

目录

第 4 章 图表	1
4.1、图形对象	1
4.2、坐标系统	5
4.2.1、坐标空间	5
4.2.2、常见转化	9
4.2.3、转化矩阵	12
4.3、图形状态	13
4.3.1、图形状态栈	17
4.3.2、图形状态参数的细节	17
4.4、路径构建和绘图	27
4.4.1、路径构造算子	28
4.4.2、路径绘制算子	31
4.4.3、剪辑路径操作	35
4.5、色彩空间	36
4.5.1、色彩值	37
4.5.2、色彩空间族	38
4.5.3、设备色彩空间	41
4.5.4、CIE-Based 色彩空间	45
4.5.5、特殊颜色空间	65
4.5.6、套印控制	90
4.5.7、颜色算子	92
4.6、模式	95
4.6.1、一般模式的性质	95
4.6.2、平铺模式	96
4.6.3、底纹图案	107
4.7、外部对象	137
4.7.1 PostScript 下的 Xobjects	138
4.8、图像	139
4.8.1 图像参数	141
4.8.2 样品表征	141
4.8.3 图像坐标系	142
4.8.4 图像词典	145
4.8.5 图像掩码	156
4.8.6 内敛图像	158
4.9、XOBJECTS 格式	162
4.9.1 字典格式	165
4.9.2 Xobjects 组	169
4.9.3 Xobjects 关联	169
4.10、可选内容	172
4.10.1、可选内容组	172
4.10.2、制作可选图像内容	177

4.10.3、可选内容的配置	182
第5章 文本.....	195
5.1、组织和字体的使用.....	195
5.1.1、基本的显示文本	196
5.1.2、实现特殊图形特效	198
5.1.3、字形定位和度量	201
5.2、文本状态参数和操作.....	203
5.2.1、字符间距.....	205
5.2.2、字间距	206
5.2.3、横向缩放.....	206
5.2.4、行距	207
5.2.5、文本渲染模式.....	207
5.2.6、文本上升.....	209
5.2.7、文本脱模.....	210
5.3、文本对象.....	211
5.3.1、文本位置操作	212
5.3.2、文本显示操作者	214
5.3.3、文本空间详解	216
5.4、字体数据结构导论.....	217
5.5、简单的字体.....	219
5.5.1、关于 1 型字体	219
5.5.2、关于 TrueType 型字体	224
5.5.3、关于字体子集	226
5.5.4、关于 3 型字体	226
5.5.5、字符编码.....	234
5.6、复合字体.....	241
5.6.1、CID 键字体的概要	242
5.6.2、CIDSystemInfo 字典	243
5.6.3、CID 字体	244
5.6.4、CMaps	250
5.6.5、字体类型 0 字典	261
5.7、字体描述符号.....	265
5.7.1、字体描述标记	267
5.7.2、CIDFonts 字体描述	270
5.8、字体嵌入程序.....	275
5.9、文本对象提取.....	279
5.9.1、映射文字符号代码得到编码值	280
5.9.2、ToUnicode CMaps	281
第6章 渲染.....	285
6.1、基于 CIE 的色彩到设备色彩的转换.....	286
6.2、设备色彩空间中的转换.....	287
6.2.1、DeviceGray 和 DeviceRGB 间的转换.....	288
6.2.2、DeviceGray 和 DeviceCMYK 间的转换.....	288
6.2.3、从 DeviceRGB 到 DeviceCMYK 的转换.....	289

6.2.4、从 DeviceCMYK 到 DeviceRGB 的转换.....	290
6.3、传递函数	291
6.4、半调	292
6.4.1、半调屏幕.....	293
6.4.2、专函数	294
6.4.3、阈值阵列.....	298
6.4.4、半调字典.....	299
6.5、扫描转换的细节	312
6.5.1、平面公差.....	312
6.5.2、平滑度公差.....	313
6.5.3、扫描转换规则	314
6.5.4、自动加横调整.....	315
第7章 透明度.....	317
7.1、透明度的概述.....	318
7.2、基本的复合计算.....	320
7.2.1 复合计算的基本符号	320
7.2.2 基本的复合公式	320
7.2.3 混合颜色空间	321
7.2.4 混合模型	323
7.2.5 Alpha 的阐述	326
7.2.6 形状和不透明性计算	327
7.2.7 基本复合计算的总结	330
7.3、透明组.....	331
7.3.1 组复合计算的符号	332
7.3.2 组结构和术语	333
7.3.3 组复合计算	334
7.3.4 孤立组	338
7.3.5 推倒组	338
7.3.6 页组	340
7.3.7、组复合计算的总结	342
7.4、软掩码.....	343
7.4.1 从组 alpha 派生一个软掩码	344
7.4.2 从组亮度得到软掩码	344
7.5 在 PDF 内指定透明度	345
7.5.1 指定源和背景颜色	345
7.5.2 指定混合颜色空间和混合模型	346
7.5.3 指定形状和不透明性	346
7.5.4 指定软掩码	349
7.5.5 透明组 XOBJECT	353
7.5.6 模型和透明度	356
7.6 颜色空间和渲染的问题	357
7.6.1 透明组的颜色空间	358
7.6.2 点颜色和透明度	359
7.6.3 重印和透明度	361

7.6.4 渲染参数和透明度	367
7.6.5 PostScript 兼容性	370
第8章 交互功能	372
8.1、用户参数	372
8.2、文档级导航	374
8.2.1、目标	374
8.2.2、文件大纲	377
8.2.3、缩略图	379
8.3、页面级导航	380
8.3.1、页面标志	380
8.3.2、协议	382
8.3.3、显示	384
8.4、注释	388
8.4.1、注释规则	390
8.4.2、注释标记	392
8.4.3、边框类型	394
8.4.4、显示流	395
8.4.5、注释类型	397
8.5、操作	419
8.5.1、操作字典	420
8.5.2、触发事件	420
8.5.3、操作类型	423
8.6、交互表格	440
8.6.1、交互表格字典	441
8.6.2、域字典	443
8.6.3、域类型	451
8.6.4、表格操作	466
8.6.5、命名页	472
8.6.6、格式数据格式	472
8.7、数字签名	485
8.7.1、转换方法	490
8.7.2、签署互用性	496
8.7.3、权限	498
8.7.4、合法内容认证	499
8.8、测量性能	501
第9章 多媒体特性	508
9.1、多媒体	508
9.1.1、可行性	510
9.1.2、简述	511
9.1.3、媒体剪辑对象	516
9.1.4、媒体播放参数	521
9.1.5、媒体屏幕参数	524
9.1.6、其他多媒体对象	529

9. 2、声音	534
9. 3、电影	536
9. 4、候补介绍	538
9. 5、3D 艺术品	541
9. 5. 1、3D 注解	542
9. 5. 2、3D 流	546
9. 5. 3、3D 视图	551
第 10 章 文件交互	561
10. 1、程序集	562
10. 2、元数据	563
10. 3、文件标识符	566
10. 4、页片 (PAGE-PIECE) 词典	567
10. 5、标记目录	569
10. 5. 1、属性列表	571
10. 5. 2、标记目录和剪切	571
10. 6、逻辑结构	575
10. 6. 1 结构层次	576
10. 6. 2、结构类型	580
10. 6. 3、结构内容	581
10. 6. 4、结构属性	595
10. 6. 5、逻辑结构例子	601
10. 7、标记 PDF	609
10. 7. 1、标记 PDF 和网页内容	610
10. 7. 2、基本布局模型	621
10. 7. 3、标准结构类型	624
10. 8、辅助功能支持	671
10. 8. 1、自然语言规范	672
10. 8. 2、备用说明	677
10. 8. 3、代替文本	678
10. 8. 4、缩略语的全写	679
10. 9、网页捕获	680
10. 9. 1、网页捕获信息词典	680
10. 9. 2、内容数据库	681
10. 9. 3、内容集合	687
10. 9. 4、源信息	690
10. 9. 5、和网页捕获相关的对象的属性	694
10. 10、打印前支持	695
10. 10. 1、页面边界	695
10. 10. 2、打印机的商标	698
10. 10. 3、分离字典	702
10. 10. 4、输出意向	704
10. 10. 5、捕获支	709
10. 10. 6、开放的打印页面	714

第 4 章 图表

用于 PDF 内容流的图形运算描述了要在光栅输出设备复制的页面外观。在这一章中所描述的设施目的是打印机和显示应用。

图形操作分成六类：

图形状态运算操作的数据结构称为图形状态，即其他的图形运算来执行的整体架构。图形状态包括当前变换矩阵 (CTM)，将使用一个 PDF 内容流的用户空间左边映射到输出设备坐标。它也包括当前颜色，当前剪辑路径，和很多绘图运算的其他隐含参数。

路径构成运算指定的路径，它定义形状，线轨迹，和各种排序的地区。它们包括开始一个新的路径、加入线段和曲线、关闭它这样一些运算。

路径绘制运算用颜色填装一个路径，沿着它绘制一个冲程，或者使用它作为分界限。

其他绘图运算绘制某些自我描述的图形对象。这些包括采样图像，几何定义的阴影，和依次包含图形操作序列的整个内容流。

文本操作选择和显示从字体到字符字形（字体的字形描述字符）。因为 PDF 把符号当做一般的图形形式来对待，所以很多文本操作可以以图形状态或者绘制操作来分类。然而，处理字符字体描述的数据结构和机制如此专业以至于第五章将聚焦于此。

标记内容操作将高水平的逻辑信息和内容流里的对象联系在一起。这个信息不回影响到内容呈现的外观（虽然它可能会决定内容是否必需得显示；见 4.10 部分，“可选内容”）；它对于使用 PDF 进行文档交换的应用程序是很有用的。10.5 部分对标记内用进行了描述，“标记内容”。

本章提出了关于 PDF 中设备无关的图形的总体说明：一个 PDF 内容流怎样描述页面的抽象外观。关于图形的设备有关部分在第 6 章里有介绍。目录列出了一些提供这些计算机图形概念和执行细节的书。

4.1、图形对象

如 3.7.1 部分所述，“内容流”，内容流里的数据被解释为一系列操作，并且它们的操作数，根据标准 PDF 句法被表示为基础数据对象。内容流可以描述一个页面的外观，或者可以摩尔些其他环境中的图形元素。

操作数和操作继续用后缀表示法写入。虽然这个表示法类似于附言语言的顺序执行模型，PDF 内容流不是一个程序解释；相反，它是一系列图形对象的静态描述。有一些特殊的规则，如下所述，针对写入描述图形对象的操作数和操作。

PDF 提供五种类型的图形对象：

一个路径对象是由直线，矩形，立方贝塞尔曲线组成的任意形状。一条路径可能自身相交并且可能有断开的部分和中断。路径对象以一个或更多绘制操作来结束，这些操作指定了路径是否被描边，填充，作为分界线使用，或者这些操作的一些组合。

文本对象包括确定绘制字形序列的一个或多个字符字符串。像路径一样，文本可以被描边，填充，或用作一个剪辑边界。

外在对象 (`XObject`) 是一种定义在内容流之外并且作为一种被参考的指定资源的对象（参见 3.7.2 部分，“资源字典”）。`Xobject` 的解释取决于它的类型。图像的 `XObject` 定义了其将被绘制的颜色样本的矩形数组；窗体的 `XObject` 是一个被视为单个图形对象的整个内容流。特殊类型窗体的 `XObjects` 用于将内容从一个 PDF 文件中导入另一个（参考 `XObjects`），和组合图形元素在一起作为一个单元用于各种目的（组合 `XObjects`）。特别的是，后者用于定义透明度组来用于透明成像模型（透明度组 `XObjects`，第 7 章详细介绍）。还有一个补充的 `XObject`，请不要使用。

内嵌图像对象使用一种特殊的语法来表示直接在内容流内的小图像数据。

阴影对象描述颜色是位置的一个任意作用在形状之内的几何形状。（阴影也被看做绘制其他图形对象时的一种颜色；在这种情形下它不被当作单独的图形对象。）

PDF 1.3 和初期版本使用每个图形对象依顺序被绘入一个不透明的想象模型，完全遮盖所有可能出现在页上的先前标记。PDF 1.4 介绍了一种对象可以有些透明的透明图像模型，允许以前绘制的标记显示。每个对象被以一种指定的不透明绘制在页上，这个指定的不透明可能在对象的形状内的每一点都恒定，或者也许从点到点变化。页面先前存在的内容行程了新对象合成的背景，这就致使了对象颜色和背景依据它们的各自不透明特性结合。页的任何特定点的对象可以被当作形成透明度堆，在堆里堆积命令被定义为命令对象指令，首先最底层的对象。堆里的所有对象根据它们的颜色，形状和不透明度可能造就结果。

PDF 的图形参数被如此安排以至于图像被以默认的完全不透明绘制，这将透

明图像模式的动作简化为不透明模式。因此，本章内容适用于不透明和透明模型，其他明确规定的地方除外；第 7 章对透明模型有充分描述概括。

尽管上述的绘制行为往往是作为一个单独操作来创造对象，它仍然经常被作为一个被绘制的整体对象。图 4.1 显示定义了图形对象的操作的预定的规则。有些操作只在图形对象之间的某些图形对象类型上被允许（称作图的页面描述水平）。没一个内容流以页面的描述水平作为开始，这里图形状态可以被更改，比如以下几节中讨论的颜色和文字属性。

在图里，箭头指向标记每种类型的图形对象开始或结束的操作。一些操作是单独确定的，其他的由一般类别确定。表 4.1 总结了所有 PDF 操作的这些类别。

表 4.1 操作类别

类别	操作	表	页码
一般图形状态	w, J, j, M, d, ri, i, gs	4. 7	189
特殊图形状态	q, Q, cm	4. 7	189
路径构造	m, l, c, v, y, h, re	4. 9	196
路径绘制	S, s, f, F, f*, B, B*, b, b*, n	4. 10	200
裁剪路径	W, W*	4. 11	205
文本对象	BT, ET	5. 4	375
文本状态	Tc, Tw, Tz, TL, Tf, Tr, Ts	5. 2	368
文本配置	Td, TD, Tm, T*	5. 5	376
文本显示	Tj, TJ, ', "	5. 6	377
类型 3 字体	d0, d1	5. 10	392
颜色	CS, cs, SC, SCN, sc, scn, G, g, RG, rg, K, k	4. 24	257
阴影模式	sh	4. 27	273
内嵌图像	BI, ID, EI	4. 42	322
外在对象	Do	4. 37	302
标记内容	MP, DP, BMC, BDC, EMC	10. 7	779
兼容性	BX, EX	3. 29	127

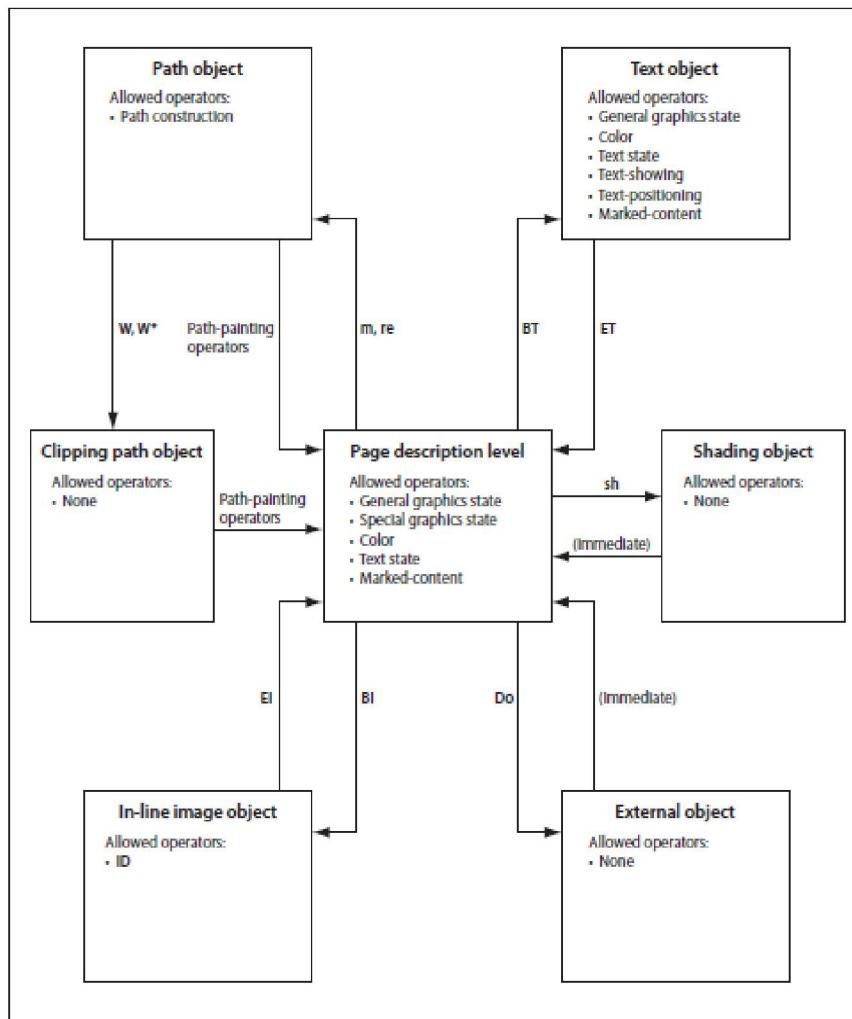


图 4.1 图形对象

例如，路径构造操作 **m** 和 **re** 标识着路径对象的开始。在路径对象内，增加路径构造操作是允许的，如同剪辑路径操作 **W** 和 **W***，但是一般的图形状态操作如 **w** 或 **J** 不允许。路径绘制操作，如 **S** 或 **f**，结束了路径对象并且返回页面描述标准。

注：操作违反这些描述图形对象规则的操作会产生不可预知的行为，即使它可能显示和打印正确。试图提取图形对象进行编辑或者其他目的的应用程序依赖于对象状态良好的形成。针对图形对象的规则对于标记内容的适当解释也很重要（见 10.5 部分，“标记内容”）。

图形对象也隐含着所有的印象其行为的图形状态参数。例如，路径对象取决于当前颜色参数的值，与此同时路径对象被定义了。其效果如同这个参数被指定为路径对象定义的一部分。然而，在页面描述级别上调用去设定图形状态参数的

操作不认为属于任何指定图形对象。图形状态参数仅当其改变时需要被指定。图形对象可能依赖于定义的早期参数。

同样，文本字符串对象的特点隐含它们所依赖的图像状态参数。这些参数大多数可能被设定在文本对象的内部或外部。其效果如同它们分别被指定给每个字符串。

重要的是没有语义适用于图形状态操作的严禁协议。读或者写一个 PDF 内容流的应用程序不要求维持这个协议，但是却可以自由将它改成可以获得每个图形对象的关于图形状态参数相同值的其他协议。一个应用程序不应该从构成一个图形对象的标志协议推论任何高级别的逻辑语义。一个独立机制，标记内容（见 10.5 部分，“标记内容”），允许这样的高级别信息明确地关联图形对象。

4.2、坐标系统

坐标系统定义了所有绘图显现的画布。它们决定了位置，方向，文本和图形的大小，和显示在页面上的图像。这一节介绍了用于 PDF 的每个坐标系统，它们如何相关，它们之间如何转换和指定。

注：本节介绍的坐标系统用适用于二维图形。PDF1.6 介绍了显示三维作品的方法，其中对象在一个三维坐标系统中被描述，如 9.5.4 部分所述，“三维坐标系注解”。

4.2.1、坐标空间

路径和位置就坐标对而言定义在直角平面上。一个坐标对是在一个二维坐标空间水平和垂直地找出一个点的一对实数 x 和 y 。二维坐标空间由关于当前页的以下属性决定：

原点位置

x 轴和 y 轴的方向

沿着每个轴的单位长度

PDF 格式定义了几个坐标空间中的坐标指定的图形对象被解释。以下各节描述了这些空间和它们之间的关系。

坐标之间的转换由转换矩阵决定，其可以指定任何二维线性映射的坐标，包括平移，缩放，旋转，反射和倾斜。转换矩阵在 4.2.2 部分有介绍，“常见转换”，

和 4.2.3, “转换矩阵”。

设备空间

页面的内容最终显示在光栅输出设备，如显示器或者打印机。这种设备在在可成像领域用来处理像素的内置坐标系统里改变巨大。一个为特定设备的坐标系统被称为它的设备空间。不同设备的设备空间来源可以集合到输出页面的不同地方；在显示上，这个来源可以根据不同的窗口系统而改变。由于纸张和其它输出媒介通过不同的打印机和网片输出机在不同方向上移动，所以它们的设备空间的坐标轴可能面向不同方向。例如，垂直 (y) 坐标可能在某些设备上从页面的顶部向底部增长而在其它设备上从底部向顶部增长。最后，不同的设备有不同的决议；有的甚至在水平和垂直方向有不同决议。

如果 PDF 文件中的坐标被指定在设备空间，文件将会依赖设备，并且将会在不同设备上显示不同。例如，指定在一个 72 像素每英寸显示器和一个 600 点每英寸打印机的典型空间里的图像将有大于 8 倍的区别；一个显示器上 8 英寸的线段将在打印机上显示出不到 1 英寸长。图 4.2 展现了在指定的设备空间里相同的图形对象怎样在不同的输出设备上显示截然不同。

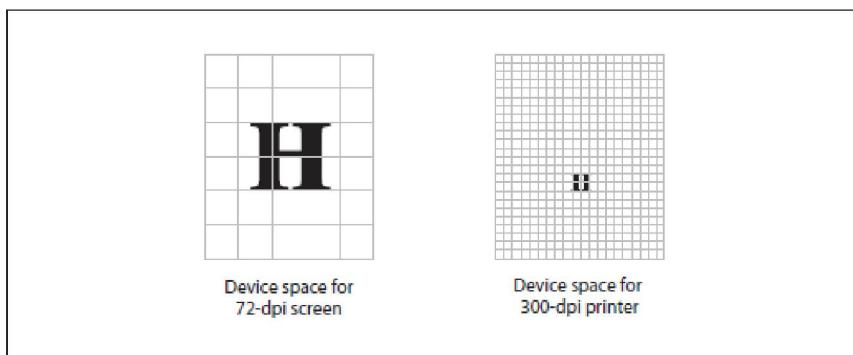


图 4.2 设备空间

用户空间

为了避免设备空间里指定对象的设备依赖影响，PDF 定义了一个始终承担着当前页面的相同关系的设备无关坐标系统，无论输出设备在打印或是显示。这个设备无关坐标系统称为用户空间。

用户空间坐标系统初始化每一个页面到默认状态。进入到页面字典 中的裁剪框指定了用户空间的矩形，其相当于预期输出工具的可见区域（显示窗体或者打印页面）。x 轴沿着水平向右，y 轴垂直 向上，如标准的运算实践（在轮流进

入页字典时候需要修改)。沿 x 轴和 y 轴的长度单位在页字典中由用户单元项 (PDF1.6) 设置 (见表 3.27)。如果该条目不存在或者不支持, 默认使用 1/72 英寸值。这个坐标系统被称为默认用户空间。

注: 在附言中, 默认用户空间的原点始终对应于左上角输出项。尽管这一公约在 PDF 文件里很常见, 但是它不是必须的; 页字典的裁剪框项可以指定默认用户空间的任何矩形在媒介上可见。

注: 在默认用户空间 (1/72 英寸) 的默认单位大小大致与一个点相同, 是一个广泛应用在印刷的单位。这并不完全一样, 然后, 没有一个点的普遍定义。

从概念上讲, 用户空间是一个无限的平面。仅仅这个平面的一小部分对应于输出设备的成像区域: 一个由裁剪框项定义在页字典里的矩形区域。用户默认空间区域, 其对于每一页查看或者打印可以不同, 在 10.10.1 部分将被介绍, “页面边界”。

注: 由于用户控件的坐标 (如同其他坐标空间中一样) 可能被指定为整型或者实数, 默认用户空间上的单元大小不限制任意网格的位置。在用户控件的该坐标协议不涉及任何设备空间上的像素决议。

从用户空间到设备空间的转化由当前变换矩阵 (CTM) 定义, 一个 PDF 图形状态的元素 (见 4.3, “图形状态”)。一个 PDF 用户应用程序可以为特定的输出设备的固有协议调整 CTM, 维护 PDF 页面描述的设备无关性。图 4.3 展示了如何在用户空间指定对象使其在不同的设备上显示相同。

默认的用户控件提供了一个使用任意输出设备时一致, 可靠的 PDF 页面说明的起点。如果有必要, 一个 PDF 内容流可能会修改用户控件来更加适应于其对左边变换操作的需求, cm (见 4.3.3, “图形状态操作”)。因此, 在一个内容流里可能是绝对坐标的并不是关于当前页面绝对的, 因为它们表示在一个可能左右滑动, 扩大或缩小的坐标系中。坐标系转换, 不仅提高了设备的独立性, 而且是一种在其自我作用中有用的工具。例如, 一个内容流通过缩小它所在的坐标系来组成占据了原本可以合并的一整个页面而没有像另外一个页面的元素一样改变。

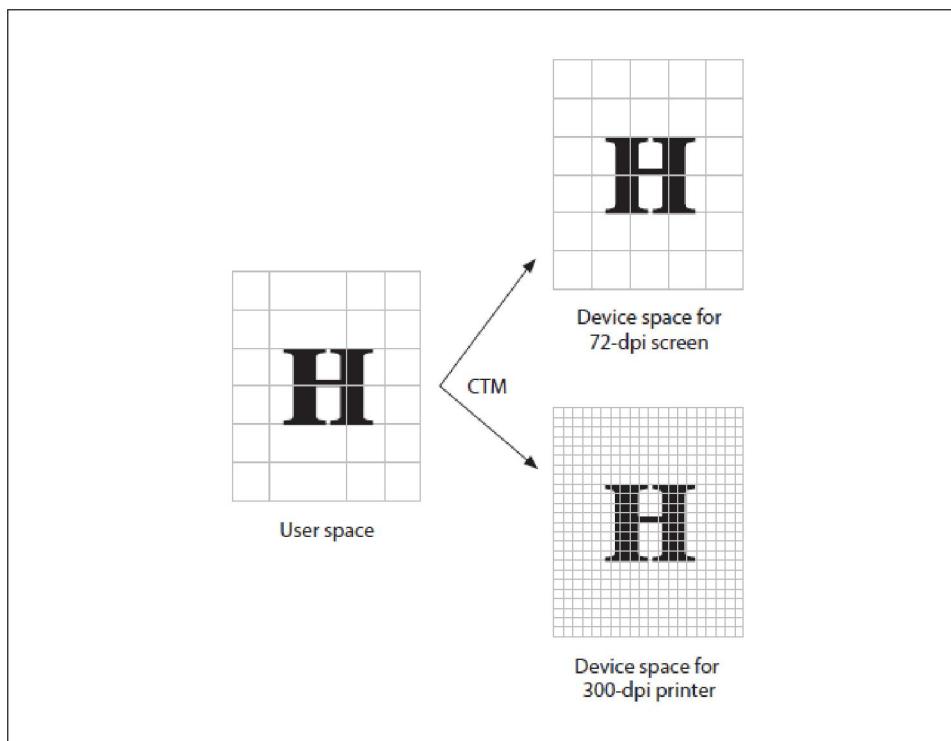


图 4.3 用户空间

其他坐标空间

除了设备空间和用户空间，PDF 针对专门用途使用了其他各种坐标空间：

文本坐标被指定在文本空间。从文本空间到用户空间的转化被一个文本矩形定义来结合图形状态中的一些文本关联参数（见 5.3.1，“文本定位操作”）。

字体中的字符字形定义在字形空间（见 5.1.3，“字形定位和衡量标准”）。从字形空间到文本空间的转化被字体矩阵定义。对于大多数类型的字体，这个矩阵式预定义映射字形空间的 1000 个单元到文本空间的 1 个单元；对于类型 3 字体，字体矩阵在字体字典里明确的给出了（见 5.5.4，“类型 3 字体”）。

所有的图像采样都定义在图像空间。从图像空间到用户空间的转化是被预置且不能更改的。所有的图像在用户空间都是一单元宽一单元高的，无论图像的样本数量。绘制时，图像暂时改变 CTM 来映射到页面上的区域。

注：补充一下，不像 PDF，图像空间和用户空间之间的关系可以被明确指定。PDF 中规定的固定转化相当于建议使用在补充说明中的公约。

一个表格外在对象（4.9 部分介绍，“表格外在对象”）是一个可作为在另一个内容流里的图形元素的独立的内容流。它被定义的空间称为表格空间。从表格空间到用户空间的转化被包含在表格外在对象里的表格矩阵所定义。

PDF1.2 定义了一种类型的颜色称为模式，4.6 部分有介绍，“模式”。模式可以通过重复调用来覆盖一个区域的内容流或者其颜色是一个位置的变量的阴影来定义。从模式空间到用户空间的转化被包含在模式里的模式矩阵来指定。

PDF1.6 介绍了嵌入式 3D 作品，其在三维坐标里被描述（见 9.5.4，“三维坐标系统说明”），这个系统预计成为一个注释到目标坐标系统（见 9.5.1 部分，“三维注解”）。

坐标空间之间的关系

图 4.4 显示了上述坐标之间的空间关系。图中每一个箭头代表了从一个坐标空间变换到另一个。PDF 允许修改这些变化中的大多数。

由于 PDF 坐标空间互相关联定义，一个转化的改变可以影响定义在某些坐标空间的对象外观。例如，一个 CTM 里的改变，其定义了从用户空间到设备空间的转化，影响了表格，文本，图像和模式，因为他们都是从用户空间来的上行数据流。

4.2.2、常见转化

变换矩阵指定了两种坐标空间之间的关系。通过修改变换矩阵，对象可以缩放，旋转，转化，或者以其他方式改变。

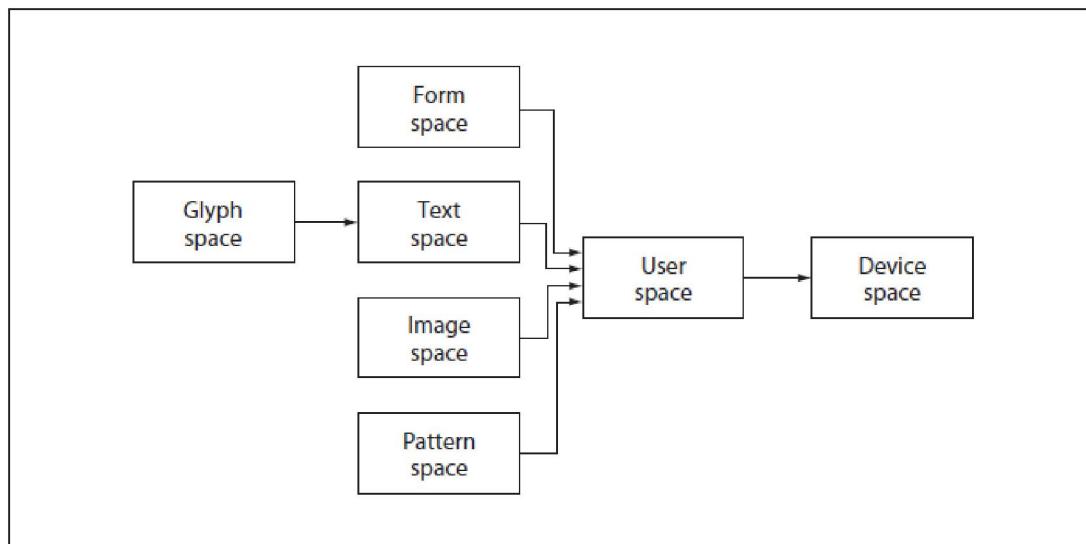


图 4.4 坐标系统之间的关系

PDF 中的变换矩阵是由通常是以包含 6 个元素的数组形式出现的 6 个号码指定。在其最一般的形式中，此数组表示为 $[a \ b \ c \ d \ e \ f]$ ；它可以表示任何从一个坐标系到另一个的线性转换。本节列出了指定最常见转化的数组；4.2.3 部分，“转型矩阵”，介绍了更多转化的运算细节，包括指明组合这里列出信息的转化的信息：

转化被指定为 $[1 \ 0 \ 0 \ 1 \ tx \ ty]$ ，其中 tx 和 ty 是分别为转化坐标系统的原点在水平和垂直尺寸上的距离。

缩放由 $[sx \ 0 \ 0 \ sy \ 0 \ 0]$ 来得到。如此衡量的坐标致使新坐标系用里水平和垂直尺寸的 1 单元分别和先前坐标系统里的 sx , sy 单元大小一样。

旋转是由 $[\cos \theta \ \sin \alpha \ -\sin \theta \cos \theta \ 0 \ 0]$ 造就，其受到 θ 角逆时针方向旋转坐标系轴的影响。

偏离由 $[1 \ \tan \alpha \ \tan \beta \ 1 \ 0 \ 0]$ 指定，其中 α 角影响偏离 x 轴且 β 角影响偏离 y 轴。

图 4.5 显示了每个转型的例子。转化，旋转以及偏离的方向的数据显示相当于相当于数组元素的正值。

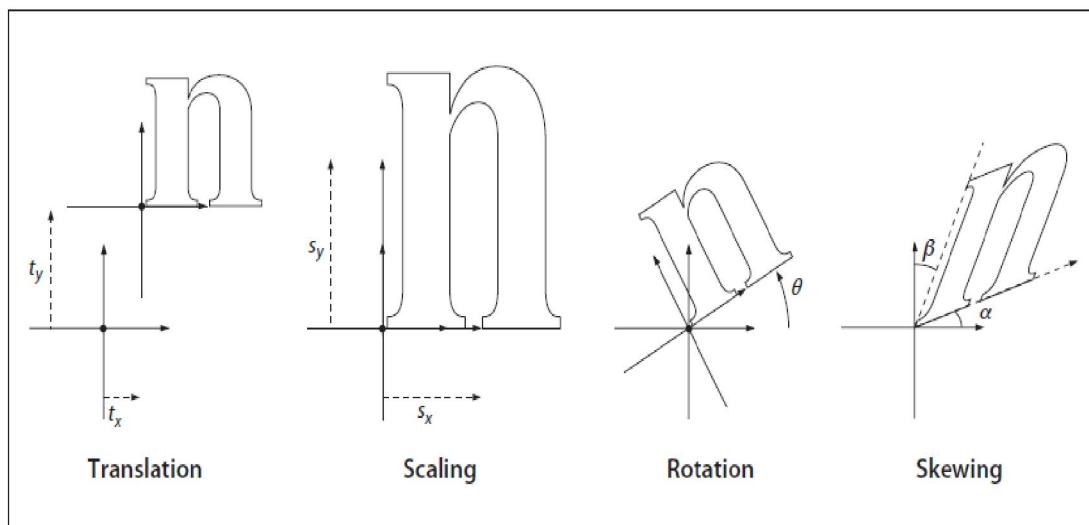


图 4.5 坐标转化的影响

如果有几个转变结合起来，它们应用的顺序是非常重要的。例如，先缩放然后转换 x 轴和先转换后缩放它是不同的。一般而言，为取得预期的结果，转换工作应该按照以下顺序：

转换

旋转

缩放或歪斜

图 4.6 显示了转化应用的次序的影响。这个图显示了应用于坐标系统的两个转化序列。在每一次连续转换后，一个字母 n 的轮廓被绘制。

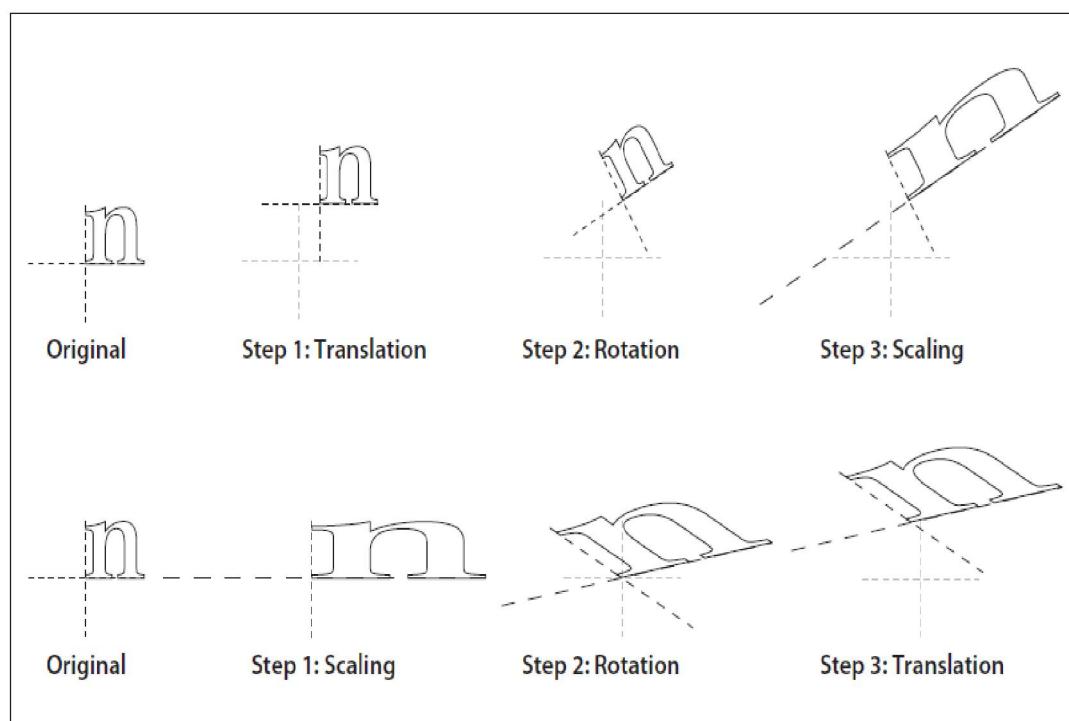


图 4.6 转换次序的影响

以下是显示在图中的转换：

在 x 方向上转化 10 个单位同时在 y 方向上转化 20 个单位

一个 30 度的旋转

一个 x 方向上的 3 倍缩放

在这个图中，两轴是以由具有 2 单位破折线和 2 单位空隙组成的短划线表示的。另外，原始的（未转化）轴以较淡的颜色标示。请注意，缩放-旋转-转化的排序导致了坐标系统的失真，使 x 与 y 轴不再垂直；而被推荐的缩放-旋转-转化排序没有导致失真。

