page

一:page description language

在其它角色，PDF 提供一个页描述语言，一种用图形描述页的形式来描述符合图形模式的描述语言。一个应用程序产生输出时通过 2 个阶段：

1: 应用使用页描述语言生成一个与设备无关的所要求的接口。

2: 一个程序控制一个特定的输出接口来将它渲染到设备里。

这两个阶段也许在不同的地方或不同的时间里被执行。页描述语言是一种紧凑的交换标准，与设备无关的交换的传输以及可打印的存储或者显示文档.

二:page tree

文档的页是通过 page tree 结构来被访问的，他在文档里定义了页。这个树结构允许 PDF 用户应用，使用仅被限制的内存，来快速的打开一个包含上千页的文档。这个树包含两种类型的节点---干节点，被称作 page tree 节点，叶节点，被称作页对象--以下部分讲述它的表现形式.应用程序应该去做好准备处理这样的节点的构造的树的结构的任何形式.最简单的结构应该是由直接引用了文档里所有页对象的单页树节点构成.然而,为了优化应用程序的性能,Adobe 以一种特殊的方式来设计树,被称作平衡树.在数据结构部分会详细介绍它的信息.

树的父节点是字典项,子节点是数组,数组里包含其它页的树信息或者页对象.count 是叶节点的数目,它表示在这个树里这个节点的后代们..

三:页对象

页树里的叶子就是页对象,每一个页对象都是字典里指定的一个文档里的单个页的属性.

page 对象的条目

关键字	类型	值
type	name	(必要的)这个字典描述的 pdf 对象的类型,必须是 page 或者 page 对象
Parent	dictionary	(必要的,必须是一个间接引用的)是这个页对象的直接父节点
LastModified	date	(当 pieceinfo 是父节点时是必选的,其它情况是可选项)页里的内容最近被修改的日期时间,当一个 page-piece 字典项是父节

		点时(即 pieceinfo),这个修改日期被用来确定应用程序数据字典项里的数据,这个数据包含符合页的当前内容.
Resources	dictionary	(必选项,可继承的)一个字典里包含所有的页的必要的资源,如果这个页不需要资源,这个条目的值就是空的,省略整个可显示的条目来使资源能够从树里的祖节点别继承.
MediaBox	rectangle	(必选的,可继承的)一个矩形,默认表示用户空间单元,定义了显示或者打印的边界
CropBox	rectangle	(可选的,可继承的)一个矩形,定义了默认用户的可见区域空间,当显示或者打印页时,它的内容就会在这个内容里裁剪或者省略,然后被加到输出媒介中,默认值是 MediaBox 的值.
Tabs	name	(可选的,PDF1.5)页里标注的命名,可能的值是 R 或 C 或 S

四：页面属性的继承

如果这样一个属性值在页对象里被省略，它的值在一个树的祖节点就是可继承的。如果它的属性值是一个必选项，那么祖节点必须提供一个值给它。如果它的属性值是可选的并且没有可继承的值，那么就使用默认值。

一个属性在整个页设置时被中间页树节点指定定义一次，并且安排页面来共用祖节点安排的属性值。例如，一个文档，可能对所有页指定相同的 media box，这个 media box 在树的路径节点里包含一个 MediaBox 条目。如果有必要，单个页对象可以用他自己的 MediaBox 条目来覆盖被继承的值。

四、Thumbnail Images

一个 PDF 文档可以定义缩略图以缩影的方式来展现它的页的内容。一个查看器阅读程序能在屏幕上展现这些图形，允许用户点击缩略图去操纵一个页。

一个页的缩略图是一个 image XObject，在 page 对象里是 Tumb 条目，它拥有一个 image 字典的常用结构，但是只有 Width, Height, ColorSpace, BitsPerComponent 和 Decode 条目是值得注意的，其它的条目目前为止都可以忽略，图形的颜色区域必须是 DeviceGray 或者 DeviceRGB，或者是一个 Indexed 是基于这些的。

common data structure

这部分主要介绍 text strings, dates, rectangles, name trees, and number trees 这些类型的数据结构，随后的两部分会介绍更复杂的数据结构。

