**Очерк по поводу создания PDF-файлов**

Максим Фокин, "Королевство Delphi"

В последнее время на просторах интернета обнаружилось очень много PDF converter'ов, reader'ов и write'ов. И подавляющее большинство из них предлагается за деньги. Сама программа от 10$ до 300$. А уж исходный код за гораздо большие деньги цена начинается от 200$ а в одном месте (заинтересовавшись этим полазил по инету) аж за 900 евро.

Данная проблемма меня заинтересовала в плане программирования и вот результаты довожу до вашего сведения. (Данные результаты получены мною при изучении внутренностей PDF файла, когда открываешь его в total commander через F3)

Обычный PDF файл состоит из четырех частей

<PDF file> := <header>

<body>

<cross-reference table>

<trailer>

Что такое такое <header>? Это обычное упоминание о версии PDF specification. Которое присутствует в первой строке PDF файла. Например "%PDF-1.3" В седьмой версии акробата которая вышла где то в начале лета этого года, этот номер "%PDF-1.7", но это не версия продукта, это версия именно спецификации. Второй строкой PDF идет небольшая аброкадабра (видимо предназначена для дальнейшего использования) " %вгПУ"

Все с первой частью PDF разобрались.

Что из себя представляет вторая часть которая называется <body>?

Ответ очень простой: это последовательность объектов, описание которых как и хедера представлены в текстовом виде.

Каждый объект это текстовой фрагмент с порядковым номером в имени например "4 0 obj"

4 это порядковый номер объекта

0 это номер (ре)генерации файла то есть когда файл обновляется (редактируется ) то данный номер увеличивается

obj это кодовое слово означающее что в теле документа нам встретился объект

Все объекты делятся на косвенные и прямые. Все косвенные, и их большинство, после слова obj имеют в своем теле делиметер "<<", означающее начало данных объекта. И в конце данных закрывающий делиметер ">>" и кодовое слово endobj

Прямые объекты не должны иметь в своем теле открывающих и закрывающих делиметеров "<<", ">>" Все косвенные объекты доступны через cross-reference table. В ней представлены ссылки в виде смещения от начала файла до начала объекта (Данные (строки) в объекте разделяются #13#10 либо #13)

Тип "самого главного" объекта в теле PDF файла носит гордое имя "/Catalog"

4 0 obj

<<

/Type /Catalog

/Pages 2 0 R

/OpenAction [ 5 0 R /XYZ null 364 1 ]

/PageMode /UseNone

>>

endobj

На самом деле в теле минимального PDF файла типа "Hello world" должно быть 3 "главных" объекта. Давайте я их перечислю по типам:

"/Catalog" содержит в себе ссылку : на дерево страниц (/Pages)

"/Pages" содержит в себе ссылку на группу страниц документа

(Например

2 0 obj

<<

/Type /Pages

/Kids [ 3 0 R ]

/Count 1

>>

endobj

)

"/Page" содержит в себе ссылку на объекты относящиеся к конкретной странице.

(Например

3 0 obj

<<

/Type /Page

/Parent 2 0 R

/MediaBox [ 0 0 612 792 ]

/CropBox [ 0 0 612 792 ]

/Contents 4 0 R

/Resources << /Font 20 0 R /ProcSet [ /PDF /Text ] >>

/Rotate 0

>>

)

И несколько "второстепенных"

Разберем объект страница:

/Rotate поле показывающее на сколько градусов изображение страницы должно быть повернуто при отображение в программе

/MediaBox и /CropBox поля описывающие размер страницы

/Parent ссылка на родительский объект "/Pages"

/Resources это поле описывает какой фонт должен быть использован при отображении страницы (фонт это отдельный объект) и установку ProcSet эта установка показывает какое содержимое потока данных данной страницы (тоже может быть определен как объект, а не как поле)

/Contents Самое интересное поле в объекте "страница", дает ссылку на объект содержимого данной страницы, причем : если это поле отсутствует в объекте "страница" значит страница пустая

Содержимое страницы:

Объект "stream"

4 0 obj << /Length 305 >> stream

BT

/F12 9 Tf

10 782 TD

0 -12.5 TD

( Max Fokin) Tj

0 -12.5 TD

( mnb) Tj

0 -12.5 TD

() Tj

0 -12.5 TD

(Max Privet) Tj

0 -12.5 TD

( 1) Tj

0 -12.5 TD

(1) Tj

0 -12.5 TD

(2) Tj

0 -12.5 TD

(3) Tj

0 -12.5 TD

(45) Tj

ET

endstream endobj

/Length 305 - это поле показыввающее сколько байт от слова stream до слова endstream

Самый простой вариант — это некодированный и несжатый поток данных в объекте stream. Он ограничивается операторами BT и ET

BT Begins a Text Object - характеризует начало текста

ET Ends a Text Object. - характеризует конец текста

/F12 9 Tf -

/F12 это кодовое имя объекта который характеризует фонт используемый на данной странице

9 это размер фонта

Tf это оператор который характеризует что данная строка в объекте steam есть установка фонта и размера

10 782 TD - это цифры откуда начинается данная строка (отсчет производиться от левого верхнего угла)

Tj - это оператор перевода на новую строку

Ну а в круглых скобках наш текст

Кодированный поток я сдесь не объясняю. Он основан на алгоритмах RC4, RC5, MD5.

