PDF file format

In order to make a blank PDF file page, we'll have to deal with two structures:

1. File structure
2. Document Structure

Table of Contents

File structure defines all the data needed to parse a file as PDF format, while the document structure defines the content of the file body.

PDF file structure

Overview of file structure

A PDF contains 4 sections:

1. Header, defines the version of PDF specification
2. Body, the actual content that will be displayed.
3. Cross-reference table, a table for PDF viewers to quickly access different objects.
4. Trailer, defines other meta info of a PDF file.

Here is an image showing the file structure of PDF file: PDF file structure

Explaination of Sample PDF file

So now I'll show a working example: files/2015-08-15-mypdf.pdf .

```
%PDF-1.7
1 0 obj
  << /Type /Catalog
     /Pages 2 0 R
  >>
endobj
```

```
2 0 obj
  << /Type /Pages
     /Kids [3 0 R]
     /Count 1
  >>
endobj

3 0 obj
  << /Type /Page
     /Parent 2 0 R
     /MediaBox [0 0 600 400]
     /Resources << >>
  >>
endobj

xref
0 4
00000000000 65535 f
00000000010 00000 n
00000000069 00000 n
00000000141 00000 n
trailer
  << /Root 1 0 R
     /Size 4
  >>
startxref
249
%%EOF
```

Object Syntax

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

The above is one object.

1. Its name is `1`, `0` is its version number, normally they are not used.
2. `obj` and `endobj` delimit the beginning and end of an object.
3. `<< >>` defines an dictionary object.

You can refer to the Adobe PDF references for details of the synatx.

Note that althrough the object names in the example are `1, 2, 3, ...`, you can choose any name(in number perhaps?).

Header

The line `%PDF-1.7` is the header and defines that this file uses PDF 1.7 specification.

Body

The contents below the header and above the line `xref` are the body. In order to correctly show up a PDF, the body should have the structure defined by Document Structure, which we will talk later.

Cross-Reference Table

Well, this part is the hardest part to understand. If uncorrectly set, the PDF viewer will give out errors.

The cross-reference table are used for quick accessing every objects appear in the body. So we need to give every object an cross-reference entry.

```
xref
0 4
0000000000 65535 f
0000000010 00000 n
0000000069 00000 n
0000000141 00000 n
```

This Cross-Reference table begins with the keyword `xref` and an `EOL`.

Then comes a line to indicate the starting object: `0` in our case, and we have `4` sequential entries corresponds to `4` objects in the body.

But we only have objects `1, 2, 3` in the body, how is that? The 0th object is the root of the body(which is different to the `Document Catalog`) and will not show.

Next comes several entries, each entry comes in the format of

```
nnnnnnnnnn ggggg n eol
```

where

1. `nnnnnnnnnn` is a 10-digit byte offset of the object starting from the beginning of the document.
2. `ggggg` is a 5-digit generation number, to indicate which generation current object is. Each time the object is deleted and

then reused, it is given a new generation number.
3. entry type, `n` for in-use, `f` for `free` (not used).
4. `<space><linefeed>` in case of Unix EOL format, `<carriage return><linefeed>` in case of Windows EOL format.

In total the entry takes up exactly 20 bytes.

So in our case, the entry

```
0000000069 00000 n
```

is the third entry, since it starts from object 0, so it refers to object:

```
2 0 obj
  << /Type /Pages
     /Kids [3 0 R]
     /Count 1
  >>
endobj
```

And the offset of the this object is `69`, and we don't use any generation related feature(`00000`), and this entry is in used(`n`).

Trailer

Trailer section gives us the overall information of the PDF documents, it must contains a dictionary, which should have at least two entries: `/Root` and `/Size`.

```
trailer
  << /Root 1 0 R
     /Size 4
  >>
startxref
249
%%EOF
```

`/Root` refers to the `Catalog` of the body(Next section). `/Size` refers to the total number of entries in the file's cross-reference table.

`startxref` follows by a line of a number, indicates the start offset of the cross-reference table. i.e. the offset of the keyword `xref`.

`%%EOF` indicates the end of the file.

The main purpose of the trailer section is that the viewer can read the file from bottom, and:

1. find out the offset of the cross-reference table by `startxref` part.
2. find out the root object by the `/Root` entry in `trailer` part.

Then the viewer will be able to find the real content of root object(`1 0 R` in our example) in offset `0000000010` with the help of cross-reference table. The root contains object (`2 0 R`) as written in `/Pages 2 0 R`. So the PDF viewer will find goto offset `0000000069` with the help of cross-reference table. And so on until the whole PDF object tree are parsed.

Document Structure

A PDF document is orgnized in a tree hierarchy. The root of the tree is called `Document Catalog` and is specified by the `/Root` entry in `trailer`.

Document Structure

Here I'll cover only the simpliest case to help us create a blank PDF document.

Document Catalog

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

1. [Must] specify the `/Type` to `/Catalog`.
2. [Must] specify the `/Pages` entry.

Pages

`Pages` node is the root node of a page tree. Below is an exmple of page tree hierarchy.

```
2 0 obj
  << /Type /Pages
     /Kids [ 4 0 R
             10 0 R
             24 0 R]
     /Count 3
  >>
endobj

4 0 obj
  << /Type /Page
  ... Additional entries describing the attributes of
this page ...
  >>
endobj

10 0 obj
  << /Type /Page
     ... Additional entries describing the attributes
of this page ...
  >>
endobj

24 0 obj
  << /Type /Page
     ... Additional entries describing the attributes
of this page ...
  >>
endobj
```

1. `/Type` must be set to `/Pages`
2. `/Parent` should be set except in root node.
3. `/Kids` array should be sed to specify the child pages.
4. `/Count` must be set to specify the number of leaf nodes(page objects).

Page Object

```
3 0 obj
  << /Type /Page
     /Parent 2 0 R
     /MediaBox [0 0 600 400]
     /Resources << >>
  >>
endobj
```

1. `/Type` should be set to `/Page`
2. `/Parent` shoudl be set to its parent object.
3. `/MediaBox` is a rectangle on the page to store media contents.
4. `/Resources` contains any resources(e.g. fonts) that are required by this page.

Summary

This post explains the bootstrap information we need to know about PDF file format.

1. PDF file structure
2. PDF document structure

Of course with this we are not able to do fancy things yet, I recommand read the Adobe PDF specification after this. So, enjoy.

References

- Make your own PDF file
- Hand Coded PDF tutorial
- Adobe PDF references

Load Disqus comments