所有的数据结构仅在作为文档层次的一部分时才有意义，它不能出现在内容流。尤其，特殊约定 string 对象的值应用在仅在 strings 的内容流的外部。一个完全不同的约定被用于在内容流里面的情况是使用 strings 来选择符号序列来描述到页里。

一.string 类型

PDF 支持 string 类型和 text string 类型,从 PDF1.7 开始,string 类型进一步限定为 PDFDocEncoded string, ASCII string, 或者 byte string,string 更深层次的应用就是被用作编码字符串和字形。

string:PDF1.6 或者更早,这个类型被用于任何 string 除了 text string,他被更深层次的定义了 PDFDocEncoded string, ASCII string 和 byte string,string.

text string:被用于可读的字符串,像 text annotation\书签名\article names\文档信息.这些 strings 都是被 PDFDocEncoding 或者 byte-order 标记的 UTF-16BE 来编码的.

PDFDocEncoded string:单字节的字符串或者字形时使用的,这种类型,比 text string 编码更加特殊.

ASCII string:用来代表用 ASCII 的单字节的字符串

byte string:用二进制数来代表一系列的 8-bit 的 bytes,每个 byte 能用 8bits 代表任何值,这个 string 能代表字符串或者字形但是它的编码不是可知的,这个 string 的字节不能代表字符串,这种类型常被用于 MD5 的哈希值,签名证书以及 Web Capture 的标示值.

Text String type

text string type 是被用于字符串,这个字符串包含的信息被用在可读性,像 text annotation\书签名\article names\文档信息,或者更广.他被术语称作 character string,被用来描述这些 strings 不依赖于编码,在一个 PDF 文档里被作为代表.

后来的 text 被写入,换码顺序就会在一个 Unicode text string 里的任何地方出现, 来象征它的语言,这对于从 text 里的 character codes 不能被定义时很有用.换码顺序由以下几个元素

组成,依次是:

1. Unicode 值为 U+001B(那就是说,比特顺序是 0 到 27);

2. 一个 2-character ISO 639 语言编码--例如,en 代表英语或则 ja 代表日语.Character 在这个环境表示比特(就像在 ASCII 字符),不是 Unicode 字符.

3. (可选的)一个 2-字符的 ISO 3166 国家编码---例如,US 代表美国或者 JP 代表日本.

4. Unicode 值为 U+001B.

编码的完整列表被 ISO 639 和 ISO 3166 定义的能从国际标准化组织中获得.

PDFDocEncode String 类型

一个 PDFDocEncoded string 类似于一个 string 对象,但是他是一个字符串,这个字符串里的字符是以一个使用 PDFDocEncoded 的单字节表示的字符.需要注意 PDFDocEncoded 不支持所有的 Unicode 字符串然而 UTF-16BE 支持.

注:这个类型不是一个真实的类型.然而,它是一个串的类型来代表数据编码的用的是一个特殊的规则.

Byte String 类型

这个比特串类型是用于表示二进制数据来代表一系列的 8-bit 字节,每个字节在一个 8 比特里能是任何能被代表的值.这个串能代表字符但是它的编码是不可知的.串的字节也许不能代表字符.

注:这个类型不是一个真实的类型,当然,他是一个串的类型来代表数据的但是它的编码是不可知的.

