



in this article i will describe how to create pdf-file by hand with a regular text-editor. i will show different very easy concepts, depending on which operating system is used.

format

the format of pdf-file can be explained quick. a clean pdf-file contains:

- header (%PDF-1.0)
- objects (0 0 obj ... endobj)
- xref (xref 0 1 0000000000 65535 f)
- trailer (trailer << ... >>)
- startxref (startxref 0)
- end of file (%%EOF)

example

a non-working minimal file looks like this:

```
%PDF-1.0
0 0 obj
endobj
xref
0 1
0000000000 65535 f
trailer
<< >>
startxref
0
%%EOF
```

content

various document-viewers in linux are able to view pdf-files without xref and startxref. this is where pdf-files can be created by hand. adobe pdf-reference shows how to create a minimalistic pdf-file.

```
%PDF-1.7

1 0 obj
  << /Type /Catalog /Pages 2 0 R >>
endobj

2 0 obj
  << /Type /Pages /Kids [3 0 R 3 0 R] /Count 2 >>
endobj

3 0 obj
  << /Type /Page /MediaBox [0 0 243 153] /Parent 2 0 R /Resources 4 0 R /Contents 5 0 R >>
endobj
```



```
4 0 obj
  << /ProcSet [/PDF /Text] /Font << /schrift_normal 6 0 R >> >>
endobj

5 0 obj
  << /Length 0 >>
stream
  BT
    70 70 Td
    (hello) Tj
    T*
    (world) Tj
    10 20 Td
    (i like it) Tj
  ET
endstream
endobj

6 0 obj
  << /Type /Font /Subtype /Type1 /BaseFont /Helvetica /Encoding /WinAnsiEncoding >>
endobj

trailer
  << /Root 1 0 R >>
```

line-breaks can be inserted for human readability. comments can also be inserted but need to be started with % sign. in this example a single page is shown twice.

binary data

binary data can be inserted in hex-encoded form. in this case /Filter /ASCIIHexDecode need to be inserted in dictionary of object too. binary data need double size in hex-encoded form.

live preview

if this pdf-file is kept open in document-viewer, chances are high, that the document-viewer recognizes any file modification and reloads the entire pdf-file. in this case, the document-viewer can be used for a live-preview of what is typed in any regular text-editor.

strategy

most document-viewers can handle corrupt files because they use a different strategy to read the pdf-file. under normal circumstances, a document is read from end of file to find startxref to find xref to find objects.

editor

any simple text-editor can be used to edit pdf-files, as long as they do not contain binary data.



header

contains information which standard is being used. valid version number range from 1.0 to 1.7.

object

documents contains multiple objects of different types. an object contains an optional dictionary, an optional stream or a value.

further instructions on how to create pdf-files can be found in [adobe pdf-reference](#).

xref

contains information where each single object in document starts. xref contains two different things.

- start-id and counter
- pointer where object start, its generation number and a marker if the object is used

first element comes first and gives information about how many elements of second type follows and which objects they belong to.

xref can be created manually by counting the size of each object. this can be done by saving each object into a separate file and get its filesize. [adobe pdf-reference](#) can help.

a manually created pdf-file can be viewed by [adobe pdf-viewer](#) on android without xref. inside [adobe pdf-viewer](#) there is an option to draw something inside the document or apply some text. this option can be used to let the document-viewer create a correct xref, trailer and startxref.

trailer

contains information where object with /Type /Catalog is saved and how many objects are saved in document.

trailer should be available to give document-viewers a chance to find object with /Type /Catalog since it can be placed anywhere inside the document like any other object.

dictionary

contains information about the current object or document. a dictionary is optional in object but needed in trailer. a dictionary contains key-value-pairs, where the key starts with / in any case.

startxref

contains information about where xref starts.