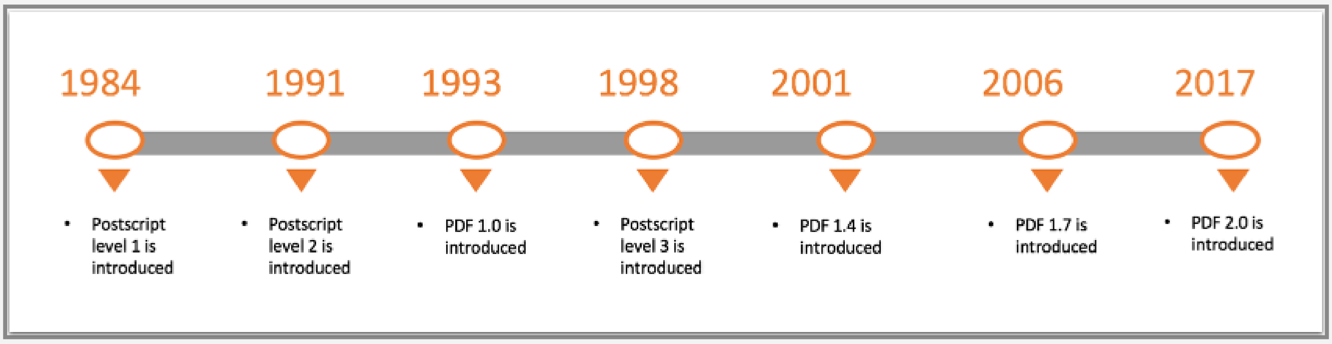
How to create a simple PDF file

Brief history of PDF

The beginning - John Warnock, an engineer at Xerox, developed a language called ‘Interpress’ that could be used to control Xerox laser printers. He along with his boss, Charles M. Geschke, tried for two years to convince Xerox to turn Interpress into a commercial product. When this failed, they decided to leave Xerox and try it on their own - by founding Adobe.

PDF started off as an internal project at Adobe by John Warnock to create a file format so that documents could be spread throughout the company and displayed on any computer using any operating system. The engineers at Adobe enhanced two technologies: Postscript and Adobe Illustrator and created both a new file format (PDF, which is really a kind of optimized PostScript) and a set of applications to create and visualize these files.



The internal structure of PDF

PDF files use a fixed structure and always contain 4 sections:

- A header, which contains information on the PDF-specifications the file adheres to. This line looks like this:

```
%PDF-1.7
```

- The body area which contains a description of the various elements that are placed on the pages.

```
1 0 obj
...
endobj
2 0 obj
...
endobj
...
```

- A cross-reference table which refers to all the elements from the body that are used on the pages of the PDF-file. In other words, the table is mainly a list of the addresses of each object in the body section.

```
xref
0 6
0000000000 65535 f
0000000010 00000 n
0000000079 00000 n
0000000173 00000 n
0000000301 00000 n
0000000380 00000 n
```

The first number after xref says that this list starts at object 0, the object number of the first object in this subsection. The second number after xref is a count of how many objects (6) are in this table and that the remaining five entries are for objects with object numbers 1, 2, 3, 4 and 5. Here object #1 is at offset 10 and is 'in use' (n).

Please note that the first ten digits (0000000000) of the first entry for object 0 points to the next free object, which is, the first object itself.

- A trailer which tells applications or RIPs where to find the cross-reference table and always ends with '%%EOF'. If this line is missing, the PDF-file is not complete and can probably not be processed by any RIP or application. This is not the case with PostScript files. If the last few lines of a PostScript file are missing (because of a lost connection while transferring the file or a computer crash) you can often still print most of the pages. With a PDF-file, you'll lose everything.

```
trailer
```

```
<<
  /Size 6
  /Root 1 0 R
>>
startxref
492
%%EOF
```

The end of a PDF file is read first by the PDF reading application. The trailer holds information about the location and details of the Cross-reference table. The trailer has three parts. The first part has the keyword *trailer* followed by a dictionary that holds values for certain fields.

The second part has the keyword *startxref*, and in the next line, a number. The number denotes how far (in bytes) the keyword *xref* (of the last section of the cross-reference table) is from the start of the file. The very next line has the value %%EOF to denote the end of the file.

Start with a simple PDF

```
%PDF-1.4
1 0 obj
<<
        /Length 51
>>
stream
1 0 0 RG
5 w
36 144 m
180 144 l
180 36 l
36 36 l
s
endstream
endobj
2 0 obj
<<
        /Type /Catalog
        /Pages 3 0 R
>>
```

```
endobj
3 0 obj
<<
        /Type /Pages
        /Kids [4 0 R ]
        /Count 1
>>
endobj
4 0 obj
<<
        /Type /Page
        /Parent 3 0 R
        /MediaBox [0 0 612 792]
        /Contents 1 0 R
>>
endobj


xref
0 4
0000000000 65535 f
0000000010 00000 n
0000000113 00000 n
0000000165 00000 n
0000000227 00000 n
trailer
<<
        /Size 4
        /Root 2 0 R
>>
startxref
344
%%EOF
```

Click to copy
Copy this code in a text editor and create your own PDF file.

NOTE: This PDF was written manually and is voluntarily simplied for the purpose of this introduction. Production PDFs are usually more complex.