二 Text Streams

一个文本刘(PDF1.5)是一个 PDF 流对象,它的不可编码的字节与一个文本串一样有相同的要求,包括编码、字节顺序和引导字节。

三 Dates

PDF 定义了一个日期格式的标准，与 ASN.1 国际标准有着密切关联，在 ISO/IEC8824

被定义。一个日期是一个 ASCII 串类型格式（D：YYYYMMDDHHmmSSOHH'mm'）

YYYY 是年

MM 是月

DD 是天（01-31）

HH 是小时（00-23）

mm 是分钟（00-59）

SS 是秒（00-59）

O 是世界时间与当地时间的关联（UT），用+，-或则 Z 字符表示

HH 与'是与 UT 时间里小时（00-23）差值的绝对值

mm 与'是 UT 时间分钟（00-59）差值的绝对值

字符串撇号（'）在 HH 和 mm 后面的是语法里的一部分。年后面的区域都是可选的。（前缀 D:也是可选的，是强烈被建议的写的）MM 和 DD 的默认值都是 01；所有其它的数字区域的默认值都是 0.一个加号标志（+）在 O 区域的表示表示本地时间比 UT 时间晚，减号(-)表示本地时间比 UT 时间早，Z 表示本地时间与 UT 时间一致。如果没有指定 UT 信息，那本地时间与 UT 时间的关系是未知的。不管时区是否可知，其余的日期应该被指定为本地时间。

例如，11 月 23 日，1998 年，在下午 7:52，U.S.太平洋标准时间，有下面的字符串代表

D：199811231952-08'00'

四 Rectangles

矩形为各种对象来描述在页里的位置和边界，像字体。一个矩形被写成一个数组的四个数字给一对对角的角落给与了坐标。代表性的，这个数组的格式

[ll_x ll_y ur_x ur_y]

指定矩形坐标的左下 x，左下 y，右下 x 和右下 y，以这个顺序。矩形两个角落坐标就是（ll_x，ur_y）和（ur_x，ll_y）。

注：虽然矩形通常被它们的左下和右下角落指定，他也是可以被任意两个对角角落指定的。PDF 应用程序应该使用标准化的矩形来指定对角坐标。

五 Name Tree

一个 name tree 与字典有类似的作用---把值和关键字结合在一起--但是有不同的含义。一个 name tree 与字典的不同处有以下几个地方：

- 与字典里的关键字不同，字典里的是对象名，name tree 里的是字符串。
- 关键字是有序的。
- 关键字的值也许是对象或者是任何类型。流对象需要被间接对象引用。建议，不是必须，字典、数组和字符串对象都能被间接对象引用，其它 PDF 对象（nulls, numbers, booleans,
and names）是被直接对象定义的。
- 数据结构能够代表任意大集合的关键字-值对，它能够被有效的查出不需要在 PDF 文件中读出整个数据结构。（与此相反，字典是受到它能包含的条目数的限制的）

一个 name tree 有节点构成，每个都是一个字典对象。节点有三种类型，这取决与他们所包含的特定条目。这个树通常正好有一个根节点，它包含一个单个条目：Kids 或者 Name 但是不可以两者都办好。如果一个根节点有一个 Names 条目，这就是这个树里的唯一节点，如果他有一个 Kids 条目，每个剩余的节点是一个中间节点，包含一个 Limits 条目和一个 Kids 条目，或者一个叶节点，包含一个 Limits 条目和一个 Names 条目。

Kids 条目在很和中间节点定义树的结构通过定义每个节点的直接子。Names 条目在叶节点（或者根节点）包含树的关键字和它的值，安排在键值对并且按照关键字的升序来排列的。短一些的关键字出现在长一些的前面，开始时使用相同的字节顺序。关键字的编码并不重要，只要它是首尾一致的就行；关键字与简单的逐字节相比是平等的。

各个节点包含的 Names 条目的关键字不重叠；每个 Names 条目的范围是树里的所有关键字。在一个叶节点，Limits 条目定义了节点的 Names 条目的最小和最大的值。在一个中间节点，它定义了任何后代节点的 Names 条目的最小和最大的关键字。遍历树就能找到给与的每个关键字的值，查找叶节点的 Names 条目也包含它的关键字。

六 Number Tree

一个 number tree 与一个 name tree 类似，除了它的关键字是整数来代替字符串和按数值的升序排列以外。叶节点（或者根节点）包含关键字-值对是命名为 Nums 来替代 Names 在一个 name tree 里。