4.2.3、转化矩阵

本节讨论的是转化矩阵的运算。没有必要阅读本节去使用之前描述的转换；这部分信息是为便利想要获得坐标变换理论基础的更深认识的读者。

为了解 PDF 里坐标转换的运算，重要的是要记住两点：

转化改变坐标系统，而不是图形对象。在一个转化应用之前绘制的所有对象都不受转化影响。在转化应用之后绘制的对象在转换坐标系中被解释。

变换矩阵指定了从新的（转换的）坐标系统到原始的（未转换的）坐标系统的转换。在转换之后使用的所有坐标都表示在了转换的坐标系中。PDF 应用转化矩阵到未转换的坐标系统中去寻找相同坐标。

注：很多计算机图形教科书更多的考虑图形对象的转换而不是坐标系的转换。尽管两种方法都是正确一致的，取决于所采取的观点的一些计算的细节不同。

PDF 代表了一个二维空间的坐标。在这样一个空间的点 (x, y) 可以表示为向量形式 $[x \ y \ 1]$ 。这个向量的恒定的第三个元素 (1) 被需要以至于这个向量可以在以下描述的计算中与 3×3 矩阵一起使用。

两个坐标系之间的转化时由一个 3×3 的转换矩阵表示如下：

$$\begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix}$$

由于一个转化矩阵只有 6 个元素是可以改变的，所以通常在 PDF 中它被指定为一个六元素的数组 $[a \ b \ c \ d \ e \ f]$ 。

坐标转换表示为矩阵乘法为：

$$[x' \ y' \ 1] = [x \ y \ 1] \times \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix}$$

由于 PDF 转换矩阵指定了从转换坐标系到原始（未转换）坐标系的转换，在这个等式里 x' 和 y' 为未转换坐标系的坐标， x 和 y 是转换了的坐标系的坐标。乘法是按照以下程序进行：

$$\begin{aligned} x' &= a \times x + c \times y + e \\ y' &= b \times x + d \times y + f \end{aligned}$$

如果一系列的转换进行，代表每一个单独转换的矩阵可以相乘一起产生一个表示复合转换的单一相等矩阵。

矩阵乘法是不可交换的——矩阵相乘的顺序非常重要。考虑两个转换的一个顺序：一个缩放转换应用到用户空间的坐标系，随后一个导致缩放用户空间到设备空间的转换。让 **MS** 作为指定缩放比例的矩阵同时 **MC** 作为转换用户空间到设备空间的当前转换矩阵。回顾坐标经常被定义在转换空间中，正确的转换顺序必须首先将比例坐标转换到默认用户空间，然后转换默认用户空间坐标到设备空间。这可以被表示为：

$$XD = XU \times MC = (XS \times MS) \times MC = XS \times (MS \times MC)$$

在此

XD 指在设备空间的坐标

XU 指在默认用户空间的坐标

XS 指在比例用户空间的坐标

这表明当一个新转换关联到一个已现有的，表示它的矩阵必须在已存在的转换矩阵前（预乘用）被乘。

一般而言这个结果对 PDF 是正确的：当一系列转换进行时，表示复合转换的矩阵(**M'**)被以预相乘代表额外转换的矩阵(**MT**)和代表所有已存在转换的矩阵(**M**)的方式计算：

$$M' = MT \times M$$

注：当绘制图形对象时，有时一个应用程序去执行相反的转换，即，查找用户空间坐标对应的特定设备空间的逆坐标对是很有必要的。然而并非所有的变化都是可逆的。例如，如果一个矩阵包含都为 0 的 **a**, **b**, **c**, **d** 元素，所有的用户坐标映射到相同的设备坐标，也没有独有的逆转换。这种不可逆转换不是很有用并且一般产生于意外操作，比如使用 0 进行缩放。绘制图形对象时使用一个不可逆的矩阵可以导致不可预测的行为。

4.3、图形状态

PDF 的用户应用程序维护的一个内部数据结构称之为图形状态，其保存了控制参数的当前图形。这些参数定义了图形操作执行的整体架构。例如，**f** (填充) 操作隐式使用了当前的颜色参数，另外 **S** (绘制) 操作使用了图形状态的当前的线宽参数。

图形状态在每一页的开头和表 4.2, 4.3 中的指定值一起初始化。表 4.2 列出了设备无关和在页说明中适当定义的图形状态参数。表 4.3 中列出的参数控制了转换（扫描转换）过程的细节并且设备相关；一个旨在设备无关的页描述不可修改这些参数。

表 4.2 独立设备图形状态参数

参数	类型	值
CTM	数组	目前的转换矩阵，其将用户坐标位置映射到设备坐标（见 4.2 部分，“坐标系统”）。这个矩阵被每一个坐标转换操作所修改， cm 。初始化值：矩阵将默认用户坐标转换到设备坐标。
裁剪路径	(内部)	定义针对所有输出的边界的当前裁剪路径将被裁剪（见 4.4.3 部分，“裁剪路径操作”）。初始化值：输出页面的整个可成像部分的边界。
颜色空间	名字或数组	颜色值将被说明的当前颜色空间（见 4.5 部分，“颜色空间”）。两个独立的颜色空间的参数：一个针对绘制令一个针对其他所有的绘制操作。初始值： DeviceGray 。
颜色	(多种)	绘制操作中所使用的当前颜色（见 4.5 部分，“颜色空间”）。这个参数的类型和说明取决于当前颜色空间；对于大多数颜色空间，对于大多数颜色空间，一个颜色值包含 1 到 4 的数量。两个独立的颜色参数：画程和其他所有绘制操作。初始值：黑色。
文本状态	(多种)	只关于到文本绘制的 9 种图形状态参数设置。这包含选择字体，衡量字形到合适大小，以及实现其他效果的参数。文本状态参数在 5.2 部分有介绍，“文本状态参数和操作”。
线宽	常数	在用户空间单元上路径宽度将被绘制（见 185 页的“线宽”）。 初始值：1.0。
线端口	整型	指定被绘制的任何开放路径终点类型的代码（见 186 页的“线端口样式”）。初始值：0，对方形顶点。

线 联 结	整型	指定连接被绘制的线段间的接口形状（见 186 页“线联结样 式”）。初始值：0，针对斜接连接。
斜 面 限制	常量	加入绘制路径的斜线的最大长度（见 187 页“斜面限制”）。 这个参数限制了“spikes”长度的产生当线段加入锐角时。 初始值：10.0，对低于大约为 11.5 度斜切截止。
虚 线 模式	数组和常 量	虚线模式的描述当路径被绘制的时候被使用（见 187 页“线 的虚线模式”）。初始值：实线。
再 现 意图	命名	当转换给予 CIE 的颜色到设备颜色时再现意图被使用（见 230 页“再现意图”）。初始值：相对色度再现。
行 程 调节	布尔型	(PDF1.2) 一个被指定是否弥补可能的光栅效果的标志当用 一个与输出设备像素些许相关的线宽来绘制路径时产生影 响（见 6.5.4 部分，“自动行程调整”）。注意这被认为是一 种独立设备的参数，即使它们细节影响是设备有关的。初始
混 合 模式	命名或数 组	(PDF1.4) 当前被用在透明成像模式中的混合模式（见 7.2.4 部分，“混合模式”，以及 7.5.2，“定义混合颜色空间和混合 模式”）。在执行一个透明度类别外在对象的开端这个参数被
Soft mask	字典或命 名	(PDF1.4) 一个 soft-mask 字典（见 520 页的“Soft-mask 字典”）指定隐含形状或者隐含的用于透明成像模式的不透 明度值（见 495 页“源形和不透明度”和 518 页的“隐形和 一透口之八”）。
阿 尔 法 常 数	常数	(PDF1.4) 其恒定形状或者恒定不透明值被用于不透明成像 模式（见 495 页的“源形与不透明度”和 519 页的“常形 与不透明度”）。两个单独的阿尔法参数：行程和其他所有的
阿 尔 法 来 源	布尔型	(PDF1.4) 一个定义是否当前的 soft mask 和阿尔法常量参 数被作为一个形状值 (true) 或者不透明度值 (false) 解 释的标志。这个标志也制约了任何如果在图像字典（见 4.4.8，“图像字典”里的 Smask 项的解释。初始值：false。

表 4.3 依赖设备的图形状态参数

参数	类型	值
套印	布尔型	(PDF1.2) 一个指定(在输出设备上支持套印控制功能)是否设置一种调色剂绘制可以引起对应的其他区域的着色剂的清除 (false) 或者保持不变 (true) 的标志。见
套印模式	常量	(PDF1.3) 一个指定在套印的时候是否一个在 DeviceCMYK 颜色空间中为 0 的颜色成分值可以清除成分
黑代	函数或命名	(PDF1.2) 一个当将 RGB 颜色转化为 CMYJ (见 6.2.3 部分, “从 DeviceRGB 转化到 DeviceCMYK”) 时计算黑色组
底色去除	函数或名称	(PDF1.2) 一个计算青色, 品红, 黄色成分级别减少以黑色增加来补偿的函数 (见 6.2.3 部分, “DeviceRGB 到
传递	函数, 数组或名称	(PDF1.2) 调整设备灰度或者颜色组件分量以补偿一个特别的输出设备里德非线性响应的函数 (见 6.3 部分,
半色调	字典, 流, 或者命名	(PDF1.2) 一个针对灰色和颜色显示的半色调屏幕, 被指定为一个半色调字典或者流(见 6.4 部分, “半色调”)。
平整度	常量	曲线要呈现在输出设备上的精确度 (见 6.5.1, “平整度公差”)。该参数值给出了在输出设备像素上测量的所能
平滑	常量	(PDF1.3) 颜色梯度被显示在输出设备上的精度 (见 6.5.2 部分, “平整度公差”)。这个参数的值给出了所允许的最大错误, 表示为一系列颜色组成部分的一小部分:

某些图像状态参数被以指定的 PDF 操作设定, 一些被包括一个在图形状态参数字典中的特定项设定, 一些可以被其他方式指定。当前线宽, 例如可以被 `w` 操作或 (在 PDF1.3) 图形状态参数字典里的 `LW` 项设定, 而当前的颜色值被指定操作设定, 且当前的半色调只被图形状态字典设定。预计所有的将来图形状

态参数将被新的图形状态参数字典里的项而不是新操作指定。

在一般情况下，设定图形状态参数的操作仅仅存储它们使它们对绘制操作以后的使用不变。然而有些参数有特殊的属性或行为：

大多数参数必须是正确类型或者在一定范围内下降。

参数是数字值的，如当前颜色，线宽，以及斜面限制，如有必要的话，被强制作正当范围。然而，它们不为反映光栅输出设备的能力而调整，比如可分辨的颜色的决议或者数量。尽管绘制操作进行这些调整，但是调整后的值不储存回图形状态中。

路径时不直接表现在 PDF 的内部对象。

注：如表 4.2 和 4.3 指出的，一些参数——色彩空间，色彩，和套印——有两个值，一个用作绘制轮廓（路径和文本对象的），一个用作所有绘制操作。两个参数值可以被独立设置，其被允许用作如用结合用不同颜色填充和绘制相同路径的操作。除特别注明外，如当前颜色这样的一个词需要针对适用于被执行的操作任何颜色参数被说明。必要时，个别颜色参数如同轮廓颜色和非轮廓颜色一样区分明确。

4.3.1、图形状态栈

一个结构良好的 PDF 文件通常包含了很多基本互相独立并且有时会多重层次嵌套的图形元素。图形状态栈允许这些元素在不干扰周围环境的图形状态的情况下对图形状态进行局部改变。这个栈是一个在其中图形状态的内容可以被保存并且之后用以下操作恢复的 LIFO（先进后出）数据结构：

q 操作将全部图形状态的一个副本压入栈内。

Q 操作以从堆栈中弹出整个图形状态的方式来恢复其原来的值。

这些图形操作可以用来封装一个图形元素以至于它可以修改图形状态的参数并且之后恢复它们到原来的值。**q** 和 **Q** 操作必须在给定的内容流中保持平衡（或者在指定在一个页字典的内容数组的流次序中）。

4.3.2、图形状态参数的细节

这一节给出了位于表 4.2 一些独立设备的图形状态参数列表的细节。

线宽

线宽制定了用作绘制一条路径的行的厚度。其在用户空间单位上表示为一个非负数；绘制一条路径要求所有的点在用户空间路径的垂直距离小于或等于线宽的一半。产生在设备空间的影响取决于在路径被绘制时候当前的转化矩阵(CTM)。如果这个 CTM 指定在水平和垂直上缩放尺寸的不同因素，设备空间里轮廓线段的厚度将随它们的方向而改变。实际线宽可以与要求宽度最多有 2 设备像素的差别，取决于关联像素网格的行的位置。自动调整绘制轮廓可用于确保行的宽度均匀，见 6.5.4，“自动调整行程”。

宽度为 0 的线表示可以被设备分辨率显示的最细的行：1 设备像素宽度。然而，一些设备不能重现 1-像素的线，并且在高分辨率的设备上，它们几乎不可见。由于显示这种 0-宽度线的结果是设备相关的，所以不推荐使用它们。

线端口样式

线端口样式指定被用在开放的子路径（和短划线，如果有的话）的两端的形状当其被绘制时。表 4.4 展示了可能的值。

表 4.4 线端口样式		
样 式	显示	描述
0		巴特端口。在路径的终点绘制被方形关闭。没有超过预测的路径末尾。
1		圆形端口。带有一条等于线宽的半圆弧被画在终点并被填充。
2		突出的方形端口。绘制仍然以一个等同于线宽的一半的距离超出了道路终点并且方形结束。

线连接样式

该线连接样式定义了被用在被绘制的路径角的形状。表 4.5 展示了可能的值。结合样式只有在一条路径的连续部分以一个角度连接在一个点时才有意义；意外接合或交叉的线段没有特殊意义。

表 4.5 线连接样式

样 式	显示	描述
0		Miter 连接。两线段轮廓的外边缘被延伸直到它们以一个角度接合，如同在一个图框里一样。如果线段接合为一个过于小的锐角（如被 <code>miter</code> 限制参数定义——见前面的“Miter Limit”），一个斜接合被用来代替。
1		圆形连接。带有一条等于线宽的圆弧被画在两线段相交的点，连接两线段轮廓的外边缘。这个扇形图被填充，产生了一个圆形的角。
2		锥形连接。这两条线段用 <code>butt</code> 端口完成（见 186 页“线端口样式”）并且导致线段末端的超出部分被一个三角形填充。

注：圆形连接的定义在 PDF1.5 中被改变。在极少数情况下，以前的规范实施可能产生意想不到的后果。

斜切极限

当两线段以一个锐角交汇并且斜切接合被指定为线结合样式，斜切延伸得远于路径轮廓线的宽度是有可能的。斜切极限规定了针对线宽的斜切长度比的最大值（见图 4.7）。当超出极限，连接将从斜切转换为斜面。

该针对线宽的斜切长度比通过以下公式直接关系到用户空间的线段之间的角度 ϕ ：

$$\frac{\text{miterLength}}{\text{lineWidth}} = \frac{1}{\sin\left(\frac{\varphi}{2}\right)}$$

例如，一个 1.414 的斜切限制转化为 ϕ 小于 90 度，2.0 转化为 ϕ 则小于 60 度，同时 10.0 转化为 ϕ 则小于大约 11.5 度。

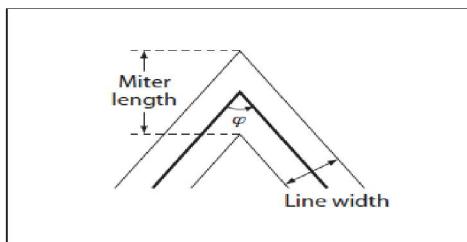


图 4.7 斜切长度

虚线模式

该虚线模式控制了用于绘制路径的虚线和间隙。它被一个虚线数组和一个虚线分割指定。这个虚线数组的元素是指定交互短划线和间隔长度的数字。在短划线阶段指定的进入开始虚线的虚线模式的距离。短划线数组和短划线模式的元素被在用户空间单元表示。

在开始绘制一个路径之前，虚线数组通过增加短划线和间隔得长度来循环。当累积长度等于被虚线数组指定的值，路径绘制开始，并且虚线数组从一个向前的点开始循环使用。表 4.6 显示了虚线牧师的例子。可以从表中看出，一个空的虚线数组和 0 阶段可以被用作恢复虚线模式到实线。

TABLE 4.6 Examples of line dash patterns

DASH ARRAY AND PHASE	APPEARANCE	DESCRIPTION
[] 0		No dash; solid, unbroken lines
[3] 0		3 units on, 3 units off, ...
[2] 1		1 on, 2 off, 2 on, 2 off, ...
[2 1] 0		2 on, 1 off, 2 on, 1 off, ...
[3 5] 6		2 off, 3 on, 5 off, 3 on, 5 off, ...
[2 3] 11		1 on, 3 off, 2 on, 3 off, 2 on, ...

虚线环绕曲线和角和实线绘制一样。每一个短划线的末端被作为当前的线端口样式，并且有短划线的角被当做样式中的当前线。一个绘制出力不对协调虚线模式和路径外观采取任何措施；它仅仅施行被虚线数组定义的模式中沿着路径的短划线和间隔。

当包含几个子路径的路径被绘制，每一个子路径被独立处理——即，虚线模式重启并且虚线模式重新应用在每一个子路径的开端。

4.3.3、图形状态操作

表 4.7 显示了设定在图形状态中的参数值的操作。(也可以参考表 4.24 中列出的颜色操作和 368 页表 5.2 的文本状态操作。)

表 4.7 图形状态操作		
操作数	操作	说明
—	q	保存图形状态堆栈中的当前图形状态(见 184 页的“图形状态堆栈”)
—	Q	通过删除堆栈中最新存储的状态来还原图形状态，并让它成为当前状态((见 184 页的“图形状态堆栈”))
a b c d e f	cm	通过一系列被指定的数组来规定当前转化数组(CTM)(见 4.2.1 部分，“坐标空间”)。尽管这个操作定义了一个数组，它们被写作 6 个独立的数字而不是一个数组。
lineWidth	w	在图形状态中设置线宽(见 185 页“线宽”)。
lineCap	J	在图形状态里设置线端口样式(见 186 页“线端口样式”)。
lineJoin	j	在图形状态里设置线连接样式(见 186 页“线连接样式”)。
miterLimit	M	在图形状态里设置斜切极限(见 187 页“斜切极限样式”)。
dashArray dashPhase	d	在图形状态里设置虚线模式(见 187 页“虚线模式”)。
Intent	ri	(PDF1.1) 在图形状态里设置图像显示方式(见 230 页“显示方式”)。
flatness	i	在图形状态中设置平面度公差(见 6.5.1, “平面度公差”)。平整度是一个介于 0 到 100 的数字；一个为 0 的值指定输出设备的默认平整度公差

dicName	gs	(PDF1.2) 设置图形状态中指定的参数。dicName 是一个在当前源字典的 ExtGState 子字典中一个图形状态参数字典的名字（见下一部分）。
---------	----	--

4.3.4、图形状态参数字典

尽管在图形状态中的一些参数可以被独立的操作设置，如表 4.7 所示，但是其他的不行。后者只能被一半的图形状态操作 gs (PDF1.2) 设定。提供给这个操作的操作说是一个包含定义一个或多个图形状态参数的图形状态参数字典的名字。这个名字在当前源字典的 ExtGState 子字典中可查找。(ExtGState, 针对延伸图形状态，是一种早期版本的 PDF 的剩余)。

注：图形状态参数字典也被 2 型模式使用，其不含有图形状态操作可以被调用的内容流（见 4.6.3，“阴影模式”）。

参数字典里的每一项都制定了一个独立的图形状态参数值，如表 4.8 中所展示的。所有的项都不必相应每个 gs 操作请求；所提供的参数字典可能包含任何参数值的组合。Gs 的结果是累计；参数值建立在之前的请求上直到被明确覆盖。注一些参数同时出现在表 4.7 和 4.8；这些参数可以被单独的图形状态操作或者 gs 设置。预计任何针对图形状态的将来的延伸将通过增加新的项到图形状态参数字典而不是引进新的图形状态操作来实现。

表 4.8 图形状态参数字典中的项

键	类型	说明
Type	name	(可选) 这个字典描述的 PDF 类型；一定是对于一个图形状态参数字典的 ExtGState。
LW	Number	(可选；PDF1.3) 线宽（见 185 页“线宽”）。

LC	Integer	(可选; PDF1.3) 线端口样式 (见 186 页“线端口样式”)。
LJ	Integer	(可选; PDF1.3) 线结合样式 (见 186 页“线结合样式”)。
ML	Number	(可选; PDF1.3) 斜切极限 (见 187 页“斜切极限”)。
D	Array	(可选; PDF1.3) 被表示为一个[下划线数组 下划线段]的数组的虚线模式，在其下划线数组本身是一个数组且下划线段是一个整型 (见 187 页“虚线模
RI	Name	(可选; PDF1.3) 显示方式的名字 (见 230 页“显示方式”)。
OP	Boolean	(可选) 指定是否应用套印的一个标志 (见 4.5.6 部分, “套印控制”)。在 PDF1.2 和更早版本, 有一个应用于所有的绘制操作的独立套印参数。伴随着
op	Boolean	(可选; PDF1.3) 对其他绘制操作而不是轮廓绘制指定是否引用套印的一个标志 (见 4.5.6 部分, “套印控制”)。如果这个项不存在, 任何 OP 项设定这个
OPM	Integer	(可选; PDF1.3) 套印模式 (见 4.5.6, “套印控制”)。
Font	Array	(PDF1.3) 一个数组结构 [字体大小], 其字体是一个对于字体词典的间接参考并且它的大小事一个在文本空间单元表示的数字。这两个对象相当于 Tf 操
BG	Function	(可选) 黑代功能, 其映射区间 [0.0 1.0] 到区间 [0.0 1.0] (见 6.2.3, “从 DeviceRGB 到 DeviceCMYK 的转化”)

BG2	Function or name	(可选; PDF1.3) 除了值可能也是默认名外与 BG 一样指定在页面开始有效地黑代功能。如果 BG 和 BG2 同时出现在相同的图形参数字典中, BG2 优先。
UCR	Function	(可选) 底色迁移功能映射区间 [0.0 1.0] 到 [-1.0 1.0] (见 6.2.3, “从 DeviceRGB 到 DeviceCMYK 的转化”)。
URC2	Function or name	(可选; PDF1.3) 除了值可能也是默认名外与 UCR 一样指定在页面开始有效的底色迁移功能。如果 UCR 和 UCR2 同时出现在相同的图形状态参数字典里,
TR	Function , array or name	(可选) 可传递函数, 其映射区间 [0.0 1.0] 到区间 [0.0 1.0] (见 6.3, “传递函数”)。该值可以是一个单独函数 (应用于所有的着色过程) 或者一个 4
TR2	Function , array or name	(可选; PDF1.3) 除了值可能也是默认名外与 TR 一样指定在页面开始有效的传递函数。如果 TR 和 TR2 同时出现在相同的图形状态参数字典里, TR2 优先。
HT	Dictionary, stream or name	(可选) 半色调字典或者流 (见 6.4, “半色调”) 或者默认名, 指定了在页面开始有效的半色调。
FL	Number	(可选; PDF1.3) 平面度公差 (见 6.5.1, “平面度公差”)。
SM	Number	(可选; PDF1.3) 平滑度公差 (见 6.5.2, “平滑度公差”)。
SA	Boolean	(可选) 一个指定是否应用自动路径绘制调节的标志 (见 6.5.4, “自动路径绘制调节”)。
BM	Name or Array	(可选; PDF1.4) 被用在透明成像模式的当前混合模式 (见 7.2.4, “混合模式”, 以及 7.5.2, “指定

Smask	Dictionary or name	(可选; PDF1.4) 当前 soft mask, 指定用于透明成像模式的 mask 形状或者 mask 不透明度值(见 495 页的“源形和不透明度”以及 518 页的“Mask 形状和不透明度”)。
CA	Number	(可选; PDF1.4) 当前绘制的阿尔法常数, 指定了在透明成像模式下用作绘制操作的形状和不透明度恒量 (见 495 页“源形和不透明度”和 519 页“常
ca	Number	(可选, PDF1.4) 与 CA 相同, 但是不针对非绘制操作。
AIS	Boolean	(可选, PDF1.4) 阿尔法来源标志 (“阿尔法是形状”), 指定了是否当前 soft mask 和阿尔法常量被作为形状值 (true) 或者不透明度值 (false) 被说明。
TK	Boolean	(可选, PDF1.4) 文本敲入标志, 决定了在透明成像模型中文本对象中的重叠字形行为(见 5.2.7,“文本敲入”)。

例 4.1 展示了两个图形状态参数字典。在第一个, 自动画程调整开启, 并且字典包含了一个反转其值的传递函数, $f(x) = 1 - x$ 。在第二个, 套印被关闭, 字典包含了一个抛物线传递函数, $f(x) = (2x - 1)^2$, 伴随一个 21 个值的样本。传递函数的定义域, 从 [0.0 1.0] 映射到 [0 20], 并且样本值的范围, 从 [0 255] 映射到传递函数的范围, [0.0 1.0]。

例 4.1

```

10 0 obj % 页对象
<< /Type /Page
/Parent 5 0 R
/Resources 20 0 R
/Contents 40 0 R
>>
endobj
20 0 obj % 页面源字典

```

```
<< /ProcSet [/PDF /Text]  
/Font << /F1 25 0 R >>  
/ExtGState << /GS1 30 0 R  
/GS2 35 0 R  
>>  
>>  
endobj  
30 0 obj % 第一个图形状态参数字典  
<< /Type /ExtGState  
/SA true  
/TR 31 0 R  
>>  
endobj  
31 0 obj % 第一个传递函数  
<< /FunctionType 0  
/Domain [0.0 1.0]  
/Range [0.0 1.0]  
/Size 2  
/BitsPerSample 8  
/Length 7  
/Filter /ASCIIHexDecode  
>>  
stream  
01 00 >  
endstream  
endobj  
35 0 obj % 第二个图形状态参数字典  
<< /Type /ExtGState  
/OP false  
/TR 36 0 R  
>>
```

```
endobj
36 0 obj %第二个传递函数
<< /FunctionType 0
/Domain [0.0 1.0]
/Range [0.0 1.0]
/Size 21
/BitsPerSample 8
/Length 63
/Filter /ASCIIHexDecode
>>
stream
FF CE A3 7C 5B 3F 28 16 0A 02 00 02 0A 16 28 3F 5B 7C A3 CE FF >
endstream
endobj
```

4.4、路径构建和绘图

路径指形状，轨迹，和各种区域。它们可以用来绘制线，确定填充区域的形状，并指定其他图形的裁剪边界。图形状态包括用来定义当前页边界的剪辑路径。在每一页的开头，剪辑路径被初始化为整个页面。

一个路径由直线和曲线线段组成，它们可能会连接到彼此也可能会断开连接。只有当一对线段连续定义时才可以说是连接，即第二段在第一段结束的地方开始。因此，一个路径中定义的各个线段的顺序是很重要的。偶然交叉的不连续线段不成为连接。

一个路径可以由一个或多个不相连接的子路径组成，每个子路径包括一组连接的线段序列。路径的拓扑结构是不受限制的：它可能是凹或凸，可能包含多个不相交的子路径，并可以按任意方式在内部交叉。将一个子路径的终点连接到起点的操作称为 `h` 算子，这种子路径称为闭环。而明确没有闭的子路径称之为开。

正如在第 4.1 节“图形对象”中所讲，路径对象用一组算子序列表来构建，这些算子可以用作绘制或者边界。`PDF` 路径包括以下三种格式：

路径构建算子（第 4.4.1）定义了路径的几何形状。一个路径通过连续使用一个或多个这种操作算子来构造。

路径绘制算子（第 4.4.2 节）完成路径对象的绘制，通常在当前页上以某种方式完成对象的绘制。

路径裁剪算子（第 4.4.3 节）在一个路径绘制操作之前使用，从而使这个路径对象也能用作随后的图形裁剪。

4.4.1、路径构造算子

一个页面描述开始于一个空路径，然后通过调用一个或者多个路径构造算子，加入线段并且建立。路径构造算子可以以任何顺序来调用，但是第一个必须是 **m** 或者 **re**，来开始一个新的路径。路径的定义以绘制算子结束，例如 **S**, **f** 或者 **b**（见地 4.4.2 章“路径绘制算子”）；这个算子之前可以选择性地加入裁剪算子例如 **W** 或 **W***（见地 4.4.3 章“路径裁剪算子”）。注意路径构造算子不会在页面上留下任何痕迹，只有路径绘制算子才会。一个路径只有到路径绘制算子使用之后才算完成定义。

目前正在构造的路径被称为当前路径。在 PDF（与 PostScript 不同）中，当前路径不是图形状态的一部分，不会随图形状态参数一起保存和恢复。PDF 路径是严格的没有明确表象的内部对象。一旦一个路径绘制完毕，当前路径就不存在了，直到有一个新的以 **m** 或者 **re** 算子开始的路径，才会有当前路径。最近加入路径的线段的结束点称之为当前点。如果当前路径是空的，那么当前点也不存在。大多数算子从当前点向当前路径加入线段；如果当前点不存在，就会导致错误。表 4.9 中显示了路径构造算子。所有的操作数都是用户空间的数字表示。

表 4.9 路径构造算子

操作数	算子	描述
x y	m	将当前点移动到坐标(x, y)以开始一个新的子路径，忽略线段的连接。如果当前路径中之前的路径构造算子也是 m ，那么新的 m 算子将代替之前的，而之前的 m 算子操作的痕迹依然保留在路径中。
x y	l (小写) L	从当前点向(x, y)点追加一条直线线段。新的当前点修改为(x, y)
x1 y1 x2 c y2 x3 y3	c	向当前路径中加入一条三次贝塞尔曲线。曲线从当前点到点 (x3, y3)，使用点(x1, y1)和(x2, y2)作为贝塞尔控

制点（见下面的“三次贝塞尔曲线”）。新的当前点更改为 (x_3, y_3) 。

x2 y2 x3 v
y3 向当前路径中加入一条三次贝塞尔曲线。曲线从当前点到点 (x_3, y_3) ，使用当前点和 (x_2, y_2) 作为贝塞尔控制点（见下面的“三次贝塞尔曲线”）。新的当前点更改为 (x_3, y_3) 。

x1 y1 x3 y
y3 向当前路径中加入一条三次贝塞尔曲线。曲线从当前点到点 (x_3, y_3) ，使用 (x_1, y_1) 和 (x_3, y_3) 作为贝塞尔控制点（见下面的“三次贝塞尔曲线”）。新的当前点更改为 (x_3, y_3) 。

操作数	算子	描述
—	h	从当前点到子路径的开始点连接一条直线段，关闭当前子路径。如果当前子路径已经关闭，那么 h 算子不做任何事情。这个算子结束当前子路径。向当前路径追加入新的元素会开始一个新的子路径，即使新的路径从 h 操作的结束点开始。
x y width re height		向当前路径中加入一个矩形作为闭的子路径。矩形的左下角为 (x, y) ，宽为 width，高为 height。这个 x y width height re 的操作相当于： $x \ y \ m$ $(x + width) \ y \ l$ $(x + width) \ (y + height) \ l$ $x \ (y + height) \ l$ h

三次贝塞尔曲线

曲线路径元素使用三次贝塞尔曲线。这些曲线使用四个点来确定：两个端点（当前点 P0 和最后一点 P3）和两个控制点 P1 和 P2。根据这四个点的坐标，就在以下方程中当参数 t 从 0.0 到 1.0 时绘制出一条曲线：

$$R(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3$$

当 $t=0.0$ 时, $R(t)$ 的值与当前点 P_0 一致; 当 $t=1.0$ 时, $R(t)$ 的值与结束点 P_3 一致。 t 的中间值就产生了曲线上的中间点。曲线一般情况下不通过两个控制点 P_1 和 P_2 。

三次贝塞尔曲线有两个有用的属性:

- 它可以非常迅速地转化为较小的部分来实现快速渲染。
- 它包含在定义它的四个点的凸壳之间, 可以很容易通过拉伸有四个点组成的面而构成多边形。此属性可以快速测试是否曲线完全在可视区域的外面, 从而决定是否需要渲染。

书后列出了几本从更深层次上介绍三次贝塞尔曲线的书。

最常用的构造曲线路径元素的 PDF 算子是 c 算子, 它明确指定了三个坐标点 P_1 、 P_2 和 P_3 , 如下图 4.8 中所示。(开始点 P_0 默认指的是当前点)

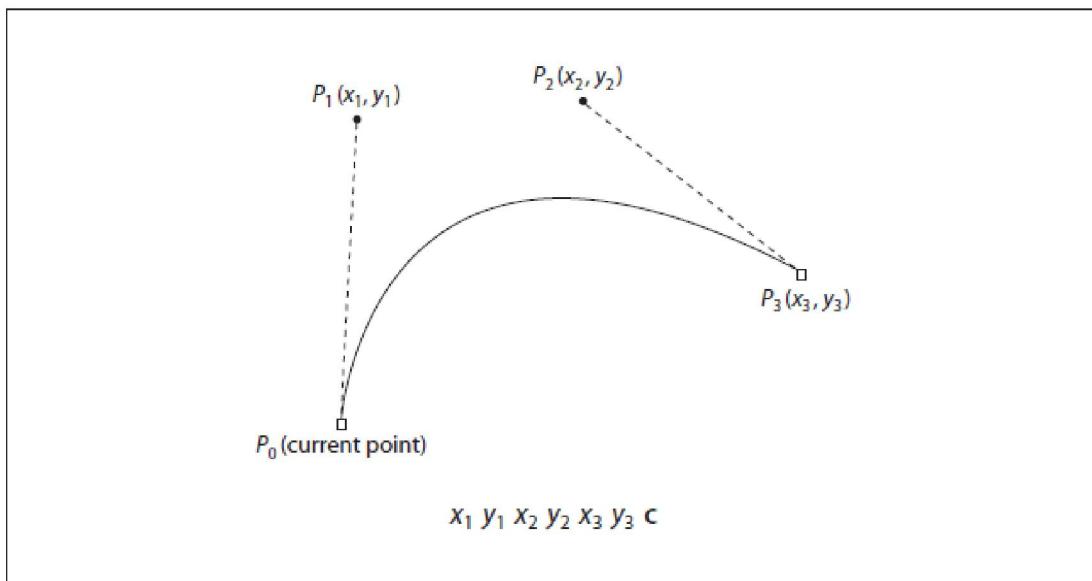


图 4.8 c 算子生成的三次贝塞尔曲线

另外两个算子, v 和 y , 每个都指定了一个控制点 (如图 4.9 所示)。这两种情况下, 曲线的一个控制点和最终点作为参数提供, 而另外一个控制点则是隐含的:

