一个简单的 PDF 文件结构的分析

Adobe 的 PDF 参考告诉我们一个 PDF 文件可以通过下面 4 个方面来理解：

1. 对象, 一个 PDF 文档是由一个由基本数据类型组成的数据结构。

2. 文件（物理结构）, 决定对象是如何存放在一个 PDF 文件中的， 它们是如何被访问的，如何被更新的。这个结构是独立于对象的语义的。

3. 文档结构, 说明一些基本的对象类型是如何来表现 PDF 文档的成分的：页，字体，批注，和另外一些内容。

4. 内容流.一个 PDF 文件内容流包含一系列的指令，描述页面的外观或其他图形实体的外观和文件内容。

但是当时对我来说要看懂这几行字是有很大的困难的，需要了解确切含义，必须看完后面的几十页上百页的内容并且要分析一下一个实际的 PDF 文件才能完全领会它的意思。

后来经过长时间的文档阅读，相关开发，并且具体地分析 PDF 文件后才把 PDF 文件的语法，文件的解析搞清楚。虽然说学习是痛并快乐着，但是对于当时我来说真的希望有一个人能够告诉我一个简单的例子，通过一个简单的例子来描述 PDF 的基本组成，它的解析原理和过程。因此下面我主要将以一个简单的例子来说明 PDF 的主要特性并给出一个简单的 PDF 文件的全景。

在继续阅读该文章前，我们先问自己下面的几个问题：

- 你了解至少一种文件格式吗？（例如 HTML）
- 为什么要学习 PDF 的相关知识？

如果你对第一个问题的答案为“是”, 并且第二个问题你能给出一个非常明确的答案，那么这篇短文是适合你的。否则，如果对任何一种格式都不了解，建议先了解一下 HTML，或 XML，你可以从这两种语言里得到很多启发，对学习 PDF 的构成有很大的好处；如果你不清楚你要学习是为了什么，那么我就认为你学习没有目的性和动力，说不定你今天学了以后明天就忘得一干二净。

1．PDF 格式和 HTML,XML 格式：

一个 PDF 文档从根本上来说是一个 8 字节序。其实 PDF 格式和我们已经熟知的 HTML，XML 等结构化的文件格式一样，包含有关键字，分隔符，数据等等。

不同的是 PDF 文件是按照二进制流的方式保存的，而 html 文件则是文本方式保存的。XML 文件一般只包含数据本身，并没有把如何显示的信息放在其中，因此要显示一个 XML 文件还需要一个 Schema 文件才能显示，否则看到的将是所有的字节流；HTML 包含了数据的同时也包含了一些关于如何显示的信息，但是 HTML 是基于文本存放的，是可读的，你打开一个 HTML 文件就能知道所有显示在浏览器里得文字。另外就是 HTML 不能包含二进制流，它对图像文件的引用都是通过链接的，全部是外部文件的方式来实现的。

2．PDF 规范的发展

PDF 规范从 1993 年到现在，已经有过 7 个版本，六次版本升级，从最初的 pdf1.0.6 版本到现在的 PDF1.6, 每次的版本升级都会加入一些新的特性，PDF 参考说明书也是从最初的 100 多页到现在的 1000 多页，但是 PDF 文件格式的主要特性还是没有改变，可以这么理解，PDF1.6 是 PDF1.0 的扩展集，学习了 PDF1.0 以后也能基本上理解 PDF1.6 的内容。因此说我下面的例子是基于一个 PDF1.0 的最简单的一个 PDF 文件的分析。

