - most Windows users will have a copy of this and I'm sure Mac users will have such a program too. On the QL, Jonathan Hudson's port of Ghostscript available from my website's Ghostscript page or from his website at www.daria.co.uk/qdos/ should be able to load PDF documents, although I've not tried it myself.

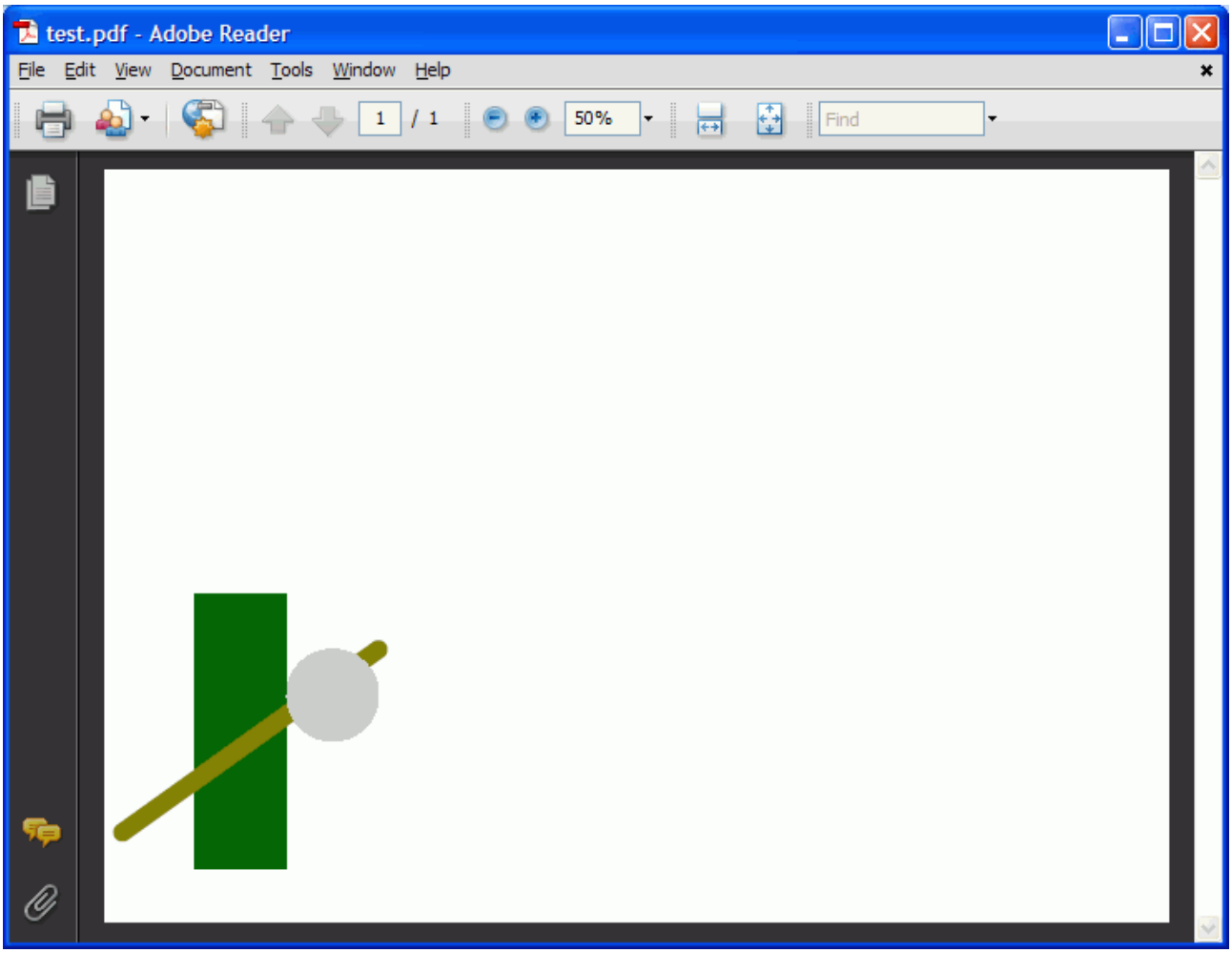
The PDF Reference was first published when Adobe Acrobat was introduced in 1993. Since then, updated versions of the PDF Reference have been made available from Adobe via the Web, and from time to time, in traditional paper documents made available from book publishers - a quote from Adobe's website. You can get more complete details of the PDF reference from their website at:

http://www.adobe.com/devnet/pdf/pdf_reference.html

In Windows programs, there are utility "printer" drivers which allow programs such as Word to generate PDF files using the program's PRINT command. In theory at least it might be possible to write printer drivers for programs like Line esign which allow output as PDF files instead of going direct to printers.