对 v 算子, 第一个控制点与曲线的起始点相同。

对 y 算子, 第二个控制点与曲线的终点相同。

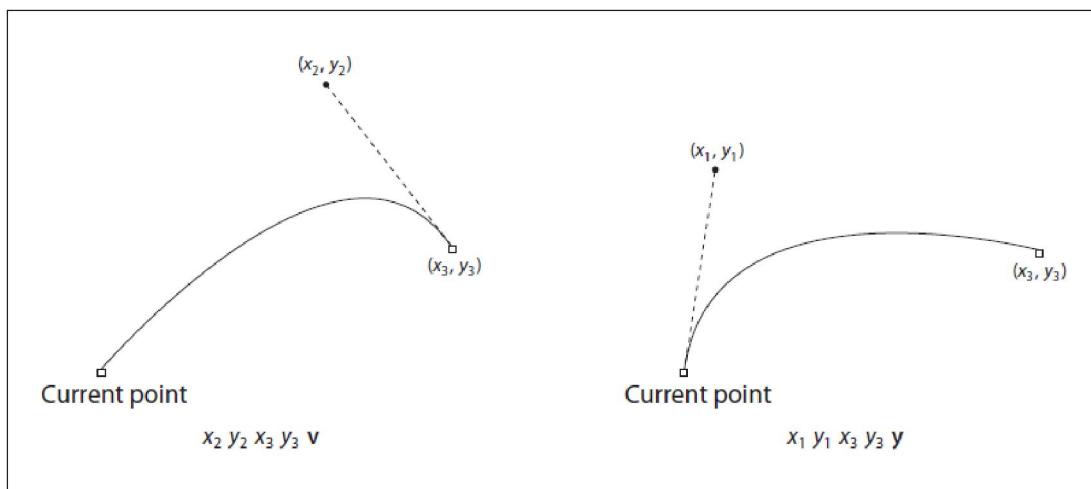


图 4.9 v 算子和 y 算子生成的三次贝塞尔曲线

4.4.2、路径绘制算子

路径绘制算子结束一个路径对象，按照算子指定的方式绘制在当前页上。最重要的路径绘制算子是 **S**（涂抹）和 **f**（填充）。将涂抹和填充以单独操作结合起来或者加上各种准则来确定填充的区域，就出现了各种不同的算子。表 4.10 中列出了这些路径绘制算子。

表 4.10 路径绘制算子

操作数	算子	描述
—	S	涂抹
—	s	涂抹并关闭路径，这个算子和 h s 的效果
—	f	填充路径，使用非零缠绕数规则来决定填充的区域（见 202 页的“非零缠绕数规则”）。任何开着的子路径在填充之前都必须关闭。
—	F	相当于 f ，为兼容性存在。尽管任何 PDF 读取程序都必须能够识别它，而 PDF 制作程序应该使用 f 代替 F 。
—	f*	填充路径，使用奇偶原则决定填充的区域（见 203 页的“奇偶原则”）。
—	B	填充然后涂抹路径，使用非零缠绕数规则来决定填充区域。这个算子和使用 f 算子和 S 算子绘制的两个路径对象结合在一起的效果一样。注意，然而填充和涂抹操作的份额造成了不同的图形状态参数值，例如当前的颜色。详见 538 页的“特殊路径绘制”。

-
- B* 填充然后涂抹路径，使用奇偶原则决定填充的区域。这个算子和 B 算子得到相同的结果，只是使用 f* 算子填充的而已。详见 538 页的“特殊路径绘制”。
 - b 关闭，填充并且涂抹路径，使用非零缠绕数规则来决定填充区域。这个算子与算子序列 h B 具有相同的效果。详见 538 页的“特殊路径绘制”。
 - b* 关闭，填充并且涂抹路径，使用奇偶原则来决定填充区域。这个算子与算子序列 h B* 具有相同的效果。详见 538 页的“特殊路径绘制”。
 - n 不填充涂抹直接结束路径对象。这个算子是一个路径绘制空算子，在改变当前裁剪路径的边界效应之前使用。
(见第 4.4.3 章的裁剪路径算子)

涂抹

S 算子在当前路径上绘制出一条线。涂抹沿着路径上的直线或者曲线进行，并与元素的边界平行。每个路径的子路径都相同地处理。S 算子处理的效果跟当前图形状态的不同参数的当前设置有关。(见第 4.3 章“图形状态”可以看到这些参数更多的信息)：

- 涂抹线的宽度由当前的线宽参数决定(将 185 页的“线宽”)。
- 线的颜色或风格由当前的颜色或者涂抹操作的颜色决定。
- 线可以被绘制成实线或者虚线，由当前的线类型决定(见 187 页的“线型”)。
- 如果子路径是开的，那么不连接的结束点会根据当前的线性风格来处理，可能是箭头、圆点或者方格(见 186 页的“线头风格”)。
- 在两个连续的元素是连接的地方，连接点会根据当前的线型处理，可能是斜接，圆形，或斜角的(见 188 页的“线连接类型”)。斜接的连接也受到斜切限制的限制(见 187 页的)。

注意：不连接的元素相交的点没有特殊的处理。特别地，使用 I 算子关闭一个子路径，而不使用 h，可能导致模糊的边角，因为使用了线头而不是线的连接。

- 涂抹适应参数(PDF1.2)指定了坐标和线宽自动生成均匀厚度的涂抹，尽管有光栅效应的存在(见第 6.5.4 章，“自适应涂抹”)。

如果子路径退化（包括单点闭路径或者同一坐标上的两个多个点），`S` 算子仅在圆线头时才绘制，产生单点上的填充圆圈。如果使用点或者方形的线头，`S` 算子没有输出，因为线头的方向不确定。（此规则只适用于以路径长度为零的子路径被涂抹，而不是零长度的虚线。后一种情况的线头会被涂抹，因为线的方向是确定的。）单点的开子路径（使用`m`算子结尾）没有输出结果。

填充

`f` 算子使用当前的非涂抹颜色来绘制当前路径中的整个区域。如果路径由几个不连接的子路径，`f` 会在所有的子路径中绘制。任何开的子路径在填充之前都会被关闭。

如果子路径退化（包括单点闭路径或者同一坐标上的多个点），`f` 将按照设备像素来绘制点；结果是跟设备相关的，并且一般没有用。一个单点的开子路径（有`m`算子结尾）没有处理结果。

对一个简单的路径，可以很直观地看到内部有什么区域。然而，对一个很复杂的路径，例如一个路径和自己交叉，或者有一个子路径把另外一个围起来，这时路径中的点不是都能明显看到的。路径体系使用这两个原则中的一个来决定哪些点在路径中：非零绕数原则和奇偶数原则，在以下将详细介绍它们。

非零绕数原则是一个比奇偶数原则更有用的规则，它是`f` 算子使用的典型原则。相似的，`W` 参数使用这个原则来决定当前裁剪路径的内部。奇偶数原则偶然对一些特殊的效果或者其他图形系统适应性有效；`f*` 和 `W*` 算子使用这种原则。

非零绕数原则

非零绕数原则使用从一个点向各个方向无限画光来确定是否一个点在路径中，然后检查路径中穿过光线的部分。从 0 开始，这个原则将从左到右每次加一个穿过光线的元素，然后从右到左每次减一个元素。在数了所有的交错点后，如果结果是 0，那么这个点是在路径外的，否则它是在路径内的。

注：被描述的方法没有定义如果一个线段与所选择的射线平行或者相切该怎么做。由于光的方向是任意的，这个规则仅仅选择了不遭遇这些问题的交点。

对于简单的凸路径，非零绕数规则界定了将被直觉期望的内部和外部。更有趣的是那些像在图 4.10 中描述的包括复杂或者自相交的路径。对于一条包含五角星的路径，由五条连接的直线段互相相交构成，这个规则将内部看做被星围起来的整体区域，包括中心的五边形。对于由两个同心圆组成的路径，被两个圆围起来的区域被认为是内部，只要都在同一个方向绘制。如果以相反方向绘制，根据规定只有它们之间的环状是内部；环形的洞是外部。

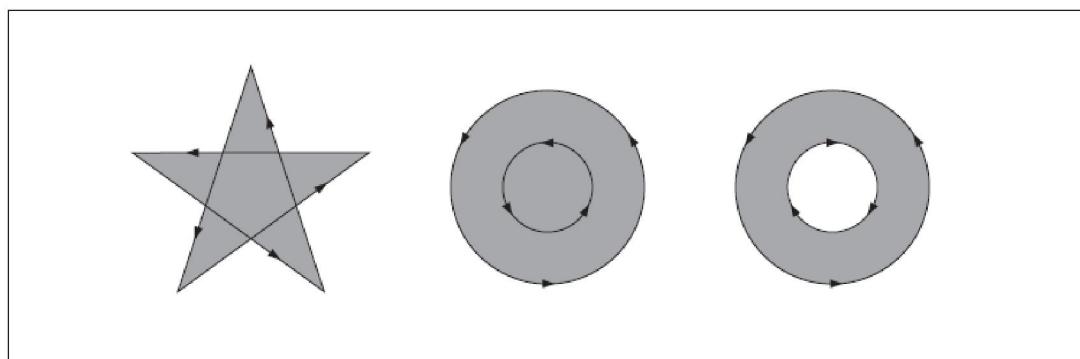


图 4.10 非零环绕数字规则

偶数-奇数原则

对于非零绕数规则的选择是旧规则。这个规则通过从这一点在任何方向上画一条射线且仅仅计算与这条射线相交的线段数目决定是否这个点在一条路径的内部，不管是什么方向。如果数目是奇数，这个点在内部；如果是偶数，则在外部。这就产生了简单形状的路径的非零绕数规则，但是对更多的复杂形状产生了不同的结果。

图 4.11 展示了对复杂路径应用奇偶规则的结果。对五角星，此规则认为三角点在此路径之内，而不是中心的五边形。对于两个同心圆，只有两圆之间的环形被认为在内部，无论圆以什么方向绘制。

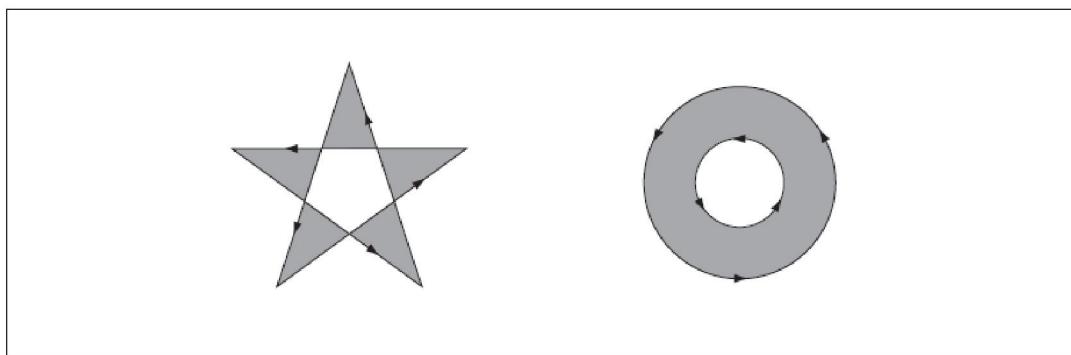


图 4.11 奇偶规则

4.4.3、剪辑路径操作

图形状态包含了通过绘制操作限制页面影响的区域的一个当前剪辑路径。这条路径的封闭子路径规定了可以被绘制的区域。区域内部的标记被应用到页面；区域外部的没有。(202 页的“填充”正是介绍了什么被认为在一条路径的内部)。

注：在透明成像模型下（PDF1.4），当前剪辑路径限制了对象的形状（见 7.1，“透明度概述”）。有效形状是对象的固有形状与剪辑路径的交集；在这个焦点之外的源形值为 0.0。同样，一个透明度组的形状（定义为其成分对象的形状单元）被当任何一个对象被绘制时的实际剪辑路径和同时绘制到其背景上的组结果的实际剪辑路径二者影响。

初始剪辑路径包括整个页面。一个剪辑路径操作（W 或 W*，表 4.11 中有给出）可能会出现在最后的路径构造操作之后和结束一个路径对象之前的路径绘制操作之前。尽管路径剪辑操作出现在路径绘制之前，它显示的那个点的剪辑路径没有改变。然而它修改了成功绘制操作的结果。在路径被绘制之后，图形状态里的剪辑路径被设置到当前剪辑路径和最新构成路径的交点。

表 4.11 剪辑路径操作		
操作数	操作	说明
—	W	通过使当前路径与其相交来修改当前剪辑路径，使用非零绕数规则来确定哪些区域在剪辑路径内。
—	W*	通过使当前路径与其相交来修改该当前剪辑路径，使用奇偶规则来确定哪些区域在剪辑路径内。

注：除路径对象之外，文本对象也可被用作剪辑；见 5.2.5，“文本显示模式”。

n 操作（见表 4.10）是一个无运算路径绘制操作；它导致没有标记被放置在页面上，但是可以和一个剪辑路径操作使用来建立一个新的剪辑路径。即，在一条路径 jianli8 之后，序列 Wn 通过当前剪辑路径与路径交叉来建立一个新的剪辑路径。

没有措施去扩大目前的剪辑路径或者不牵涉到目前的去设置一条新的剪辑路径。然而，由于剪辑路径是图形状态的一部分，它的结果可以通过在一对操作 q 和 Q 之间内附剪辑路径的限制和哪些对象的绘制限制指定的图形对象（见 4.3.1，“图形状态堆栈”）。Q 操作的执行引起剪辑路径恢复了在剪辑路径被修改前 q 操作保存的值。

4.5、色彩空间

PDF 在页面的彩色图像生成方面提供了强大的功能。色彩功能可以分成两个部分：

色彩规范：一个 PDF 文件可以独立规定设备上抽象颜色。颜色可以用属于任意的色彩系统或色彩空间的表示法来描述。某些色彩空间是属于设备相关的色彩表示 (grayscale, RGB, CMYK)，另一些则基于人类的视觉 (CIE-based)。确定的特性也像色彩空间一样的模型化了：矢量，色彩映射，分隔，以及高保真的色彩。

色彩转换：在一个由多种色彩转换方式组合在一起的多步骤流程中，通过光栅设备的输出完成图像的还原：`gamma` 校准，半色调技术，以及扫描转换。上述流程中会用到某些 PDF 包含的信息。尽管如此，与色彩规范不同的是，色彩转换的功能是依赖于设备的并且一般并不在页面的描述中出现。

在第 208 与 209 页的图 4.12 和 4.13 显示了色彩规范与色彩转换之间的划分。这部分描述了色彩规范的特性，涵盖了大多数 PDF 文档需要用到的颜色规范。色彩转换功能的控制方式将会在第六章中描述；一个 PDF 文档应该使用这些功能配置或者校准一个设备上的输出或者得到特定设备的效果。

4.5.1、色彩值

如同在 4.4.2 中描述的，路径着色操作，页面上的 `f` 及 `s` 标记具有的颜色会被当前图像的状态的色彩参数决定。一个色彩值包括了一个或者多个部分，这些部分通常有数字组成。例如，一个灰度阶数可以被一个值在 0.0 至 1.0 之间的参数代表。全色彩的值可以被有很多规定方式；一个普遍采用的方式是用红，绿，蓝三个值来定义。

颜色值可以通过图像状态的另一个参数——当前色彩空间来确定。一个 PDF 流首先会通过调用 `CS` 操作符来选择色彩空间。`G, g, RG, rg, K` 以及 `k` 也是很方便的，在一个步骤中既可以选择色彩空间也可以选择色彩值的操作符。在 257 页的表 4.24 标出了所有的色彩操作符。

抽样图像（见 4.8 节，“图像”）规定的色彩值是有个体图像的抽样色彩空间决定的。尽管这些值是独立于当前色彩空间和图像状态的色彩参数，所有后续的色彩操作流程会采用和使用 `sc` 操作符相同的操作方式对待它们。

4.5.2、色彩空间族

色彩空间可以被抽象成色彩空间族。同族的色彩空间共享同样的主要特性；在色彩空间确立以后，它们由其提供的参数值来区分。色彩族被分为三个大类：设备色彩空间直接规定了输出设备产生的灰度阶数。它们提供了各种色彩规范方法，包括灰度，RGB, CMYK，分别与色彩空间 DeviceGray, DeviceRGB, 及 DeviceCMYK 相关联。由于每个色彩族包括了一个没有参数的色彩空间，它们通常很少涉及 DeviceGray, DeviceRGB, 及 DeviceCMYK。

CIE-based 色彩空间是基于一个由 de l' Eclairage 国际会议标准色彩规范的。这些规范的色彩通过一种独立于任何特定输出设备方式来定义。这类色彩空间族包括 CalGray, CalRGB, Lab 以及 ICCBased。每个具有这些色彩族的个体色彩空间由具有定义该色彩空间参数值组成的字典来规范。

特殊色彩空间为一个基础的色彩空间加入了新特性或新属性，它们包括向量，色彩映射，分割，高质量图像的方面的功能。相关的色彩空间的族被定义为，Pattern, Indexed, Separation, 以及 DeviceN 族。具有这些族的单独色彩空间由附加参数来规范。

表 4.12 总结了 PDF 支持的色彩族。

TABLE 4.12 Color space families

DEVICE	CIE-BASED	SPECIAL
DeviceGray (PDF 1.1)	CalGray (PDF 1.1)	Indexed (PDF 1.1)
DeviceRGB (PDF 1.1)	CalRGB (PDF 1.1)	Pattern (PDF 1.2)
DeviceCMYK (PDF 1.1)	Lab (PDF 1.1)	Separation (PDF 1.2)
	ICCBased (PDF 1.3)	DeviceN (PDF 1.3)

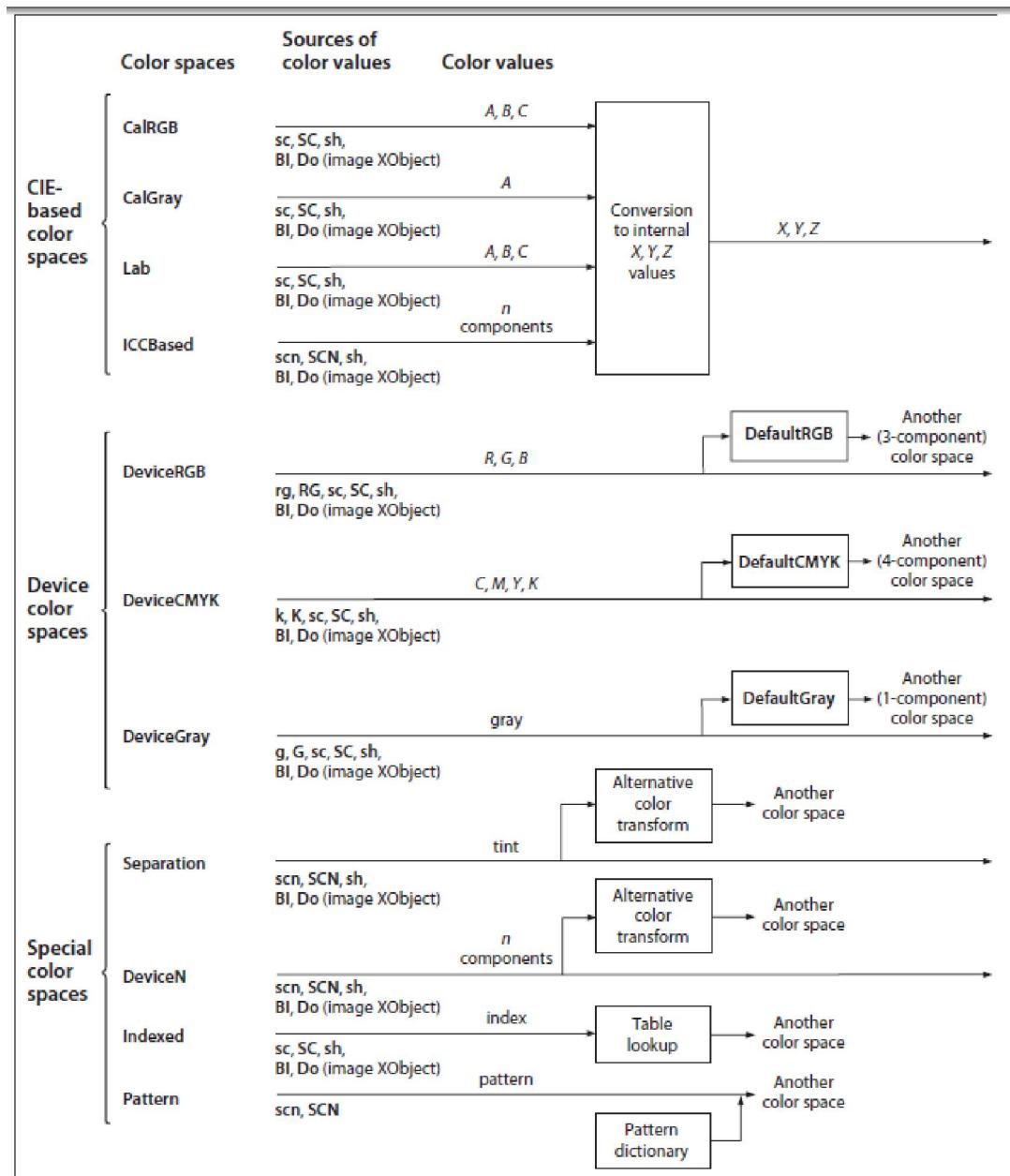
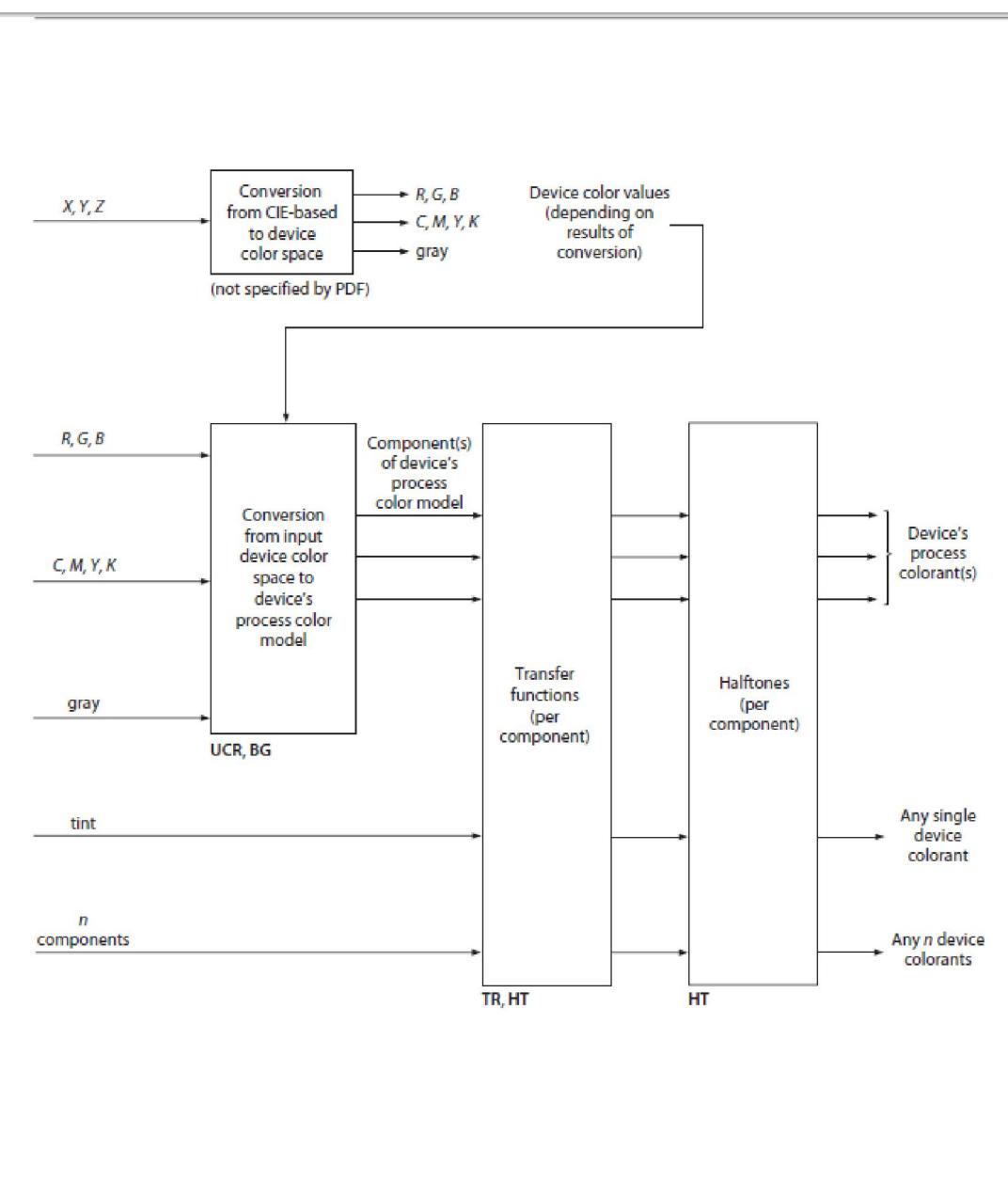


图 4.12 颜色规格



4.13 颜色方式

色彩空间被定义为一个数组实体，该数组的首个元素是该色彩族的名字。其余的数组元素，如果存在的话，是该色彩空间的其他特性参数；对于特定的色彩族，它们的数值与类型也会随之变化。对于不需要参数的色彩族，色彩空间可以被简单的定义成色彩族的族名，而无需使用数组定义。

色彩空间的规范主要有两种方式：

在一个容量流中，**cs** 及 **CS** 操作符建立了当前图像的色彩空间参数。操作数通常为一个命名目标，要么规定一个不需要参数的色彩空间(**DeviceGray**, **DeviceRGB**, **DeviceCMYK** 或者某些 **Pattern** 的情况)，要么作为当前源字典的“色彩空间子字典”的关键字。在后一种情况中，字典接口的值返回值为一个色彩空间数组或者名字。一个色彩空间数组永远不许包含在容量流中。

在容量流之外，确定对象——例如 **Xobject** 图像，用一个明确的参数来定义色彩空间，通常会与 **ColorSpace** 关键词相关联。在这种情况下，色彩空间数组通常被直接定义成一个 PDF 对象，而不是由色彩空间源的子字典的接口定义。这一协定也在色彩空间被其他的色彩空间定义时使用。

下面的操作符可以设定当前色彩空间以及图像的当前色彩参数：

CS 用来设定冲击色彩空间；**cs** 设定非冲击色彩空间。

SC 以及 **SCN** 设定冲击色彩；**sc** 以及 **scn** 设定非冲击色彩。根据色彩空间的不同，这些操作需要 1 个或多个操作数。

G, **RG** 以及 **K** 隐式的设定冲击色彩空间以及通过被操作数指定的色彩值；**g**, **rg** 以及 **k** 对非冲击色彩实现相同的操作

4.5.3、设备色彩空间

设备色彩空间使用一个页面描述来规定直接与当前设备输出相关的色彩值。这些空间的色彩值直接与设备的着色值相映射，例如墨水的质量或者显示的强度等。这使得一个 PDF 文档可以精确的控制特定输出设备上的图像颜色，不过该显示结果无法在不同设备上保持一致。

输出设备制造颜色的方式无非分为将光信号组合或者减低明亮源的光效果两种方式。电脑的显示或者电影胶片通常为增色方式；打印墨水一般为减色方式。这两种方式发展了两种互补的图像色彩规范，分别称为加性色彩与减性色彩。对于这两种色彩规范类型，最广泛采用是我们熟知的 **RGB** 以及 **CMYK**，各自为它们的

基础色的简称。它们代表下面的设备色彩空间：

`DeviceGray` 控制了一束非彩色光的强度，介于黑与白之间。

`DeviceRGB` 控制了红，绿，蓝光的强度，三种加性的基础颜色。

`DeviceCMYK` 控制了青色，洋红色，黄色以及黑色墨水——4 种减性颜色的强度。

虽然颜色空间的概念是 PDF 1.1 的特性，但是在设备色彩空间的规定操作符——`G`, `g`, `RG`, `rg`, `K` 以及 `k` 在每个版本的 PDF 中都是可用的。从 PDF1.2 开始，在设备色彩种的规定色彩值可以在其他的色彩空间做系统性的重映射；参见“默认色彩空间”，227 页。

注：在透明成像模型（PDF1.4）中，设备颜色空间的使用是服从组颜色空间是 CIE-based 的透明组的特殊处理（见 7.3，“透明度组”和 7.5.5，“透明度组外在对象”）。特别是，该设备颜色空间的操作仅当设备颜色空间通过默认颜色空间机制被重新映射到 CIE-based 空间时应该被使用。否则，结果是依赖执行且不可预料。

设备灰度色彩空间

黑，白，以及中性的灰色，是全色彩的一种特例。一个灰度阶的值，被一个介于 0.0 至 1.0 之间的值所设定，其中 0.0 代表黑色，而 1.0 则代表白色，中间的值则代表不同程度的灰。例 4.2 显示了设备灰度色彩空间的两种选择方式。

例子 4.2

```
/DeviceGray CS          % 设定设备灰度色彩空间
gray SC                % 设定灰度的阶
gray G                 % 用一个操作符实现上述两种操作
CS 以及 SC 操作符分别选择了当前冲击色彩空间以及冲击色彩；G 则同时可以设定两者。设定任意的当前色彩空间将会初始化相应的颜色值到 0.0。
```

DeviceRGB 色彩空间

在 DeviceRGB 色彩空间中的颜色，是被加性的 RGB 色彩模型所规定的，模型中的颜色值定义了三种基础加性色彩的强度。每个基础色彩的强度由一个 0.0-1.0 之间的值定义，0.0 代表该色的部分不存在，而 1.0 代表了该色的最大值。如果三种颜色的强度值相等，那么所得结果将是理论上的介于黑与白之间的纯灰色。如果三种强度不等，那么所得结果将比纯灰显得更有颜色性。

例子 4.3 显示了两种 DeviceRGB 色彩空间的操作方式

例子 4.3

```
/DeviceRGB CS          % 设定 DeviceRGB 色彩空间
red green blue SC      % 设定颜色
red green blue RG      % 用一个操作符实现以上两种操作
```

CS 与 SC 操作符分别选择了当前的色彩空间与颜色; RG 则可以同时实现以上两种操作。(cs, sc, 以及 rg 对非冲击色彩空间实现相同操作)。

DeviceCMYK 色彩空间

DeviceCMYK 色彩空间准许色彩被四种减性的颜色模型所定义，通常是用于打印机或者基于纸张输出的设备上。理论上，三种减性的标准的着色（青色，杨红，黄色）将分别吸收三种加性的颜色（红，绿，蓝）。黑，第四种标准着色，用相同的强度吸收所有的加性颜色。四种 DeviceCMYK 颜色值代表了这些颜色的强度。每种值的范围是 0.0-1.0, 0 为最小，1 为最大。

例子 4.4 显示了对于 DeviceCMYK 色彩空间规范颜色的方法

例子 4.4

```
/DeviceRGB CS           % 设定 DeviceRGB 色彩空间  
red green blue SC       % 设定色彩  
red green blue RG       % 用一个操作符设定以上两者
```

CS 与 SC 操作符分别选择了当前的色彩空间与颜色; RG 则可以同时实现以上两种操作。(cs, sc, 以及 rg 对非冲击色彩空间实现相同操作)。

4.5.4、CIE-Based 色彩空间

PDF 使用的色彩是基于在图像艺术，电视，印刷工业中的国际标准协议来指定的。CIE-based 色彩空间用一个页面描述来规定色彩的值，并且与人类的视觉系统相联系。目标是，对于相同的色彩规范，在不同的设备上制造出相同的色彩效果。模板 2 显示了各种不同设备上的非规范颜色再现。PDF1.1 支持三种 CIE-based 色彩族，CalGray，CalRGB，以及 Lab；PDF 增加到了第四种，ICCBased。

CIE 细节和理论基础已经超过了本书的讨论范围；更多的信息请参见术后的参考书目。CIE-based 色彩空间在语义上被定义为由 CIE 1931 XYZ 的三色值 X, Y, Z 组成。该色彩空间的公式可以由一个简单的区域理论所描述，由非线性的三色阶段组成。这个公式支持高精度的数字化颜色，是一种重要的抽样图像方式。

CIE-based ABC 色彩空间值具有三个组成部分，称为 A, B, C. 第一步，将这三个值赋值后，转换到一个特定的值域，然后使用解码函数，然后将结果乘以一个 3×3 的矩阵，制造出三个中间值 L, M, N。第二步将这三个中间值用相似的方式，转换成最终的 CIE 1931 XYZ 色彩空间中的 X, Y, Z 分量（见图 4.14）。

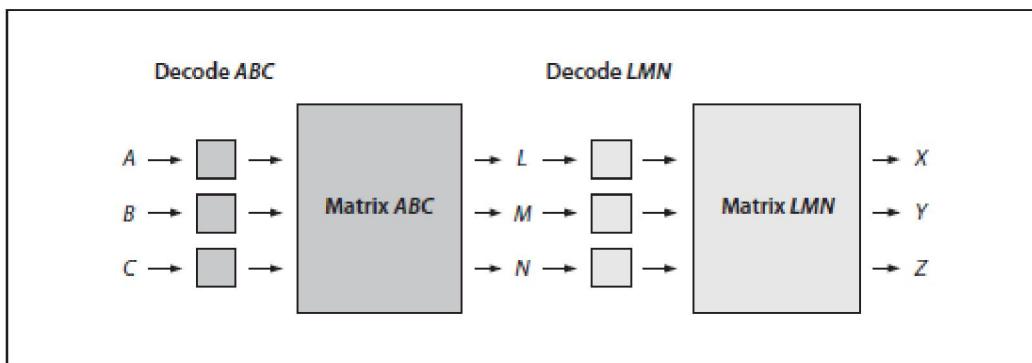


图 4.14 在 CIE-based ABC 颜色空间的元件变化

CIE-based 族的色彩空间可以被一个数组定义

[name dictionary]

name 代表族名而 dictionary 作为一个字典则包括了该空间的更多特性参数。该字典的入口根据色彩空间有着其特定规范；某些入口是必须的，而另一些则是可选的。关于特定色彩空间族的细节如下：

对于任何的 CIE-based 色彩空间做初始化，将会把相应的分量值设置为 0.0。

CalGray 色彩空间

一个 CalGray 色彩空间 (PDF1.1) 是单一 CIE-based 色彩空间——也称为 CIE-based A 色彩空间的特殊情况。这种类型的空间是基于 ABC 空间的一维模拟空间。单一 CIE-based 色彩空间的色彩值具有单一的分量，命名为 A。图 4.15 显示了分量 A 如何转换到 CIE 1921 XYZ 空间。

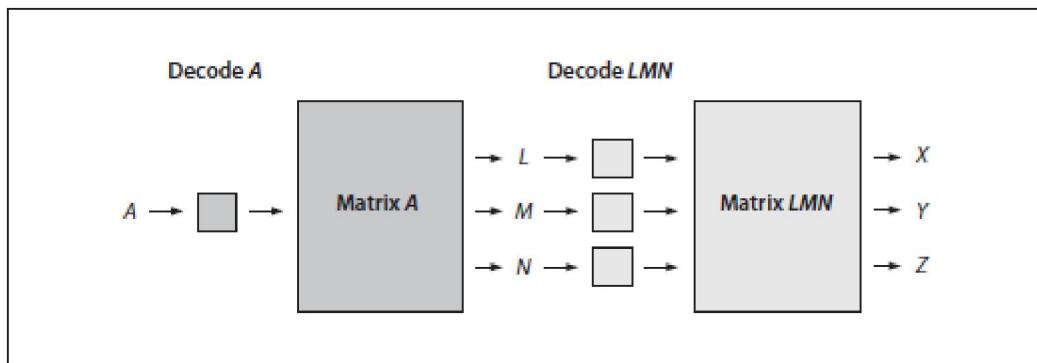


图 4.15 在 CIE-based A 颜色空间的元件转化

一个 CalGray 色彩空间是只具有一个转换步骤的 CIE-based 的单一色彩空间。在这种类型的空间中，A 代表标准灰色空间中的灰色分量。该分量的取值范围必须为 0.0 至 1.0。解码函数（在图 4.15 中用 Decode A 表示）为一个 gamma 函数，该函数的系数被色彩空间字典的 gamma 入口所指定（见表 4.13）。图中，Matrix A 代表的转移矩阵被字典的 WhitePoint 入口接收。因为只有一个转换步骤，“Decode LMN” 以及 “Matrix LMN” 隐式的执行在转换中。

表 4.13 Calgray 色彩空间的字典入口

关键字	类型	值
WhitePoint	数组	(必须) 该数组是由 CIE 1931 XYZ 空间指定的三维数组，用来描述 whitepoint；更多的信息参见下面的 CalRGB 色彩空间。Xw 以及 Zw 必须为正数，Yw 必须等于 1.0。
BlackPoint	数组	(可选) [XB, YB, ZB] 是由 CIE 1931 XYZ 空间指定的三维数组，用来描述 blackpoint；更多的信息参见下面的 CalRGB 色彩空间。三个分量必须为非负值，默认值为 [0.0, 0.0, 0.0]。
Gamma	数值	(可选) 数值 G 定义了灰度 A 分量的 gamma 值。G 必须是正数，并且不小于 1。默认值为 1

Gamma 以及 WhitePoint 的入口可以用来定义转换

$$\begin{aligned} X &= L = X_W \times A^G \\ Y &= M = Y_W \times A^G \\ Z &= N = Z_W \times A^G \end{aligned}$$

