

pdf reference 格式详细说明

1. PDF概要

1.1. 图像模型

PDF能以平台无关、高效率的方式描叙复杂的文字、图形、排版。

PDF 用图像模型来实现设备无关。图像模型允许应用程序以抽象对象描叙文字、图像、图标，而不是通过具体的像素点。把描述抽象对象的语言称作“页描述语言”。PDF文档中存储都是以“页描述语言”表示的各种抽象对象。

要把PDF文档输出到具体设备上，产生具体像素点，需要有个翻译软件，把抽象对象渲染到具体输出设备上，称这个“翻译软件”为“规则扫描”软件。

1.2. 属性

PDF 具有设备无关、可压缩、安全、增量修改、可扩展等特性。

2. 语法

2.1. 字符集

PDF是大小写敏感的语言的，及A和a是不一样的。

PDF字符集分文三种：

1).空白字符

空白字符包括Null(NUL)、Tab(HT)、Line Feed(LF)、Form Feed(FF)、Carriage return(CR)、Space(SP)。除了在comment、string、streams中，所有的空白符都是一样的。PDF把多个连续的空白符看作一个。

2).分割字符

分隔字符包括(,), <, >, [,], {, }, /, and %。分割字符是用来分开需实体对象的，包括string、comment、arrays、names。

3).常规字符

除了上面的两种字符之外的字符都称作常规字符。

2.2. 注释 (comment)

注释没有语义，只是注释说明作用。注释以%开头，到本行结尾。例如

```
abc% comment {/%) blah blah blah
```

123

“comment {/%) blah blah blah” 为注释。

2.3. 对象

3.3.1 Boolean 对象

Boolean 对象只有**true**和**false**两种值，可以用在**array**和**dictionary**中。**true**和**false**是关键字。

3.3.2 Numeric 对象

Numeric对象是数组对象，它的取值范围和具体的机器有关，具体范围可以参考“《PDF reference》 appendix C”。

Numeric 对象分为**Integer**和**Real**对象。**Integer**对象就是证书对象，可以在数字前面加+/-的符号，标识正负。例如。

```
123 43445 +17 -98 0
```

Real对象是实数对象。需要注意的是PDF的**Real**对象是定点小数，而不是浮点小数，不支持像“**6.02E23**”这样的标识。正确标识例如：

```
34.5 -3.62 +123.6 4. -.002 0.0
```

如果**Integer**对象超出的标识范围，会自动转化为**Real**对象。但是如果**Real**对象超出标识范围，会报错。

所有可以用**Real**对象的地方都可以用**Integer**对象，**Integer**对象会自动转换为**Real**对象。所以对于需要**Real**对象**1.0**的地方，直接写**1**就可以，不用写成**1.0**。

3.3.3 String 对象

String对象是一个字节数列的集合，字节的取值范围为**0~255**。字符串的长度范围跟具体平台有关，具体参考“《PDF reference》 appendix C”。

String对象分为**Literal string**和**Hexadecimal string**。

Literal string对象是用（）括起来的一些字符，除了不对称的括号（如“(“或者”）”）或者反斜杠（“\”）外的字符，都可以出现在**Literal string**中。例如：

(This is a string)

(Strings may contain newlines

and such.)

(Strings may contain balanced parentheses () and

special characters (*!&}^% and so on).)

(The following is an empty string.)

()

(It has zero (0) length.)

在Literal String 可以使用转意字符，如表1

TABLE 3.2 Escape sequences in literal strings

SEQUENCE	MEANING
\n	Line feed (LF)
\r	Carriage return (CR)
\t	Horizontal tab (HT)
\b	Backspace (BS)
\f	Form feed (FF)
\(Left parenthesis
\)	Right parenthesis
\\	Backslash
\ddd	Character code ddd (octal)

所以不在表1中的转义字符会被忽略。

如果一个字符串太长，放在一行不在方便，可以分开写在多行，然后用反斜杠连接起来，如下。

(These \

two strings \

are the same.)

(These two strings are the same.)

如果不加反斜杠，就会相当在末尾加了一个\n。如下

(These

two strings

are the same.)

(These \n

two strings \n

are the same.)

\ddd 标识是一个八进制数，提供一种标识不可显示的ASCII的方式。如

(This string contains \245two octal characters\307.)

\ddd 只需 “\” 后后面三位数，包括0，例如

\0053是表示\005和3。而不是\053或者\53。

Hexadecimal string 是用尖括号<>括起来的0-9和A-F（a-f）的字符，如

<4E6F762073686D6F7A206B6120706F702E>

每一对表示一个字节。如果字符是奇数，在字符最后字符补零，组成一对。例如：

<901FA3> 标识的90、1F、A3。如果是<901FA>会是90、1F、A0。

3.3.4 Name objects

Name对象一串原子、唯一的字符，用来标识一个对象。原子性是说Name对象没有内部结构了。唯一性是说如果两个Name是一样的，说明它们标识的师动一个对象。

Name对象以 “/” 开头，后面紧跟对象名，“/” 和第一个字符之间不能有空格。“/” 并不是Name对象的一部分，只是用来说明 “/” 后面的字符是Name对象。例如：

`/Name1`

`/ASomewhatLongerName`

`/A;Name_With– Various***Characters?`

`/1.2`

`/$$`

`/@pattern`

`/.notdef`

Name对象一般是由“常规字符”组成，不能是“空白字符”和“分割字符”。

但是在PDF1.2中，可以包括“空白字符”和“分隔字符”，但是必须以“#”开头，紧跟相应字符的两位十六进制编码。例如表2。`#20`代表的一个字符，而不是三个字符。例如`/A#20B`是四个字符，而不是六个。

Literal Name	Result T
<code>/Adobe#20Green</code>	Adobe Green
<code>/PANTONE#205757#20CV</code>	PANTONE 5757 CV
<code>/paired#28#29parentheses</code>	paired()parentheses
<code>/The_Key_of_F#23_Minor</code>	The_Key_of_F#_Minor
<code>/A#42</code>	AB

3.3.5 Array 对象

Array对象是一个一维对象集合，集合可以包括String对象、Name对象等任何其他对象。如果要组成多维数组，可通过Array对象中包含Array对象的方式。

Array对象的标识方式：

`[549 3.14 false (Ralph) /SomeName]`

3.3.6 Dictionary 对象

Dictionary对象是键值对的集合，用来描述复杂对象。键必须是**name**对象，值可以是任何对象，包括**Dictionary**对象。

Dictionary对象是用“<<>>”括起来的表示，例如：

```
<< /Type /Example
```

```
/Subtype /DictionaryExample
```

```
/Version 0.01
```

```
/IntegerItem 12
```

```
/StringItem (a string)
```

```
/Subdictionary << /Item1 0.4
```

```
/Item2 true
```

```
/LastItem (not!)
```

```
/VeryLastItem (OK)
```

```
>>
```

```
>>
```