PDF 规范的发展升级：

1.1 1995 加入了文档加密（40 字节），线索树，名字树，链接，设备独立色彩资源。

1.2 1996 表单, 半色调屏幕，和其他的一些高级色彩特性, 对中文，日文和韩文的支持

1.3 2000 数字签名, 逻辑结构， JavaScript, 嵌入式文件，Masked Images, 平滑阴影, 支持 CID 字体的附加色彩。

1.4 2001 文件加密 (128 字节), 标签式 PDF, 访问控制，透明，元数据流

1.5 2003 文档加密 (公钥), JPEG 2000 压缩, 可选的内容组，附加的注解类型

1.6 2005 文档加密 (AES),增加最大文件支持，加入 3D 支持，额外的注解类型

3．PDF 文件的基本组成：

一个 PDF 文件从大的方面来说分 4 个部分：

l 文件头，指明了该文件所遵从的 PDF 规范的版本号，它出现在 PDF 文件的第一行。

l 文件体，PDF 文件的主要部分，由一系列对象组成。

l 交叉引用表，为了能对间接对象进行随机存取而设立的一个间接对象的地址索引表。

l 文件尾，声明了交叉引用表的地址，即指明了文件体的根对象（Catalog），从而能够找到 PDF 文件中各个对象体的位置，达到随机访问。另外还保存了 PDF 文件的加密等安全信息（以后详细讨论）。

如下图：

图 1

4．PDF 文档的逻辑结构

作为一种结构化的文件格式，一个 PDF 文档是由一些称为“对象”的模块组成的。并且每个对象都有数字标号，这样的话可以这些对象就可以北其他的对象所引用。这些对象不需要按照顺序出现在 PDF 文档里面，出现的顺序可以是任意的，比如一个 PDF 文件有 3 页，第 3 页可以出现在第一页以前，对象按照顺序出现唯一的好处就是能够增加文件的可读性，如果你不会用文本编辑器来阅读 PDF 结构，那么大可不必关心。正是因为页与页之间的不相关性，就可以对 PDF 文件的页码进行随机的访问。

文件尾（Trail），说明根对象的对象号，并且说明交叉引用表的位置，通过对交叉引用表的查询可以目录对象(Catalog)。这个目录对象是该 PDF 文档的根对象，包含 PDF 文档的大纲(outline)和页面组对象（pages）引用。大纲对象是指 PDF 文件的书签树；页面组对象（pages）包含该文件的页面数，各个页面对象(page)的对象号。