换句话说，A 分量首先被 `gamma` 函数解码，然后结果被具有 L, M, N 分量的 `whitepoint` 向量相乘。因为只有一个转换步骤，L, M, N 分量也就是最终的 X, Y, Z 分量。

下面的例子显示了 CalGray 色彩空间的优点和特性。例 4.5 建立了一个包含 Y 维度的 CCIR XA/11 - recommended D65 white point 色彩空间。

例 4.5

```
[ /CalGray
<< /WhitePoint [0.9505 1.0000 1.0890] >>
]
```

例 4.6 建立了一个标准的 CCIR XA/11 - recommended D65 white point 灰度色彩空间。

```
[ /CalGray
<< /WhitePoint [0.9505 1.0000 1.0890]
/Gamma 2.222
>>
]
```

CalRGB 色彩空间

一个 CalRGB 色彩空间是只具有一个转换步骤的 CIE-based ABC 色彩空间。在这种类型的色彩空间中，A, B, C 代表标准的红，绿，蓝分量。这三个分量必须取值范围在 0.0 至 1.0 之间；分量的值如果超过了上述范围，则要被调整到最接近的取值。解码函数（215 页图 4.14 中的解码 ABC）为 `gamma` 函数组，该函数组的系数被色彩空间字典的 `gamma` 入口所指定（表 4.14）。在图 4.14 中由“矩形 ABC”指定的转型矩阵被字典得矩阵项指定。由于没有第二次转型阶段，“解码 LMN”和“矩阵 LMN”被含蓄地作为特性转换。

表 4.16 在一个 CalRGB 色彩空间字典的项

键	类型	值
White point	array	(必填) 对三元素数组 [XW YW ZW] 指定的三色值，在 CIE 1931 XYZ 的空间，扩散白点的；见下述。数字 XW 和 ZW 必须明确，且 YW 必须等于 1.0。
Black point	array	(可选) 对三元素数组 [XB YB ZB] 指定的三色值，在 CIE 1931 XYZ 空间，扩散黑点的，见下述。所有的这些数字都必须是不确定的。默认值 [0.0 0.0 0.0]。
Gamma	array	(可选) 一个 3 元素数组 [GR GG GB] 指定了一个颜色空间红，绿，蓝 (A, B, C) 元件的对比度。默认值：[1.0 1.0 1.0]。
Matrix	array	(可选) 9 元素数组 [XA YA ZA XB YB ZB XC YC ZC] 使用关于最终的 XYZ 的表示定义了关于颜色空间的被译码的 A, B 和 C 制定了线性说明。默认值：特性矩阵 [1 0 0 0 1 0 0 0 1]。

颜色空间字典里的白点和黑点项控制了 6.1 部分描述的 CIE-based 范围内的映射函数的所有影响，“CIE-Based 颜色到设备颜色”。同城情况下，由白点和黑点指定的颜色映射到输出设备能够以保持颜色显示和色差的显示的最轻和最深非彩色的颜色。

白点是假定代表广泛的非彩色部分，而不是一个镜面亮点。镜面亮点，无色或者否则经常被比广泛亮点复制的更轻。黑点是假定代表广泛无色阴影；它的值一般被动态的输入设备范围限制。在摄影系统产生的图像中，白点和黑点的值随曝光，系统响应，艺术意图而改变；因此，它们的值是依赖图片的。

在 GaIRGB 颜色空间字典中被 Gamma 和 Matrix 项定义的转换为

$$\begin{aligned} X &= L = X_A \times A^{G_R} + X_B \times B^{G_G} + X_C \times C^{G_B} \\ Y &= M = Y_A \times A^{G_R} + Y_B \times B^{G_G} + Y_C \times C^{G_B} \\ Z &= N = Z_A \times A^{G_R} + Z_B \times B^{G_G} + Z_C \times C^{G_B} \end{aligned}$$

换句话说，A, B, C 元素都先各自被伽马函数解码。其结果被作为一个三元素的食量被矩阵（3X3 矩阵）乘来获得中间值 L, M, N 元素。由于没有第二阶段，X, Y 和 Z 元素也在最后表达中。

例 4.7 展示了一个使用 1.8 伽马和 Sony Trinitron ® 荧光色度针对 CCIR XA/11-recommended D65 白点的 CaIRGB 颜色空间的例子。

例 4.7

```
[ /CaIRGB
<< /WhitePoint [0.9505 1.0000 1.0890]
/Gamma [1.8000 1.8000 1.8000]
/Matrix [ 0.4497 0.2446 0.0252
0.3163 0.6720 0.1412
0.1845 0.0833 0.9227
]
>>
]
```

在一些情形下，一个 CaIRGB 色彩空间的参数可能被各组指定在 CIE 1931 红色，绿色和蓝色荧光粉染色坐标(xR, yR), (xG, yG), (xB, yB)，且广泛白点染色性(xW, yW)对应一些线性 RGB 值 (R, G, B)，在其通常 R=G=B=1.0。注意，使用标准智能库符号小写字母来指定色肚子表和大写字母来指定三色值。见于此信息，矩阵和白点可以被发现如下：

$$z = y_W \times ((x_G - x_B) \times y_R - (x_R - x_B) \times y_G + (x_R - x_G) \times y_B)$$

$$Y_A = \frac{y_R}{R} \times \frac{(x_G - x_B) \times y_W - (x_W - x_B) \times y_G + (x_W - x_G) \times y_B}{z}$$

$$X_A = Y_A \times \frac{x_R}{y_R} \quad Z_A = Y_A \times \left(\frac{1 - x_R}{y_R} - 1 \right)$$

$$Y_B = -\frac{y_G}{G} \times \frac{(x_R - x_B) \times y_W - (x_W - x_B) \times y_R + (x_W - x_R) \times y_B}{z}$$

$$X_B = Y_B \times \frac{x_G}{y_G} \quad Z_B = Y_B \times \left(\frac{1 - x_G}{y_G} - 1 \right)$$

$$Y_C = \frac{y_B}{B} \times \frac{(x_R - x_G) \times y_W - (x_W - x_G) \times y_R + (x_W - x_R) \times y_G}{z}$$

$$X_C = Y_C \times \frac{x_B}{y_B} \quad Z_C = Y_C \times \left(\frac{1 - x_B}{y_B} - 1 \right)$$

$$X_W = X_A \times R + X_B \times G + X_C \times B$$

$$Y_W = Y_A \times R + Y_B \times G + Y_C \times B$$

$$Z_W = Z_A \times R + Z_B \times G + Z_C \times B$$

实验色彩空间

一个实验色彩空间是一个有两个转换阶段的智能色彩空间（见图 4.14, 215 页）。在这种空间类型下，A, B, C 代表了一个 1976 智能库 L*a*b* 空间的元件 L*, a* 和 b*。第一个 (L*) 元件的范围总是从 0 到 100；第二个和第三个元件 (a* 和 b*) 的范围被颜色空间字典中的范围项定义（见表 4.15）。

板块 3 显示了一个典型的实验色彩空间坐标；板块 4 L*a*b*, RGB 和 CMYK 空间的范围（一系列可能被描绘的颜色）。

表 4.16 在一个实验色彩空间字典的项

键	类型	值
White point	array	(必填) 对三元素数组 [XW YW ZW] 指定的三色值，在 CIE 1931 XYZ 的空间，扩散白点的；见即将讨论的 217 页“CalRGB 颜色空间”。数字 XW 和 ZW 必须明确，且 YW 必须等于 1.0。
Black point	array	(可选) 对三元素数组 [XB YB ZB] 指定的三色值，在 CIE 1931 XYZ 空间，扩散黑点的，见即将讨论的 217 页的“CalRGB 颜色空间”。所有的这些数字都必须是不确定的。默认值 [0.0 0.0 0.0]。
Range	array	(可选) 一个 4 元素数组 [amin amax bmin bmax] 指定了一个颜色空间的 a* 和 b* 成分 (B 和 C) $a_{\min} \leq a^* \leq a_{\max}$ and $b_{\min} \leq b^* \leq b_{\max}$ 的有效值范围——即在指定范围外的元件值在没有错误指示的情况下调整到最近的有效值。默认值：[-100 100 -100 100]。

实验色彩空间并不对从 L*a*b* 空间到 XYZ 空间的转换阶段（在 215 页的图 4.14 中由“Decode ABC,” “Matrix ABC,” “Decode LMN,” 和 “Matrix LMN” 指定）指定明确的解码函数或者矩阵系数。相反，这些参数有恒定的隐含值。转化的第一步被方程式定义：

$$L = \frac{L^* + 16}{116} + \frac{a^*}{500}$$

$$M = \frac{L^* + 16}{116}$$

$$N = \frac{L^* + 16}{116} - \frac{b^*}{200}$$

第二个转换阶段被给出：

$$\begin{aligned} X &= X_W \times g(L) \\ Y &= Y_W \times g(M) \\ Z &= Z_W \times g(N) \end{aligned}$$

在此函数个 $g(x)$ 被定义为

$$\begin{aligned} g(x) &= x^3 && \text{if } x \geq \frac{6}{29} \\ g(x) &= \frac{108}{841} \times \left(x - \frac{4}{29} \right) && \text{otherwise} \end{aligned}$$

例 4.8 定义了 1976 智能库 L*a*b*空间与 CCIR XA/11 - recommended D65 白点。a*和 b*元件，虽然理论上无界，但是被定义才在与有效范围-128 到+127

例 4.8

```
[ /Lab
<< /WhitePoint [0.9505 1.0000 1.0890]
/RANGE [-128 127 -128 127]
>>
]
```

ICCBased 色彩空间

ICCBased 色彩空间 (PDF1.3) 基于一个跨平台的颜色概括被定义在国际色彩协会 (ICC)。不像 CalGray, CalRGB 和实验色彩空间，其是色彩空间子点里德特征项，一个 ICCBased 色彩空间的特点是字节序列标准格式，配置文件的详细信息可以再 ICC 说明中被找到 (见参考文献)。

一个 ICCBased 色彩空间被定义为一个数组：

```
[/ICCBased 流]
```

这个流包含了 ICC 的配置文件。除了一般对所有流的常见项 (见表 3.4, 38 页)，配置文件流增加了列在表 4.16 中的项。

表 4.16 指定到一个 ICC 配置流字典的增加项		
键	类型	值
N	Integer	(要求) 在被 ICC 配置数据描述的颜色空间的颜色成分数量。这个数目必须匹配在 ICC 配置中的实际元件数量。如 PDF1.4, N 必须为 1, 3, 4。
Alternate	array or name	(可选择的) 一个备用颜色空间在数据流数据被指定的情形下不被支持 (例如, 通过为早期 PDF 版本设计的应用程序)。这个备用空间可能是有被 N 定义的元件数量的任意有效颜色空间 (除了一个模式颜色空间)。如果这个项被省略并且应用程序并不理解 ICC 配置文件数据, 被颜色空间为 DeviceGray, DeviceRGB 或者 DeviceCMYK, 取决于是否 N 值分别是 1, 3, 4。 注: 没有来源颜色值的转换, 如色彩变换, 当使用代替颜色的空间。范围内的 ICCBased 色彩空间范围内的颜色值可能不属于颜色的备用空间范围。这种情况下, 在空间范围内的备用空间范围。这种情况下, 用备用空间范围内的最近值取代。
Range	array	(可选) $2 \times N$ 的数字数组 [min0 max0 min1 max1 ...] 制定相应的颜色分量的最小和最大有效值。这些值必须匹配在 ICC 配置文件里。默认值: [0.0 1.0 0.0 1.0 ...]。
Metadata	stream	(可选; PDF1.4) 一个包含针对颜色空间的包含元数据的元数据流 (见 10.2.2, “元数据流”)。

ICC 说明是一个不断发展的规范。表 4.17 展示 ICC 说明的版本在其上 ICCBased 颜色空间被 PDF1.3 版本所支持并且之后基于其。(早期的 ICC 说明版本也被支持)。

TABLE 4.17 ICC specification versions supported by ICCBased color spaces

PDF VERSION	ICC SPECIFICATION VERSION
1.3	3.3
1.4	ICC.1:1998-09 and its addendum ICC.1A:1999-04
1.5	ICC.1:2001-12
1.6	ICC.1:2003-09

PDF 生产者和用户应该遵循以下原则：

一个支持给定的 PDF 版本的用户被要求支持符合相应版本的 ICC 配置(和更早版本)如以上所述。它可能选择性的支持 ICC 后期版本。

对于大多数可预见和一致的结果，一个给定 PDF 版本的产生者应该符合 ICC 说明书版本的仅有配置文件。

一个 PDF 产生者可能嵌入式符合一个后期 ICC 版本的配置文件，通过了解结果将会取决于用户的能力。当忽略新特点的时候用户可能会处理配置文件，否则它可能完全失败。因此，建议产生者提供一个可供替代的色彩空间 (ICCBased 色彩空间字典中的替换项) 包含一个适用于 PDF 版本的配置文件。

PDF 仅支持的配置文件类型展示在表 4.18 中；其他类型可能将来会被支持。(尤其是，注意 XYZ 和 16-bit L*a*b*配置文件不被支持)。每一个特征域都必须在表的第二列中有这个范围的其值的列表。(配置必须满足表里所示的两种规范)。这个术语从 ICC 中提取。

TABLE 4.18 ICC profile types

HEADER FIELD	REQUIRED VALUE
deviceClass	icSigInputClass ('scnr') icSigDisplayClass ('mntr') icSigOutputClass ('prtr') icSigColorSpaceClass ('spac')
colorSpace	icSigGrayData ('GRAY') icSigRgbData ('RGB ') icSigCmykData ('CMYK') icSigLabData ('Lab ')

用于 PDF 色彩空间和 ICC 色彩配置的术语是相似的，但是有时候这些相同的项目是以不同的含义来使用。例如，在一个 ICCBased 色彩空间的原件的每个默认值都是 0 每个颜色原件的范围是一个被配置定义的指定颜色空间的函数并且在 ICC 说明中被指出。一些 ICC 颜色空间的范围被展示在表 4.19 中。

TABLE 4.19 Ranges for typical ICC color spaces

ICC COLOR SPACE	COMPONENT RANGES
Gray	[0.0 1.0]
RGB	[0.0 1.0]
CMYK	[0.0 1.0]
L*a*b*	$L^*: [0 \text{ } 100]; a^* \text{ and } b^*: [-128 \text{ } 127]$

由于 ICCBased 色彩空间被用作源颜色空间。仅“to CIE”配置信息 (AToB 在 ICC 术语中) 被使用; “from CIE”(BtoA) 信息当前被忽略。一个 ICC 配置文件可能也指定了一个显示意图, 但是 PDF 用户应用忽略了这些信息; 这个现实一天呀被一个独立参数在 PDF 中指定 (见“现实”, 230 页)。

注: 上述规定适用于一 ICCBased 色彩空间, 用于指定颜色的图形对象的来源。当这样的空间是用来作为一个透明的透明成像模型组混合色彩空间 (参见 7.2.3 节, “混合颜色空间”, 7.3, “透明度组”; 及 7.5.5, “透明度集团 XObjects”), 它必须同时“的智库”(AToB) 及“智库”(BToA) 信息。这是因为这个集团的颜色空间作为目的地的对象都被用来画内集团和本集团的业绩来源。 ICC 配置文件还用于指定一个匹配与目标输出设备或生产环境中的 PDF 文档的颜色特征的输出意向。在这种情况下使用时, 他们仍然受到了“以智库”和“从智库”信息, 见第 10.10.4, “输出意向, 为细节”其他方面的限制。

ICCBased 色彩空间的表现的申述小于 CalGray, CalRGB 和 Lab, 但是可以比表示一个色彩空间更宽的范围。在一个给定的色彩空间可以通过多于一个 CIE-Based 色彩空间系被表示的情形下, 结果颜色被期望相同显示, 无论挑选来显示的方法怎样。

一个特定的颜色空间是所谓的“标准 RGB”或者 sRGB，被定义在国际电工委员会（IEC）的文件颜色测量和电工系统和设备管理（见文献）。在 PDF 中，sRGB 颜色空间可以仅作为一个 ICCBased 空间被精确显示，尽管它近似于 ICCBased 空间。例 4.9 展示了一个针对典型的 3 元素 RGB 空间的 ICCBased 色彩空间。配置数据可读的 16 进制显示中被译码；实际上，如 FlateDecode 的无损减压过滤器被使用。

Example 4.9

```
10 0 obj % Color space
[ /ICCBased 15 0 R ]
endobj

15 0 obj % ICC profile stream
<< /N 3
/Alternate /DeviceRGB
/Length 1605
/Filter /ASCIIHexDecode
>>
stream
00 00 02 0C 61 70 70 6C 02 00 00 00 6D 6E 74 72
52 47 42 20 58 59 5A 20 07 CB 00 02 00 16 00 0E
00 22 00 2C 61 63 73 70 41 50 50 4C 00 00 00 00
61 70 70 6C 00 00 04 01 00 00 00 00 00 00 00 02
00 00 00 00 00 00 F6 D4 00 01 00 00 00 00 D3 2B
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 09 64 65 73 63 00 00 00 F0 00 00 00 71
72 58 59 5A 00 00 01 64 00 00 00 14 67 58 59 5A
00 00 01 78 00 00 00 14 62 58 59 5A 00 00 01 8C
00 00 00 14 72 54 52 43 00 00 01 A0 00 00 00 0E
67 54 52 43 00 00 01 B0 00 00 00 0E 62 54 52 43
00 00 01 C0 00 00 00 0E 77 74 70 74 00 00 01 D0
00 00 00 14 63 70 72 74 00 00 01 E4 00 00 00 27
64 65 73 63 00 00 00 00 00 00 00 00 17 41 70 70 6C
65 20 31 33 22 20 52 47 42 20 53 74 61 6E 64 61
72 64 00 00 00 00 00 00 00 00 00 00 00 00 17 41 70
70 6C 65 20 31 33 22 20 52 47 42 20 53 74 61 6E
64 61 72 64 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00 58 59 5A 58 59 5A 20 00 00 00 00 00 00 00 00 63 0A  
00 00 35 0F 00 00 03 30 58 59 5A 20 00 00 00 00  
00 00 53 3D 00 00 AE 37 00 00 15 76 58 59 5A 20  
00 00 00 00 00 40 89 00 00 1C AF 00 00 BA 82  
63 75 72 76 00 00 00 00 00 00 00 01 01 CC 63 75  
63 75 72 76 00 00 00 00 00 00 00 01 01 CC 63 75  
63 75 72 76 00 00 00 00 00 00 00 01 01 CC 58 59  
58 59 5A 20 00 00 00 00 00 00 F3 1B 00 01 00 00  
00 01 67 E7 74 65 78 74 00 00 00 00 20 43 6F 70  
79 72 69 67 68 74 20 41 70 70 6C 65 20 43 6F 6D  
70 75 74 65 72 73 20 31 39 39 34 00 >  
endstream  
endobj
```

默认颜色空间

指定在一个设备颜色空间（DeviceGray, DeviceRGB, 或者 DeviceCMYK）的颜色是设备相关的。通过设定默认颜色空间（PDF1.1），一个 PDF 文件可以要企鹅这些颜色系统地转化（重新被映射）到设备无关 CIE-based 颜色空间。这种能力在很多情况下很有用：

原打算针对一个输出设备的文件被重定向到不同设备。

一份文件意图与为早期版本的 PDF 设计兼容并且因此不能直接指定 CIE-based 颜色空间。

色彩更正或者显示意图被需要去应用到设备色彩（见“显示意图”，230 页）。

一个颜色空间被为绘制图形对象而选择。这是当前图形状态中颜色空间参数或者作为一个图像外在对象，在线对象，或者阴影字典中的项给出的色彩空间。无论色彩空间怎么被指定，它可能属于如下重新映射。

当一个设备色彩空间被选择，色彩空间的当前源字典（见 3.7.2，“源字典”的子字典被一个指定相应的默认颜色空间（DefaultGray, DefaultRGB, 或者 DefaultCMYK，各自对应 DeviceGray, DeviceRGB, 或者 DefaultCMYK）的项的显示所检查。如果这样一个项背提出，它的值被用作针对当前被执行操作的颜色空间（如果应用程序没有识别出此空间，没有重新映射产生；原始的设备色彩空间被使用）。

在原始设备色彩空间的颜色值被不变地传递到默认色彩空间，其必须有和源空间相同数量的元素。定义元素范围和是否元素被增加或减少的默认色彩空间应该被选择区兼容原始的。如果一个颜色值在默认色彩空间的范围之外，它被调整到最近的有效值。

任何一个不同于实验的色彩空间，索引或者图案色彩空间可能被用作一个被提供的默认色彩空间，其如上所述可与原始设备色彩空间兼容。

如果选择的空间是基于基础设备色彩空间的特殊色彩空间，默认的色彩空间被用于底层空间。这适用于以下色彩空间：

一个图形色彩空间格局的基本色彩空间

一个索引色彩空间的基本色彩空间

一个 Separation 的或者 DeviceN 的替代色彩空间或者 DeviceN 的色彩空间（但是仅当替代色彩空间被选择）

见 4.5.5，“特殊色彩空间，“针对这些色彩空间上的细节。

注：有没有颜色值的转换，如色彩变换，当使用默认的色彩空间。颜色值范围内的设备的色彩空间范围，可能不属于默认的色彩空间（特别是如果默认是 ICCBased 色彩空间）的范围。在这种情况下，在默认的空间范围内最接近值使用。基于这个原因，Lab 色彩空间是不允许的 DefaultRGB 色彩空间。

CIE-Based 颜色空间的隐式转换

PDF 文件被打算显示在一个特别的目标输出设备的工作流程（如一个有特殊油墨和媒体的打印印刷机），指定针对匹配校准目的设备的一个 CIE-based 色彩空间的所有文件对象的源颜色非常有用。结果文件，尽管是针对目标设备的具体特点，但是仍然设备评测独立，并且将产生合理的结果如果重定位到一个不同的输出设备。然而，预期的是如果文件被在目标设备上打印，源颜色将在一个匹配校准的色彩空间中被指定如果设备将不变传递，而不转换且从 215 页的图 4，。

14 描绘的中间智库 1931 XYZ 的空间。

尤其是，当针对一个 CMYK 输出设备的颜色被指定在一个使用匹配 CMYK 打印配置，将这些颜色从四个元素转换到三个且返回不可用且导致黑色元素的一个失真的 ICCBased 颜色空间。在这种情形下，PDF 用户应用程序可能提供供用户指定一个用于打印，证明或者预映的特定标准的能力。这个校准之后被认识是目的输出设备（一般是 DeviceCMYK）的原本颜色空间，并且颜色被表示在一个 CIE-based 源颜色空间匹配他可以被当做它们被直接指定在一个设备的原本颜色空间。注意，在这个隐式转换下的条件不能被在 PDF 中指定，因为 PDF 中没有输出设备的校准的项（尽管一个输出目的字典，如果被提出，可能提出了这样一个校准；见 10.10.4，“输出目的”）。这个转换通过应用程序完全隐藏并且没有参与 PDF 颜色空间的说明。

当这种类型是隐形转换时，所有的设备色彩空间语义也应该适用，及时它们不适用于一般 CIE-based 空间。特别是：

非零套印模式（见 4.5.6，“套印控制”）决定空间的颜色成分值的说明。

如果空间被用作针对透明成像模式的透明度组的混合色彩空间（见 7.2.3，“混合色彩模式”；7.3，“透明度组”；和 7.5.5 “透明度组外在对象”），空间成分，如 Cyan，可以被选择和这个组使用在一个 Separation 获知 DeviceN 颜色空间（见 234 页“Separation 颜色空间”和 238 页“DeviceN 颜色空间”）。

同样，任何核这样一个透明度组一起对对象的设备色彩空间的使用有明确的到组色彩空间的转换。

注：一个源色彩空间可以被直接指定（如何一个 **ICCBased** 色彩空间）或者间接指定使用在默认色彩空间机制中（例如，**DefaultCMYK**；见 227 页“默认色彩空间”）。一个 **CIE-based** 色彩空间的隐式转换为一个设备空间不应该取决于是否这个 **CIE-based** 空间被直接或者间接指定。

绘制目的

虽然 **CIE-based** 颜色规格理论上设备无关，但是它们受到输出设备的色彩再现能力中的实际限制。这样的限制可能有时候要求颜色规格的多样性属性的协调为一个给定设备绘制颜色时。指定一个绘制目的（PDF1.1）允许一个 PDF 文件设定关于那些属性来保存和哪些属性舍弃的优先权。例如，PDF 文件可能要求在输出设备范围内的颜色（即它可以复制的颜色范围）被不完全显示当舍弃溢出颜色的精确性或者一个被扫描的图像如图像被在严格的色度精度上以一个感性愉悦的方式呈现。

绘制目的被 **ri** 操作（见 4.3.3，“图形状态操作”），一个图形状态参数字典里的 **RI** 项（见 4.3.4），和图像字典里的目的项（4.8.4，“图像字典”）指定。值是一个定义绘制目的的名字。表 4.20 中列出的规范绘制目的在 PDF 查看应用程序中被从 Adobe 系统中初始化清空；板块 5 说明了它们的影响。这些目的被刻意地选择密切相应于被国际色彩协会（ICC）定义的，一个发展了设备无关色彩的规范的行业组织。注意，然而，绘制目的支持的准确设置可能从一个输出设备改变到另一个；一个特定的设备可能不支持所有可能的目的或者在表中列出以外增加的项。如果应用程序不认可指定名，它则默认使用 **RelativeColorimetric** 目的。

见 7.6.4，“绘制参数和透明度”，且特别是 543 页的“绘制意图和颜色转换”，针对将来的透明度成像模式里的绘制目的角色讨论。

表 4.20 绘制目的

名字	说明
AbsoluteColorimetric	颜色仅仅用关于光源来表示；没有修改被应用到输出设备的白点上（如未打印页面的颜色）。因此，例如，一个显示器的白点，其比较打印纸上的近乎蓝色，将备用一个蓝色部分重造。在色域中色彩完全再现；在色域之外颜色在可复写的范围里被映射为最接近的颜色。这种再现方式有益于提供比从一个输出媒体到另一个更加精确地颜色。它不益于使用在媒体白点和 1.0 之间的 Y 值产生色域之外的颜色。一个要求根据不同媒体的精确复制的特别使用可能是针对对标志和实色的。
RelativeColorimetric	颜色使用光源和输出媒体白点的结合来表示（如未打印页面的颜色）。因此，例如，一个显示器的白点将被重现在一个打印机上当仅仅留下未标记页面，忽略两种媒体间颜色的不同。在色域中色彩精确再现；色域外颜色在可复制的范围被映射到最接近的颜色。重现风格有益于为不同的输出设备改写变化的白点。同时它不利于不提供比较从一个媒体到另一个的精确颜色。典型用法作用可能会针对矢量图形。
Saturation	颜色以维持或者加强饱和状态的方式被代表。颜色范围内的复制可能活着可能不是无律颜色。典型用法可能针对任务图形，在其 saturation 是最重要的颜色属性。
Perceptual	颜色以一种令人愉悦的感性出现的方式被代表。为维持颜色关系，颜色范围内和范围外一般都从它们的精确度颜色被改变着。典型用法可能针对扫描图像。

4.5.5、特殊颜色空间

特殊颜色空间对基本色彩空间增加了功能或属性。有四种特殊的色彩空间系：模式，索引，Separation 和 DeviceN。

模式色彩空间

一个模式色彩空间（PDF1.2）使 PDF 内容流能够使用模式而不是一个单一的颜色来绘制一个区域。这个 mode 可能是一个平铺模式（类型 1）或者一个底纹模式（类型 2）。4.6，“模式”，细节上介绍了模式。

索引颜色空间

一个索引颜色空间允许一个 PDF 内容流使用小整数作为进入一个颜色地图或者在其他空间任意颜色的颜色表的索引。一个 PDF 用户应用程序将每一个样本值作为一个进入色彩表的索引且使用在此找到的颜色值。这个功能可以相当减少要求表示一个样本图像的数据数量——例如，用 8-bit 的索引值作为样本代替 24-bit RGB 色彩值。

一个索引颜色空间可以被一个四元素数组定义：

[/Indexed base hival lookup]

第一个元素是颜色空间系的名字索引。剩余的元素是一个索引颜色空间要求的参数；它们的含义如下所述。设定当前冲程或者无冲程色彩空间到一个索引色彩空间初始化相应的当前颜色到 0。

基本参数是一个数组或者定义颜色表中的值被解释的基本颜色空间的名。它可以使任意设备或者 CIE-based 色彩空间或者（在 PDF1.3）一个 Separation 或者 DeviceN 空间，但非模式空间或者其他的空间。例如，如果基础色彩空间是 DeviceRGB，色彩表的值被表示为红色，绿色和蓝色成分；如果基础色彩空间是一个 CIE-based ABC 空间如一个 CalRGB 或者 Lab 空间，值则被显示为 A, B, C 元素。

注：在 PDF1.2 中企图使用一个 Separation 或者 DeviceN 颜色空间作为一个索引色彩空间的基础引发错误。

该 hival 参数是一个整数制定了最大索引值。换句话说，颜色表被从 0 到 hival 的整数索引。Hival 不能比 255 大，其是一个要求索引一个表通过 8-bit 的索引值。

颜色表被查找参数定义，其可以是一个流或者（在 PDF1.2）中的一个字符串。它提供了索引值和在基础颜色空间里的相应值的映射。

颜色表数据必须是 $m \times (hival + 1)$ 字节长，在此 m 是基本颜色空间的颜色元素的数目。每一个字节是一个在 0 到 255 范围内缩放到在基础颜色空间内的相应颜色元素上的非指定整数；即，0 相应于元素范围的最小值，且 255 相应最大值。

注：元素使用了一个关于索引色彩空间色彩表的不同说明。附言中，元素值总是所嚷道 0.0 到 1.0 的范围，无论基础色彩空间的颜色值的范围。

表中每一项的颜色元素连续的在字符串或者流中显示。例如，如果基础颜色空间是 DeviceRGB 并且索引颜色空间包含两种颜色，字符串或者流中的字节顺序是 R0 G0 B0 R1 G1 B1，在其字母指示了颜色元素且数值下标指示表中项。

例 4.10 显示了一个索引色彩空间的指定，其在 DeviceRGB 色彩空间映射 8-bit 的索引值到 3 元素颜色值。

例 4.10

```
[ /Indexed
/DeviceRGB
255
<000000 FF0000 00FF00 0000FF B57342 ...>
]
```

这个例子值显示了查找字符串的前 5 个颜色值；总之，应该是 256 颜色值并且字符串应该是 768 字节长。已建立在这个颜色空间，程序现在可以指定颜色作为单独元素值在一个 0 到 255 的范围。例如，一个 4 颜色值选择一个元素被指定为十六进制整数的 RGB 颜色 B5, 73 和 42。用 255 区分这些并且缩放结果到 0.0 到 1.0 的范围产生一个各为 0.710, 0.451 和 0.259 的红，绿和蓝色元素。

尽管一个索引颜色空间主对图像值很有用，索引值也可以被和颜色选择操作 `SC`, `SCN`, `sc`, 和 `scn` 一起使用。例如：

`123sc`

选择一个相同颜色作为设定一个图像样本值为 123。索引值应该是一个从 0 到 `hival` 的整数。如果这个值是一个实数，它在最接近的整数附近；如果它在 0 到 `hival` 范围之外，它被调整到该范围内最接近的值。

Separation 色彩空间

颜色输出设备通过结合初始颜色或者产生不同数量的着色剂来产生所有的颜色。在作为添加颜色设备如显示器中，初始着色剂包括红，绿和蓝色荧光粉；在一个输出设备如打印机中，它们一般包括青色，品红，黄色，有时还有黑色墨。另外，一些设备可以应用特殊的着色剂，常被称为稀有着色剂，产生了不能通过标准处理染色剂单独获得。例子包括金属和荧光色以及特殊质感。

当打印一个页面时，大多数设备产生了一个单独的混合页面，在其上所有的处理染色剂（和特殊染色剂，如果有的话）被结合。然而，一些设备，如图像设置，产生了一个 `Separation`，单色的页面显示，称之为 `Separation`，针对每一种着色剂。当 `Separation` 在之后结合——在一个打印机上，例如——适当的油墨或者其他着色剂被应用到它们之上，结果是一个全彩色页面。

一个 `Separation` 的颜色空间 (PDF1.2) 提供了一种定义增加的着色剂或者对一个输出设备隔离设备颜色空间的单独颜色元素的方式。当这样一个空间是当前的颜色空间时，当前颜色是一个单独的元素值，称作 `tint`，仅控制给定的着色剂或者颜色元素的应用。

注：长期 **Separation** 经常被误用为一个独立设备着色剂的代名词。在这一讨论中，一个产生 **Separation** 的打印系统导致了一个针对每一个着色剂的 **Separation** 物理媒介（一般薄膜）。这些物理媒介部分被正确地认为是 **Separation**。一个特定的染色剂正好构成了一个 **Separation** 仅当设备产生了物理 **Separation**，其中一个相等于被给定的着色剂。这个 **Separation** 颜色空间虽然为历史原因命名，但是它延伸出拓宽控制一般独立染色应用程序的目的，无论它们是否真实地被作为物理 **Separation**。

同时注意一个 **Separation** 色彩空间的操作本身是任何特定的输出设备特性的独立、根据不同的设备，这个空间可能或者可能不相等于一个正确自然的 **Separation** 或者一个实际的着色剂。例如，一个 **Separation** 颜色空间可能被用作控制一个独立处理染色剂（如青色）的应用程序在一个不产生自然 **Separation** 的混合设备上，或者表示了一个颜色（比如橙色）针对没有指定的着色剂存在于设备上。一个 **Separation** 颜色空间提供了恒定可预测的行为，及时在不能直接产生要求颜色的设备上。

一个 **Separation** 颜色空间被如下定义：

[/Separation name alternateSpace tintTransform]

换句话说，第一个元素是颜色空间家族姓氏 **Separation** 的是一个 4 元素的数组。其余元素是一个 **Separation** 色彩空间要求的参数；它们的意义在下面被讨论。

一个在 **Separation** 颜色空间里的颜色值包含了一个单独的 **tint** 元素在 0.0 到 1.0 的范围里。值 0.0 代表了可以被应用的着色剂的最小值；1.0 代表了最大值。**Tint** 总是被当做消退颜色，即使当设备产生通过增加的方式对指定元素的输出。这样，一个 0.0 的 **tint** 值指定了可以由给定的着色剂获得的最浅的颜色，且 1.0 是最深的。（这个公约对 **DeviceCMYK** 颜色元素也相同但是与 **DeviceGray** 以及 **DeviceRGB** 的相反）。这个针对在图形状态中绘制和未绘制的颜色的初始值是 1.0。**SCN** 和 **scn** 操作分别设定了当前的绘制和非绘制颜色到一个着色值。一个带有单独样本的样本图像也可以被用作染色值的一个来源。

命名参数是一个命名对象定义了其 **Separation** 空间被打算表示（或者特殊的命名 **All** 或者 **None**，见下面）的着色剂的名称。这样的着色剂名称是任意的，并且它们的任何数字，遵守执行限制。

特殊的染色剂命名 **All** 全部引用所有的可用的着色剂在一个输出设备上，包括那些针对处理染色剂的规范。当带有这个着色剂名的 **Separation** 空间是当前颜色空间时，绘制操作立刻应用染色值到所有可能的着色剂。这对于在相同地方如同绘制记录目标的目的在任意 **Separation** 上非常有用。这样的标记一般被作为构建一个页面的最后一步以至于它们不会被以后的绘制操作覆盖。

特殊的染色剂命名 **None** 从不产生任何可见的输出。在一个 **Separation** 空间的带有这些染色剂命名的绘制操作对当前页面没有影响。

所有的设备支持带有着色剂命名 **All** 和 **None** 的 **Separate** 颜色空间，即使它们不支持其他的。带有这些着色剂命名中的任何一个的 **Separate** 空间忽略了备用空间和染色转换参数（以下有述），尽管有效值仍然被提供。

目前颜色空间被设置到一个 **Separation** 空间，用户称呼决定是否设备有一个可用的着色剂相应于被要求的空间的命名。如果这样，这个应用程序忽略了备用空间和染色转换参数；空间里的之后的绘制操作直接应用了指定的着色剂，依据提供的染色值。

注：先前的段落只适用于如打印机或者图像设置设备这样的输出设备。对于一个增加的设备如一个电脑显示器，一个 **Separation** 颜色空间从不会直接应用一个处理染色剂；它总是如下所述复位到备用颜色空间。这是因为独立使用处理染色剂的模式不会像有意在添加设备上工作那样；例如，绘制红色元素染色剂在一个白色背景上产生了红色到青色转化的结果。

请注意，这个例子只适用于针对增加设备的着色剂，而不是特殊的命名红，绿和蓝色。对比而言，一个打印机可能有一个（输出的）油墨命名，例如，红色，其应该如同任何其他的被支持的着色剂运作在一个 **Separation** 颜色空间。