每个**Dictionary**对象代表一个复杂对象，每个键值对表示对象的一个属性。

按照惯例，每个**Dictionary**都会有一个**Type**键值对，用来标识这个**Dictionary**描述的对象类别。一般还会有个**SubType**键值对，用来标识**Dictionary**标识对象子类别。例如一个字体的**Dictionary**的**Type**键的值是**Font**，**subType**键的值是**Type1**，**TrueType**，或者别的字形名称。

3.3.7 Stream 对象

Stream对象和**String**对象一样也是一个字节序列，但是不一样的地方，**Stream**对象可以被**PDF**应用程序递增的读取，而不是像**String**对象要整个读入内存。**Stream**对象可以不限长度的，而**String**对象是长度后受平台最大字符长度的限制。

Stream对象包括一个**Dictionary**对象和

stream和**endstream**关键字，以及**stream**和**endstream**中间包含的一行或者多行字节。格式如下：

```
dictionary
```

stream

...Zero or more lines of bytes...

Endstream

Stream对象必须包含在Indirect对象中，并且Dictionary对象不是Indirect对象。

关键字stream必须以“\r\n”或者“\r”结尾，不能以“\n”结尾。组成Stream对象的字节位于关键字stream及endstream中间。Dictionary对象一个属相用来说明字节的长度。在PDF1.2中，组成Stream对象的字节也可以来自文件，在Stream对象的Dictionary对象中指定文件的路径。在这种情况下，关键字stream和endstream中间的字节是被忽略的。

在表3中列出了，Stream对象中的Dictionary对象常有一些属性。

KEY	TYPE	VALUE
Length	integer	(Required) The number of bytes from the beginning of the line following the keyword stream to the last byte just before the keyword endstream. (There may be an additional EOL marker, preceding endstream, that is not included in the count and is not logically part of the stream data.) See “Stream Extent,” above, for further discussion.
Filter	name or array	(Optional) The name of a filter to be applied in processing the stream data found between the keywords stream and endstream, or an array of such names. Multiple filters should be specified in the order in which they are to be applied.
DecodeParms	dictionary or array	(Optional) A parameter dictionary, or an array of such dictionaries, used by the filters specified by Filter. If there is only one filter and that

		<p>filter has parameters, DecodeParms must be set to the filter' s parameter dictionary unless all the filter' s parameters have their default values, in which case the DecodeParms entry may be omitted. If there are multiple filters and any of the filters has parameters set to nondefault values, DecodeParms must be an array with one entry for each filter: either the parameter dictionary for that filter, or the null object if that filter has no parameters (or if all of its parameters have their default values). If none of the filters have parameters, or if all their parameters have default values, the DecodeParms entry may be omitted. (See implementation note 7 in Appendix H.)</p>
F	file specification	<p>(Optional; PDF 1.2) The file containing the stream data. If this entry is present, the bytes between stream and endstream are ignored, the filters are specified by FFilter rather than Filter, and the filter parameters are specified by FDecodeParms rather than DecodeParms. However, the Length entry should still specify the number of those bytes. (Usually there are no bytes and Length is 0.)</p>
FFilter	name or array	<p>(Optional; PDF 1.2) The name of a filter to be applied in processing the data found in the stream' s external file, or an array of such names. The same rules apply as for Filter.</p>
FDecodeParms	dictionary or array	<p>(Optional; PDF 1.2) A parameter dictionary, or an array of such dictionaries,</p>

		used by the filters specified by FFilter. The same rules apply as for DecodeParms.
--	--	--

3.3.8 Null 对象

Null对象不等于任何对象，用关键字**null**代表。**Indirect**对象引用不存在对象等价与引用空对象。

3.3.9 Indirect 对象

任何的对象都可以声明为**Indirect**对象，**Indirect**对象用一个**Object Identifier**标识这个对象，以便其他对象可以通过**Object Identifier**引用它。

Object Indirect有两部分组成：

- 1).Object Number 由一个正数表示，一般PDF中的**Object Number**是连续的，但不是必须的，他可以是任意顺序。
- 2).Generation Number由一个非负数标识。在一个新建立的PDF文档中，**Generation Number**是0。当文档被更新或者修改以后，**Generation Number**会被修改为非零值。

除了**Object Identifier** ,**Indirect** 对象还包括关键字**obj**和**endobj**以及他们中间包含的对象。例如：

12 0 obj

(Brillig)

Endobj

12 为**Object Number**，0为**Generation Number**，**Indirect**对象值为**(Brillig)**。

这个对象可以被引用通过**Object Indirect** 和关键字**R**。例如：

12 0 R

如果引用一个没有定义的**Indirect**对象，并不报错，而是等价于引用了一个**Null**对象（值为**null**）。

例子**Example 3.1** 定义了一个**Indirect**对象，它的**Object Identifier**是**7 0**，

它长度有引用了一个**Object Identifier**为**8 0**的**Indirect**对象。

Example 3.1

7 0 obj

<< /Length 8 0 R >> % An indirect reference to object 8

stream

BT

/F1 12 Tf

72 712 Td

(A stream with an indirect length) Tj

ET

endstream

Endobj

2.4. 过滤器 (Filter)

Filter 是Stream对象中引入的，可以通过Stream对象中的Dictionary对象的

Filter键表示（如果字节通过文件引入，是FFilter键），有一写Filter需要一些参数可以通过DecodeParms（如果字节通过文件引入，是FDecodeParams）

/Filter [/ASCII85Decode /LZWDecode]

在创建PDF过程中，应用程序用对PDF中的数据做了压缩或或者转码，等到有应用程序读取PDF的时候，在用对应的Filter对数据进行解码，然后输出到显示设备，进行显示。

如果存在多个Filter，这些Filter会组成一个管道，即第一个Filter输出，会作为第二个Filter的输入，第二个Filter的输出回作为第三个Filter的输入，一次类推。

Filter的种类大概分为两类：

ASCII Filter用来解码恢复已经倍编码为ASCII文本的数据。

Decompression filters 用来解压那些被压缩的数据。

关于具体的Filter的种类和解码方式等细节参考《PDF reference》3.3.1-3.3.7节中介绍。

2.5. 文档结构

档结构如图1:

PDF的文

Header只有一行，用来表示文件遵循的**PDF**协议版本号。

Body 中存放的是组成文件的所有的**Objects**。

Cross-reference table 中存放的是关于**Indirect**对象的信息。

Trailer 中指明了**Cross-reference table**的位置和在**Body**中特别指定的对象的位置。

Incremental Updates 是在**Trailer**后面添加的对象。如果一个文件被修改了，在文件的末尾还会包含增加的对象。

文件中一行的结尾符可以是‘\r’、‘\r\n’、‘\n’。除了在**Stream**对象中的字节，一般一行不超过**255**个字节。

3.5.1 Header

在**PDF**文档开头会包含此文件所遵循的**PDF**协议版本，格式如下：

```
%PDF-1.4
```

查看**PDF**文档的软件，如**Adobe Reader**、**Adobe Acrobat**等等，都能向后兼容。

比如软件是按**1.4**版本实现，对于遵循版本**1.4**之前的版本，比如**1.3**、**1.2**，都可以查看。

对于文档遵循版本比软件遵循版本高的情况，软件会忽略那些它不认识的新特性。

3.5.2 Body

PDF文档的**Body**是有好多代表文档内容的**Indirect**对象组成。这些对象代表文字、图像、表格等组成**PDF**文档的**Objects**。

3.5.3 Cross-reference table

一个**PDF**可以包含一个或者多个**Cross-reference sections**，一般**PDF**文档创建的时候，**Cross-reference table** 就由一个**Cross-reference section**组成。如果是**Linearized PDF**（线性

PDF用于网络浏览PDF格式，具体可以参见《PDF reference》Appendix F）会有两个。如果此PDF添加新的对象，就会添加新的Cross-reference section。

每个Cross-reference section，以关键字xref独占一行开头。后边跟着一个或者多个Cross-reference subsection。

每一个Cross-reference subsection 第一行是两个以空格分割的数字开头，第一个数字代表此Cross-reference subsection包含的起始Object的Object Identifier；第二个数字代表这个Cross-reference subsection包含的Object的个数。所以这个特性决定，每个Cross-reference subsection 中的Object编号都是连续的。比如，第一行是

28 5

标识此Cross-reference subsection 包含Object Number从28到32的5个Object。

在第一行之后包含一个或者多个Entries（条目），每个Entry独占一行，一共占20个字节，有两种格式，一种是代表Entry正在使用称作Used Entry，第三个字段使用关键词n表示，另一种代表Entry不在使用，代表Free Entry，第三个字段用f表示。Used Entry的格式为：

nnnnnnnnnn ggggg n eol

nnnnnnnnnn 10个字节代表对象在文档中的偏移量（offset）。如果偏移量不够10前面补零。

ggggg 5个字节代表对象Generation，作用后边做进一步解释。如果数字不够五位，也是前面补零凑齐。

n 一个字节代表这个对象还是使用。

eol 两个字节标识行结束（end of line）。可以是‘\r\n’、‘\r’、‘\n’，如果是‘\r’、‘\n’，在前面不一个空格组成两个字节。

每个字段用一个空格分割。总共加一起是20个字符。

Free Entry的格式为：

nnnnnnnnnn ggggg f eol

nnnnnnnnnn 是一个10字节的数字，标识的是下一个Free Entry的Object identifier。这样所有 Free Entry可以组成一个链表。

ggggg 是一个5字节的数字，标识Generation。作用后面解释。

f 是一个字节，代表这是一个 **Free Entry**。

eol 是两个字节行结束符。

Free Entry 的第一个字段表示下一个 **Free Entry** 的 **Object Number**，这样所有的 **Free Entry** 都可以组成一个链表结构。如果是链表就需要有个链表头，所以规定 **Cross-reference subsection** 第一个 **Entry** 是 **Free Entry**，它的第一个字段是 0，第二个字段 **Generation** 是 65536，作为 **Free Entry** 链表头。链表最后一个 **Free Entry** 的第一个字段设置为 0，指向第一个 **Object**。

除了第一个 **Entry**，PDF 文档中所有的 **Generation** 初始值都是 0。如果 **Indirect Object** 被删除了，相应的 **Used Entry** 修改为 **Free Entry**，这个 **Entry** 的 **Generation** 被加一。所以当 一个 **Entry** 被重新使用，这个 **Entry** 都会有一个新的 **Generation**。

Generation 的最大值为 65536，当达到最大值以后，这个 **Entry** 就不能在重新使用了。

Example 3.5

xref

0 6

0000000003 65535 f

0000000017 00000 n

0000000081 00000 n

0000000000 00007 f

0000000331 00000 n

0000000409 00000 n

Example 3.5 显示了一个 **Cross-reference section**，这个 **Cross-reference section** 由一个 **Cross-reference subsection** 组成。**Cross-reference subsection** 有六个 **Entries** 组成。其中有四个 **Used Entry**，**Object Number** 分别是 1，2，4，5。还有两个 **free Entry**，**Object Number** 分别是 0，3。**Object Number** 为 3 的对象已经被删除了，所以 **Object Number 0** 的 **Free Entry** 的第一个字段是 3。而 **Object Number** 为 3 的 **Free Entry** 的第一个字段为 0，指向第一个 **Entry**。这样 **Free Entry** 就是连起来了。

Example 3.6

xref

0 1

0000000000 65535 f

3 1

0000025325 00000 n

23 2

0000025518 00002 n

0000025635 00000 n

30 1

0000025777 00000 n

Example 3.5 显示了一个**Cross-reference section**，这个**Cross-reference section**由4个**Cross-reference subsection**组成。

3.5.4 File Trailer

一个**Trailer**允许PDF应用程序快速的查找**Cross-reference section**和一些特定的**Object**。**Trailer**的格式是：

trailer

<< key1 value1

key2 value2

...

keyn valuen

startxref**Byte_offset_of_last_cross-reference_section****%%EOF**

一个**Trailer**关键字后面跟着一个**Dictionary**对象，然后跟着**startxref**关键字占一行，然后跟着一个**offset**（指的从文件开头到最后一个**Cross-reference section**的偏移量），最后跟着**Trailer**的一行结束标志**%%EOF**。

表4显示**Trailer**中的**Dictionary**对象的成员。

Key	Type	Value
Size	Integer	(Required) The total number of entries in the file’ s cross-reference table, as defined by the combination of the original section and all update sections. Equivalently, this value is 1 greater than the highest object number used in the file.
Prev	Integer	(Present only if the file has more than one cross-reference section) The byte offset from the beginning of the file to the beginning of the previous cross-reference section.
Root	Dictionary	(Required; must be an indirect reference) The catalog dictionary for the PDF document contained in the file (see Section 3.6.1, “ Document Catalog ”).
Encrypt	Dictionary	(Required if document is encrypted; PDF 1.1) The document’ s encryption dictionary (see Section 3.5, “ Encryption ”).
Info	Dictionary	(Optional; must be an indirect reference) The document’ s information dictionary (see Section 9.2.1, “ Document Information Dictionary ”).
ID	Array	(Optional; PDF 1.1) An array of two strings constituting a file identifier (see Section 9.3, “ File Identifiers ”) for the file.