一个 PDF 文档有下图所示的层次关系：

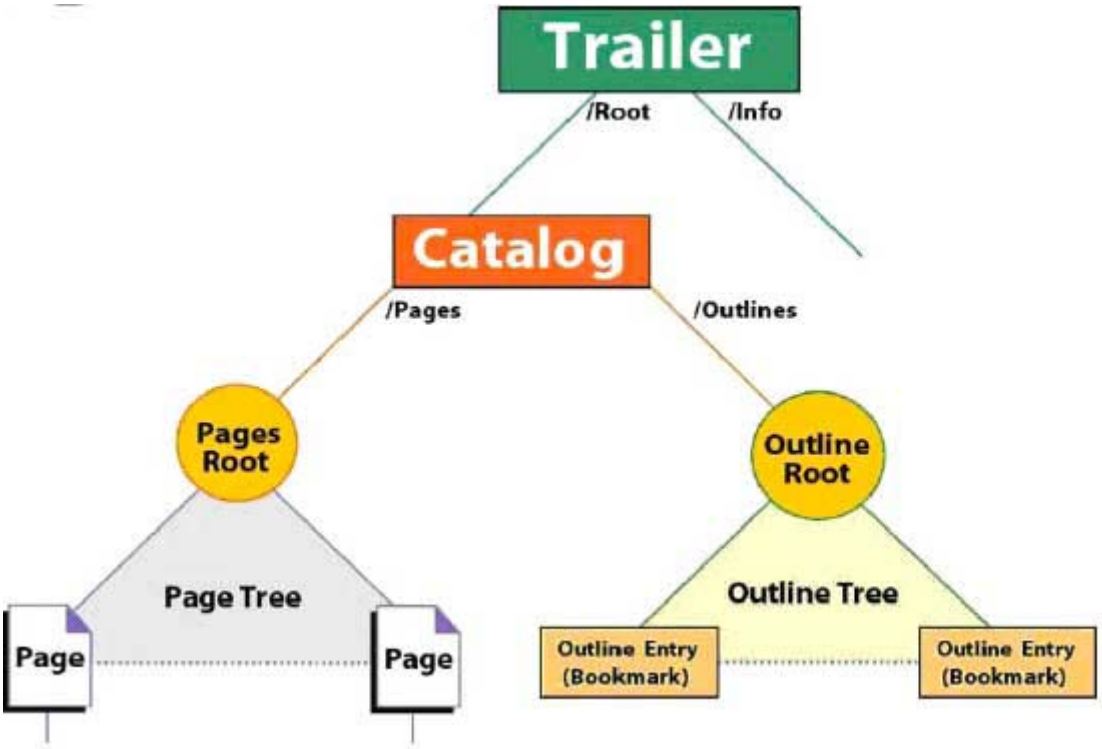

图 2

页面（page）对象作为 PDF 中最重要的对象，包含如何显示该页面的信息，例如使用的字体，包含的内容（文字，图片等），页面的大小。当然里面的子项也可以是其他对象的引用。页面中包含的信息是包含在一个称为流（stream）的对象里，这个流的长度（字节数）必须直接给出或指向另外一个对象。如下图：

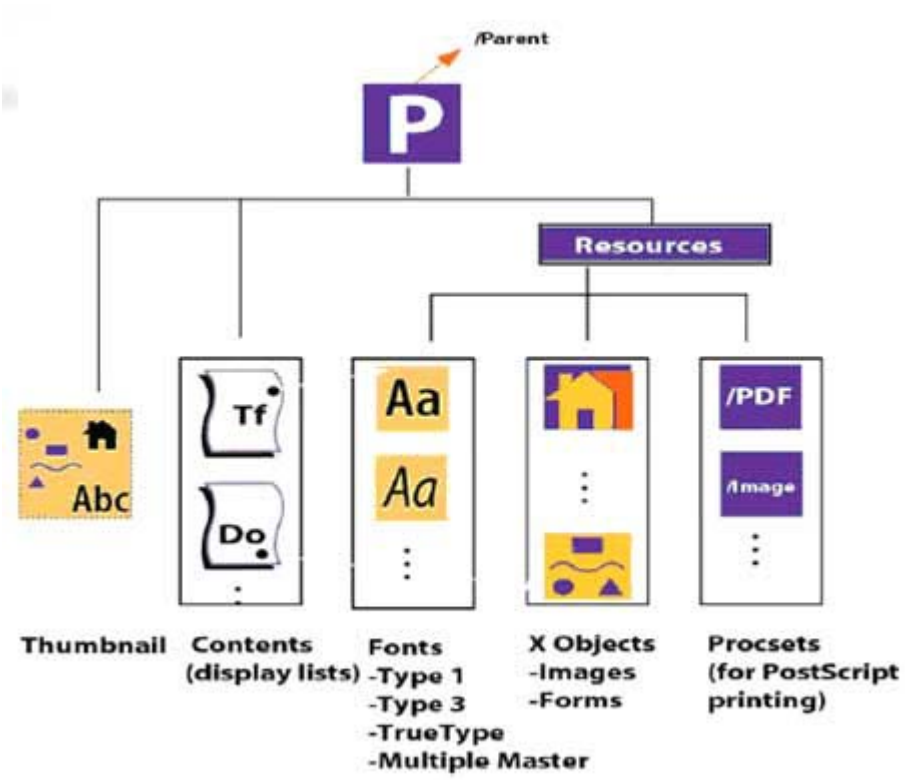

图 3

5． PDF 的基本语法：

文件的第一行是文件头，指明了该文件所遵从的 PDF 规范的版本号，它出现在 PDF 文件的第一行。

一个对象的第一行一般有两个数字和关键字“obj”。例如：

3 0 obj

<<

/Type /Pages

/Count 1

/Kids [4 0 R]

>>

endobj

第一个数字称为对象号，来唯一标识一个对象的，第二个是产生号，是来表明它在被创建后的第几次修改，所有新创建的 PDF 文件的对象号应该都是 0，即第一次被创建以后没

有被修改过。上面的例子就说明该对象的对象号是 3，而且创建后没有被修改过。

对象的内容应该是包含在<< 和>>之间的，最后以关键字 endobj 结束.