如果关联着一个 **Separation** 颜色空间的着色剂命名不相应于设备上的一个可用着色剂，为随后的绘制操作的应用程序将被在一个备用颜色空间里执行。目的颜色近似于在一个设备或者 CIE-based 色彩空间的颜色，其被以一般初级或者处理过的染色剂显示：

备用空间参数必须是一个定义备用颜色空间的数组或者命名对象，这个颜色空间可以是任意设备或者 CIE-based 颜色空间但是不是另一个特别的颜色空间 (Pattern, Indexd, Separation, 或者 DeviceN)。

染色转化参数必须是一个函数 (见 3.9, “函数”)。在随后的绘制操作，一个应用程序调用这个函数去转化一个染色值为备用颜色空间里的颜色值。这个函数被和染色值一起调用并且必须返回相应的颜色元素值。即，元素数量和他们值的说明依赖于备用空间。

注：在一个备用颜色空间的绘制可能产生一个目的颜色的较好近似值仅当不透明对象被绘制。然而，它并没有恰当的表示对象和它的背景之间的合作当对象透明绘制或者当套印被激活 (见 4.5.6, “套印控制”)。

例 4.11 阐述了一个 **Separation** 颜色空间 (对象 5) 的指定，其想要沉声一个命名为 **LogoGreen** 的颜色。如果输出设备没有着色剂相应于这个颜色，**DeviceCMYK** 被用作备用颜色空间，并且染色转换函数 (对象 12) 线性映射染色值到一个近似 **LogoGreen** 颜色的 CMYK 颜色值的阴影里。

Example 4.11

```
5 0 obj % 颜色空间
[ /Separation
/LogoGreen
/DeviceCMYK
12 0 R
]
endobj
```

```
12 0 obj % 染色转换函数
<< /FunctionType 4
/Domain [0.0 1.0]
/Range [0.0 1.0 0.0 1.0 0.0 1.0 0.0 1.0]
/Length 62
>>
stream
{ dup 0.84 mul
exch 0.00 exch dup 0.44 mul
exch 0.21 mul
}
endstream
endobj
```

见 7.6.2, “专用颜色和透明度”, 针对在透明成像模式下 **Separation** 颜色空间的角色的进一步讨论。

DeviceN 颜色空间

DeviceN 颜色空间 (PDF1.3) 包含任意颜色元素的数目。它们提供了比可能在规范颜色空间如 **DeviceCMYK** 或者独立的 **Separation** 颜色空间更大的灵活性。例如, 可能去建立一个 **DeviceN** 颜色空间包含仅仅青色, 品红, 黄色成分, 同时黑色成分被排除。

DeviceN 颜色空间被如这些用于应用程序中:

高精度颜色更甚于规范 CMYK 处理着色剂区产生一个延伸得全部或者范围的颜色的用法。一个通俗的例子是 PANTONE Hexachrome 系统, 其使用了 6 种染色剂: 通常的青色, 品红, 黄色和黑色, 橙色和绿色。

多频彩色系统使用单独元素的图像去指定多重颜色元素。例如, 在一个网版中, 一个单独元素的图像可以被用作指定黑素元素和稀有颜色元素。色调复制一般而言对于不同的元件是不同的。例如, 黑色元件可能是哟个单个元件图像的精确样本数据绘制; 专色元件可能通过一种加强阴影的方式作为图像数据的非线性函数。板块 6 展示了一个使用黑色和品红元件的例子。板块 7 内, 一个单独原色灰度级图像被用作产生一个使用四个染色剂的 quadtone 结果: 黑色和三个 PSNTPNE

专属性。见 252 页的例 4.21 针对用来产生这个图像的代码。

DeviceN 被设计为包含多种成分的颜色空间。这些成分相应于一些指定设备的染色剂。正如 **Separation** 颜色空间, PDF 用户程序必须能够近似着色剂如果它们不能在当前输出设备上可用时, 如一个显示器。为习惯这, 颜色空间定义提供了一个专属性转换函数, 这个函数可以被用作将所有的元件转换到备用颜色空间。**PDF1.6** 扩展 **DeviceN** 的意义去包含被认为是 **Nchannel** 颜色空间的颜色空间。这样的颜色空间可能包含了任意的数量的专属性和执行原件, 其可能或可能不相应于指定的设备着色剂 (这个处理成分必须来自一个单个处理的颜色空间)。它们提供了关于允许应用程序在转化颜色中更加灵活的信息。例如, 它们可能使用它们自身的混合算法到一个屏幕显示并且混合打印, 而不是被要求使用一个指定的专属性转换函数。这些颜色空间被一个针对属性字典的图表类型项的 **Nchannel** 值定义 (见表 4.21)。一个对于图表类型项的 **DeviceN** 的值, 意味着不仅先前的外形被支持。不支持 **PDF1.6** 的 PDF 用户应用把这些颜色空间作为标准 **DeviceN** 颜色空间并且使用专属性转换函数作为一个参考。使用 **NChannel** 外观的生产者应用需要遵循某些准则, 如注意在这一节的每一处, 获得好的向后兼容。

DeviceN 颜色空间被与 **Separation** 颜色空间以一个相同的方式定义——事实上, 一个 **Separation** 颜色空间可以仅用一种元件被定义为一个 **DeviceN** 颜色空间。一个 **DeviceN** 颜色空间被如下定义:

[/DeviceN names alternateSpace tintTransform]

或者

[/DeviceN names alternateSpace tintTransform attributes]

这是一个 4 元素或者 5 元素的数组其第一个元素是颜色空间家族姓氏 **DeviceN**。

其实元素是一个 **DeviceN** 颜色空间要求的参数。

这个命名参数是一个命名对象的数组定义了单独的颜色元件。数组的长度决定了在 **DeviceN** 颜色空间的元件数目，其遵守执行限制；见附录 C。元件命名必须全都互不相同，除了命名 **None**，其可以被重复如这一节之后所述。（这个特殊的命名 **All**，被 **Separation** 空间所使用的，不被允许）。

颜色值是在范围 0.0 到 1.0 的淡色元件。

对于没有一个 **Nchannel** 图表类型的 **DeviceN** 颜色空间，0.0 总是表示着色剂的最小数量；1.0 代表最大数量。淡色经常被作为负颜色，及时设备产生了输出队增加方式制定的原件。这样，一个 0.0 的淡色值指定了可以使用给定着色剂获得的最浅的颜色，和最深的 1.0。（这一协议和 **DeviceCMYK** 颜色元件相同但是和 **DeviceGray** 和 **DeviceRGB** 相反）。

低于 **Nchannel** 颜色空间，针对增加处理颜色的值（如 **RGB**）被指定在它们的自然结构，在这 1.0 代表了颜色的最大值。

当这个空间被设备到当前颜色空间（使用 **CS** 或者 **cs** 操作），每一个元件被给定一个为 1.0 的初始值。提供给 **SCN** 或 **scn** 操作数值被解释为颜色元件值在以颜色被给定在一个命名数组中的秩序那样，如在使用一个 **DeviceN** 颜色空间的采样图像中的值一样。

这个备用空间参数是一个数组或者命名对象，其可以是任何设备或者 CIE-based 颜色空间但是不是另一个特殊的颜色空间（**Pattern**, **Indexed**, **Separation**, 或者 **DeviceN**）。当颜色空间被设置到一个 **DeviceN** 空间时，如果任何在这个颜色空间的元件命名不相应用于一个在设备上的可用染色剂时，PDF 用户应用可以执行随后的绘制操作在被这个参数指定的备用颜色空间中。

注：对于 **NChannel** 颜色空间，元件被各自评估；这即是，只有不显示在设备上的那些使用备用颜色空间。

这个浅色转换参数指定了一个函数（见 3.9，“函数”），其被使用来将浅色值转化到备用颜色空间中。它被 **n** 浅色值调用并且返回 **m** 颜色元件值，在此 **n** 是需要去定义一个 **DeviceN** 颜色空间的颜色的元件数目，**m** 是备用空间要求的数目。

注：在备用空间上的绘制可能产生了一个目标颜色的较好的近似值仅当不透明对象被绘制。然而，它没有恰当的代表了一个对象和它的背景的相互作用当对象被使用透明度绘制或者套印（见 4.5.6，“套印控制”）被激活。

命名为 **None** 的颜色成分，可能被只为没有 **NChannel** 的图表类型的 **DeviceN** 颜色空间显示，指示的相应的颜色元素从不绘制在页面上，如在一个针对 **None** 着色剂的 **Separation** 颜色空间。（然而，见 Appendix H 中的执行记录 47）。当一个 **DeviceN** 颜色空间正在直接回执一个命名的设备着色剂，颜色元件相应于 **None** 着色剂被舍弃。然而，当 **DeviceN** 颜色空间回复到它的备用空间时，哪些元件被传递到淡色转换函数，其可以如意向使用它们。

注：一个其元件着色命名全为 **None** 的 **DeviceN** 颜色空间总是舍弃它的输出，正如一个针对 **None** 的 **Separation** 一样；它从不回复到备用设备空间。仅当至少一种颜色元素（除 **None** 外）被指定并且在设备上不可用时复原重现。

可选属性的参数是一个包含关于 PDF 用户程序可能使用的颜色空间元件增加信息字典（见表 4.21）。PDF 用户没有被要求去使用备用空间和淡色转化空间的参数，并且可能代替使用混合算法的惯例，随着被提供在属性字典里的其它信息如果现实的话。（如果在一个属性字典里 **Subtype** 项的值是 **NChannel**，这些信息必须被现实。）然而，备用空间和单色转化不惜总是为想使用它们或者不支持 PDF1.6 的应用程序提供。

Subtype 入口的 **Nchannel** 值表示了某些字典中的必须具有的入口不如可选的入口。**Colorants** 入口规定了包括了色彩空间中所有污点着色剂的字典；它们被特定的 **Separation** 色彩空间所定义。**Process** 入口是一个规定了表征色彩空间使用的色彩过程以及分量名称的过程字典。如果子类型为 **Nchannel** 并且色彩空间具有过程分量的话，那么它将是必须的。（一个 **Nchannel** 色彩空间可能包含了至多一个过程色彩空间的分量。）

表 4.21 DeviceN色彩空间属性字典的入口

关键字	类型	值
Subtype	Name	(可选; PDF1.6) 用来指定对色彩空间推荐处理方式的命名。可能的值为 DeviceN 和 NChannel 。默认值: DeviceN 。
Colorants	字典	(如果子类型是 Nchannel 或者色彩空间包括污点着色，那么该关键字是必须的；其他情况是可选的)。 字典描述了特定的 DeviceN 色彩空间所使用的着色。对于该字典的每个入口，关键字为着色名，并且值为一个定义了 Separation 色彩空间的数组（见“ Separation 色彩空间”，234 页）。关键字必须与给定色彩空间的着色名相匹配。 字典提供了一些应用特定着色剂可能用到的信息。在特定情况下，在一个 Separation 色彩空间所属的色彩空间和染色转换函数单独描述了着色剂，但是 DeviceN 色彩空间只描述了它自身的着色剂。 如果子类型是 Nchannel ，在这个色彩空间中的字典必须为所有的污点着色剂准备了入口。字典也有可能包括了一些在该色彩空间中不会出现的加性色彩。
Process	Dictionary	(如果子类型是 Nchannel 并且色彩空间包括了过程色彩空间的分量，那么该关键字是必须的，其他情况下则是可选的; PDF1.6)。该字典（见表 4.22）描

		述了该空间中过程色彩分量所属的色彩空间。
MixingHints	Dictionary	(可选的; PDF1.6)。当某墨水被使用在染色转换函数中, 该字典(见表 4.23) 规定了该墨水的可选属性。

对于属性字典中 `Subtype` 入口的值为 `NChannel` 的色彩空间，下面的限制被应用到过程色彩里：

至多存在一个过程色彩空间的色彩分量，这个过程色彩空间可以是任何的设备或者 `CIE-based` 过程色彩空间。

对于非 `CMYK` 色彩空间，过程分量的名字必须以正常的色彩排序（例如，红，绿，蓝）顺序的出现在 `names` 数组中。尽管如此，`names` 数组中的名称不需要和真实色彩空间的名称相匹配（例如，红分量不需要叫做红）。名称的映射是被过程字典所规范的（见表 4.22 以及下面的讨论），且这种规范是必须出现的。

过程着色剂的定义不是必须出现在着色剂字典中。如果着色剂出现在过程字典中，那么该字典中任何定义可以被忽略。过程字典中不应出现指定任何被指定为污点着色剂的分量的规范。

对于一个 `CMYK` 色彩空间，分量的子集可能被显示出来，并且可能是以任何的顺序出现在 `names` 数组中。保留字 `Cyan`, `Magenta`, `Yellow`, `Black` 被设置为过程颜色，而并不是必须与规定设备上的着色剂对应；在过程字典中，它们并不是必须具有入口的。

过程分量的值必须按照它们的自然格式来存储（包括 `CMYK` 的减性色彩值以及 `RGB` 的加性色彩值），因此在用户使用过程字典时，这些过程值可以被直接的转换。

（对于加性色彩空间，正如在前面讨论 `names` 参数时描述的，这刚好是如何为 `DeviceN` 规定色彩值的反向过程。

属性字典中的 `MixingHints` 入口规定了当目标设备无法提供真实着色剂时，`minxinhints` 字典中着色剂使用时的特性信息。这种信息的应用并不是必须的。

表格 4.22 请自行完善

表格 4.23 请自行完善

每个 `mixing hints` 字典入口针对着一个着色剂名称组，该组包括了 `Colorants` 字典中的点着色剂。在某些情况下，当 `DeviceCMYK` 被过程色彩空间字典规范以后，它们也许还会代表一个或者多个独立的过程分量 (`Cyan`, `Magenta`, `Yellow`, `Black`)。尽管如此，如果具有 `ICC` 文件提供的信息，应用程序应该忽略掉这些过程分量的入口。

注释：`mixing hints` 子字典（向着色剂字典一样）也许会制定某些该 `DeviceN` 色彩空间中未必使用的着色剂。这允许它们被用于多个 `DeviceN` 色彩空间，一致的定义使得文件的规模变得更小了。

为了色彩的一致性，PDF 使用者应该遵循如下的原则：

对于文件中的所有 `DeviceN` 实例，使用者应该使用想用的混合算法或者使用标准的染色转换函数。

注释：当染色装换函数被调用，难点在于，制造者需要保证所有的 `DeviceN` 实例所选择的特定函数保证会制造出相同的色彩表现。

当分离的色彩或者一个 `DeviceN` 色彩空间分量调用混合算法时，它们应该制造出相似的效果。

例 4.12 显示了一个有三个分量 Orange, Green, None 的 DeviceN 色彩空间。在这个例子中, DeviceN 色彩空间, 标号 30, 具有一个 Colorants 入口间接指向标号 45 实例 (该实例也可能被其他的属性字典所引用) 的属性字典。 tintTransform1, 定义未必给出, 将三个色彩分量 (Orange, Green, None 的染色) 映射到一个四维色彩空间, DeviceCMYK。 tintTransform2, 将一个单一的颜色分量 (一个橘染色) 映射到四位色彩空间 DeviceCMYK, tintTransform4 将 PANTONE 染色映射到 DeviceCMYK。

Example 4.12

```
30 0 obj % Color space
[ /DeviceN
[/Orange /Green /None]
/DeviceCMYK
tintTransform1
<< /Colorants 45 0 R >>
]
endobj
45 0 obj % Colorants dictionary
<< /Orange [ /Separation
/Orange
/DeviceCMYK
tintTransform2
]
/Green [ /Separation
/Green
/DeviceCMYK
tintTransform3
]
/PANTONE#20131 [ /Separation
/PANTONE#20131
/DeviceCMYK
tintTransform4
```

]
>>
endobj

例 4.13 至 4.16 显示了 NChannel 色彩空间的应用。例 4.13 显示了标准 CMYK 过程分量的应用。例 4.14 则显示了 Lab 过程分量的应用。

Example 4.13

```
10 0 obj % Color space
[ /DeviceN
[/Magenta /Spot1 /Yellow /Spot2]
alternateSpace
tintTransform1
<< % Attributes dictionary
/Subtype /NChannel
/Process
<< /ColorSpace [/ICCBased CMYK_ICC profile ]
/Components [/Cyan /Magenta /Yellow /Black]
>>
/Colorants
<< /Spot1 [/Separation /Spot1 alternateSpace tintTransform2]
/Spot2 [/Separation /Spot2 alternateSpace tintTransform3]
>>
>>
]
Endobj
```

Example 4.14

```
10 0 obj % Color space
[ /DeviceN
[/L /a /b /Spot1 /Spot2]
alternateSpace
tintTransform1
<< % Attributes dictionary
/Subtype /NChannel
/Process
<< /ColorSpace [ /Lab << /WhitePoint ... /Range ... >> ]
```

```
/Components [/L /a /b]
>>
/Colorants
<< /Spot1 [/Separation /Spot1 alternateSpace tintTransform2 ]
/Spot2 [/Separation /Spot2 alternateSpace tintTransform3]
>>
>>
]
```

例子 4.15 说明了一种推荐的方法，这种方法处理了一个问题，即点着色和过程着色所用到的组件拥有相同的名字。因为名称数组不可以拥有相同的名字，因此过程着色应该被赋予不同的名字，这些不同的名字映射到过程组件，过程组件位于程序库的组件入口处。在这种情况下，红色引用了点着色；过程红，过程绿，以及过程蓝被映射到了 RGB 颜色空间中的组件中。

例子 4.15

```
10 0 obj % Color space
[ /DeviceN
[/ProcessRed /ProcessGreen /ProcessBlue /Red]
alternateSpace
tintTransform1
<< % Attributes dictionary
/Subtype /NChannel
/Process
<< /ColorSpace [ /ICCBased RGB_ICC profile ]
/Components [/ProcessRed /ProcessGreen /ProcessBlue]
>>
/Colorants
<< /Red [/Separation /Red alternateSpace tintTransform2 ] >>
>>
]
```

例子 4.16 表明了混合字典的用法

例子 4.16

```
10 0 obj % Color space
[/DeviceN
[/Magenta /Spot1 /Yellow /Spot2]
alternateSpace
tintTransform1
<<
/Subtype /NChannel
/Process
<< /ColorSpace [ /ICCBased CMYK_ICC profile ]
/Components [/Cyan /Magenta /Yellow /Black]
>>
/Colorants
<< /Spot1 [/Separation /Spot1 alternateSpace tintTransform2 ]
/Spot2 [/Separation /Spot2 alternateSpace tintTransform2 ]
>>
/MixingHints
<<
/Solidities
<< /Spot1 1.0
/Spot2 0.0
>>
/DotGain
<< /Spot1 function1
/Spot2 function2
/Magenta function3
/Yellow function4
>>
```

```
/PrintingOrder [/Magenta /Yellow /Spot1 /Spot2]
>>
>>
]
```

见第 7.6.2 章节，“点色及透明”，为了深一步地讨论在透明图像模型中 DeviceN 颜色空间的作用。

多色调例子

下面的例子说明了各种有趣的和有用的特殊情况，即在组合索引和 DeviceN 色彩使用的空间，产生多频颜色。

例子 4.17 和 4.18 的说明了 DeviceN 用于创建双色调的色彩空间。在例子 4.17 中，色彩空间索引将范围是 0 到 255 的索引值映射到在青色和黑色的双色调 DeviceN 空间。实际上，索引值犹如是双色调色彩空间，然后双色调色彩空间再被映射到色彩的两个基本着色剂。仅仅是为了被显示的索引色而查找字符表的开始；全表将包含 256 个双字节条目，每个指定为青色和黑色色彩值，一共有 512 字节。如果 DeviceN 色彩空间的可选择色彩空间被选择，色彩转换功能（例子中对象 15）将两个青色和黑色的色彩组件映射到 4 个组件，这 4 个组件支持 DeviceCMYK 色彩空间，通过给其他两个组件提供零值。例子 4.18 显示了另一个双色调色彩空间的定义，这次使用黑色和金色着色剂（金色是个着色点），并使用 CaIRGB 空间作为备用的色彩空间。这些可以以相同的方法定义，就如同之前的例子一样，包括色彩转换功能，此功能将两个色彩组件转换为可选择 CaIRGB 色彩空间里的色彩。

例子 4.17

```
10 0 obj % Color space
[ /Indexed
[ /DeviceN
[/Cyan /Black]
```

```
/DeviceCMYK
15 0 R
]
255
<6605 6806 6907 6B09 6C0A ...>
]
endobj
15 0 obj % Tint transformation function
<< /FunctionType 4
/Domain [0.0 1.0 0.0 1.0]
/Range [0.0 1.0 0.0 1.0 0.0 1.0 0.0 1.0]
/Length 16
>>
stream
{0 0 3 -1 roll}
endstream
endobj
```

例子 4.18

```
30 0 obj % Color space
[ /Indexed
[ /DeviceN
[/Black /Gold]
[ /CalRGB
<< /WhitePoint [1.0 1.0 1.0]
/Gamma [2.2 2.2 2.2]
>>
]
35 0 R % Tint transformation function
]
255
```

…Lookup table…

]

Endobj

考虑到转换任何黑色和金色的组合色为校准的 RGB 的准则，一个 2 输入，3 输出标准 4（附加计算）功能应该被用作色彩变换中。或者，一个 0 类型（采样）功能应该被采纳，但是这样将需要大量的采样点才能准确代表这些功能；例如，在 256 个 0.0 和 0.1 之间的色彩值中采样每个不同输入将需要 $256^2 = 65,536$ 个样本。但自从 DeviceN 色彩空间被用作索引色空间的基地，实际上只有 256 种黑色和金色色调值重组的可能。代表这种信息更紧凑的方式是将替代色的值直接放入 DeviceN 色彩值旁边的查找表中，如例 4.19。

例 4.19

```
10 0 obj % Color space
[ /Indexed
[ /DeviceN
[/Black /Gold /None /None /None]
[ /CaIRGB
<< /WhitePoint [1.0 1.0 1.0]
/Gamma [2.2 2.2 2.2]
>>
]
20 0 R % Tint transformation function
]
255
…Lookup table…
]
Endobj
```

在这个例子中，查找表中的每个条目有 5 个组成部分：2 个黑色和金色着色剂以及 3 个以上（如无指定）等效的 CaIRGB 色彩组件分量。如果在输出设备，黑色

和金色着色剂是现存的，没有组件将被忽略；如果没有黑色和金色可用，色彩转换功能将被用作转换 5 分量色彩为 3 分量等同的候补 CaiRGB 色彩空间。但因为，通过组建，第三，第四，第五组件是 CaiRGB 元件，色彩转换功能可以只丢弃前两个组件并返回后三个组件。这可以很容易地表述为类型 4 功能（附加计算器），如例 4.20。

例子 4.20

```
20 0 obj % Tint transformation function
<< /FunctionType 4
/Domain [0.0 1.0 0.0 1.0 0.0 1.0 0.0 1.0 0.0 1.0]
/Range [0.0 1.0 0.0 1.0 0.0 1.0]
/Length 27
>>
stream
{5 3 roll pop pop}
endstream
endobj
```

例子 4.21 利用了上述技术的延伸，用来产生四色泽（4 组件部分）图像，如图板块 7 所显示的。

例子 4.21

```
5 0 obj % Image XObject
<< /Type /XObject
/Subtype /Image
/Width 288
/Height 288
/ColorSpace 10 0 R
/BitsPerComponent 8
/Length 105278
/Filter /ASCII85Decode
```

```
>>  
  
stream  
...Data for grayscale image...  
endstream  
endobj  
10 0 obj % Indexed color space for image  
[ /Indexed  
15 0 R % Base color space  
255 % Table has 256 entries  
30 0 R % Lookup table  
]  
Endobj
```

```
15 0 obj % Base color space (DeviceN) for Indexed space  
[ /DeviceN  
[ /Black % Four colorants (black plus three spot colors)  
/PANTONE#20216#20CVC  
/PANTONE#20409#20CVC  
/PANTONE#202985#20CVC  
/None % Three components for alternate space  
/None  
/None  
]  
16 0 R % Alternate color space  
20 0 R % Tint transformation function  
]  
Endobj
```

```
16 0 obj % Alternate color space for DeviceN space  
[ /CaIRGB  
<< /WhitePoint [1.0 1.0 1.0] >>
```

]

Endobj

```
20 0 obj % Tint transformation function for DeviceN space
<< /FunctionType 4
/Domain [0.0 1.0 0.0 1.0 0.0 1.0 0.0 1.0 0.0 1.0 0.0 1.0 0.0 1.0 0.0 1.0]
/Range [0.0 1.0 0.0 1.0 0.0 1.0]
/Length 44
>>
stream
{ 7 3 roll % Just discard first four values
pop pop pop pop
}
endstream
endobj
```

```
30 0 obj % Lookup table for Indexed color space
<< /Length 1975
/Filter [/ASCII85Decode /FlateDecode]
>>
stream
8;T1BB2"!M7*!"psYBt1k\gY1T<D&t0]r*F7Hga*
…Additional data ( seven components for each table entry)…
endstream
endobj
```

正如在前面的例子，一个基于 DeviceN 空间上的索引色彩空间被用作绘制灰度级的图像，灰度级的图像如左侧板所示：黑色以及三个潘通专色。备用色彩空间是一个简单的校准的 RGB。因此，DeviceN 色彩空间有 7 个部分组成：四着色剂再加上所需的备用空间的三个组件。例子说明了 XObject 图像（见第 4.8.4 章节，“图像词典”）代表了四色泽图像，四色泽图像用来解释随后的色彩空间图像数

据（见附录 H 附加说明 48）。

4.5.6、套印控制

图形状态包含套印参数，由在图形状态参数字典内的 `OP` 和 `OPM` 入口控制。套印控制在产生物理分离的设备上是有用的，但是也可以用于一些合成设备。虽然这个参数的作用是取决于设备的，在这里描述而不是渲染，因为它与设备色彩空间的绘画方面有关，因此对于许多应用程序很重要。

任何绘画行为都与一些特定的设备着色剂有关，取决于发生绘图行为的彩色空间。在分离的或 `DeviceN` 色彩空间，被标示的着色剂应被明确地指定；在一个设备或者 `CIE` 的色彩空间，他们被输出设备的过程色彩模型所隐含（见第 6 章）。该套印参数是一个布尔标志，此标志决定了绘画操作是如何影响着色剂而不是这些被目前色彩空间所明确或含蓄地指定的东西。

如果套印参数为 `false`（默认值），在任何色彩空间绘制颜色的行为会导致相应的不明色素区域的删除。效果是在页面上任何位置被绘制产生的颜色，这些颜色与不透明的图像模型的一般绘制行为保持一致。

如果套印参数为 `true`，输出设备支持叠印，将不会出现这种擦除行为；任何之前的在其他着色剂中的绘制会原封不动的被保留。因此，在一个页面上的指定位置的颜色可能是几个不同着色剂绘制操作的综合结果。这种叠加产生的效果是依靠设备的以及在 PDF 语言中没有被定义。

注意：并非所有的设备都支持叠印。此外，许多 PostScript 输出机制仅支持正在分离并且不是复合式的输出。如果不支持叠印，套印的参数值将被忽略。

另外一个图形状态参数，套印模式（PDF 格式 1.3），当叠印使能时，将会影响在 `DeviceCMYK` 色彩空间里色彩值为 0.0 的色彩组成。这个参数是被图形状态参数字典内的 `OPM` 入口控制；当只有套印参数为真时，它才能起效果，如上所述。

当色彩被指定为 `DeviceCMYK` 色彩空间并且输出设备的原始色彩空间也是

`DeviceCMYK`, 每个资源色组件直接控制着相应的着色设备。一般来说, 每个原始色彩组件值代替了之前为了相应设备的着色值, 无论新的值是多少; 这是默认的行为, 由套印模式 0 指定。

当套印模式是 1 (也称为非零叠印模式), 一个 0.0 的色彩值离开了相应的之前绘制出的色彩组件。其效果相当于在画一个 `DeviceN` 色彩空间, `DeviceN` 色彩空间仅包括值不为 0 的组件。例如, 如果叠印参数为真并且叠印的模式是 1, 则

`0.2 0.3 0.0 1.0 k`

相当于

`0.2 0.3 1.0 scn`

在色彩空间如同例子 4.22

例子 4.22

```
10 0 obj % Color space
```

```
[ /DeviceN
```

```
[/Cyan /Magenta /Black]
```

```
/DeviceCMYK
```

```
15 0 R
```

```
]
```

```
Endobj
```

```
15 0 obj % Tint transformation function
```

```
<< /FunctionType 4
```

```
/Domain [0.0 1.0 0.0 1.0 0.0 1.0]
```

```
/Range [0.0 1.0 0.0 1.0 0.0 1.0 0.0 1.0]
```

```
/Length 13
```

```
>>
```

```
stream
```

```
{0 exch}
```

```
endstream
```

```
endobj
```

非零叠印模式只适用于当前色彩空间为 `DeviceCMYK` 在图形状态下利用当前色彩的绘制操作（或是隐式转换为 `DeviceCMYK`；见“CIE-Based 色彩空间的隐式转换”第 228 页）。它并不适用于图像的绘制或者任何计算出来的色彩，比如底纹图案或来自其他色彩空间的转换。它也不适用于设备的原始色彩空间不是 `DeviceCMYK`，在这种情况下，原色彩必须转换成设备的原始色彩空间，并且所有组件参与这种转换，无论他们的值如何。（这明确地表述了可选色彩空间和例子 4.22 中 `DeviceN` 色彩空间的色彩转换功能。）

参见 7.6.3 节，“叠印和透明度”，进一步讨论在透明成像模型中叠印的作用。

4.5.7、颜色算子

表 4.24 列出了 PDF 中控制颜色空间和颜色值的参数。（与色彩相关的是图像状态参数 `ri`，在 189 页的表 4.7 中列出，相关讨论在 230 页的“色域转换的模式”。）色彩参数可能会出现在页面描述层面或者文本对象内部（参见图 4.1 第 167 面）。

表 4.24 色彩参数

对象	参数	描述
名字	<code>CS</code>	(PDF1.1) 设置当前色彩空间。操作数的名称必须是一个对象的名称。如果颜色空间是一个可以指定一个名称，也没有额外的参数 (<code>DeviceGray</code> , <code>DeviceRGB</code> , <code>DeviceCMYK</code> ，与模式某些情况下)，这样名称可以直接指定。否则，它必须在当前资源字典中的色彩空间子字典里被定义名字（见 3.7.2 节，“资源字典”）；关联的值是一个描述色彩空间的数组（见 4.5.2 章节，“色彩空间家族”）。

注意：名称为 `DeviceGray`, `DeviceRGB`, `DeviceCMYK` 及一些模式往往直接确定相应的色彩空间；它们从不在色彩空间子字典里涉及到资源。

参数 `CS` 还设置它的初始值，初始值取决于色彩空间：
• 在 `DeviceGray`, `DeviceRGB`, `CalGray`, 或 `CalRGB` 色彩空间，最初的色彩拥有所有等于 0.0 的组件。

• 在 DeviceCMYK 色彩空间，最初的颜色是 [0.0 0.0 0.0 1.0]。

• 在实验室或 ICCBased 色彩空间，最初的颜色等于 0.0，除非超出了空间的指定时间间隔的距离，在这种情况下，最近且有效的值将被替换。

• 在索引色彩空间，最初的颜色值是 0。

• 在分离的或 DeviceN 色彩空间，对于所有着色剂，色彩的初始值是 1.0。

对象	参数	描述
名字	CS	• 在色彩空间模式，初始颜色是一个模式对象，导致了白板的出现。
名字	cs	(PDF1.1) 跟 CS 一样但利用非 stroking 技术。
C1…Cn	SC	(PDF1.1) 在设备上颜色设置为使用 stroking, CIE-based (除 ICCBased)，或收录的色彩空间。操作数的数量是必须的，以及它们的解释取决于当前 stroking 色彩空间：
		• 对于 DeviceGray, CalGray，色彩空间索引，需要一个操作数 (n=1)。
		• 对于 DeviceRGB, CalRGB，色彩空间，3 个操作数是必须的 (n=3)。
C1…Cn	SCN	• 对于 DeviceCMYK，需要 4 个操作数 (n=4)。 (PDF 1.2) 跟 SC 一样，但同样也支持模式，分离，DeviceN，和 ICCBased 色彩空间。

如果目前的行程是一个色彩空间分离，DeviceN，颜色或 ICCBased

颜色空间，操作数 C1…Cn 是数字。操作数的大小以及它们的解释取决于色彩空间。

如果目前的行程是一个色彩空间模式的色彩空间，名称是在当前资源字典内的子字典里德入口处（见 3.7.2 节，“资源词典”）。对于一个本色的平铺模式 (PatternType = 1 和 PaintType = 2)，c1…cn 是指定的元件值。对于其他类型，这些操作数必须不被指定。

C1…Cn sc (PDF 1.1) 跟 CS 一样但利用非 stroking 技术。

C1…Cn	scn	(PDF 1.2) 跟 SCN 一样但利用非 stroking 技术。
灰色	G	色彩空间设置为 DeviceGray (或 DefaultGray 色彩空间; 见“默认颜色空间”页 227) 并设置灰度操作。灰色是一个介于 0.0 (黑色) 和 1.0 (白色) 的颜色。
灰色	g	跟 G 一样但利用非 stroking 技术。
Rgb	RG	色彩空间设置为 DeviceRGB (或 DefaultRGB 色彩空间; 见“默认颜色空间”页 227) 和颜色设置为使用抚摸行动。每个操作数必须是介于 0.0 (最小密度), 1.0 (最大强度)。
Rgb	rg	跟 RG 一样但利用非 stroking 技术。
Cmyk	K	色彩空间设置为 DeviceCMYK (或 DefaultCMYK 色彩空间; 见“默认颜色空间”页 227) 和颜色设置为 stroking。每个操作数必须是介于 0.0 数 (零浓度) 和 1.0 (最高浓度)。这个操作行为被套印模式影响 (见第 4.5.6, “套印控制”)。
Cmyk	k	跟 K 一样但利用非 stroking 技术。

调用参数如特定颜色或其他颜色相关的参数在图形状态某些情况下是限制的。此限制下定义时, 其颜色会发生将被分别指定每次使用它们的图形数据。具体来说, 限制适用于这些情况:

- 在任何字形描述, 使用 d1 操作数 (见 5.5.4 节, “类型 3 字体”)
- 在一个无色的窗口配置模式 (见“无色的窗口配置模式” 268 页)

在这种情况下, 采取以下行动导致一个错误:

- 调用下面的操作之一:

CS	scn	K
cs	G	k
SC	g	ri
SCN	RG	sh
sc	rg	

- 调用与该图形下面的任何操作状态参数字典:

TR	BG	UCR
TR2	BG2	UCR2
HT		

- 绘制一幅图像。然而, 绘画图像掩码 (见“模具掩蔽”页 320) 是允许的, 因

为它没有指定颜色的，相反，它指定当前绘制颜色。

4.6、模式

用当前色彩绘制一个区域，他们通常适用于一个单一的颜色，面积也要一致。它也可以适用“绘制”说的一个重复的数字或图形，而不是顺利组成一个简单的渐变颜色。这种重复的数字或平滑梯度称为一种模式。模式是相当普遍，并有许多用途，例如，它们可用于创建各种图形纹理，如织，砖墙，太阳，以及类似几何和色效果。（见执行情况的说明在附录 H 49）

有两种不同的模式：

- 平铺模式包括一个小的图形数字模式，（称为模式细胞），是在固定的水平和垂直间隔复制，以填补绘制区域。图形对象使用平铺流内容描述。
- 底纹图案定义一个渐变填充，产生平稳过渡在整个区域之间的颜色。颜色使用被指定为一个函数位置使用了各种方法的任何。

注：画有图案的能力是一个功能的 PDF 1.2（平铺模式）和 PDF 格式 1.3（着色模式）。随着一些努力，就可以实现有限形式平铺的 PDF 1.1 模式定义为字符字形他们在特殊的字体与绘画和经营他们一再对 TJ。另一种方法，确定作为半色调屏幕模式，是不建议的，因为生产的影响依赖于设备的。

模式是在一个特定的颜色命名空间定局特殊的组合。这些空间利用，而不是颜色值相当于模式对象的数字元件值与其他空间使用。一个模式对象可能是一个字典或流，这取决于模式类型；术语词典一般用来模式在本节中提及任何字典对象或字典部分流对象。（这些模式对象的流具体确定为这种模式在特定类型的描述，除非另有注明外，他们理解为，而不是简单的字典。）本部分介绍了图案颜色空间和色彩的价值规范内。参见第 4.5 节，“颜色空间”，有关的色彩空间和色彩信息一般条和第 7.5.6 价值观“模式和透明度”，为进一步讨论透明成像模型。