Example 3.7

trailer

<< /Size 22

/Root 2 0 R

/Info 1 0 R

/ID [<81b14aafa313db63dbd6f981e49f94f4>

<81b14aafa313db63dbd6f981e49f94f4>]

>>

startxref

18799

%%EOF

3.5.5 增量更新（Incremental Updates）

增量更新允许PDF文件的修改可以不用修改整个文件，只是把改变、增加的对象添加到文件末尾。优点如下：

- 1).提高对大文件的小的修改的速度。
- 2).减少数据丢失的风险。
- 3).提高HTTP connection修改文件的效率。
- 4).对于一些已经签名（**signed**）文件，是不可能修改原来的数据，因为只要修改原来的数据，之前的签名就无效了。

增量更新一个PDF文件的时候，增加、修改的对象会被添加到文件末尾，形成一个**Update Body**，然后包含增加、修改和删除队形的**Cross-reference section**会被添加到**Update Body**的后面。最后会包含的一个新的**Trail**，它的**offset**会指导新添加的**Cross-**

reference section，他的**Dictionary**对象中的**Prev**属性值之处之前**Cross-reference section**的位置。这样就和之前文档结构关联起来了。

经过多次增量更新之后，一个PDF文件格式就可能如图。

2.6. 加密 (encryption)

PDF可以通过加密保护文档内容，阻止未授权的访问。**PDF**至加密**String**对象和**Stream**对象，不会加密**Number**对象和**Boolean**对象等标识文档结构，而不是文档内容的对象。

PDF的机密信息，用**Trailer**中的**Dictionary**对象中的**Encrypt**属性标识，**Encrypt**属性也是个**Dictionary**对象，称这个**Dictionary**为**Encryption Dictionary**。下表显示**Encryption Dictionary**的属性。

Key	Type	Value
Filter	Name	(Required) The name of the security handler for this document; see below. Default value: Standard , for the built-in security handler. (Names for other security handlers can be registered using the procedure described in Appendix E.)
V	Number	(Optional but strongly recommended) A code specifying the algorithm to be used in encrypting and decrypting the document: 0 An algorithm that is undocumented and no longer supported, and whose use is strongly discouraged. 1 Algorithm 3.1 on page 73, with an encryption key length of 40 bits; see below. 2 (PDF 1.4) Algorithm 3.1 on page 73, but allowing encryption key lengths greater than 40 bits. 3 (PDF 1.4) An unpublished algorithm allowing encryption key lengths ranging

		<p>from 40 to 128 bits. (This algorithm is unpublished as an export requirement of the U.S. Department of Commerce.)</p> <p>The default value if this entry is omitted is 0, but a value of 1 or greater is strongly recommended.</p> <p>(See implementation note 15 in Appendix H.)</p>
Length	Number	<p>(Optional; PDF 1.4; only if V is 2 or 3) The length of the encryption key, in bits. The value must be a multiple of 8, in the range 40 to 128. Default value: 40.</p>
R	Number	<p>(Required) A number specifying which revision of the standard security handler should be used to interpret this dictionary. The revision number should be 2 if the document is encrypted with a V value less than 2 (see Table 3.13) and does not have any of the access permissions set (via the P entry, below) that are designated “Revision 3” in Table 3.15; otherwise (that is, if the document is encrypted with a V value greater than 2 or has any “Revision 3” access permissions set), this value should be 3.</p>
O	String	<p>(Required) A 32-byte string, based on both the owner and user passwords, that is used in computing the encryption key and in determining whether a valid owner password was entered. For more information, see “Encryption Key Algorithm” on page 78 and “Password Algorithms” on page 79.</p>
U	String	<p>(Required) A 32-byte string, based on the user password, that is used in determining whether to prompt the user for a password and, if so, whether a valid user or owner password was entered. For more information, see “Password Algorithms” on page 79.</p>
P	Integer	<p>(Required) A set of flags specifying which operations are permitted when the document is opened with user access (see Table 3.15).</p>

不像其他对象中的String对象，Encryption Dictionary中的String对象不能用常规的加密方法，进行加密。Encryption Dictionary中的Filter属性指定Security Handler负责对其加密。默认的情况Filter的值为Standard。

3.6.1 标准加密算法（General Encryption Algorithm）

PDF通用加密使用MD5作为hash函数，选用RC4作为加密算法。RC4是对称加密算法，他的密钥可以通过各种方式计算出来，下节讨论生成密钥的方面。先看一下加密的过程。

- 1).取得要加密对象的Object Identifier（包括Object Number和Object Generation组成）。对于不是Indirect String对象，取包涵它的Indirect对象的Object Identifier。
- 2).用n字节的key，加上由Object Number的后三位加上Object Generation后两位组成的5位，组成n+5个字节。然后用这（n+5）个字节作为hash函数的输入参数。
- 3).使用hash输出结果的（n+5）个字节，最多不超过16个字节，作为key，调用RC4，对要加密数据进行加密。

构建PDF文档时候，对于Stream对象是先加密，然后在用编码Filter进行编码。查看PDF文档时候，是用Filter进行解码，然后在解密。

对于String对象先进行escape-sequence processing and hexadecimal decoding，然后在进行解密。

3.6.2 密钥的生成

PDF的加解密过程是通过两个密码进行的：Owner Password和用户Password。

PDF生成是的时候是更具是否用户提供密码和一些权限限制，如果是就是对文档进行加密。

PDF读取软件，读取PDF的时候，如果是加密文档，会要求用户输入两个密码中的任意一个。如果输入的是Owner Password，那么用户就有文档权限，除了修改文档和密码的权限。

权限通过Encryption Dictionary中的P属性值进行指定，P值是一个32为的Number对象，其中bit代表不同的权限。在不同的版本（Reversion）中，P每个bit代表的权限值，是不一样的。具体值可以参考《PDF Reference》中第77页。

版本是通过Encryption Dictionary 中的R属性值指定，可以是2或者3。

Encryption Dictionary中的O属性值是Owner Password通过一定加密过程生成值，而U属性是User Password通过一定加密过程生成的。对于R=2和R=3的情况，O属性值和U属性值的生成过程是不一样的。具体可以参考《PDF Reference》79页。

上文提到的RC4加密之后用到的密钥，也是通过Owner Password和User Password 按照一定方法生成的具体过程可以参考《PDF Reference》78页。

2.7. 文档结构

PDF文档可以被看成一个有PDF Body部分的对象组成的一个Tree。Tree的根就是Catalog Dictionary（目录结构），而Catalog Dictionary结构中有包含很多属性。比如Page Tree包含了所有组成文档Page的对象。

下图可以标识PDF文档的层级结构。

3.7.1 目录结构（document catalog）

Key	Type	Value
Type	Name	(Required) The type of PDF object that this dictionary describes; must be Catalog for the catalog dictionary.
Version	Name	(Optional; PDF 1.4) The version of the PDF specification to which the document conforms (for example, 1.4), if later than the version specified in the file’s header (see Section 3.4.1, “File Header”). If the header specifies a later version, or if this entry is absent, the document conforms to the version specified in the header. This entry enables a PDF producer application to update the version using an incremental update; see Section 3.4.5, “Incremental Updates.” (See implementation note 18 in Appendix H.)