Here's Malcolm's example routine which writes out a very simple PDF file as shown in the illustration. For those who don't enjoy typing in listings, I'll see if I can get it onto the Quanta website for download. Malcolm has used good meaningful variable names etc and useful REM statements so it should be reasonably easy to follow, though obviously not a subject for beginners! I've added a touch of indenting to Malcolm's listing to try to make it a little easier to follow in print.

It writes a file called Test.pdf onto the DATA_USE default drive (lines 1020 and 1030) which you can then transfer to wherever you wish to try to view the file. In my case I moved it to Windows and successfully loaded it into the Adobe reader.



Screen dump of results of loading the test.pdf into Adobe Reader in Windows.

```
1000 DEFine PROCedure CreatePDF
1010    SetupPDF
1020    FileOut$ = 'Test' : Author$ = 'Unknown' : Creator$ = 'Unknown' : PaperSize = 4 :
WhtBG = 1
1030    OPEN_NEW#3,FileOut$&'.PDF'
1040    PDFWidth$ = PDFOut$(PaperW(PaperSize)) : PDFHeight$ = PDFOut$(PaperH(PaperSize))
1050    PrtHeader
1060    IF NOT WhtBG THEN PDFColour 0,0,0 : PDFBlock
0,0,PaperW(PaperSize),PaperH(PaperSize)
1070    REMark Graphics Start Here ****************************
1080    PDFColour 0,90,0 : REMark Colour is Red, Green, Blue 0-255
1090    PDFBlock 1000,3000,1000,600 : REMark A Dark Green 1000x3000 mil Block at 1000,600
1100    PDFColour 128,128,0 : REMark Dark Yellow
1110    PDFLine 200,1000,3000,3000,200 : REMark 200 mil Line from 200,1000 to 3000,3000
1120    PDFColour 200,200,200 : REMark Grey
1130    PDFCircle 2500,2500,1000 : REMark 1000 mil Diameter Circle at 2500,2500
1140    REMark and Finish Here ********************************
1150    PrtTrailer
1160    CLOSE#3
1170 END DEFine CreatePDF
1175 :
1180 REMark -------------------------------
1185 :
1190 DEFine PROCedure SetupPDF
1200    DIM PDFWidth$(6) : DIM PDFHeight$(6) : DIM FileOut$(20) : DIM Author$(20) : DIM
Creator$(20)
1205    DIM In$(2) : DIM Obj(8) : DIM PaperH(5) : DIM PaperW(5) : In$ = CHR$(13) &
CHR$(10)
1210    PaperH(0) = 33000 : PaperH(1) = 23333 : PaperH(2) = 16500 : PaperH(3) = 11666 :
PaperH(4) = 8250 : PaperH(5) = 5833
1220    PaperW(0) = 46666 : PaperW(1) = 33000 : PaperW(2) = 23333 : PaperW(3) = 16500 :
PaperW(4) = 11666 : PaperW(5) = 8250
```

```
1230 END DEFine SetupPDF
1235 :
1240 REMark --------------------------------
1245 :
1250 DEFine PROCedure PrtHeader
1260   BytesSent = 0 : xref = 1262
1270   PDFPrint('%PDF-1.3') : Obj(1) = BytesSent
1280   PDFPrint('1 0 obj'&In$&'<<')
1290   PDFPrint(' /Title ('&FileNmExp1$&')'&In$&' /Author ('&Author$&')')
1300   PDFPrint(' /Creator ('&Creator$&' '&sbasRev$&')'&In$&' /Producer (Malcolm Lear
UK)')
1310   PDFPrint(' /CreationDate ('&DATE$&')'&In$&'>>'&In$&'endobj') : Obj(2) = BytesSent