4.6.1、一般模式的性质

一个模式字典包含描述性的信息定义外观与一个模式的属性。所有模式字典

包含一个模式类型，其值标识图案的字典描述：模式 1 为平铺图案或者模式 2 为底纹图案。其余的内容字典的依赖型模式，并在以下各节详细个别模式类型。

所有的模式都被视为色彩；一个模式色彩空间被 CS 或 cs 参数确定，就像其他色彩空间那样，一个特定的模式被当前色彩中的 SCN 或者 scn 参数所装载（见表 4.24 257 面）。

模式描述为关于其自身的内部协调系统。每个模式有一个模式矩阵，转换矩阵，映射模式及内部协调体系，以默认的坐标系统为模式的内容流（流中的内容定义为模式的资源）。此次模式矩阵级联流模式建立了协调的空间，在其中所有的图形对象中建立了模式解释。

例如，如果一个模式是在页面上使用的模式出现在该页面模式 subdictionary 资源字典内，地图和模式矩阵模式空间的默认（初始）协调本页面空间。页面的变化成转换矩阵，在网页的内容流，出现如旋转与缩放，不会对格局的影响，它保持了原有的关系网页上的任何地方使用它一页。同样，如果一个模式是在 form 中使用了 XObject（见 4.9 节，“表格 XObjects”），该模式矩阵地图模式空间到窗体的默认用户空间（即表格坐标在形式与经营者做画的时间空间）。甲模式可用于在另一个模式；内部格局的矩阵定义其关系到外花纹图案的空间。

注：PostScript 允许一种模式被定义在一个页面，但在另一个下使用。举例来说，这种模式可能会被定义在页面上（即它的模式矩阵图该模式坐标空间到用户空间的页面），但在一个窗体上使用该页面，使页面的关系是每个人独立安置的形式。PDF 格式不支持此功能，在 PDF 格式，所有模式在本地定义它们的范围。

4.6.2、平铺模式

平铺模式由一个小型的图形称为模式图子细胞。画与在固定的模式复制水平和垂直间隔，以填补一个地区。其效果是，如果这个数字是：在一个明确的玻璃表面涂瓦，其中规定了当时覆盖面积阵列，而相同的副本并精简其边界。这一过程被称为瓦片该地区。

该模式细胞可以包括图形元素，如填充区域，文本和采样图像。它的形状不是长方形，与瓷砖间距从细胞自身的尺寸来说可以不同。当执行操作，例如画为

S (中风) 或 F (填充)，应用程序涂料对当前的一页，许多细胞必要时填写的区域。该命令在个别瓷砖（实例细胞）的画是不确定和不可预测的，它是为数字不宜在邻近的瓦片重叠。

该模式的细胞形态的定义是一个包含内容的流画商需要油漆一单元格的实例。除了通常的条目共同所有流（见表 3.4 第 38 页），此流的字典在表 4.25 中列出的其他条目。

表 4.25 附加的 1 型模式字典入口

关键字	类型	值
类型	名字	(可选) PDF 类型的描述，正如这字典介绍的；如果存在，必须是一个模式字典模式。
类型模式	整数	(必须的) 代码识别字典所描述的模式类型；必须使平铺模式为 1。
绘制模式	整数	(必需的) 代码来确定单元格颜色模式是如何确定的：1，彩色模式。该模式的内容流指定的颜色用于绘制模式细胞。当内容流开始执行，当前的颜色是一个实际上最初在模式的内容流。（这是类似的定义该模式矩阵，见 4.6.1 节，“一般性质的模式。”） 2，未着色平铺模式。该模式的内容流不指定任何颜色的信息。相反，整个细胞，画上图案一个单独指定的颜色，每次使用的模式。从本质上讲，内容流通过它描述了一个模具当前颜色要倒。内容流必须不以经营者指定颜色或其他颜色相关的图形参数状态；否则，一个错误发生（见第 4.5.7，“颜色算”）。该内容流可能油漆图像面具，但是，它不指定任何色彩信息（见 320 页上的模具遮蔽）。
TilingType	整数	(必需的) 代码控制调整间距相对于该设备的像素网格： 1，常数间距。细胞的间距始终模式，也就是说，由一个多一个设备像素。要做到这一点，应用程序可能需要歪曲模式通过使细胞稍微小调整特步，YStep 和变换矩阵。失真的距离不超过 1 设备像素。 2，不失真。该模式细胞不被扭曲，但间距细胞之间的格局可能会有所不同多达 1 设备像素，无论水平方向和垂直方向，在模式是画。因此可以达到由特步和平均水平，但不一定 YStep 要求间距为每个模式细胞。

3, 恒间距。细胞的间距始终模式但允许一个额外的失真，使 1 型更有效的执行。

Bbox 矩形 (必须的) 四个数字中的坐标系统给予阵列模式坐标左部, 底部, 右侧, 顶部和边缘, 分别是格局细胞的边界框。这些边界是用来定位模式细胞

Xstep 数字 (必须的) 所需的细胞之间的测量空间, 在模式坐标系统下测量。

Ystep 数字 (必须的) 所需的细胞之间的测量模式, 由模式坐标系统测量。请注意, XStep 和 YStep 不同的尺寸由 BBox 暗地的方式进入细胞。这使得产生不规则形状的数字。XStep 和 YStep 可以是正的或负的, 但不是零。

关键字	类型	值
Resources	字典	(必须的) 一个资源字典, 包含所有已命名的资源即该模式所要求的内容流 (见 3.7.2 节, “资源词典”)。
矩阵	数组	(可选) 指定的 6 个数字阵列模式矩阵 (见第 4.6.1, “一般性质的模式”)。默认值: 单位矩阵 [1 0 0 1 0 0]。

该模式字典为 BBox, XStep, YStep 值被模式坐标系统解释, 并在该模式下内容流的图形对象是指有关该坐标系统的。细胞的安置模式在平铺的基础是一个重要的模式, 然后由 XStep 和 YStep 倍数流离失来复制模式的位置。该模式的主要细胞来源坐标系统。该阶段的瓦片可以控制由矩阵的模式中字典项产生的翻译内容。

在绘制平铺模式的第一步是建立模式流为彩色图形状态。随后的绘制行为伴随着模式的内容流所描述的模式细胞。要获取模式细胞, 应用程序执行这些步骤:

1. 保存当前的图形状态 (如如果引用的 Q 算子)
2. 安装图形状态实际上是在模式的开头改变矩阵的内容流于当前的状态, 矩阵模式在 4.6.1 节所述, “一般模式的性质”
3. 图形对象中指定模式的内容流
4. 恢复已保存的图形状态 (如如果引用的 Q 算子)

注: 该模式的内容流, 在图形状态参数不应设置任何设备的依赖 (见表 4.3 页 182), 因为它可能会导致不正确的输出。

彩色窗口模式

一个窗口图案是一个模式，其颜色是独立的。当然，在绘画的模式细胞，该模式的内容流明确地设置颜色每个图形的元素点。单一模式单元格可以包含的元素画的颜色是不同的，它也可以包含采样灰度或彩色图像。这种模式的类型，是由模式 1 类型和绘制类型为 1 的鉴定字典模式。

如果当前的色彩空间是一个空间模式，彩色模式可以平铺作为当前的颜色。此名称必须是在当前资源字典模式的子字典关键入口（见 3.7.2 节，“资源词典”），其值是对象值，表示该模式流。由于该模式定义了自己的颜色信息，没有代表颜色组件的其他操作数指定的 SCN 或者 scn。例如，如果 P1 是一个名字在当前资源型词典的资源，下面的代码建立了它作为当前 nonstroking 颜色：

```
/Pattern cs  
/P1 scn
```

随后执行 nonstroking 绘制操作，例如 F（填充），Tj（显示文本），或做一个图像遮掩（绘制外部对象），使用指定的模式去标志待绘制的区域。

示例 4.23 定义了一个页面（对象 5）在一个黄色的背景下利用彩色绘制模式绘制了 3 个圈和一个三角形（对象 15）。该模式包括四组扑克牌（黑桃，红心，钻石和俱乐部），这是从 ZAPFDINGBATS 排序中采取的字符字体（见第草 4，“ZAPFDINGBATS 排序集与编码”），该模式的内容流指定每一个字形的颜色。第 8 板显示结果。

例子 4.23

```
5 0 obj % Page object  
<< /Type /Page  
/Parent 2 0 R  
/Resources 10 0 R  
/Contents 30 0 R  
/CropBox [0 0 225 225]  
>>  
endobj
```

```
10 0 obj % Resource dictionary for page
<< /Pattern << /P1 15 0 R >>
>>
Endobj
15 0 obj % Pattern definition
<< /Type /Pattern
/PatternType 1 % Tiling pattern
/PaintType 1 % Colored
/TilingType 2
/BBox [0 0 100 100]
/XStep 100
/YStep 100
/Resources 16 0 R
/Matrix [0.4 0.0 0.0 0.4 0.0 0.0]
/Length 183
>>
stream
BT % Begin text object
/F1 1 Tf % Set text font and size
64 0 0 64 7.1771 2.4414 Tm % Set text matrix
0 Tc % Set character spacing
0 Tw % Set word spacing
1.0 0.0 0.0 rg % Set nonstroking color to red
(\001) Tj % Show spade glyph
0.7478 -0.007 TD % Move text position
0.0 1.0 0.0 rg % Set nonstroking color to green
(\002) Tj % Show heart glyph
-0.7323 0.7813 TD % Move text position
0.0 0.0 1.0 rg % Set nonstroking color to blue
(\003) Tj % Show diamond glyph
0.6913 0.007 TD % Move text position
```

```
0.0 0.0 0.0 rg % Set nonstroking color to black
(\004) Tj % Show club glyph
ET % End text object
endstream
endobj
16 0 obj % Resource dictionary for pattern
<< /Font << /F1 20 0 R >>
>>
Endobj
20 0 obj % Font for pattern
<< /Type /Font
/Subtype /Type1
/Encoding 21 0 R
/BaseFont /ZapfDingbats
>>
Endobj
21 0 obj % Font encoding
<< /Type /Encoding
/Differences [1 /a109 /a110 /a111 /a112]
>>
Endobj
30 0 obj % Contents of page
<< /Length 1252 >>
stream
0.0 G % Set stroking color to black
1.0 1.0 0.0 rg % Set nonstroking color to yellow
25 175 175 - 150 re % Construct rectangular path
f % Fill path
/Pattern cs % Set pattern color space
/P1 scn % Set pattern as nonstroking color
99.92 49.92 m % Start new path
```

```
99.92 77.52 77.52 99.92 49.92 99.92 c % Construct lower-left circle
22.32 99.92 -0.08 77.52 -0.08 49.92 c
-0.08 22.32 22.32 -0.08 49.92 -0.08 c
77.52 -0.08 99.92 22.32 99.92 49.92 c
B % Fill and stroke path
224.96 49.92 m % Start new path
224.96 77.52 202.56 99.92 174.96 99.92 c % Construct lower-right circle
147.36 99.92 124.96 77.52 124.96 49.92 c
124.96 22.32 147.36 -0.08 174.96 -0.08 c
202.56 -0.08 224.96 22.32 224.96 49.92 c
B % Fill and stroke path
87.56 201.70 m % Start new path
63.66 187.90 55.46 157.32 69.26 133.40 c % Construct upper circle
83.06 109.50 113.66 101.30 137.56 115.10 c
161.46 128.90 169.66 159.50 155.86 183.40 c
142.06 207.30 111.46 215.50 87.56 201.70 c
B % Fill and stroke path
50 50 m % Start new path
175 50 l % Construct triangular path
112.5 158.253 l
b % Close, fill, and stroke path
endstream
endobj
```

例 4.23 几个特点值得注意：

- 这三个圈和三角形的绘制是利用相同的模式。该模式下相点是对齐的，即使圆和三角形在图案的单元格是不尊重对齐。举例来说，蓝色的钻石位置相对于 3 园各不相同。
- 该模式细胞并不完全覆盖：它叶片间的空间字形上漆。当平铺模式作为一种颜色，现有的背景（黄色矩形）显示通过这些上漆的地方。

本色平铺模式

一个本色的平铺模式是一种固有的模式，没有颜色：颜色必须指定单独使用时的模式。它提供了一种不同的方式与模式，具有相同的状态，但不同的页颜色。这种类型的模式是由 1 型涂料类型确定模式 2 字典。该模式的内容流不明确指定任何颜色，它可以绘制图像掩码（见“模具掩蔽”320 页），但没有其他类型的图像。

一个模式的色彩空间较本色平铺模式需要一个参数：确定基本颜色空间中的实际颜色，该模式是被指定的。基本的颜色空间，给出的第二个元素的数组定义图案颜色空间。例如，数组

[/Pattern /DeviceRGB]

定义了 DeviceRGB 一个图案颜色空间作为其基本的色彩空间。

注：基础色彩空间不能是另一个模式的色彩空间。

在这样一个操作数提供的色彩空间的 SCN 或 scn 必须在基本的色彩空间里的包括色彩值，指定一个或多个数字颜色分量，以及作为一个模式对象的名称较本色平铺模式。例如，如果当前的资源字典（见第 3.7.2，“资源词典”）CS3 的定义为一个色彩空间资源，其值是以上模式的色彩空间作为一种范式的资源名称显示的 P2 指在一个本色的平铺模式，代码

/Cs3 cs

0.30 0.75 0.21 /P2 scn

CS3 的确定为当前 nonstroking 色彩空间和色彩的 P2，将在由指定的组件所代表的颜色绘制在 DeviceRGB 色彩空间。随后执行 nonstroking 绘制，例如 F(填充)，Tj (显示文本)，并完成图像遮掩（绘制外部对象），使用指定的图案和绘制好的色彩平铺的领域。该相同的模式每次可通过不同的颜色重复使用。

265 页的例子 4.24 和例子 4.23 是类似的，除了它使用一个本色平铺图案绘制三个圈，三角形，每个都使用不同的颜色（见片 9）。为此，它每次提供 4 操作数的 scn：3 个数字表明在基础的 DeviceRGB 色彩空间下的色彩组件伴随着该模式的名称。

例子 4.24

```
5 0 obj % Page object
<< /Type /Page
/Parent 2 0 R
/Resources 10 0 R
/Contents 30 0 R
/CropBox [0 0 225 225]
>>
endobj

10 0 obj % Resource dictionary for page
<< /ColorSpace << /Cs12 12 0 R >>
/Pattern << /P1 15 0 R >>
>>
endobj

12 0 obj % Color space
[/Pattern /DeviceRGB]

Endobj

15 0 obj % Pattern definition
<< /Type /Pattern
/PatternType 1 % Tiling pattern
/PaintType 2 % Uncolored
/TilingType 2
/BBox [0 0 100 100]
/XStep 100
/YStep 100
/Resources 16 0 R
/Matrix [0.4 0.0 0.0 0.4 0.0 0.0]
/Length 127
>>
Stream

BT % Begin text object
/F1 1 Tf % Set text font and size
```

```
64 0 0 64 7.1771 2.4414 Tm % Set text matrix
0 Tc % Set character spacing
0 Tw % Set word spacing
(\001) Tj % Show spade glyph
0.7478 - 0.007 TD % Move text position
(\002) Tj % Show heart glyph
- 0.7323 0.7813 TD % Move text position
(\003) Tj % Show diamond glyph
0.6913 0.007 TD % Move text position
(\004) Tj % Show club glyph
ET % End text object
endstream
endobj
16 0 obj % Resource dictionary for pattern
<< /Font << /F1 20 0 R >>
>>
endobj
20 0 obj % Font for pattern
<< /Type /Font
/Subtype /Type1
-Encoding 21 0 R
/BaseFont /ZapfDingbats
>>
Endobj
21 0 obj % Font encoding
<< /Type /Encoding
/Differences [1 /a109 /a110 /a111 /a112]
>>
endobj
30 0 obj % Contents of page
<< /Length 1316 >>
```

stream

0.0 G % Set stroking color to black
1.0 1.0 0.0 rg % Set nonstroking color to yellow
25 175 175 - 150 re % Construct rectangular path
f % Fill path
/Cs12 cs % Set pattern color space
0.77 0.20 0.00 /P1 scn % Set nonstroking color and pattern
99.92 49.92 m % Start new path
99.92 77.52 77.52 99.92 49.92 99.92 c % Construct lower-left circle
22.32 99.92 - 0.08 77.52 - 0.08 49.92 c
- 0.08 22.32 22.32 - 0.08 49.92 - 0.08 c
77.52 - 0.08 99.92 22.32 99.92 49.92 c
B % Fill and stroke path
0.2 0.8 0.4 /P1 scn % Change nonstroking color
224.96 49.92 m % Start new path
224.96 77.52 202.56 99.92 174.96 99.92 c % Construct lower-right circle
147.36 99.92 124.96 77.52 124.96 49.92 c
124.96 22.32 147.36 -0.08 174.96 - 0.08 c
202.56 - 0.08 224.96 22.32 224.96 49.92 c
B % Fill and stroke path
0.3 0.7 1.0 /P1 scn % Change nonstroking color
87.56 201.70 m % Start new path
63.66 187.90 55.46 157.30 69.26 133.40 c % Construct upper circle
83.06 109.50 113.66 101.30 137.56 115.10 c
161.46 128.90 169.66 159.50 155.86 183.40 c
142.06 207.30 111.46 215.50 87.56 201.70 c
B % Fill and stroke path
0.5 0.2 1.0 /P1 scn % Change nonstroking color
50 50 m % Start new path
175 50 l % Construct triangular path
112.5 158.253 l

```
b % Close, fill, and stroke path
endstream
endobj
```

4.6.3、底纹图案

底纹图案 (PDF 格式 1.3) 提供不同绘制区域色彩的跨颜色之间的平稳过渡，在任何未经指定的特定输出设备做色彩过渡步骤。图谱由模式字典描述为 2 型这种类型的模式。表 4.26 表明这种类型的词典内容。

表 4.26 类型 2 的字典模式入口

关键字	类型	值
类型	名字	(可选) 这字典介绍的 PDF 类型对象，如果存在，则必须是一个字典模式。
类型模式	整数	(必须的) 代码识别的模式，正如这本字典类型描述；必须是 2 的底纹图案。
底纹	字典或流	(必需的) 着色对象 (见下文) 定义底纹图案的梯度填充。字典内容组成了表 4.28 的入口，它们在表 4.29 至表 4.23 中。
矩阵	数组	(可选) 指定的 6 个数字阵列模式矩阵 (见第 4.6.1，“一般性质的模式”)。默认值：单位矩阵 [1 0 0 1 0 0]。
ExtGState	字典	(可选) 图形状态参数词典 (见第 4.3.4 节，“图形状态参数词典”) 包含图形状态参数，当底纹图案绘制时图形状态参数才有效。从图形说明来看任何参数是不是指定的继承，实际上是内容流在该模式下定义的开始资源。

最重要的入口是底纹，其值是一个渐变的对象，该对象定义了底纹在底纹图

案的渐变填充的属性。这是一个复杂的“绘制”，决定了色彩过渡类型的底纹图案在绘制时产生在区域。一个底纹对象可能是一本字典或流，这取决于遮掩的类型；术语遮掩词典一般用来遮掩整个本节中提及任何字典对象或一本字典的一部分流对象。（那些阴影对象的具体确定在特定类型的阴影，除非另有说明，他们被理解为简单的字典。）

底纹算子

当绘制的区域是一个相对简单的几何形状，该几何的形状与作为渐变填充本身一致，sh 参数可以用来代替通常的绘制参数。sh 接受渐变字典作为操作数，并直接为当前用户空间请求相应的渐变填充。这算子不要求模式字典的建立以及工作在图像状态里没有当前色彩值的情况下。表 4.27 描述了 sh 算子。

注：被字典模式 2 定义的模式没有列出。要创建一个包含一个渐变填充的瓦片模式，在类型 1（平铺）模式下调用 sh 操作数。

表 4.27 底纹算子

操作数	算子	描述
名字	sh	(PDF 格式 1.3) 绘制的形状和色彩是被底纹字典所描述，受当前剪辑路径控制。在当前图形的颜色状态，既不使用也不改变。作用是从不同的绘画路径使用遮阳模式作为当前的颜色。
		名称是一种底纹子字典在字典资源的当前名称（见 3.7.2 节，“资源词典”）。所有坐标的渐变子词典解释是相对于当前的用户空间。（与此相反，当一个渐变词典用于模式 2，相应的应该在模式空间中予以描述。）各种颜色的色彩在色彩空间内予以解释，色彩空间被渐变字典的色彩空间入口予以确认（见表 4.28）。背景入口，如果存在，将被忽略。
		这个算子应该仅用于有界或者几何定义的阴影。如果应用到无限的阴影，它绘制所有渐变填充剪辑区域的阴影，这将会很费时。

底纹字典

底纹字典指定某一梯度的详细情况，包括将要用到的底纹类型，该区域的几何将要被重塑，几何图像在渐变填充。各种底纹类型可用，取决于字典的底纹类型入口：

- 基于函数的渐变（类型 1）在区域内定义每个点的颜色使用数学函数功能（不一定平稳或连续）。
- 轴向渐变（类型 2）定义一个两点间的混色线，通过继续在边界填充颜色来跨越了边界点。
- 放射状渐变（三型）定义两个圆圈之间的融合，可以选择扩展径向阴影作为超越边界的颜色，继续沿用边界色。这种渐变类型很常见地用于代表三位领域和锥形体。
- 自由形式的高氏着色三角网格（类型 4）定义一个共同的结构，这种结构被许多三维应用程序中使用，用来代表复杂的色彩以及阴影形状。顶点在自由形式的几何中指定。
- 格形式高氏着色三角网格（类型 5）是基于相同的几何构架，这种几何构架和类型 4 相同，但指定为 `pseudorectangular` 格。
- 孔斯片三角网格（类型 6）建立一个或更多颜色底纹补丁，每个由 4 个三次 Bézier 曲线定边界。
- 张量积三角网格补丁（类型 7）跟类型 6 相似，但是在每个补丁有额外的控制点，从而更好地控制色彩映射。

表 4.28 显示，所有底纹字典的共享入口是相同的；特定的底纹类型入口将在接下来的章节中描述。

注：远期目标坐标空间，在许多以下描述中用到，指的是在其中一个坐标空间绘制阴影。对于使用阴影与类型 2 的模式字典，这是协调的空间格局，讨论于 4.6.1 节，“一般性质的模式。”对于底纹，直接利用 `sh` 算子，它是当前用户的空间。

表 4.28 底纹字典里共同的入口

关键字	类型	值
底纹类型	整形	(必须的) 底纹类型： 1, 具有某些功能的阴影 2, 轴阴影 3, 径向阴影 4, 自由格式高氏着色三角网格 5, 格形式高氏着色三角网格 6, 孔斯片三角网格 7, 张量积三角网格补丁
色彩空间	名字或者数组	(必须) 由色彩值表示的色彩空间。这可能是任何设备, CIE 为基础的, 或者除了一个模式空间的特殊色彩空间。见“色彩空间：特别注意事项” 276 页的更多资料。
背景	数组	(可选) 一个色彩组件的数组是专属于色彩空间的, 并指定一个背景颜色值。如果存在, 这种颜色被使用, 在任何阴影绘制操作之前, 以填补有待绘制阴影对象边界的部分区域。在不透明的成像模型, 效果犹如两次执行绘制操作: 第一次执行背景的颜色, 然后执行阴影。 注: 背景颜色是只适用于当阴影是用来作为底纹图案的一部分, 而不是通过 sh 算子直接绘制。
BBOX	矩形	(可选) 一个由四个数字阵列组成的数组提供了左, 下, 右, 和上的坐标, 分别是阴影的边界框。该坐标用来定位阴影目标坐标空间。如果存在, 当绘制阴影时这个包围盒的应用是作为临时剪辑边界, 同时剪辑路径和其他任何剪辑边界。
平滑	布尔	(可选) 一个标志位指示是否过滤底纹功能以便阻止运动模糊。

该底纹功能操作的速度决定于输出设备的分辨率。如果功能不平滑会会出现混叠，也就是说，是否它有一个相对高的空间频率采样率。反锯齿计算昂贵，通常是不必要的，因为大多数底纹功能能够顺畅或利用了足够高的采样频率，以避免混叠效应。反锯齿在一些输出设备上不可取，在这种情况下该标志将被忽略。默认值：0。

底纹类型 4 至 7 被定义为一个包含定性描述数据流阴影的渐变填充区。在这种情况下，底纹词典也是流字典，可以包含所有流的共同标准入口（见表 3.4 第 38 页）。特别是，它总是包含一个长度项，这是要求的是所有的流。

此外，一些阴影字典，也包括一个函数的入口，其价值是一个函数对象（字典或流）确定如何跨越不同阴影区域。在这种情况下，阴影词典通常定义几何的阴影，并在该函数定义色彩过渡几何。该函数对某些阴影类型来说是必需，对其他阴影是可选的。函数将在 3.9 节中详细描述，“函数。”

注：不连续的色彩过渡，或具有高空间频率，可能在低效率绘制时会出现混淆现象。

色彩空间：特别注意事项

从概念上讲，一个阴影决定了在绘制区域每个单独点的颜色值。然而在实践中，阴影实际上可能被用来只计算一些目标地区的一个子集的点，以及颜色值所确定的插值之间的中间点的颜色值的计算。消费者应用软件是免费使用的这种策略，只要对颜色值近似的范围内对这些阴影平滑段中指定的图形状态进行定义（见第 6.5.2 节，“光滑 容限”）。对于所有底纹字典使用的色彩空间通用入口不只定义的色彩空间中指定的阴影，也决定了色彩解析中的色彩空间。

注：有些底纹类型（4 至 7）提供一个参数值插补作为阴影的输入的色彩功能，所述的有关章节在下文介绍。这种插值形式跟这里所描述的在概念上有所不同，这对输出的颜色值的颜色产生功能发生在底纹的目标色彩空间。

渐变填充颜色之间大多数阴影定义是使用一各种插值算法，这些算法很敏感的颜色空间的特点。线性插值，例如，可能会有差异显时，比在 DeviceCMYK 色彩空间应用于色彩空间，即使开始和结束颜色感知相同。这些差异造成，因为两个色彩空间彼此间不是线性关系。底纹根据以下规则出现：

- 如果色彩是一个设备的色彩空间从本土色彩空间不同输出设备，在底纹颜色值转换为本土色彩空间使用标准的 6.2 节中描述的转换公式，“转换其中设备颜色空间。“为了优化性能，这些转换可在任何之前或之后的任何时间的颜色插值（地方在阴影值）。因此，阴影的色彩空间定义的设备可能有颜色渐变填充不那么准确，有些设备的依赖。（这并不适用于轴向和径向遮荫，遮荫 2 型和 3，因为这些类型的着色上进行渐变填充一个变量的计算，然后转换为参数的颜色。）
- 如果色彩是一个色彩空间的 CIE 库，所有的计算进行渐变填充在这个空间。颜色转换到设备后，才发生所有的计算了。因此，颜色梯度是取决于设备的。
- 如果色彩是一个分离或 DeviceN 色彩空间和指定的着色剂支持，没有颜色转换计算是必要的。如果指定的着色剂不支持（这样的空间的替代颜色空间必须使用），进行渐变填充计算在指定的分离 DeviceN 色彩空间之前或转换到备用空间。因此，非线性色彩转换功能被安置在尽可能具有代表性的阴影。
- 如果色彩是一个索引色彩空间，所有的颜色值中指定的阴影立即转换为基础的色彩空间。这取决于基础色彩空间是设备或智库的空间，梯度填充计算完成上述指令。插值从来没有发生在索引颜色空间，这是量子化的，因此需要承担不恰当的计算方法，连续的颜色范围。出于同样的原因，索引颜色空间不得以任何阴影的颜色值是由函数生成的；这一规则适用于任何阴影字典，其中包含一个函数的入口。

底纹类型

除了在表 4.28 所列的项目，所有底纹字典具体到他们所代表的底纹类型，以他们的底纹类型值表示。以下各节描述了可用的底纹类型且具体到每一个字典条目。

类型 1（功能型）底纹

在 1 型（功能型）阴影，在每个域中点颜色定义一个指定的数学函数。该函数不需要顺利或连续。这种类型是可用的阴影类型，是最普遍有用的，不能充分描述的阴影与任何其他类型。表 4.29 显示了特有的底纹这种类型的词典条目遮光，除了这些共同所有（表 4.28）底纹词典。

注：此类型的底纹不能当作一个索引色彩空间使用。

表 4.29 类型 1 底纹字典的附加特定项

关键字	类型	值
域名	数组	(可选) 一个由四个数字 [xmin xmax ymin ymax] 的数组指定矩形域坐标，该矩阵的颜色功能已被定义。 默认值：[0.0 1.0 0.0 1.0]。
矩阵	数组	(可选) 对 6 个数字的数组指定一个变换矩阵映射坐标空间进入底纹的目标域项中指定的坐标空间。例如，要绘制矩形域 [0.0 1.0 0.0 1.0] 为 1 英寸在左下角坐标 (100, 100) 的默认用户空间，矩阵的值将为 [72 0 0 72 100 100]。 默认值：确认矩阵 [1 0 0 1 0 0]。
功能	功能	(必须) 2 输入, n 输出功能或 2 输入 1 输出的矩阵（其中 n 是彩色底纹字典中的组件的彩色空间的数字）。每函数的域必须是一个字典的阴影。如果该值该函数返回一个给定的颜色成分是超出范围，它要调整到最近的有效值。

矩形域（域）建立了一个内部协调的空间底纹是独立的目标坐标空间中。颜色功能 (S) (功能) 指定在每个颜色的底纹点在这一领域的矩形。该变换矩阵 (矩阵)，则到相应的矩形或平行四边形矩形域映射目标坐标空间。在阴影的边框点 (BBox) 表示，外域的矩形转化为涂上阴影的背景颜色 (背景)，如果没有背景底纹字典项，这些点是无用的。如果函数未定义在任何时候矩形域范围内宣布，可能会出现错误，即使相应的转换点不属于阴影的边界框。

类型 2 (轴向) 底纹

类型 2 (轴向) 底纹定义一种混合颜色，沿直线轴之间两个差异端点并无限延长该垂直轴。该底纹可以随意延长至由一个或两个端点到颜色边界。表 4.30 显示了底纹词典条目针对这种类型的阴影，除了这些共同所有阴影字典 (表 4.28)。

注：此类型的底纹不能用一个索引色彩空间使用。

表 4.30 附加类型 2 底纹字典入口

关键字	类型	值
坐标	数组	(必须) 一个由四个数字数组 [x0 y0 x1 y1] 指定的起点和结束点坐标轴，在目标阴影表示其坐标空间。
域名	数组	(可选) 两个数字数组 [T0 T1 的] 指定一个限制参变量值的变量被认为是这些直线之间变化两个不同的颜色梯度值之间的起始点和结束点轴。变量 t 成为颜色功能 (S) 的输入参数。默认值：[0.0 1.0]。
功能	功能	(必需) 1 输入，n 输出功能或 1 个 1 输入，1 输出功能的矩阵（其中 n 是彩色底纹字典中的组件的彩色空间的数字）。该功能 (S) 与被称为在定义域参变量 t 值的域条目。每个函数的域必须是这种阴影字典。如果该函数返回一个给定的颜色值超出范围，则它要调整到最接近的有效值。
扩展	数组	(可选) 两个布尔值数组定义了是否扩展底纹功能至起始和结束轴点。默认值：[0 0]。

颜色混合线性映射是通过每个点 (x, y) 的沿端点之间的轴 (x0, y0) 和 (x1, y1) 间的一个点，由相应的阴影字典的域名项中指定其中的域。在点 (0, 0) 和 (1, 0) 对应的域名分别为 (x0, y0) 和 (x1, y1) 的所有点上沿域空间中的线垂直轴。因为从 (0, 0) (1, 0) 具有相同的颜色，只有新的 x 值需要计算：

$$x' = \frac{(x_1 - x_0) \times (x - x_0) + (y_1 - y_0) \times (y - y_0)}{(x_1 - x_0)^2 + (y_1 - y_0)^2}$$

参变量的 t 值确定 x 如下：

对于 $0 \leq x' \leq 1$, $t = t0 + (t1 - t0) * x'$.