		Note: The value of this entry is a name object, not a number, and so must be preceded by a slash character (/) when written in the PDF file (for example, /1.4).
Pages	Dictionary	(Required; must be an indirect reference) The page tree node that is the root of the document' s page tree (see Section 3.6.2, “ Page Tree ”).
Pagelabels	NumberTree	(Optional; PDF 1.3) A number tree (see Section 3.8.5, “ Number Trees ”) defining the page labeling for the document. The keys in this tree are page indices; the corresponding values are page label dictionaries (see Section 8.3.1, “ Page Labels ”). Each page index denotes the first page in a labeling range to which the specified page label dictionary applies. The tree must include a value for page index 0.
Names	Dictionary	(Optional; PDF 1.2) The document' s name dictionary (see Section 3.6.3, “ Name Dictionary ”).
Dests	Dictionary	(Optional; PDF 1.1; must be an indirect reference) A dictionary of names and corresponding destinations (see “ Named Destinations ” on page 476).
ViewerPreferences	Dictionary	(Optional; PDF 1.2) A viewer preferences dictionary (see Section 8.1, “ Viewer Preferences ”) specifying the way the document is to be displayed on the screen. If this entry is absent, viewer applications should use their own current user preference settings.

PageLayout	Name	<p>(Optional) A name object specifying the page layout to be used when the document is opened:</p> <p>SinglePage Display one page at a time.</p> <p>OneColumn Display the pages in one column.</p> <p>TwoColumnLeft Display the pages in two columns, with oddnumbered pages on the left.</p> <p>TwoColumnRight Display the pages in two columns, with oddnumbered pages on the right.</p> <p>(See implementation note 19 in Appendix H.)</p> <p>Default value: SinglePage.</p>
PageMode	Name	<p>(Optional) A name object specifying how the document should be displayed when opened:</p> <p>UseNone Neither document outline nor thumbnail images visible</p> <p>UseOutlines Document outline visible</p> <p>UseThumbs Thumbnail images visible</p> <p>FullScreen Full-screen mode, with no menu bar, window controls, or any other window visible</p> <p>Default value: UseNone.</p>
Outlines	Dictionary	<p>(Optional) A name object specifying how the document should be displayed when opened:</p> <p>UseNone Neither document outline nor thumbnail images visible</p> <p>UseOutlines Document outline visible</p> <p>UseThumbs Thumbnail images visible</p> <p>FullScreen Full-screen mode, with no menu bar, window controls, or any other window visible</p> <p>Default value: UseNone.</p>
Threads	Array	<p>(Optional; must be an indirect reference) The</p>

		outline dictionary that is the root of the document’ s outline hierarchy (see Section 8.2.2, “ Document Outline”).
OpenAction	Array or dictionary	(Optional; PDF 1.1) A value specifying a destination to be displayed or an action to be performed when the document is opened. The value is either an array defining a destination (see Section 8.2.1, “ Destinations”) or an action dictionary representing an action (Section 8.5, “ Actions”). If this entry is absent, the document should be opened to the top of the first page at the default magnification factor.
AA	Dictionary	(Optional; PDF 1.4) An additional-actions dictionary defining the actions to be taken in response to various trigger events affecting the document as a whole (see “ Trigger Events” on page 514). (See also implementation note 20 in Appendix H.)
URI	Dictionary	(Optional) A URI dictionary containing document-level information for URI (uniform resource identifier) actions (see “ URI Actions” on page .
AcroForm	Dictionary	(Optional; PDF 1.2) The document’ s interactive form (AcroForm) dictionary (see Section 8.6.1, “ Interactive Form Dictionary”). 523)
Metadata	Stream	(Optional; PDF 1.4; must be an indirect reference) A metadata stream

		containing metadata for the document (see Section 9.2.2, “ Metadata Streams ”).
StructTreeBoot	Dictionary	(Optional; PDF 1.3) The document’ s structure tree root dictionary (see Section 9.6.1, “ Structure Hierarchy ”).
MarkInfo	Dictionary	(Optional; PDF 1.4) A mark information dictionary containing information about the document’ s usage of Tagged PDF conventions (see Section 9.7.1, “ Mark Information Dictionary ”).
Lang	Text String	(Optional; PDF 1.4) A language identifier specifying the natural language for all text in the document except where overridden by language specifications for structure elements or marked content (see Section 9.8.1, “ Natural Language Specification ”). If this entry is absent, the language is considered unknown.
SpiderInfo	Dictionary	(Optional; PDF 1.3) A Web Capture information dictionary containing state information used by the Acrobat Web Capture (AcroSpider) plugin extension (see Section 9.9.1, “ Web Capture Information Dictionary ”).
OutputIntents	Array	(Optional; PDF 1.4) An array of output intent dictionaries describing the color characteristics of output devices on which the document might be

3.7.2 页面树（Page Tree）

组成PDF文档的所有页都通过Tree结构进行组织，称作Page Tree，Tree结构可以允许读取PDF内容的应用程序，在有限的内存上，读取很大文档。

组成Page Tree有两种类型的节点：一种是节点他自己也是一个page tree，称作page tree node。另一种节点叶子节点，称作page objects。

为了进一步优化性能，好多创建PDF文件的软件，把page tree组成一个平衡数（balanced tree）。

3.7.2.1 page tree nodes

KEY	TYPE	VALUE
Type	Name	(Required) The type of PDF object that this dictionary describes; must be Pages for a page tree node.
Parent	Dictionary	(Required except in root node; must be an indirect reference) The page tree node that is the immediate parent of this one.
Kids	Array	(Required) An array of indirect references to the immediate children of this node. The children may be page objects or other page tree nodes.
Count	Integer	(Required) The number of leaf nodes (page objects) that are descendants of this node within the page tree.