6. 文件 Hello World 的文件分析：

6．1．文件的具体分析

%PDF-1.0

文件头，说明符合 PDF1.0 规范

1 0 obj

<<

/Type /Catalog

/Pages 3 0 R

/Outlines 2 0 R

>>

endobj

Catalog 对象（根对象）

2 0 obj

<<

/Type /Outlines

/Count 0

>>

endobj

outline 对象（此处它的计数为 0，说明没有书签）

3 0 obj

<<

/Type /Pages

/Count 1

/Kids [4 0 R]

>>

endobj

pages 对象（页面组对象），/Type /Pages 说明自身的属性，对象的类型为页码，/Count 1 说明页码数量为 1，/Kids [4 0 R]说明页的对象为 4, 这里要说明的是如果有多个页面，就多个页面直接连续下去，比如说/Kids [4 0 R 10 0 R], 就说明该 PDF 的第一页的对象号是 4, 第二页的对象号是 10。

4 0 obj

<<

/Type /Page

/Parent 3 0 R

/Resources << /Font << /F1 7 0 R >> /ProcSet 6 0 R >>

/MediaBox [0 0 612 792]

/Contents 5 0 R

>>

endobj

页对象，/Parent 3 0 R 说明其父对象的对象号为 3，/Resources << /Font << /F1 7 0 R >> /ProcSet 6 0 R >>说明该页所要包含的资源，包括字体和内容的类型，/MediaBox [0 0 612 792]说明页面的显示大小（以象素为单位），/Contents 5 0 R 说明页面内容对象的对象号为 5。

5 0 obj

<< /Length 44 >>

stream

BT

/F1 24 Tf

100 100 Td (Hello World) Tj

ET

endstream

endobj

<< /Length 44 >>说明 stream 对象为字节数，从 BT 开始，ET 结束，包括中间的行结束符。

Stream 说明一个流对象的开始。

BT 说明一个文字对象的开始。

/F1 24 Tf，Tf 说明 True font 对象，字体明为 F1, 大小为 24 个象素。

100 150 Td (Hello World) Tj，100 100 说明这一行文字放置的位置，对于 Td, 我们可以这样理解，我们的当前 X,Y 坐标分别加上 100 和 150 就是文本的位置，因为在该例子中只有一个对象，那么它的位置就是(100,150), 如果下个对象位置信息为 100, 50 Td, 那么它的位置应该就是（100+100, 150+50）也就是（200，200）。（Hello World） Tj 说明文本的内容，当然，如果这里是文本的内容可以写成 16 进制，用<>包含。

ET 说明文字对象的结束

endstream 流对象的结束

6 0 obj

[/PDF /Text]

Endobj

[/PDF /Text]说明 PDF 的内容类型仅仅为文本，如果有图片则为[/PDF /Image]

7 0 obj

<<

/Type /Font

/Subtype /Type1

/Name /F1

/BaseFont /Helvetica

>>

endobj

Object six defines the

字体对象，不再多作解释。

所有的对象之后是下面的交叉引用表：

xref

0 8

0000000000 65535 f

0000000009 00000 n

0000000074 00000 n

0000000120 00000 n

0000000179 00000 n

0000000322 00000 n

0000000415 00000 n

0000000445 00000 n

xref 说明一个交叉引用表的开始，交叉引用表的第一行 0 8 说明下面各行所描述的对象号是从 0 开始，并且有 8 个对象。

0000000000 65535 f，一般每个 PDF 文件都是以这一行开始交叉应用表的，说明对象 0 的起始地址为 0000000000，产生号（generation number）为 65535，也是最大产生号，不可以再进行更改，而且最后对象的表示是 f,表明该对象为 free, 这里，大家可以看到，其实这个对象可以看作是文件头。

0000000009 00000 n 就是表示对象 1，也就是 catalog 对象了，0000000009 是其偏移地址，00000 为 5 位产生号（最大为 65535），0 表明该对象未被修改过, n 表示该对象在使用，区别与自由对象，不可以更改。