对于 $x' < 0$, 如果扩展数组第一个元素为真, 则 $T = T0$ 的, 否则, t 是未定义, 点就为空白。

对于 $x' > 1$, 如果扩展数组的第二部分是真, 则 $T = T1$ 的, 否则, t 是未定义, 点就为空白。

t 的结果值传递给函数作为阴影中定义字典的输入 (s), 产生的颜色与元件值确定画点 (x, y)。

版 10 显示了一个例子, 使用轴向底纹填充一个矩形和显示文本。该地区应填补超越了底纹的覆盖。阴影是共有的, 只有 3 个背景值在底纹词典扩展项。在第一个例子, 阴影是没有两端延伸, 没有指定的背景颜色, 因此, 底纹是截断其覆盖的两端。第二个例子无背景色指定, 但底纹是两端延伸; 结果是填补了在定义的颜色填充的区域, 其余部分两端的阴影。在第三个例子, 阴影是两端延伸和背景色指定, 因此, 背景颜色是用来超越底纹的结束部分的填充区域。

类型 3 (径向) 底纹

类型 3 (径向) 底纹定义两个各不相同圆圈间颜色的混合。这种类型的底纹常用来描绘三维领域和视锥细胞。这种类型的底纹字典包含了表 4.31 所示的入口, 并且所有这些共同的底纹字典 (表 4.28)。

注: 此类型的底纹不能当作一个索引色彩空间使用。

表 4.31 附加类型 3 底纹字典入口

关键字	类型	值
-----	----	---

坐标	数组	(必须) 对 6 个数字的数组 [x0 y0 r0 x1 y1 r1] 指定了中心和半径界的开始和结束点, 用阴影目标表示坐标空间。半径 r0 和 r1 的必须同时大于或等于 0。如果一半径为 0, 相应的圆圈被视为一个点, 如果两个都是 0, 什么都没有。
域名	数组	(可选) 两个数字数组 [t0 t1] 指定一个限制参变量的值, 该变量被认为是这些直线之间变化两个不同的颜色梯度值之间的开始和结束点。该变量 t 成为颜色功能 (S) 的输入参数。默认值: [0.0 1.0]。
功能	功能	(必需) 1 个 1 输入, n 输出函数或 1 个的 1 输入, 1 输出函数 (其中 n 是彩色底纹字典组件的彩色空间数字)。该功能 (S) 被称为由定义域参变量 t 值决定的底纹字典的网点。每个函数的域必须是一对阴影词典的超集。如果该函数的返回值对于一个给定的颜色成分超出范围, 则它要调整到最接近的有效值。
扩展	数组	(可选) 两个布尔值数组定义了是否扩展底纹功能至圆的起始和结束点。默认值: [0 0]。

颜色混合是基于一个共混圈族中的开始和结束圈, 由该底纹字典及其坐标入口定义的。共混圈是带一个附属条件的参变量界。

$$s = \frac{t - t_0}{t_1 - t_0}$$

其中 0.0 和 1.0 之间变化呈线性变化, 变量 t 横跨从 T0 到域 T1 的, 并作为指定的字典的域名项。该中心和每个半径混合循环都是通过以下参数方程:

$$\begin{aligned}x_c(s) &= x_0 + s \times (x_1 - x_0) \\y_c(s) &= y_0 + s \times (y_1 - y_0) \\r(s) &= r_0 + s \times (r_1 - r_0)\end{aligned}$$

每个变量 S 的 0.0 和 1.0 之间的值决定了 t 相应的值，通过作为输入参数的功能 (S) 定义阴影字典的函数的入口。这就产生了颜色与其中的元件值相应的混合循环。对于值不是介于 0.0 和 1.0 的，阴影字典的扩展布尔数组元素确定其是否延长阴影和如何延长阴影。如果两个参数中前面一个是真的，阴影已经超出了已经定义好的起始圆，s 值小于 0.0，如果第二个元素是真的，阴影已经超出了已经定义好的终点圆，s 值大于 0.0。

请注意，任何的开始和结束圆可能比其他的大。如果阴影是在较小的底延长，共混圆继续让成员中 s 值，即共混圆 $r(s) = 0$ 。如果阴影最后是在更大的延长内，共混圆继续尽可能的让 s 值大到足以涵盖遮荫的整个边界框 (BBox)。可扩展着色绘画从而导致延续由两个圆自己界定的领域。最右边的板 11 两个例子描绘延长较小和阴影的结果较大的两个情况。

从概念上讲，对混合画界都在 S 的增加值的顺序，从小到大。混合界以外的圈子扩大是开始

画中的阴影字典的功能项中定义的相同的颜色为起始圆 (吨= T0 的，下= 0.0)。混合结束循环圈以外的延伸是画在截至圆 (吨= T1 的定义的颜色，下= 1.0)。该绘画是不透明的每个圆圈的颜色，完全覆盖的前面它。因此，如果一个点在于融合在一个以上的循环，它的最终颜色是封闭的圈子最后被画，对应于 s 的最大值。

请注意以下几点：

- 如果开始和结束完全包含在一个圈子内，另一方面阴影描绘了一个球，如 12 和 13 板所示。在板 12，内圆半径为零；它是在左边的数字开始循环，在循环结束图右侧。无论在阴影或延长较小或较大的结束区。在 13 板块，在这两个数字内圈有一个非零半径和底纹是在较大规模的扩展结束。在每一个板，背景颜色指定为右边的数字，但不是在左边的数字。
- 如果没有任何圆包含其他圆，只描绘了阴影锥。如果起始圆更大，锥似乎点出了页面。如果结束圆更大，锥似乎点进了页面（见板 11）。

例子 4.25 绘制了叶覆盖分支，如板 14 所示。每个叶子是充满具有相同的径

向底纹（对象数 5）。颜色函数（对象 10）1 拼接功能（3.9.3 节中所述，“第三类（拼接）函数”）他的两个子功能（对象 11 和 12）都插指数功能（见 3.9.2 节，“2 型（指数插值）函数”）。每片叶子被画成一个路径，然后用阴影填充，利用代码，示例 4.26 所示（其中 Sh1 的名字是与对象 5 相关联，对象 5 是当前资源字典的底纹子字典；见第 3.7.2，“资源词典”）。

例子 4.25

```
5 0 obj % Shading dictionary
<< /ShadingType 3
/ColorSpace /DeviceCMYK
/Coords [0.0 0.0 0.096 0.0 0.0 1.000] % Concentric circles
/Function 10 0 R
/Extend [true true]
>>
endobj

10 0 obj % Color function
<< /FunctionType 3
/Domain [0.0 1.0]
/Functions [11 0 R 12 0 R]
/Bounds [0.708]
/Encode [1.0 0.0 0.0 1.0]
>>
Endobj

11 0 obj % First subfunction
<< /FunctionType 2
/Domain [0.0 1.0]
/C0 [0.929 0.357 1.000 0.298]
/C1 [0.631 0.278 1.000 0.027]
/N 1.048
>>
```

```
endobj
12 0 obj % Second subfunction
<< /FunctionType 2
/Domain [0.0 1.0]
/C0 [0.929 0.357 1.000 0.298]
/C1 [0.941 0.400 1.000 0.102]
/N 1.374
>>
Endobj
```

例子 4.26

```
316.789 140.311 m % Move to start of leaf
303.222 146.388 282.966 136.518 279.122 121.983 c % Curved segment
277.322 120.182 l % Straight line
285.125 122.688 291.441 121.716 298.156 119.386 c % Curved segment
336.448 119.386 l % Straight line
331.072 128.643 323.346 137.376 316.789 140.311 c % Curved segment
W n % Set clipping path
q % Save graphics state
27.7843 0.0000 0.0000 - 27.7843 310.2461 121.1521 cm % Set matrix
/Sh1 sh % Paint shading
Q % Restore graphics state
```

类型 4 底纹（自由格式高氏着色三角网格）

类型 4 底纹（自由格式高氏着色三角网格）通常用来表示复杂的颜色和阴影的三维形状。该阴影部分面积是完全是由组成的三角形路径定义。在颜色每个三角形的顶点指定，并作为一种技术称为古罗插值是用来内饰颜色的。插值函数定义阴影可能是线性或非线性的。表 4.32 显示了特定的条目这种类型的底纹词典，除了这些共同所有底纹词典（表 4.28）和流字典（第 38 页表 3.4）。

表 4.32 附加类型 4 底纹字典入口

关键字	类型	值
-----	----	---

BitsPerCoordinate	整数	(必须) 位数来表示每个顶点坐标。有效的值是 1, 2, 4, 8, 12, 16, 24 和 32。
BitsPerCoordinate	整数	(必须) 位数来表示每个颜色分量。有效的值是 1, 2, 4, 8, 12 和 16。
BitsPerFlag	整数	(必须) 位数来表示每个顶点的边标志 (见下文)。 BitsPerFlag 有效值为 2, 4 和 8, 但至少每个标志值中两个重要的标志被使用。对于边缘的有效标志值是 0, 1 和 2。
解码	数组	(必须) 对指定如何映射顶点坐标的数字数组并在适当范围值颜色分量。解码方法类似于在图像字典所用的(见“解码阵列” 314 页)。指定的范围如下: [xmin xmax ymin ymax c1,min c1,max ... cn,min cn,max] 如果一个函数的入口已经存在, 请注意, 只有一对的 C 值应指明。
功能	功能	(可选) 1 输入, n 输出的函数或 1 个 1 输入, 1 输出功能的数组 (其中 n 是颜色组件数量, 在底纹字典的色彩空间)。如果此项存在, 每个顶点的颜色数据必须由一个参变量指定, 而不是单独的颜色由 N 组件。指定的功能 (S) 称为与每个插变量的参数值, 以确定每个实际的颜色点。每个输入值是被迫给指定的范围区间在遮荫字典的解码阵列相应的颜色分量。每个函数的域必须是该区间的超集。如果返回值该函数为一个给定的颜色成分是超出范围, 它是调整至最接近的有效值。 本入口不得用于索引的色彩空间。

跟底纹类型 1 至 3 不同, 类型 4 至 7 表示为流。每个流包含了顶点坐标和颜色序列数据, 它定义三角形网格。在 4 型着色, 每个顶点由指定下列值, 值如下:

$f\ x\ y\ c_1\dots c_n$

其中

f 为边量顶点（下面讨论）

x 和 y 是其水平和垂直坐标

$c_1\dots c_n$ 是它的颜色分量

所有的顶点坐标是表现在阴影目标的坐标空间。如果阴影字典包含一个函数的入口，只有一个参数值， t ，色彩组件 $c_1\dots c_n$ 的空间顶点是允许的。

边量与每个顶点相关决定的方式连接到其他的三角形网格顶点。一个具有标志值 va 顶点边 $fa=0$ 开始一个新的三角形，无关的任何其他。至少有两个顶点（VB 中和 VC）必须提供，但他们的信号量被忽略。这三个顶点定义一个三角形 (va, vb, vc) 的，如图 4.16 所示。

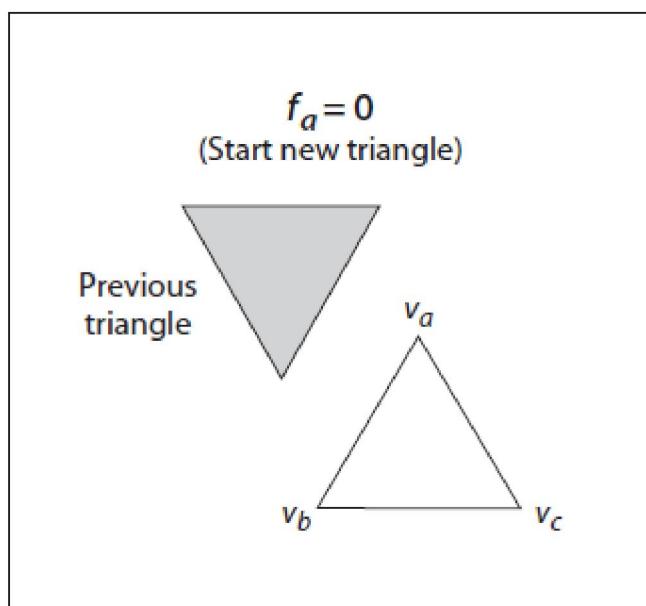


图 4.16 在自由三角网格形成一个新的三角

随后的三角形是由一个新的定义有两个顶点并合并前面的顶点三角形。鉴于三角形 (va, vb, vc)，其中 va 是之前顶点， vb 在 vc 之前，一个新的顶点 vd 可以形成一个新的三角形侧面 vbc 或 vac ，如图 4.17 所示。（侧 VAB 型假设为分享了前面的三角形，因此不继续使用网格。）如果边缘标志 $fd=1$ (vbc)，下一个顶点形成三角形 (vb, vc, vd) 的；若边缘标志 $fd=2$ (vac)，下一个顶点构成的三角形 (va, vc, vd)。边缘标志 $fd=0$ 的将启动一个新的三角形，如上所述。

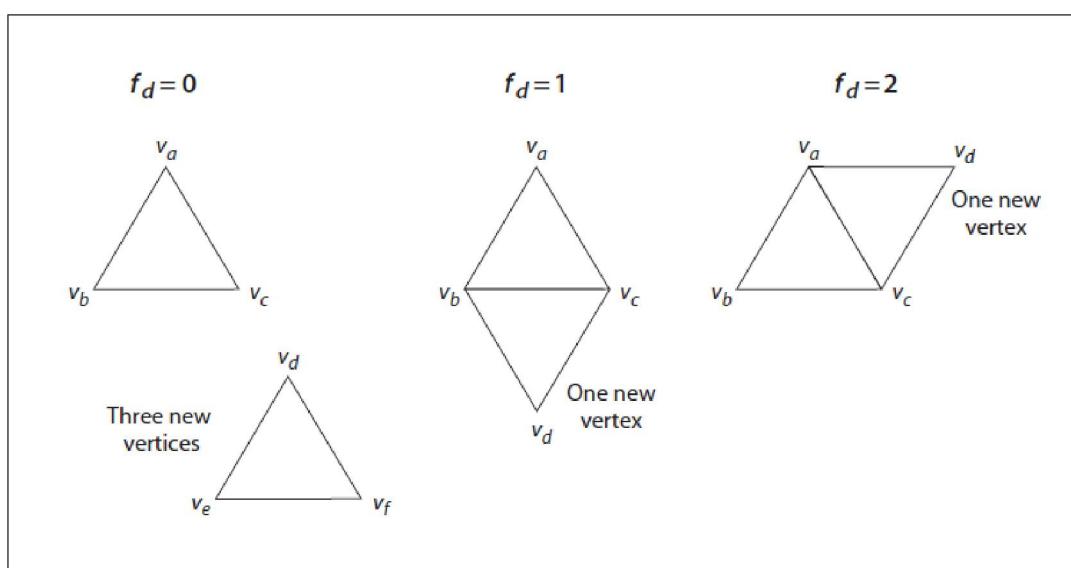


图 4.17 在自由三角网格连接三角网格

可以创建复杂的形状使用边缘标志来控制的边缘上形成的三角形。图 4.18 显示了两个简单的例子。网一开始的三角形 1 和使用下列边旗成功绘制每个三角形：

$$1 \ (f_a = f_b = f_c = 0)$$

$$7 \ (f_i = 2)$$

$$2 \ (f_d = 1)$$

$$8 \ (f_j = 2)$$

$$3 \ (f_e = 1)$$

$$9 \ (f_k = 2)$$

$$4 \ (f_f = 1)$$

$$10 \ (f_l = 1)$$

$$5 \ (f_g = 1)$$

$$11 \ (f_m = 1)$$

$$6 \ (f_h = 1)$$

网 2 又开始与三角形 1 使用以下标志：

$$1 \ (f_a = f_b = f_c = 0)$$

$$4 \ (f_f = 2)$$

$$2 \ (f_d = 1)$$

$$5 \ (f_g = 2)$$

$$3 \ (f_e = 2)$$

$$6 \ (f_h = 2)$$

流必须提供整数的三角形顶点数据与适当边旗，否则就会出错。

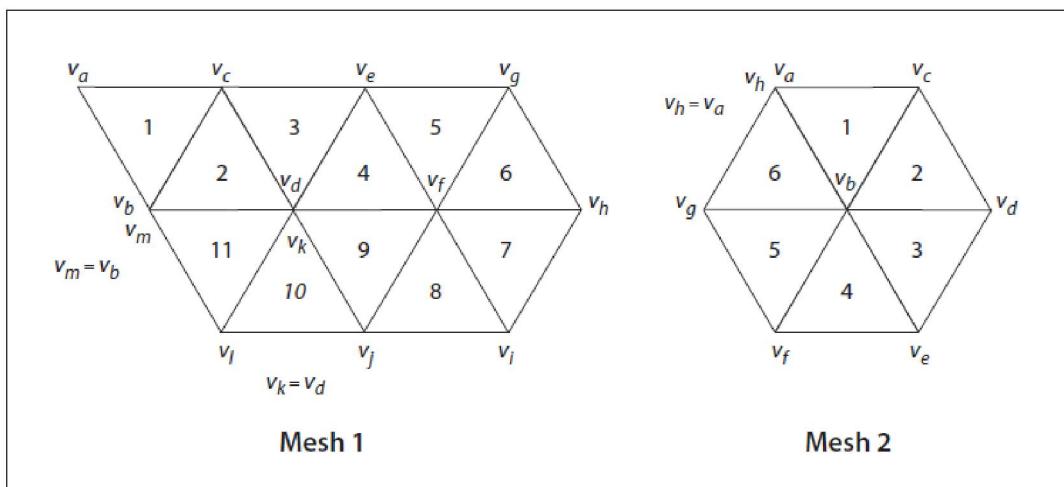


图 4.18 不同的标志值的创造不同的形状

每个顶点的数据包括以下项目，按顺序读取从高阶到低阶位的位置：

- 一个边量，表示在 BitsPerFlag 位
- 一个横向和纵向坐标，表示在 BitsPerCoordinate 的每个位
- n 个颜色分量（其中 n 是组件的数量底纹的颜色空间），在每个分量帧予以表示，该命令被 sc 算子表示。

每个顶点数据集必须占用的字节整数。如果该总数所需的 8 位不整除，最后一个数据字节为每个顶点是在填充与额外的比特，这是被忽略的末尾。坐标和颜色值解码按照同样的方式解码图像中的数组字典（见解码阵列“314 页”）。

如果阴影字典包含一个函数的入口，每个顶点的颜色数据必须指定一个参数值 t ，而不是单独的颜色由 N 组件。三角网格内的所有线性插值是使用吨值。结果被传递给函数（附注）在该函数的入口，以确定在每一点的颜色。

类型 5 底纹（格形式高氏着色三角网格）

5 型阴影（格形式高氏着色三角网格）与 4 型相似，而采用自由形式几何，其顶点都被排列在 pseudorectangular 格，这是拓扑等价于一个长方形的网格。顶点被组织为行，而不必是几何直线（见图 4.19）。

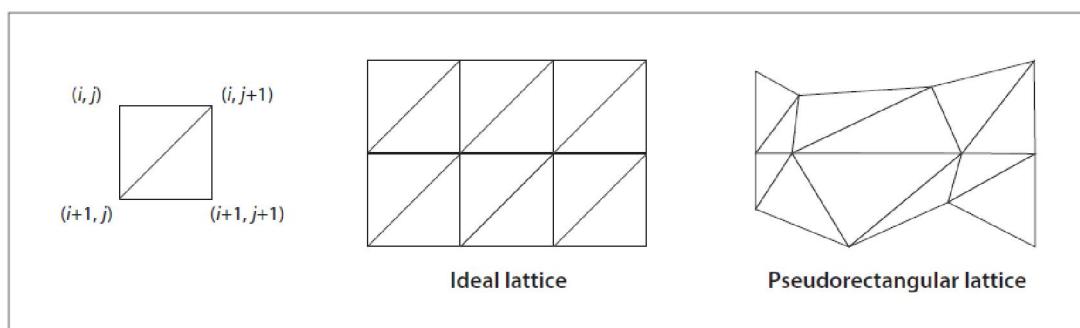


图 4.19 格形式高氏着色三角网格

表 4.33 显示了这种特有的底纹底纹，型词典条目除了这些共同所有（表 4.28）和流字典（表 3.4 底纹词典第 38 页）。

对于一个 5 型着色数据流具有为 4 型相同，只不过格式 5 型不使用边缘标志来界定的三角形网格的几何形状。因此，为每个顶点的数据包括下列值，顺序显示为：

$$x \ y \ c_1 \dots c_n$$

其中

x 和 y 是顶点的水平和垂直坐标

c1...cn 是它的颜色分量

表 4.33 附加的类型 5 底纹字典入口

关键字	类型	值
BitsPerCoordinate	整数	(必须) 位数来表示每个顶点坐标。有效的值是 1, 2, 4, 8, 12, 16, 24 和 32。
BitsPerCoordinate	整数	(必须) 位数来表示每个颜色分量。有效的值是 1, 2, 4, 8, 12 和 16。
VerticesPerRow	整数	(必须) 顶点在每一格列数的值必须大于或等于 2。行数不必指定。
解码	数组	(必须) 对指定如何映射顶点坐标的数字数组到适当颜色分量范围值。解码方法类似于在图像字典所用的(见

“解码阵列” 314 页)。指定的范围如下:

[xmin xmax ymin ymax c1,min c1,max ... cn,min cn,max]

请注意, 如果一个函数的入口已经存在, 只有一对 C 值应指定。

功能	功能	(可选) 1 输入, n 输出的功能或 1 个 1 输入, 1 输出的 n 功能数组 (其中 n 是颜色组件数量, 在底纹字典的色彩空间)。如果此项存在, 每个顶点的颜色数据必须由一个参变量指定, 而不是单独的 N 组件颜色空间。指定的功能 (S) 称为与每个插变量的参数值, 以确定每个实际的颜色点。每个输入值是被指定的范围区间在底纹字典的解码阵列相应的颜色分量。每个函数的域必须是该区间的超集。如果该函数返回值为一个给定的颜色成分值且是超出范围, 它应调整至最接近的有效值。 这个入口不能使用索引的色彩空间。
----	----	---

所有的顶点坐标是表现在目标坐标空间阴影处。如果阴影字典包含一个函数的入口, 只有一个参数值, t, 对于每一个部件的颜色 c1...cn 的顶点是允许的。

在底纹词典 **VerticesPerRow** 项给出了顶点数量在每一格的行。在一排的顶点都是指定顺序, 其次是由下一行。由于每个顶点的 k 列 m 行, 该网的建成要使用以下三个三角形顶点, 如图 4.19 所示:

$$(V_{i,j}, V_{i,j+1}, V_{i+1,j}) \quad \text{for } 0 \leq i \leq m-2, 0 \leq j \leq k-2$$

$$(V_{i,j+1}, V_{i+1,j}, V_{i+1,j+1})$$

见“类型 4 底纹 (格形式高氏着色三角网格)”在页面 284, 将对顶点数据格式做进一步说明。

类型 6 底纹 (孔斯片三角网格)

第 6 类阴影 (孔斯片网格) 构造一个或更多的颜色补丁, 每 4 个界三次 Bézier 曲线。退化贝塞尔曲线对于某些允许和图形效果非常有用。至少有一个完整的修

补程序可以指定。

阿昆斯补丁一般有两个独立的方面：

•颜色指定为每个单位正方形的角落，和双线性插值用来填充的颜色对整个单位正方形（见上面的数字板 15）。

•从坐标映射到一个单位正方形四边的补丁双方都不是线性（参见下图板 15）。

映射是连续的：该单位的角广场地图的补丁和角落该单位双方平方映射到片方，如图 4.20 所示。

修补程序的双方都是通过 4 次 Bézier 曲线， C_1 , C_2 的， D_1 和 D_2 中，参数定义了一个变量， U 和 V ，是不同水平和垂直穿过单位正方形。昆斯补丁的四角满足下面的公式：

$$C_1(0) = D_1(0)$$

$$C_1(1) = D_2(0)$$

$$C_2(0) = D_1(1)$$

$$C_2(1) = D_2(1)$$

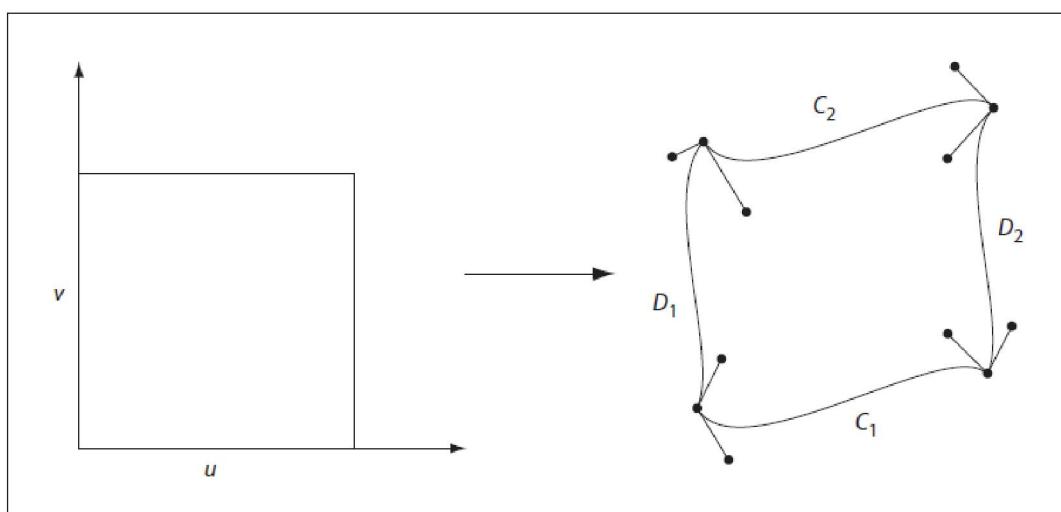


图 4.20 坐标从一个单位正方形映射到四面昆斯补丁

可以说两个表面之间的边界是线性插值曲线。沿 U 轴，表面 SC 由以下定义

$$S_C(u, v) = (1 - v) \times C_1(u) + v \times C_2(u)$$

沿 V 轴，表面 SD 是这样被赋值的

$$S_D(u, v) = (1 - u) \times D_1(v) + u \times D_2(v)$$

第三面是四角双线性插值：

$$\begin{aligned} S_B(u, v) = & (1 - v) \times [(1 - u) \times C_1(0) + u \times C_1(1)] \\ & + v \times [(1 - u) \times C_2(0) + u \times C_2(1)] \end{aligned}$$

坐标映射到底纹是由曲面 S 提供的，定义为：

$$S = S_C + S_D - S_B$$

这里定义了每个补丁的几何形状。一个补片是从构建序列的一个或多个颜色的补丁。

补丁有时似乎相对自己对折的，例如，如果一边界曲线相交本身。作为参数 u 或 v 值增加

参数空间，在设备空间的相应像素的位置可能会改变方向，使新像素到像素的映射已经过去映射。如果超过 1 点 (u 和 v 在参数空间) 被映射到相同的在设备空间中的点，点选择的是与最大的价值，如果一个诉多点具有相同的五，与 u 的最大价值之一被选中。如果另外一个补丁重叠，出现了补丁后，在数据流涂料在较早的一个。

还要注意的是补丁而不是绘画表面几何控制。该纲要预计的平方 (即绘制区域) 可能不一样如果这个补丁的边界，例如，补丁倍以上的本身，显示在图 4.21。

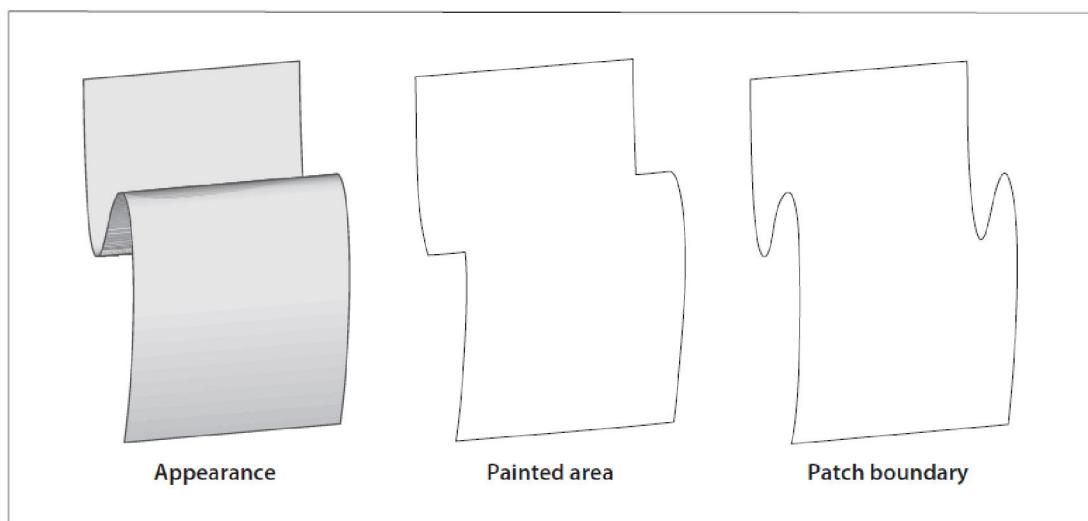


图 4.21 彩绘的昆斯区和边界补丁

表 4.34 显示了这种特有的底纹字典入口，附加的这些共同所有的词典条目（表 4.28）和流字典（表 3.4 第 38 页）。

表 4.34 附加类型 6 底纹字典入口

关键字	类型	值
BitsPerCoordinate	整数	(必须) 的位数来表示每个几何坐标。有效的值是 1, 2, 4, 8, 12, 16, 24 和 32。
BitsPerCoordinate	整数	(必须) 位数来表示每个颜色分量。有效的值是 1, 2, 4, 8, 12 和 16。
BitsPerFlag	整数	(必须) 位数来代表每个标志位修补程序（见下文）。对 BitsPerFlag 有效值为 2, 4 和 8, 但只有最重要的 2 位标志值被使用。对于边缘的有效值是 0, 1, 2 和 3。
解码	数组	(必需) 对指定如何映射顶点坐标的数字数组到适当颜色分量范围值。解码方法类似于在图像字典所用的（见“解码阵列” 314 页）。指定的范围如下： [xmin xmax ymin ymax c1,min c1,max ... cn,min cn,max] 请注意，如果一个函数的入口已经存在，则只有一对的 C 值被确定。

功能	功能	(可选) 一个 1 输入, n 输出功能或 1 个 1 输入, 1 输出的功能数组 (其中 n 是颜色组件数量, 在底纹字典的色彩空间)。如果此项存在, 每个顶点的颜色数据必须由一个参变量指定, 而不是单独的颜色组件。指定的功能 (S) 称为与每个插变量的参数值, 以确定每个实际的颜色点。每个输入值是被迫给指定的范围区间在底纹字典的解码阵列相应的颜色分量。每个函数的域必须是该区间的超集。如果返回值该函数为一个给定的颜色成分是超出范围, 它是调整至最接近的有效值。 这个入口不得用于索引的色彩空间。
----	----	---

数据流提供了一个控制点的 *Bézier* 和颜色值的序列定义的形状和颜色的每个补丁。一个修补程序的控制点都由于第一, 其次是它的角的颜色值。请注意, 这不同于三角形网 (底纹类型 4 和 5), 其中的坐标和颜色每个顶点给出在一起。所有的控制点坐标是表现在阴影的目标坐标空间。见“4 型底纹 (自由格式高氏着色三角网格) 284 页的”关于数据的格式进一步的细节。

正如自由形式三角网格 (类型 4), 每个补丁有一个标志位, 如果有的话指示标志, 它与以前的补丁。标志为 0 的优势开始一个新的补丁, 任何其他无关的。这必须遵循由 12 个坐标对, $x_1 \ y_1 \ x_2 \ y_2 \dots \ x_{12} \ y_{12}$, 指定控制点的贝塞尔确定四个边界曲线。图 4.22 显示了如何将这些控制点对应到三次 *Bézier* 曲线的 C1, C2, D1 和图 4.20 确定的 D2。颜色值给出的补丁软件中的四个角落, 以相同的顺序为控制点对应的角落。因此, C1 是在色坐标 (x_1, y_1), C2 在 (x_4, y_4), C3 在 (x_7, y_7), 和 C4 在 (x_{10}, y_{10}), 如图所示。

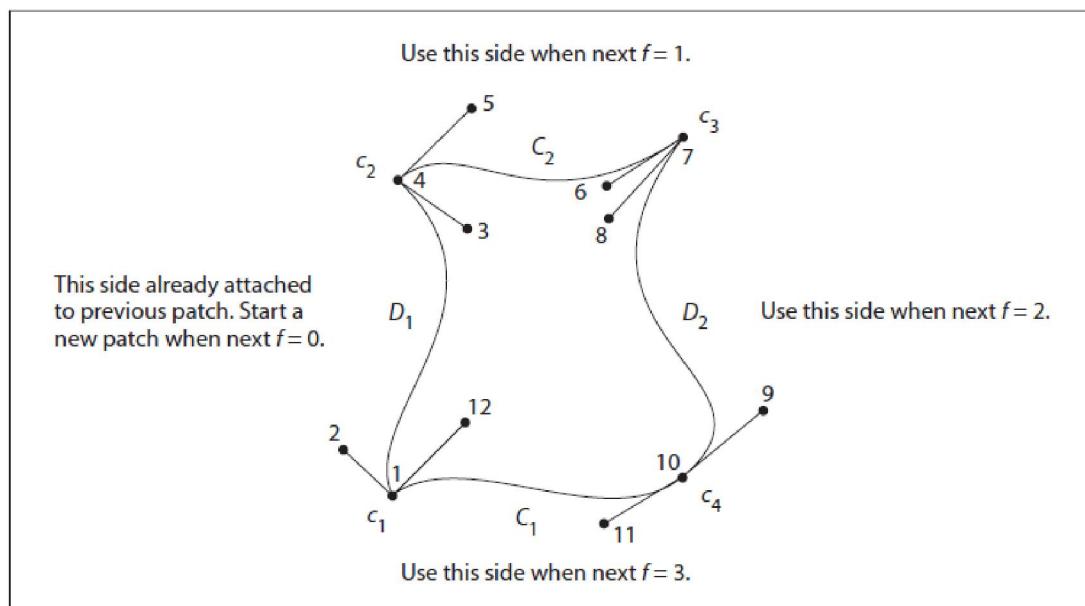


图 4.22 在昆斯网的颜色值和边缘斑块

图 4.22 还显示如何在标志位 ($f=1, 2$ 或 3) 连接一个新的补丁到以前的补丁的边缘之一。在这种情况下，一些以前的修补程序的控制点为含蓄为控制点的新补丁以及（见图 4.23），因此在没有明确的数据重复流。表 4.35 总结了所需的各值的数据值。

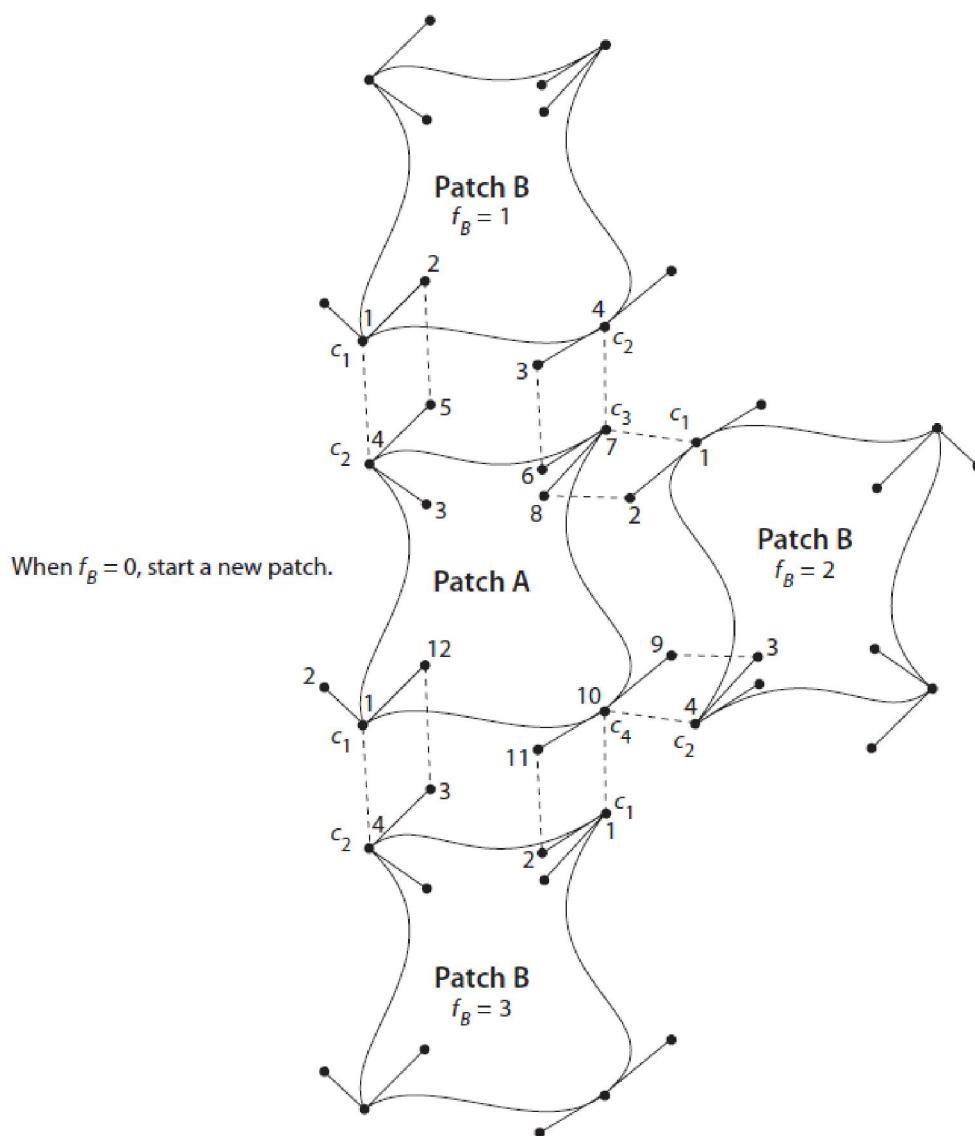


表 4.23 在补丁边缘连接网孔斯片网格

如果阴影字典包含一个函数的入口，每个颜色数据的修补程序，必须指定一个参数值 t ，而不是单独由 N 颜色分量 $c_1 \dots c_n$ 。网内的所有线性插值完成使用 T 值。其结果被传递给函数在指定的函数的入口处，以确定在每一点的颜色。

表 4.35 孔斯片网格里的数据值

边界量	下一组数据值
f=0	$x_1 \ y_1 \ x_2 \ y_2 \ x_3 \ y_3 \ x_4 \ y_4 \ x_5 \ y_5 \ x_6 \ y_6$ $x_7 \ y_7 \ x_8 \ y_8 \ x_9 \ y_9 \ x_{10} \ y_{10} \ x_{11} \ y_{11} \ x_{12} \ y_{12}$ $c_1 \ c_2 \ c_3 \ c_4$ 新的补丁；没有隐藏的值
f=1	$x_5 \ y_5 \ x_6 \ y_6 \ x_7 \ y_7 \ x_8 \ y_8 \ x_9 \ y_9 \ x_{10} \ y_{10} \ x_{11} \ y_{11} \ x_{12} \ y_{12}$ $c_3 \ c_4$ 隐藏值： $(x_1, \ y_1) = (x_4, \ y_4) \text{ previous} \quad c_1 = c_2 \text{ previous}$ $(x_2, \ y_2) = (x_5, \ y_5) \text{ previous} \quad c_2 = c_3 \text{ previous}$ $(x_3, \ y_3) = (x_6, \ y_6) \text{ previous}$ $(x_4, \ y_4) = (x_7, \ y_7) \text{ previous}$
f=2	$x_5 \ y_5 \ x_6 \ y_6 \ x_7 \ y_7 \ x_8 \ y_8 \ x_9 \ y_9 \ x_{10} \ y_{10} \ x_{11} \ y_{11} \ x_{12} \ y_{12}$ $c_3 \ c_4$ 隐藏值： $(x_1, \ y_1) = (x_7, \ y_7) \text{ previous} \quad c_1 = c_3 \text{ previous}$ $(x_2, \ y_2) = (x_8, \ y_8) \text{ previous} \quad c_2 = c_4 \text{ previous}$ $(x_3, \ y_3) = (x_9, \ y_9) \text{ previous}$ $(x_4, \ y_4) = (x_{10}, \ y_{10}) \text{ previous}$
f=3	$x_5 \ y_5 \ x_6 \ y_6 \ x_7 \ y_7 \ x_8 \ y_8 \ x_9 \ y_9 \ x_{10} \ y_{10} \ x_{11} \ y_{11} \ x_{12} \ y_{12}$ $c_3 \ c_4$ 隐藏值： $(x_1, \ y_1) = (x_{10}, \ y_{10}) \text{ previous} \quad c_1 = c_4 \text{ previous}$ $(x_2, \ y_2) = (x_{11}, \ y_{11}) \text{ previous} \quad c_2 = c_1 \text{ previous}$ $(x_3, \ y_3) = (x_{12}, \ y_{12}) \text{ previous}$ $(x_4, \ y_4) = (x_1, \ y_1) \text{ previous}$

类型 7 底纹（张量积三角网格补丁）

类型 7 底纹（张量积三角网格补丁）跟类型 6 是相同的，它们都是基于双三次张量，由 16 个控制点而不是定义好的补丁 12 个控制点定义一个昆斯补丁。阴影词典代表两个不同类型的补丁，底纹类型项的每个控制点数目在指定的数据流片。

虽然孔斯片网格补丁更简洁，更容易使用，张量积三角网格在彩色绘图中表现更好。

注：底纹类型 7 的数据格式（如类型 4 至 6）跟 PDF 相似，因为它是 PostScript 体系。但是，控制点的编号和顺序化在第一次打印 PostScript 相关语言的时候会被错误的描述。第三版已得到纠正。