Example 3.9

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

Example 3.9 展示了一个文档有三个页，对于每个页的内容，具体查看下一节的介绍。

3.7.2.2 Page Objects

Page Objects 是Page Tree的叶子节点，包含了文档中一个页的所有属性。下表列出Page Object的所有属性，在表中属性，标有（**inheritable**）的条目，是说明其值可以从他的父对象中继承（具体可以参考下一节）。关于表中**rectangle**和**date**对象3.9小节进行介绍。

KEY	TYPE	VALUE
Type	Name	(Required) The type of PDF object that this dictionary describes; must be Page for a page object.
Parent	Dictionary	(Required; must be an indirect reference) The page tree node that is the immediate parent of this page object.
LastModified	Date	(Required if PieceInfo is present; optional otherwise; PDF 1.3) The date and time (see Section 3.8.2, “ Dates ”) when the page’ s contents were most recently modified. If a page-piece dictionary (PieceInfo) is present, the modification date is used to ascertain which of the application data dictionaries that it contains correspond to the current content of the page (see Section 9.4, “ Page-Piece Dictionaries ”).
Resources	Dictionary	(Required; inheritable) A dictionary containing any resources required by the page (see Section 3.7.2, “ Resource Dictionaries ”). If the page requires no resources, the value of this entry should be an empty dictionary; omitting

		the entry entirely indicates that the resources are to be inherited from an ancestor node in the page tree.
MediaBox	Rectangle	(Required; inheritable) A rectangle (see Section 3.8.3, “ Rectangles ”), expressed in default user space units, defining the boundaries of the physical medium on which the page is intended to be displayed or printed (see Section 9.10.1, “ Page Boundaries ”).
CropBox	Rectangle	(Optional; inheritable) A rectangle, expressed in default user space units, defining the visible region of default user space. When the page is displayed or printed, its contents are to be clipped (cropped) to this rectangle and then imposed on the output medium in some implementation-defined manner (see Section 9.10.1, “ Page Boundaries ”). Default value: the value of MediaBox.
BleedBox	Rectangle	(Optional; PDF 1.3) A rectangle, expressed in default user space units, defining the region to which the contents of the page should be clipped when output in a production environment (see Section 9.10.1, “ Page Boundaries ”). Default value: the value of CropBox.
TrimBox	Rectangle	(Optional; PDF 1.3) A rectangle, expressed in default user space units, defining the intended dimensions of the finished page after trimming (see Section 9.10.1, “ Page Boundaries ”). Default value: the value of CropBox.
ArtBox	Rectangle	(Optional; PDF 1.3) A rectangle, expressed in default user space units, defining

		<p>the extent of the page’ s meaningful content (including potential</p> <p>white space) as intended by the page’ s creator (see Section 9.10.1, “ Page</p> <p>Boundaries”). Default value: the value of CropBox.</p>
BoxColorInfo	Dictionary	<p>(Optional) A box color information dictionary specifying the colors and</p> <p>other visual characteristics to be used in displaying guidelines on the</p> <p>screen for the various page boundaries (see “ Display of Page Boundaries”</p> <p>on page 679). If this entry is absent, the viewer application should use its</p> <p>own current default settings.</p>
Contents	Stream or array	<p>(Optional) A content stream (see Section 3.7.1, “ Content Streams”) describing</p> <p>the contents of this page. If this entry is absent, the page is empty.</p> <p>The value may be either a single stream or an array of streams. If it is an</p> <p>array, the effect is as if all of the streams in the array were concatenated, in</p> <p>order, to form a single stream. This allows a program generating a PDF</p> <p>file to create image objects and other resources as they occur, even though</p> <p>they interrupt the content stream. The division between streams may</p> <p>occur only at the boundaries between lexical tokens (see Section 3.1, “ Lexical</p> <p>Conventions”), but is unrelated to the page’ s logical content or organization.</p> <p>Applications that consume or produce PDF files are not required</p> <p>to preserve the existing structure of the Contents array. (See implementation</p> <p>note 22 in Appendix H.)</p>

Rotate	Integer	(Optional; inheritable) The number of degrees by which the page should be rotated clockwise when displayed or printed. The value must be a multiple of 90. Default value: 0.
Group	dictionary	(Optional; PDF 1.4) A group attributes dictionary specifying the attributes of the page’ s page group for use in the transparent imaging model (see Sections 7.3.6, “ Page Group, ” and 7.5.5, “ Transparency Group XObjects ”).
Thumb	Stream	(Optional) A stream object defining the page’ s thumbnail image (see Section 8.2.3, “ Thumbnail Images ”).
B	array	(Optional; PDF 1.1; recommended if the page contains article beads) An array of indirect references to article beads appearing on the page (see Section 8.3.2, “ Articles ” ; see also implementation note 23 in Appendix H). The beads are listed in the array in natural reading order.
Dur	Number	(Optional; PDF 1.1) The page’ s display duration (also called its advance timing) : the maximum length of time, in seconds, that the page will be displayed during presentations before the viewer application automatically advances to the next page (see Section 8.3.3, “ Presentations ”). By default, the viewer does not advance automatically.
Trans	Dictionary	(Optional; PDF 1.1) A transition dictionary describing the transition effect to be used when displaying the page during presentations (see Section 8.3.3, “ Presentations ”).

Annots	Array	(Optional) An array of annotation dictionaries representing annotations associated with the page (see Section 8.4, “ Annotations ”).
AA	Dictionary	(Optional; PDF 1.2) An additional-actions dictionary defining actions to be performed when the page is opened or closed (see Section 8.5.2, “ Trigger Events ” ; see also implementation note 24 in Appendix H).
Metadata	Stream	(Optional; PDF 1.4) A metadata stream containing metadata for the page (see Section 9.2.2, “ Metadata Streams ”).
PieceInfo	Dictionary	(Optional; PDF 1.3) A page-piece dictionary associated with the page (see Section 9.4, “ Page-Piece Dictionaries ”).
StructParents	Integer	(Required if the page contains structural content items; PDF 1.3) The integer key of the page’ s entry in the structural parent tree (see “ Finding Structure Elements from Content Items ” on page 600).
ID	String	(Optional; PDF 1.3; indirect reference preferred) The digital identifier of the page’ s parent Web Capture content set (see Section 9.9.5, “ Object Attributes Related to Web Capture ”).
PZ	Number	(Optional; PDF 1.3) The page’ s preferred zoom (magnification) factor: the factor by which it should be scaled to achieve the “ natural ” display magnification (see Section 9.9.5, “ Object Attributes Related to Web Capture ”).
SeperationInfo	Dictionary	(Optional; PDF 1.3) A separation dictionary containing information needed

		to generate color separations for the page (see Section 9.10.3, “ Separation Dictionaries ”).
--	--	---

Example 3.10 显示文档中一个**Page**，这个**Page**由一个**thumb**，两个**Annotation**。

MediaBox指定这个**Page**将要一个信纸大小的纸上。**Resources**指定这个**Page**使用了三种字体，分别是**F3**、**F5**、**F7**。

Example 3.10

```
3 0 obj
```

```
<< /Type /Page
```

```
/Parent 4 0 R
```

```
/MediaBox [0 0 612 792]
```

```
/Resources << /Font << /F3 7 0 R
```

```
/F5 9 0 R
```

```
/F7 11 0 R
```

```
>>
```

```
/ProcSet [/PDF]
```

```
>>
```

```
/Contents 12 0 R
```

```
/Thumb 14 0 R
```

```
/Annots [ 23 0 R
```

```
24 0 R
```

```
]
```

```
>>
```

```
endobj
```