Что такое объект Font

12 0 obj

<<

/Type /Font

/Subtype /Type1

/Name /F7

/BaseFont /Courier-Oblique

/Encoding /WinAnsiEncoding

>>

/Type /Font Естественно название типа

/Subtype /Type1 название подтипа

/Name /F7 F7 это кодовое имя

PDF поддерживает несколько видов фонтов. Они перечисленны ниже

Type 1, including subsets and Multiple Master "snapshots"

Type 3

TrueType, including subsets

Type 0

Честно говоря, я не разбирался с Type 3, TrueType, including subsets, Type 0 ничего по ним сказать не могу.

А Type 1 — это следующие фонты

Courier

Courier-Bold

Courier-BoldOblique

Courier-Oblique

Helvetica

Helvetica-Bold

Helvetica-BoldOblique

Helvetica-Oblique

Times-Roman

Times-Bold

Times-Italic

Times-BoldItalic

Symbol

ZapfDingbats

20 0 obj

<<

/F1 6 0 R

/F2 7 0 R

/F3 8 0 R

/F4 9 0 R

/F5 10 0 R

/F6 11 0 R

/F7 12 0 R

/F8 13 0 R

/F9 14 0 R

/F10 15 0 R

/F11 16 0 R

/F12 17 0 R

/F13 18 0 R

/F14 19 0 R

>>

endobj

Это объект с названиями кодовых имен для фонтов первого типа. По этому кодовому имени можно легко получить сам объект фонт.

6 0 obj

<<

/Type /Font

/Subtype /Type1

/Name /F1

/BaseFont /Helvetica

/Encoding /WinAnsiEncoding

>>

ВСЕ: то есть минимальное <Body> Состоит из следующих объектов: "catalog" , "pages", "page", "Resources" (опиционально может присутствовать, как поле в объекте страница), нетипизированный объект "stream", группа объектов "font"

Что такое <cross-reference table>?

На самом деле это обычная текстовая таблица, она начинается со слова xref и своем теле имеет ссылки на все косвенные объекты в документе. Вот пример

xref

0 27

0000000021 65535 f

0000000016 00000 n

0000000105 00000 n

0000000169 00000 n

0000000356 00000 n

0000000713 00000 n

0000000892 00000 n

0000001006 00000 n

0000001125 00000 n

0000001247 00000 n

0000001373 00000 n

0000001486 00000 n

0000001604 00000 n

0000001725 00000 n

0000001850 00000 n

0000001967 00000 n

0000002084 00000 n

0000002203 00000 n

0000002326 00000 n

0000002439 00000 n

0000002558 00000 n

0000000024 00001 f

0000002751 00000 n

0000002831 00000 n

0000000000 00001 f

0000002915 00000 n

0000002955 00000 n

0 27 Эти цифры обозначают следующее :

0 - первый object number в таблице

27 - количество элементов таблицы

Первый элемент таблицы всегда иммет вид "XXXXXXXXXX 65535 f" где X это цифра, а 65535 это значение по умолчанию для первого элемента в таблице. Символ "f" обозначает "free", то есть объект не используется Ссылки на объекты, которые используются, в конце имеют символ "n"

Разберем элемент данной таблицы.

Первые 10 цифр — это смещение от начала файла до начала объекта.

0000000016 означает что через 16 байт от начала файла Вас встретит первое упоминание об объекте то есть, например, 4 0 obj

Вторые пять цифр — это номер генерации файла. Если файл только что создан, то они всегда нули. Если файл модифицируется, то это число увеличивается на единицу. То есть, 0000000024 00001 f

Канонический, только что созданный PDF файл, имеет только одну таблицу. Но, если файл редактируется, то таких таблиц может быть очень много.

Взаимосвязь таблиц осуществляется при помощи последнего элемента <trailer> и кодового слова startxref

Канонический, только что созданный PDF файл, имеет только одну таблицу, после таблицы идет элемент trailer

А после трайлера идет кодовое слово startxref, указывающее на смещение от начала файла до начала таблицы, вот пример.

trailer

<<

/Size 3

>>

startxref

173

%%EOF

Это значит, что через 173 байта от начала документа, будет присутствовать кодовое слово xref. Но, если файл был отредактирован, то последний в файле трайлер будет иметь вид:

xref

0 3

0000000000 65535 f

0000003609 00000 n

0000003832 00000 n

trailer

<<

/Size 3

/ID[<7a15ab3ed3999575ff2f3034104a82c1><7a15ab3ed3999575ff2f3034104a82c1>]

>>

startxref

173

%%EOF

Но, если мы обратимся к таблице, куда указывает ссылка startxref 173, то мы найдем следующую таблицу, а за ней трайлер, который будет иметь поле /Prev 3896

3 16

0000000016 00000 n

0000000664 00000 n

0000000936 00000 n

0000001106 00000 n

0000001133 00000 n

0000001250 00000 n

0000001395 00000 n

0000001811 00000 n

0000001992 00000 n

0000002180 00000 n

0000002360 00000 n

0000002760 00000 n

0000003438 00000 n

0000003516 00000 n

0000000776 00000 n

0000000916 00000 n

trailer

<<

/Size 19

/Info 1 0 R

/Root 4 0 R

/Prev 3896

/ID[<7a15ab3ed3999575ff2f3034104a82c1><7a15ab3ed3999575ff2f3034104a82c1>]

>>

startxref

567

%%EOF

Данное поле /Prev 3896 указывает нам на предыдущую таблицу, а ссылка startxref 567 указывает на следующую таблицу и так практически до бесконечности, пока в очередном поле startxref мы не увидим 0. Это значит, мы прочитали все таблицы.

В данном очерке, конечно, не хватает исходного кода. Вот и он: представлены два модуля основной "PDFDocument" и вспомогательный "PDFBaseFonts"