1320   PDFPrint('2 0 obj'&In$&'<<'&In$&' /Type /Catalog'&In$&' /Outlines 3 0 R'&In$&'
/Pages 4 0 R'&In$&'>>'&In$&'endobj') : Obj(3) = BytesSent
1330   PDFPrint('3 0 obj'&In$&'<<'&In$&' /Type /Outlines'&In$&' /Count
0'&In$&'>>'&In$&'endobj') : Obj(4) = BytesSent
1340   PDFPrint('4 0 obj'&In$&'<< /Type /Pages'&In$&' /Kids [5 0 R]'&In$&' /Count
1'&In$&'>>'&In$&'endobj') : Obj(5) = BytesSent
1350   PDFPrint('5 0 obj'&In$&'<<'&In$&' /Type /Page'&In$&' /Parent 4 0 R'&In$&'
/MediaBox [0 0 '&PDFWidth$&' '&PDFHeight$&']'&In$&' /Contents 6 0 R'&In$&' /Resources <<
/ProcSet 8 0 R >>'&In$&'>>'&In$&'endobj') : Obj(6) = BytesSent
1360   PDFPrint('6 0 obj'&In$&'<< /Length 7 0 R >>'&In$&'stream'&In$&'1 j 1 J')
1370 END DEFine PrtHeader
1375 :
1380 REMark --------------------------------
1385 :
1390 DEFine PROCedure PrtTrailer
1400   PDFPrint('endstream'&In$&'endobj') : Obj(7) = BytesSent
1410   PDFPrint('7 0 obj'&In$&'639'&In$&'endobj') : Obj(8) = BytesSent
1420   PDFPrint('8 0 obj'&In$&'[/PDF]'&In$&'endobj') : xref = BytesSent
1430   PDFPrint('xref'&In$&'0 9')
1440   PDFPrint('0000000000 65535 f')
1450   FOR x = 1 TO 8 : Temp$ = Obj(x) : PDFPrint(FILL$('0',10-LEN(Temp$))&Temp$&' 00000
n') : END FOR x
1460   PDFPrint('trailer << /Size 9 /Root 2 0 R /Info 1 0 R >>'&In$&'startxref
'&xref&In$&'%%EOF')
1470 END DEFine PrtTrailer
1475 :
1480 REMark --------------------------------
1485 :
1490 DEFine PROCedure PDFColour(r,g,b)
1500   PDFRed = INT(r/25.5+.5)/10 : PDFGreen = INT(g/25.5+.5)/10 : PDFBlue =
INT(b/25.5+.5)/10
1510 END DEFine PDFColour
1515 :
1520 REMark --------------------------------
1530 DEFine PROCedure PDFCircle(x,y,d)
1540   PDFPrint(PDFRed&' '&PDFGreen&' '&PDFBlue&' RG'&En$)
1550   PDFPrint(PDFOut$(d)&' w '&PDFOut$(x)&' '&PDFOut$(y)&' m '&PDFOut$(x)&'
'&PDFOut$(y)&' l S')
1560 END DEFine PDFCircle
1565 :
1570 REMark --------------------------------
1575 :
1580 DEFine PROCedure PDFLine(x1,y1,x2,y2,d)
1590   PDFPrint(PDFRed&' '&PDFGreen&' '&PDFBlue&' RG'&En$)
1600   PDFPrint(PDFOut$(d)&' w '&PDFOut$(x1)&' '&PDFOut$(y1)&' m '&PDFOut$(x2)&'
'&PDFOut$(y2)&' l S')
1610 END DEFine PDFLine
1615 :
1620 REMark --------------------------------
1625 :
1630 DEFine PROCedure PDFBlock(w1,h1,x1,y1)
1640   x2 = w1 + x1 : y2 = h1 + y1
1650   PDFPrint(PDFRed&' '&PDFGreen&' '&PDFBlue&' rg '&PDFOut$(x1)&' '&PDFOut$(y2)&' m
'&PDFOut$(x2)&' '&PDFOut$(y2)&' l '&PDFOut$(x2)&' '&PDFOut$(y1)&' l '&PDFOut$(x1)&'
'&PDFOut$(y1)&' l h f*')
1660 END DEFine PDFBlock
1665 :
1670 REMark --------------------------------
1675 :
```

```
1680 DEFine FuNction PDFOut$(Coord)
1690   IF Coord > 1 OR Coord < -1 THEN RETurn INT(Coord*7.2)/100 : ELSE RETurn 0
1700 END DEFine PDFOut$
1705 :
1710 REMark --------------------------------
1715 :
1720 DEFine PROCedure PDFPrint(a$)
1730   BytesSent = BytesSent + LEN(a$) + 1
1740   PRINT #3,a$
1750 END DEFine PDFPrint
```