3.7.2.3 页属性的继承（Inheritance of Page Attributes）

上表可以看到有些页的属性值被指定为**inheritable**，**inheritable**说明**page object**的属性值如果被省略了，它的值可以从他的父对象**page tree object**中继承。如果这个属性在**page object**中是**required**，那么父对象中，一定要提供这个值。如果这个属性是**optional**的，那么使用父对象的中的属性值或者取默认值。

3.7.3.4 名字对照表（Name Dictionary）

在PDF文档，有些对象是通过**Name**直接引用的，而不是通过**Indirect Object**。**Name**和对应的对象之间的关系是通过**Name Dictionary**表建立，**Name Dictionary**可以通过**Trailer**中文件目录中的**Names**属性值找到。

3.8 Content Streams and Resources

3.8.1 Content Stream

Content Stream 是一个**Stream**对象，其数据部分是一个指令（**instruction**）队列，用来描述一个页是怎么被显示的。每一个页可能由多个**Content Stream**组成，这种情况所有的**Content Stream**组成一个指令队列，相当于一个大的**Content Stream**。

Content Stream 中的之类，在被解释执行之前，先用指定的**Filter**对**content Stream**的内容进行解码。

每一个指令由**operand**（操作数）和**operator**（操作码）组成。**Operand**位于**Operator**前面

Operands可以由任意的PDF基本对象组成，除了**Stream**对象。有一些特殊的**Operators**需要**Dictionary**作为他的**Operands**。**Indirect**对象不允许做**Operands**。

Operators 是PDF文档中的关键字，它和**Name**对象的区别是，不以‘/’开头。

4、5、6、9章进步说明PDF支持的**Operators**。

通常情况下，如果一个PDF读取软件的时候遇到不认识的**Operators**，会报错。如果不想让其报错，可以用**BX**和**EX**关键字包含那些指令。这样PDF读取软件会忽略不认识的**Operators**。

3.8.2 Resources Dictionary

有的时候，**Operator**需要引用自身**Content Stream**之外的**Indirect**对象，例如字体、一个包含图片的**Stream**对象。这种情况就需要需要把引用到的**Indirect**对象声明为**Named Resource**。**Resources Dictionary**中定义各种**Named Resource**。

Resource Dictionary 通过三种方式和Content Stream关联在一起：

- 1).Content Stream 是页内容的Contents属性的值，这种情况，对应的Resource Dictionary是页内容的Resources的属性的值。
- 2).对于除了页内容外的其他Content Stream，对应的Resource Dictionary是通过Stream中的Dictionary对象的Resource属性的值指定的。
- 3).对于XObject和Type 3 font会忽略Content Stream中的Resource属性，这种情况，Resource是使用包含XObject或Type 3 font的页的Resource的值。

每一个Resource Dictionary中Key值是资源类型的名字，Key对应Value也是一个Dictionary对象，我们称作SubDictionary。SubDictionary对象的Key值指定类型的一个资源名称，SubDictionary对象Value值是引用的资源对象。

下表列出了所有的资源类型名称。

ExtGState	Dictionary	(Optional) A dictionary mapping resource names to graphics state parameter dictionaries (see Section 4.3.4, “Graphics State Parameter Dictionaries”).
ColorSpace	Dictionary	(Optional) A dictionary mapping each resource name to either the name of a device-dependent color space or an array describing a color space (see Section 4.5, “Color Spaces”).
Pattern	Dictionary	(Optional) A dictionary mapping resource names to pattern objects (see Section 4.6, “Patterns”).
Shading	Dictionary	(Optional; PDF 1.3) A dictionary mapping resource names to shading dictionaries (see “Shading Dictionaries” on page 233).
XObject	Stream	(Optional) A dictionary mapping resource names to external objects (see Section 4.7, “External Objects”).

Font	Dictionary	(Optional) A dictionary mapping resource names to font dictionaries (see Chapter 5).
ProSet	Array	(Optional) An array of predefined procedure set names (see Section 9.1, “ Procedure Sets ”).
Properties	Dictionary	(Optional; PDF 1.2) A dictionary mapping resource names to property list dictionaries for marked content (see Section 9.5.1, “ Property Lists ”).

Example 3.11

```
<< /ProcSet [/PDF /ImageB]

/Font << /F5 6 0 R

/F6 8 0 R

/F7 10 0 R

/F8 12 0 R

>>

/XObject << /Im1 13 0 R

/Im2 15 0 R

>>

>>
```

Example 定一个包含 **Font**、**ProcSet**、**XObject**类型资源的**Resource Dictionary**。**Font**资源包含四个对象分别是**F5**、**F6**、**F7**、**F8**。

3.9 Common Data Structures

这节介绍四种数据对象**Text String**、**Date**、**Rectangles**、**Name Trees**、**Number Trees**。这五种数据类型，不能用在**Content Stream**对象中。

3.9.1 Text Strings

Text Strings 对象是**PDFDocEncoding**或者**Unicode**编码。**PDFDocEncoding**是**ISO Latin 1**编码的一个超集。

如果用**Unicode**编码，**Text Strings**前两个字节必须是**254**和**256**，代表**Text String**是**Unicode**编码。

任何使用**Unicode**编码的地方，都会有一个转义字符串来表名字符串使用哪国语言写的。转移字符串有下列部分顺序组成。

1. The Unicode value U+001B (that is, the byte sequence 0 followed by 27)
2. A 2-character ISO 639 language code—for example, **en** for English or **ja** for Japanese
3. (Optional) A 2-character ISO 3166 country code—for example, **US** for the United States or **JP** for Japan
4. The Unicode value U+001B

3.9.2 Dates

PDF文档的日期格式是：

(D:YYYYMMDDHHmmSSOHH'mm')

I YYYY代表年

I MM代表月

I DD代表日

I HH代表小时

I mm代表分

I SS代表秒

I O表示是本地时间和标准国际标准时间（**UT time**）的关系，O如果是‘+’标识本地时间比国际标准时间慢，O如果是‘-’标识本地时间比国际标准时间快，具体的差值由O后面的**HH**和**mm**标识。

l HH' 表示本地时间和国际标准时间查多少个小时。

l mm' 表示本地时间和国际标准时间查多少个分钟。mm后面的单引号是语法的一部分，不能省略。

YYYY后面所有的项都是可以省略的，MM和DD的默认值是01，其他项的默认值是00。

For example, December 23, 1998, at 7:52 PM, U.S. Pacific Standard Time, is represented by the string

D:199812231952-08'00'

3.9.3 Rectangles

Rectangles 是一个Array对象，通过两个对角坐标来描述矩形位置，通常格式是：

[llx lly urx ury]

llx lly 表示左下角（lower-left）的坐标x，y。

urx ury 标识右上角（up-right）的坐标x，y。

也可以通过左上交和右下角坐标确定一个矩形。

3.9.4 Name Trees

Name Trees 由几种节点组成，每个节点都是一个Dictionary对象。Name Trees有三种类型的节点，Root节点、Kids节点、Name节点（相当于叶子节点）。

Root节点中包含对Kids节点或者Name节点的引用，但是不能同时两种都存在。

Kids节点中包含对Name节点应用。

这样由这三种节点就可以组成一颗逻辑上的数。

Name Trees 各种节点Dictionary中有三种属性。

Key	Type	Value
Kids	Array	(Root and intermediate nodes only; required in intermediate nodes; present in the root node if and only if Names is not present) An array of indirect references to the immediate children of this node. The children may be intermediate or leaf nodes.