下面的几行相信大家就可以告诉我含义了。

Trailer

<<

/Size 8

/Root 1 0 R

>>

startxref

553

%%EOF

trailer

说明文件尾 trailer 对象的开始。

/Size 8 说明该 PDF 文件的对象数目。

/Root 1 0 R 说明根对象的对象号为 1。

Startxref

553 说明交叉引用表的偏移地址，从而可以找到 PDF 文档中所有的对象的相对地址，进而访问对象。

%%EOF 为文件结束标志。

6．2．PDF 解析过程

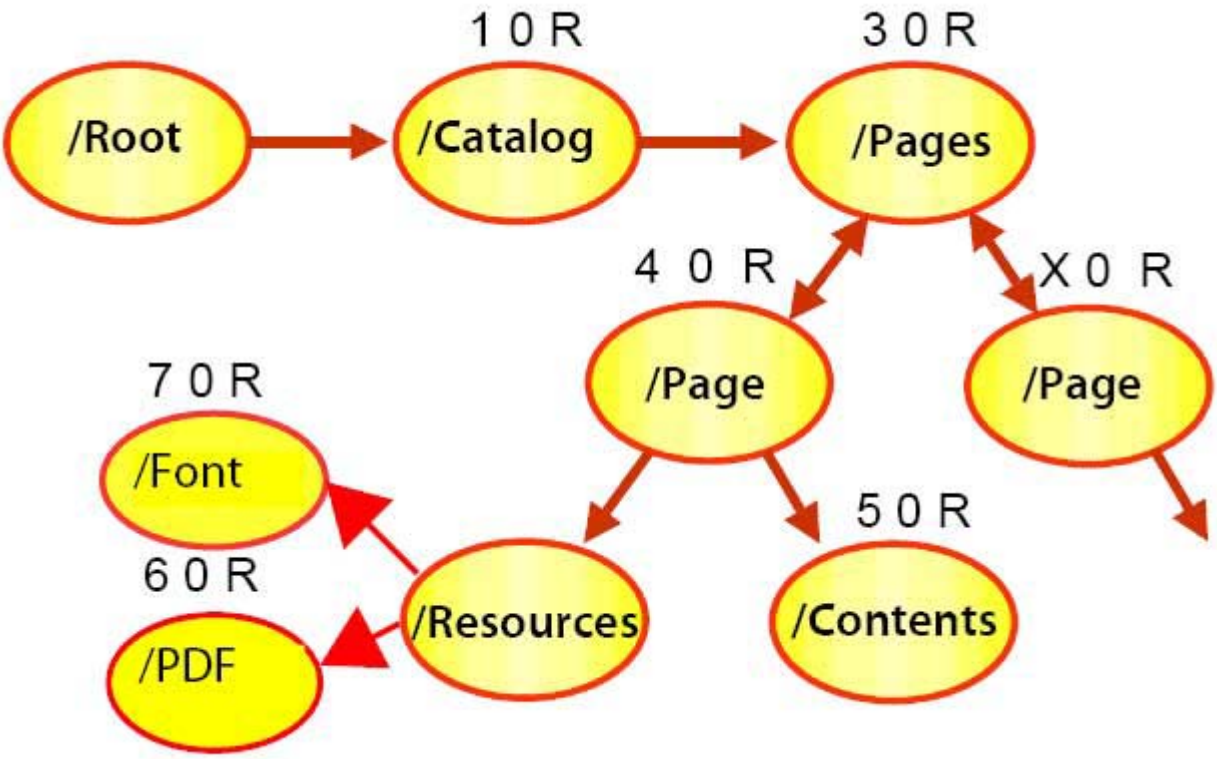

图 4

7．结束语：

到这里，我们对一个最简单的 PDF 文件的介绍就结束了，我想大家对 PDF 文件的格式和特定应该已经有所了解了。

当然，我这里介绍的是不完整的，完整的信息，请访问 adobe 的网站下载：

http://partners.adobe.com/public/developer/pdf/index_reference.html

下次介绍 PDF 的加密过程及原理。