Names	array	(Root and leaf nodes only; required in leaf nodes; present in the root node if and only if Kids is not present) An array of the form [key1 value1 key2 value2 ... keyn valuen] where each keyi is a string and the corresponding valuei is an indirect reference to the object associated with that key. The keys are sorted in lexical order, as described below.
Limits	Array	(Intermediate and leaf nodes only; required) An array of two strings, specifying the (lexically) least and greatest keys included in the Names array of a leaf node or in the Names arrays of any leaf nodes that are descendants of an intermediate node. The

Name属性的的Key是String对象，而且Key是按字母顺序排序的。

Limit对象是一个由两个String对象组成，标识Limit属性所在节点（包括它的子节点）中，最小和最大的Name属性的Key值。

Example 3.24用Name Trees结构描述如下图分类的元素周期表。

```
1 0 obj
/Kids [ 2 0 R % Root node
3 0 R
4 0 R
]
>>
endobj
```

2 0 obj

<< /Limits [(Actinium) (Gold)] % Intermediate node

/Kids [5 0 R

6 0 R

7 0 R

8 0 R

9 0 R

10 0 R

11 0 R

]

>>

endobj

3 0 obj

<< /Limits [(Hafnium) (Protactinium)] % Intermediate node

/Kids [12 0 R

13 0 R

14 0 R

15 0 R

16 0 R

17 0 R

18 0 R

19 0 R

]

>>

endobj

40 obj

<< /Limits [(Radium) (Zirconium)] % Intermediate node

/Kids [20 0 R

21 0 R

22 0 R

23 0 R

24 0 R

]

>>

endobj

5 0 obj

<< /Limits [(Actinium) (Astatine)] % Leaf node

/Names [(Actinium) 25 0 R

(Aluminum) 26 0 R

(Americium) 27 0 R

(Antimony) 28 0 R

(Argon) 29 0 R

(Arsenic) 30 0 R

(Astatine) 31 0 R

]

>>

endobj

...

24 0 obj

<< /Limits [(Xenon) (Zirconium)] % Leaf node

/Names [(Xenon) 129 0 R

(Ytterbium) 130 0 R

(Yttrium) 131 0 R

(Zinc) 132 0 R

(Zirconium) 133 0 R

]

>>

endobj

25 0 obj

89 % Atomic number (Actinium)

endobj

...

133 0 obj

40 % Atomic number (Zirconium)

Endobj

|

3.9.5 Number Trees

Number Trees 和Name Trees结构相似，区别在于Number Trees节点Name属性中的Key值是Integer对象，而不是String对象。

3.9 Functions（函数）对象

虽然PDF不是一个编程语言，但是提供了一些函数对象，用于数学、采样还原等等计算。所有的函数对象都会有一些通用的属性如下表：

Key	Type	Value
FunctionType	Integer	(Required) The function type: 0 Sampled function

		2 Exponential interpolation function 3 Stitching function 4 PostScript calculator function
Domain	Array	(Required) An array of $2 \times m$ numbers, where m is the number of input values. For each i from 0 to $m - 1$, Domain2i must be less than or equal to Domain2i+1 , and the i th input value, x_i , must lie in the interval $\text{Domain2i} \leq x_i \leq \text{Domain2i+1}$. Input values outside the declared domain are clipped to the nearest boundary value.
Range	array	(Required for type 0 and type 4 functions, optional otherwise; see below) An array of $2 \times n$ numbers, where n is the number of output values. For each j from 0 to $n - 1$, Range2j must be less than or equal to Range2j+1 , and the j th output value, y_j , must lie in the interval $\text{Range2j} \leq y_j \leq \text{Range2j+1}$. Output values outside the declared range are clipped to the nearest boundary value. If this entry is absent, no clipping is done.

每种类型的函数，还会有特定的函数属性和计算过程，具体可以参考《PDF reference》3.9.1到3.9.4节的介绍。

3.10 文件引用（File Specifications）

我们可以通过文件引用（File Specifications），在PDF文档中已用别的文件中的内容。文件引用分文简单文件引用（simple file specification）和全文件引用(full file specification)。简单文件引用可以是String对象也可以是Dictionary对象。全文件引用只能是Dictionary对象。

文件引用是平台无关的，当在具体平台上读取PDF文档的时候，会转换为平台相关的文件引用。

3.10.1 文件引用字符串（File Specification Strings）

文件引用字符串通过多个子字符串组成，每个部分用‘/’进行分割。如果‘/’是子字符串的一部分，就需要转义，在前面加两个反斜杠“\\”。

(in\Vout) 代表的是（in/out）

文件引用还可以分文相对文件引用和绝对文件引用，其含义和相对路劲和绝对路径差不多。以‘/’开头的是绝对文件引用，否则为相对文件引用。

在PDF1.2以后的协议中，允许组成文件引用的子字符串可以是多字节字符，比如中文。这种必须要有String对象的十六进制的表示方法。由于‘/’（ASCII码为0x2f）和‘\’（ASCII码为0x5c），但多自己字符中包含2f和5c的时候，要在前面添加一个5c，比如有个两个字节的编码为<89 5C>,要写成<89 5C 5C>。

3.10.2 文件引用字典（File Specification Strings）

文件引用字典是个Dictionary对象。它允许多个文件对应多个不同的文件系统。

下边是Dictionary对象包含的属性：

Key	Type	Value
Type	Name	（required）值为FileSpec指明这是个文件引用对象。
FS	Name	（optional）文件系统的文成，PDF只有一个标准文件系统URL。
F	String	（Required if Dos, Mac, Unix entries are all absent）文件描述字符串。当Dos, Mac, Unix属性都没有的时候使用。
Dos	String	（optional）指定Dos系统中使用的文件引用。
Mac	String	（optional）指定Mac系统中使用的文件引用。
Unix	String	（optional）指定Unix系统中使用的文件引用。
ID	Array	（optional）文件标识（具体作用参考《PDF reference》9.3节）。
V	Boolean	（optional PDF1.2）指定引用文件是是否是易变（volatile）。如果是True，PDF读取软件不要缓冲引用文件的内容。
EF	Dictionary	嵌入（embedded）引用文件Dictionary对象。

RF

Dictionary

引用文件Dictionary对象。