像孔斯片网格补丁映射，张量积三角网格补丁映射由 4 名的位置和形状三次 Bézier 曲线的边界标志此程序。不过，张量积三角网格补丁有四个额外的“内部”控制点来调整映射。16 个控制点，可安排在一 4×4 的阵列来索引，如下（见图 4.24）：

$$\begin{array}{cccc} p_{03} & p_{13} & p_{23} & p_{33} \\ p_{02} & p_{12} & p_{22} & p_{32} \\ p_{01} & p_{11} & p_{21} & p_{31} \\ p_{00} & p_{10} & p_{20} & p_{30} \end{array}$$

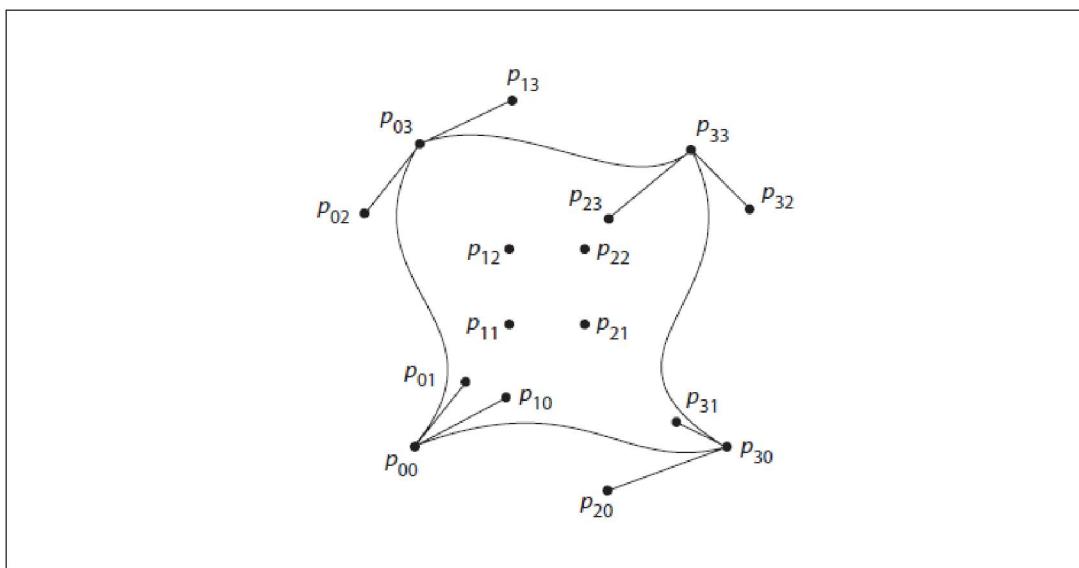


表 4.24 张量积三角网格补丁的控制点

正如在孔斯片网格补丁，张量积三角网格补丁被几何参数变量 U 和 V 描述，在整个单位空间， U 和 V 水平和垂直都不相同。表面的定义是由方程来完成的。

$$S(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 p_{ij} \times B_i(u) \times B_j(v)$$

当中 P_{ij} 在坐标的行 i 和列 j 中起到控制的作用， B_i 和 B_j 是伯恩斯坦多项式

$$B_0(t) = (1-t)^3$$

$$B_1(t) = 3t \times (1-t)^2$$

$$B_2(t) = 3t^2 \times (1-t)$$

$$B_3(t) = t^3$$

因为每个 P_{ij} 点实际上是一对坐标 (X_{ij}, Y_{ij}) ，表面可以这样予以描述

$$x(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 x_{ij} \times B_i(u) \times B_j(v)$$

$$y(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 y_{ij} \times B_i(u) \times B_j(v)$$

张量积三角网格补丁的几何特性可以通过贝兹曲线从底部边界到顶端的移动来予以虚拟化。在底部和顶部，相应的曲线的控制点是配合着那些补丁的底部 (P00...P30) 和顶部 (P03...P33) 边界曲线。当曲线由补丁的底部边界移动到顶部，它的每一个 4 控制点服从一个轨迹，这个轨迹由 4 个控制点在相应的行和列经贝塞尔曲线的定义。也就是说，开始点的运动曲线的控制点定义如下轨迹 p00... p03 时，轨迹的终点定义为点 p30 ... P33 和两个中间控制点，这些由 P10... P13 和 P20 ... p23。等价地，补丁可以被认为是由一个三次贝塞尔曲线移动追踪从左侧到右侧边缘与控制点，以下的运动轨迹定义坐标数组而不是列的行。

孔斯片网格补丁（类型 6）实际上是张量积三角网格补丁（类型 7）的一个特殊，其中 4 个内部控制点 (p11, p12, p21, p22) 被边界曲线隐性定义。内部控制点的值由这些方程给出：

$$p_{11} = 1/9 \times [-4 \times p_{00} + 6 \times (p_{01} + p_{10}) - 2 \times (p_{03} + p_{30}) + 3 \times (p_{31} + p_{13}) - 1 \times p_{33}]$$

$$p_{12} = 1/9 \times [-4 \times p_{03} + 6 \times (p_{02} + p_{13}) - 2 \times (p_{00} + p_{33}) + 3 \times (p_{32} + p_{10}) - 1 \times p_{30}]$$

$$p_{21} = 1/9 \times [-4 \times p_{30} + 6 \times (p_{31} + p_{20}) - 2 \times (p_{33} + p_{00}) + 3 \times (p_{01} + p_{23}) - 1 \times p_{03}]$$

$$p_{22} = 1/9 \times [-4 \times p_{33} + 6 \times (p_{32} + p_{23}) - 2 \times (p_{30} + p_{03}) + 3 \times (p_{02} + p_{20}) - 1 \times p_{00}]$$

在更一般的张量积三角网格补丁，这 4 个点的值是不受限制的。

在张量积三角网格补丁里的控制点坐标实际上在以下底纹数据流中被指定：

4	5	6	7
3	14	15	8
2	13	16	9
1	12	11	10

所有的控制点坐标是表现在阴影的目标坐标空间。这些都是该补丁下的，在四个角落的相应的颜色值作为自己的角落坐标。如果补丁的边量 f 是 0，所有 16 个控制点和四个角落的颜色必须明确指定的数据流。如果 f 是 1, 2 或 3，控制点和修补程序的共享优势颜色隐式理解为是为沿指定边相同以前的补丁，并没有重复的数据流。表 4.36 总结了各边量的数据值，表示该行的条件和列指数如图 4.24 “类型 4 底纹（自由形式的高氏着色三角网格）” 第 284 页。

表 4.36 张量积三角网格里的数据值

边界量	下一组数据值
$f=0$	x00 y00 x01 y01 x02 y02 x03 y03 x13 y13 x23 y23 x33 y33 x32 y32 x31 y31 x30 y30 x20 y20 x10 y10 x11 y11 x12 y12 x22 y22 x21 y21 c00 c03 c33 c30 新的补丁；没有隐藏的值
$f=1$	x13 y13 x23 y23 x33 y33 x32 y32 x31 y31 x30 y30 x20 y20 x10 y10 x11 y11 x12 y12 x22 y22 x21 y21 c33 c30 隐藏值： (x00, y00) = (x03, y03) previous c00 = c03 previous (x01, y01) = (x13, y13) previous c03 = c33 previous (x02, y02) = (x23, y23) previous (x03, y03) = (x33, y33) previous

f=2	x13 y13 x23 y23 x33 y33 x32 y32 x31 y31 x30 y30 x20 y20 x10 y10 x11 y11 x12 y12 x22 y22 x21 y21 c33 c30
	隐藏值： (x00, y00) = (x33, y33) previous c00 = c33 previous (x01, y01) = (x32, y32) previous c03 = c30 previous (x02, y02) = (x31, y31) previous (x03, y03) = (x30, y30) previous
f=3	x13 y13 x23 y23 x33 y33 x32 y32 x31 y31 x30 y30 x20 y20 x10 y10 x11 y11 x12 y12 x22 y22 x21 y21 c33 c30
	隐藏值： (x00, y00) = (x30, y30) previous c00 = c30 previous (x01, y01) = (x20, y20) previous c03 = c00 previous (x02, y02) = (x10, y10) previous (x03, y03) = (x00, y00) previous

4.7、外部对象

一个外部对象（通常称为了 XObject）是一个图形对象，其内容是指由一个独立的内容流，从内容分开流中使用它。有三种类型的外部对象：

- 图像了 XObject（第 4.8.4，“图像词典”）代表一个样本视觉图像，如照片。
- 一个表格了 XObject（第 4.9 节，“表格 XObjects”）是一个独立的描述一个图形对象的任意序列。
- 一个 PostScript 体系的 XObject（第 4.7.1，“PostScript 体系的 XObjects”）载有片段代码，用 PostScript 页面描述语言。PostScript XObjects 不再推荐使用。

两个外部对象，组 XObjects 和 XObjects 扩展（都是 PDF 格式 1.4），实际上是 XObjects 类型的额外属性。见第 4.9.2，“XObjects 类”和 4.9.3，“XObjects

参考”了解更多信息。

任何描绘了 XObject 可以作为另一个内容流部分（见表 4.37）。这个操作符适用于任何 XObject 图像类型，或 PostScript 形式。语法是一样的，虽然详细介绍的行为有所不同的。（执行情况见附录 H 中标注 50）

表 4.37 XObject 算子		
操作数	算子	描述
名字	DO	绘制指定的 XObject。操作数的名称必须出现在关键的 XObject 子字典，在当前资源字典（见 3.7.2 节，“资源词典”）内。相关的值必须是一个流，其类型项，如果存在，是 XObject。其影响取决于 XObject 的亚型项值，这可能是图像（见第 4.8.4，“图像词典”），表（第 4.9 节，“XObjects 表格”），或 PS（第 4.7.1，“的 PostScript XObjects”）。

4.7.1 PostScript 下的 XObjects

与 PDF 1.1，一开始可以包括的内容流 PostScript 语言片段。这些片段仅用于当打印到 PostScript 输出设备；他们没有任何效果或者当查看文档在屏幕上或当打印到非 PostScript 设备。此外，应用程序能够理解 PDF 格式是不太可能能够解释的 PostScript 片段。因此，这种能力应使用极为谨慎且仅当有没有其他办法达到同样的效果。不适当的 PostScript XObjects 使用，可能使 PDF 格式文件打印不正确。

注：由于 PDF 格式的 Adobe 1.4 包括成像模型的所有功能 PostScript 语言，不再有任何理由使用 PostScript XObjects。这功能很可能是在未来的版本中删除。

PostScript 的 XObject 是 XObject 流，它的子类型入口有 PS 值。PostScript 的 XObject 字典可以包含在表 4.38 所示的条目以及一般常见的所有作品流（见

表 3.4 第 38 页)。

表 4.38 附加的 PostScript 的 XObject 字典入口		
关键字	类型	值
类型	名字	(可选) PDF 类型的对象, 这字典已介绍, 如果存在, 必须是 PostScript 的 XObject。
子类型	名字	(必须), 这种类型的 XObject 字典已描述; 必须是 PostScript XObject 的 PS.
级别 1	流	(可选) 流的内容是代替 PostScript 的 XObject 流时, PostScript 解释的目标是已知唯一支持 LanguageLevel 1。

当一个 PDF 内容流翻译成 PostScript 语言, 任何操作引用了 XObject, 一个 PostScript 是由内容所取代了 XObject 流本身。该流不被复制的。在 PostScript 碎片可以使用 Type 1 字体和 TrueType 字体子字典目前的资源词典 (见 3.7.2 节, “资源词典”), 访问他们通过自己的 BaseFont 使用 PostScript findfont 经营者的名称。该片段可能不能使用在字体的子字典. 不应引用的 PostScript 到 PDF 程序定义相应的其他类型的字体集 (见 10.1 节, “程序集”), 这可能会有所变化。

4.8、图像

PDF 格式的算子包括图像采集与处理一般设施。一个采样图像 (或只是短期) 是一个矩形阵列的样品图片值, 分别代表一种颜色。该图像可能近似的外观一些自然场景获得通过输入扫描仪或摄像头, 或可能综合的产生。

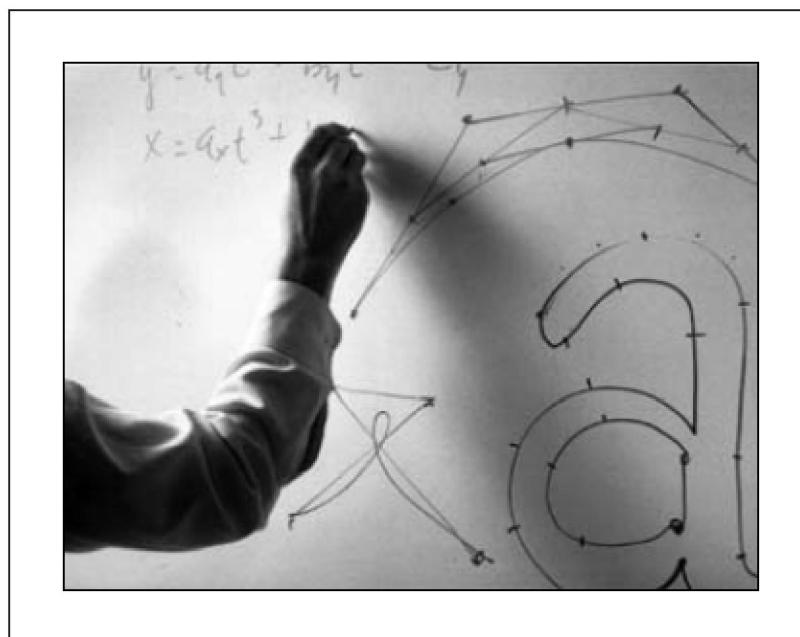


图 4.25 典型抽样图像

图像被定义为一个由扫描得到的图像序列样本，样本阵列中进行行或列顺序扫描。数组中的每个样本由许多色彩作为其色彩空间，他们需要的组件指定为，一个 DeviceRGB 组件，四个 DeviceGray，三个 DeviceCMYK，无论数量或所要求的特定 DeviceN 空间。每个组件是一个 1 - 2 - ， 4 - ， 8 - ，或 (PDF 1.5) 的 16 位整数，允许代表 2, 4, 16, 256, 或 (PDF 1.5) 65536 为每个组件产生不同的值。(其他组件的大小可以容纳滤器当 JPXDecode 使用时；参阅第 3.3.8，“JPXDecode 过滤器。）

PDF 格式的图像提供了两种方式来指定：

- 图像了 XObject (4.8.4 节中所述，“图像词典”) 是一流对象，它的字典指定图像的属性和其数据中包含的图像样本。像所有的外部对象，它是画在通过调用做好内容流操作页面 (见第 4.7 节，“外对象“)。图像 XObjects 以及有其他用途，如用于替代图像 (见备用图片“页 317”), 图像面具 (第 4.8.5, “戴面具的图像”), 和缩略图 (第 8.2.3, “缩略图图像“)。
- 内嵌的图像是一个很小的图像，完全确定，这两个属性和数据直接内嵌在内容流。图像的种类，可以以这种方式代表的是有限的，见第 4.8.6, “内联图像,”

为细节。

4.8.1 图像参数

图像的分辨率的性质，定位，扫描秩序，等等规定，是完全由光栅输出设备的特点决定的。使用者通常使用的 PDF 渲染图片由一个抽样技术，试图接近的颜色值在尽可能准确的条件下。实际的精度取决于实现关于分辨率和输出设备。

画一个图像，四个相互关联的项目，必须指定：

- 图像的格式：列数（宽度），行数（高度），每个样品的数量颜色分量，而且每一个颜色的位数组成部分
- 样本数据构成图像的视觉内容
- 坐标之间的对应用户空间和那些在图像的内部协调的空间，界定了用户空间将接收区域
- 图像
- 从图像中的数据组件的颜色成分值映射图像的色彩空间。

所有这些项目都是由明确或暗示的图像 XObject 内联形象指定。

注：为方便起见，下面的章节提到的对象界定为图像字典形象。虽然这个词只是指流字典部分的一个图像 XObject 对象，应理解为同样适用于流的数据部分或参数和数据内嵌的形象。

4.8.2 样品表征

对于一个图像源格式可以说是由 4 个参数组成：

- 样品图像的宽度
- 样品图像的高度
- 每个样本的颜色组件的数量

- 在每种颜色的位数组成部分

图像字典指定的宽度，高度，以及每个组件的位数应明确。色彩的元件数量可以推断出从颜色空间中指定的字典。

注：对于使用 `JPXDecode` 过滤图像（见第 3.3.8，“`JPXDecode` 过滤器”），每一点的数目取决于图像数据，并没有指定在图像字典。颜色空间可能不会列明在字典中。

样本数据表示为一个字节流，为 8 位无符号整数，范围为 0 到 255。与每个字节的第一次高序位构成一个连续的字节码流。该位流，又分为单位，对每一个 n 位，其中 n 是每个组件的位数。每个单元编码颜色成分值，高序位第一考虑；单位的 16 位给出最重要的字节开始。字节边界被忽略，但每个样本数据行必须开始一个字节的边界。如果数据位每号码

行不是 8 的倍数，该行的末尾有额外比特填充填写最后一个字节。使用者应忽略 PDF 这些填充比特。

每个比特流中的一个无符号整数， n 位单位范围 0 到 $2n - 1$ 。图像字典的解码将这个整数映射到彩色部分的值，相当于可以使用颜色，如 SC 或颜色组件的抽样样本，例如，在一个三分量 RGB 图像，红色，绿色和样本蓝色组件其次是下一个的红色，绿色和蓝色分量。

通常，在一个图像中的颜色样本，在没有参考图形状态的颜色参数下根据颜色在图像空间指定词典解析（见第 4.5 节，“颜色空间”）。但是，如果图像字典的图像掩码项是真实的，样本数据被解析为一个模具掩码的图形状态的非平滑颜色参数（见“模具掩蔽”320 页）。

4.8.3 图像坐标系

每个形象或图像的空间都有自己的内部协调制度。该图像在图像中占有空间 W 单位宽和 H 单位高， w 和 h 是样本中的宽度和高度。每个样本中占有一方单位。

坐标原点 $(0, 0)$ 在图像的左上角，坐标范围横向从 0 到 W ，纵向 0 至 H 。

图像的样本数据是按行排列，在横向坐标变化最迅速。如图 4.26 所示，在空间内的数字

说明了该样品的顺序，从 0 计数。该左上角第一个示例是在坐标 $(0, 0)$ ，在第二 $(1, 0)$ ，等最后通过示例的第一行，其左上角位于 $(w - 1, 0)$ ，其 *upperright* 角是在 $(w, 0)$ 。接下来的样本之后，是在坐标 $(0, 1), (1, 1)$ ，所以到的图像，其左上角是在最后的样本 $(w - 1, h - 1)$ ，其右下角是 (w, h) 。

注：根据规定图片坐标系统和扫描整理 PDF 格式，是不排除在实际图像中有所不同。坐标转换可用于从其他公约映射到 PDF 公约。

图像之间和用户空间的空间对应常数：单位平方米的用户空间，由用户界坐标 $(0, 0)$ 和 $(1, 1)$ ，对应到图像空间图像边界（见图 4.27）。用户空间的正常惯例，空间坐标 $(0, 0)$ 在左下角，对应图像空间的坐标 $(0, h)$ 。从图像空间到用户空间可以由矩阵 $[1 \ w \ 0 \ 0 \ -1 \ h \ 0 \ 1]$ 转换。

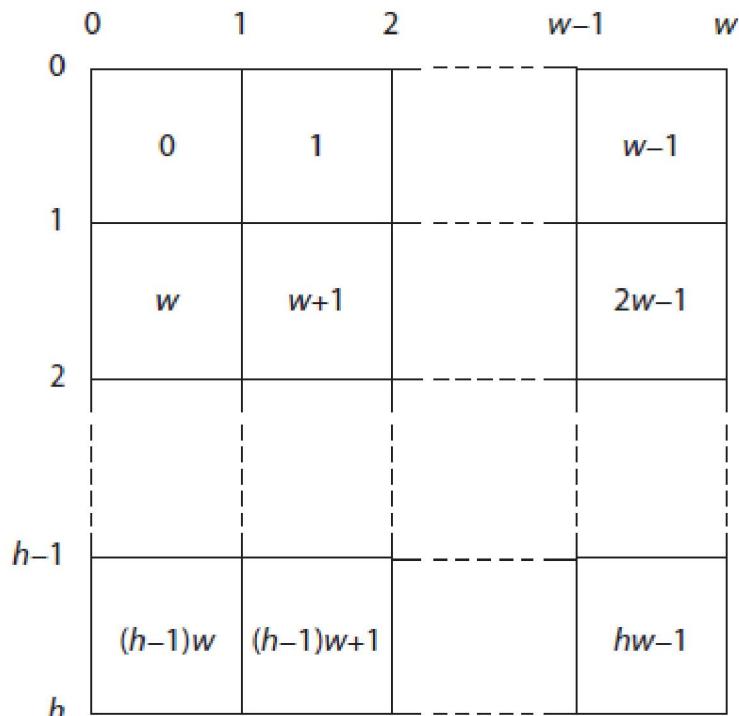


图 4.26 源图像坐标系

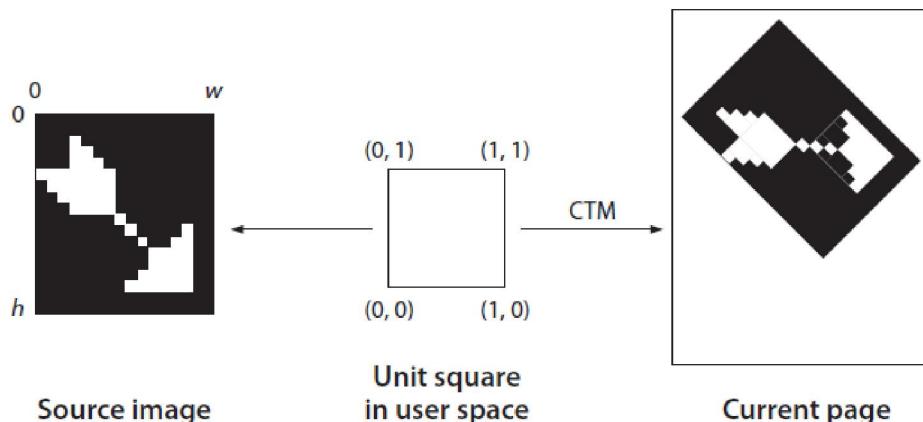


图 4.27 源图像映射

图像可以被放置在输出页面的任何位置，方向和规模通过使用 `cm` 参数来修改当前变换着的矩阵(`ctm`)，以便使用户空间映射到单位正方形的矩形或平行四边形，其中该图像是要绘制的。通常，完成参数对 `q` 和 `Q` 的赋值来孤立转化效果，它可以包括平移，旋转，反射和倾斜（见 4.2 节，“协调制度”）。例如，如果当前资源字典的 `XObject` 子字典定义图像 1，用来代表 `XObject` 图像，如例所示代码 4.27 绘制的图像在一个长方形的左下角是在坐标 (100, 200)，即逆时针旋转 45 度，并为 150 个单位和 80 个单位宽高。

例子 4.27

```
q % Save graphics state
1 0 0 1 100 200 cm % Translate
0.7071 0.7071 -0.7071 0.7071 0 0 cm % Rotate
150 0 0 80 0 0 cm % Scale
/Image1 Do % Paint image
Q % Restore graphics state
```

（正如 4.2.3 节中所讨论，“转型矩阵，”这三个转变可以合而为一。）当然，如果例子中的原始图像的高宽比（宽度比上高度）不是 150:80，结果会被扭曲。

4.8.4 图像词典

字典的图像，也就是说，一个流字典部分描述了 XObject 图像，可以包含除了一般所有流通用的入口（见 38 页表 3.4）之外，还包含表 4.39 所列的入口。这些入口之间有很强关联性，当前色彩空间可能对它们来说有所限制。试图用这种图像的字典会引起错误，这种图像字典的入口对彼此或对当前色彩空间不够兼容。

注：这里所描述的项是基本图像，是一个通过 Do 参数直接调用的入口。而这些入口有些是不相关的用于其他方面的图像，例如备用图像，（见“备用图片”页 317），图像掩码（第 4.8.5，“掩码图像”），或缩略图（组 8.2.3，“缩略图”）。除非指出，这种无关痛痒的入口却易被忽视。

表 4.39 附加的图像字典指定入口

关键字	类型	值
类型	名字	(可选) 正如这个字典所描述的 PDF 对象类型；如果存在，则必须是 XObject 图形。
子类型	名字	(必须) 正如这个字典所描述的，XObject 类型必须是 XObject 图像。
宽	整数	(必须) 样品图像的宽度。
高	整数	(必须) 样品图像的高度。
色彩空间	名字或数组	(所需的图像，除了那些使用 JPXDecode 滤波器；不得使用图像掩码) 样本图像指定的颜色空间可以是任何类型的色彩空间除开 Pattern 图形。 如果图像使用 JPXDecode 滤波器，入口是可选的： •如果色彩空间存在，在 JPEG2000 的数据中任何规格的色彩空间将被忽略。

- 如果色彩不存在，在 JPEG2000 的数据中任何规格的色彩空间将被使用。解码数组也被忽略除非 `ImageMask` 的值为真。

分量帧	整数	<p>(只要不是图像掩码和使用了 <code>JPXDecode</code> 滤波的图像，这些是必须的) 位的数值通常代替每个色彩分量。只有单独的值可能会被指定；位的数值对于所有色彩分量来说是相同的。固定值为 1, 2, 4, 8 以及（在 PDF 1.5 中）16。如果 <code>ImageMask</code> 值为真，这个入口是可选的，如果被指定，其值必须为 1。</p> <p>如果图像流使用一个过滤器，分量帧的值必须由数据样本组成，数据样本由滤波器提供。特别的是，<code>CCITTFaxDecode</code> 和 <code>JBIG2Decode</code> 滤波器总是提供 1 位的样本，<code>RunLengthDecode</code> 或者 <code>DCTDecode</code> 滤波器提供 8 位的样本，如果预测功能的函数被用到，<code>LZWDecode</code> 或者 <code>FlateDecode</code> 滤波器提供指定的样本。</p> <p>如果该图像流使用 <code>JPXDecode</code> 滤波器，这种入口是可选的，而且如果存在将会被忽略。位深度将在解析 JPEG2000 图像时被确定。</p>
<code>Intent</code>	名字	(可选; PDF 1.1) 色彩渲染的名字意在用于渲染图像（见“渲染目的”，230 面）默认值：在图形状态里的渲染值。
图像掩码	布尔	(可选) 一个标志决定了图像是否被当作图像掩码（见 4.8.5 节，“图像掩码”）。如果此标志为真，分量帧的值必须为 1，掩码以及色彩空间不能被指定；未被掩码的区域通过利用当前非平滑色彩进行绘制。默认值：0。
掩码	流或者数组	(除开图像掩码之外是可选的；对于图像掩码来说是不允许的；PDF1.3) 一个 <code>XObject</code> 图像定义了一个用于此图像（见 321 页）的图像掩码，或者一个指定了色彩范

围的数组用于此图像就如同一个色彩掩码（见“色彩掩码”在 321 页）。如果参数 `ImageMask` 的值是 1，则这个入口必须被调出（见附录 H 的说明 51）

解码	数组	(可选) 一个数字的数组描述了怎样把图像样本映射到图像色彩空间值的范围内（见“数组解码”314 页）。如果参数 <code>ImageMask</code> 的值为 1，则数组必须要么是 [0 1] 要么是 [1 0]；否则，它的长度必须是色彩组件数量的 2 倍，色彩组件提供给色彩空间。如果图像利用 <code>JPXDecode</code> 滤波器并且参数 <code>ImageMask</code> 为 0，解码将会失效。默认值：见“解码数组”第 314 页。
插入值	布尔	(可选) 一个标志，决定了是否进行图像插值（见“图像插值”316 页）。默认值：0。
替补	数组	(可选; PDF 1.3) 可选图像的数组是这个图像的字典（见“候选图像”在 317 页）。在数组里的有序元素没有意义。这个入口可能在 <code>XObject</code> 图像中不存在，它本身就是一个可选图像。
<code>Smask</code>	流	(可选; PDF 格式 1.4) 辅助的 <code>XObject</code> 图像定义一个软掩码图像（见“软掩码图像”第 522 页），软掩码图像被当作掩码定型的根源或者非透明掩码值，其在透明的图像模型中。该阿尔法在图形源参数的状态决定掩码值是否被解释为成型或不透明。如果存在，此项目如果有的话会覆盖了目前在软掩模图形的状态以及图像的掩码项。（但是，其他于透明有关的图形状态参数—混合模式和 <code>a</code> 常数—仍然有效。）如果没有 <code>Smask</code> ，图像没有相关的软掩码（尽管当前的图形状态仍可申请软掩码）。
<code>SmaskInData</code>	整数	(对于使用 <code>JPXDecode</code> 过滤器的图像来说是可选的，否则毫

无意义；PDF 1.5）代码确定软掩码图像（见“软掩码图像”第 522 页）是如何对图像样本进行编码这个问题应该被确定为：

0 如果存在，对软掩码图像信息的编码应该被忽略。

1 图像数据流包含软掩码编码值。一个应用程序可以创造软掩码图像。软掩码图像由一些信息得到，这些信息在透明图像模型中被当作掩码模型或者非透明掩码。

2 图像数据流包含事先混合了背景色的色彩渠道；图像数据也包含非透明渠道。一个应用程序可以通过来自非透明渠道信息的色板入口来创造一个软掩码图像，非透明渠道信息在透明模型中常被用作掩码成型或者非透明掩码。如果这个入口是个非零值，**SMask** 不应该被指定。见 3.3.8 节，“JPXDecode 滤波器”

默认值：0。

名字 名字 (在 PDF1.0 里是必须的；其他的可选) 此名字与在当前资源字典的 **XObject** 子字典里的 **XObject** 图像有关联。(见 3.7.2 章，“资源字典”)

注：这个入口是废弃的并且它的用法不再推荐（见附录 H 的说明 52）

StructParent 整形 (如果图像是一个结构性的内容项则是必须的；PDF 1.3) 在结构父项树中图像入口的整形节点。(见“从内容要素项目中寻找结构元”797 页)。

ID 串 (可选；PDF 格式 1.3；间接性优先) 图像父网捕捉内容设定中的数字标识符（见第 10.9.5，“对象有关网络抓取属性”）。

OPI	字典	(可选; PDF 1.2) OPI 版本的图像词典 (见 10.10.6 节, “开放前印界面 (OPI)”)。如果参数 ImageMask 为 1, 此入口被忽略。
元数据	流	(可选; PDF 1.4) 元数据流中包含的图像元数据 (参见 10.2.2 节, “元数据流”)。
OC	字典	(可选; PDF 1.5) 一个可选的内容组成员或可选的内容字典 (见第 4.10 节, “自选内容”), 为这种 XObject 图像指定可选内容属性。在图像处理之前, 其可见度是取决于这种入口的。如果它被确定为不可见的, 则入口图像会被跳过, 就像没有 Do 参数来调用它。

例子 4.28 定义了一个图像样本, 256 宽, 256 高, 在 DeviceGray 色彩空间每个样本有 8 比特。它在页面上绘制了图像, 图像的左下角定位在坐标 (45, 140), 位于当前用户空间, 并调整为用户空间的 132 个单位高度。

例子 4.28

```
20 0 obj % Page object
<< /Type /Page
/Parent 1 0 R
/Resources 21 0 R
/MediaBox [0 0 612 792]
/Contents 23 0 R
>>
Endobj

21 0 obj % Resource dictionary for page
<< /ProcSet [/PDF /ImageB]
/XObject << /Im1 22 0 R >>
>>
Endobj
```

```
22 0 obj % Image XObject
<< /Type /XObject
/Subtype /Image
/Width 256
/Height 256
/ColorSpace /DeviceGray
/BitsPerComponent 8
/Length 83183
/Filter /ASCII85Decode
>>
stream
9LhZ19h\GY9i+bb;, p:e;G9SP92/) X9MJ>^:f14d;, U(X8P;c0;G9e];c$=k9Mn\]
...Image data representing 65,536 samples...
8P;c0;G9e];c$=k9Mn\] ^>
endstream
endobj

23 0 obj % Contents of page
<< /Length 56 >>
stream
q % Save graphics state
132 0 0 132 45 140 cm % Translate to (45, 140) and scale by 132
/Im1 Do % Paint image
Q % Restore graphics state
endstream
endobj
```

解码数组

图像的最初数据流分解成整数域 0 至 $2^n - 1$ ，其中 n 是图像字典 BitsPerComponent 项的值。图像的解码阵列指定的每个整数线性映射组件值为

一个数字，在适当图像的色彩空间将作为一个组件的值。

在解码阵列的每个数值对样本值的范围指定了上限和下限，这些样本值是在图像中有被映射。解码数组包含一对数值，其样本值为图像中的域名映射。数组包含一个解码对，其中的每个数字由指定的颜色空间部分图像的色彩项确定。对于每种颜色分量的映射是线性转变的，也就是说，它使用的线性插值公式如下：

$$\begin{aligned}y &= \text{Interpolate}(x, x_{\min}, x_{\max}, y_{\min}, y_{\max}) \\&= y_{\min} + \left((x - x_{\min}) \times \frac{y_{\max} - y_{\min}}{x_{\max} - x_{\min}} \right)\end{aligned}$$

一般来说，这个公式是用来将 x 值转换为相应的 y 值， x 的范围是 x_{\min} 到 x_{\max} ， y 的范围是 y_{\min} 到 y_{\max} ，由点 (x_{\min}, y_{\min}) 和点 (x_{\max}, y_{\max}) 确定这个转换。虽然这个转换的值范围是 x_{\min} 到 x_{\max} 并且不需要 $x_{\min} < x_{\max}$ ，需要注意的是内插法用于色彩转换，比如解码数组，需要 $x_{\min} < x_{\max}$ 并且裁剪 x 的值以至于对于所有的 $x \leq x_{\min}$, $y = y_{\min}$ ，并且对于所有的 $x \geq x_{\max}$, $y = y_{\max}$ 。

对于形式为 $[D_{\min} D_{\max}]$ 的解码数组，可以写成

$$\begin{aligned}y &= \text{Interpolate}(x, 0, 2^n - 1, D_{\min}, D_{\max}) \\&= D_{\min} + \left(x \times \frac{D_{\max} - D_{\min}}{2^n - 1} \right)\end{aligned}$$

其中

n 是分量帧的值

x 是输入值，范围是 0 到 $2^n - 1$

D_{\min} 和 D_{\max} 是被解码数组确定的值

Y 是输出值，在图像色彩空间里被诠释

为 0 值的样品映射到 D_{min} , 同时为 $2n - 1$ 值的样品映射到 D_{max} , 中间值之间的线性映射范围是 D_{min} 到 D_{max} 。表 4.40 列出了解码与使用数组的各种默认颜色空间。对于大多数的色彩空间, 列在表中的解码阵列映射到允许的完整范围元件值内。对于一个索引颜色空间中的默认解码阵列, 确保组件的索引颜色值不变。

表 4.40 默认的解码数组

色彩空间	解码数组
DeviceGray	[0. 0 1. 0]
DeviceRGB	[0. 0 1. 0 0. 0 1. 0 0. 0 1. 0]
DeviceCMYK	[0. 0 1. 0 0. 0 1. 0 0. 0 1. 0 0. 0 1. 0]
CalGray	[0. 0 1. 0]
CalRGB	[0. 0 1. 0 0. 0 1. 0 0. 0 1. 0]
Lab	[0 100 amin amax bmin bmax], 其中 amin, amax, bmin, 和 bmax 的值对应着图像色彩空间范围数组的值。
ICCBased	跟色彩空间的图像 ICC 曲线的范围值相同。
Indexed	[0 N], 其中 N = 2n - 1
Pattern	(不容许有图像)
Separation	[0. 0 1. 0]
DeviceN	[0. 0 1. 0 0. 0 1. 0 … 0. 0 1. 0] (一对每种颜色的组成部分)

指定一种映射是可能的，这种映射通过指定一个 `Dmin` 值大于 `Dmax` 来使样本色强度反转。例如，如果图像的色彩空间是 `DeviceGray` 并且解码数组是 `[1.0 0.0]`，为 0 的输入值被映射到 1.0（白）；一个输入值是 $2^n - 1$ 映射到 0.0（黑色）。

色彩组件参数 `Dmin` 和 `Dmax` 在那种组件中不用降低值的范围。例如，如果应用程序作为它的本机映像格式样本的 6 位数字，它可以代表那些 PDF 格式的样本中 8 位，设置两个未使用的每一个高序位的样本为 0。然后字典图像应指定一个解码阵列 `[0.00000 4.04762]`，将输入值从 0 到 $2^n - 1$ 映射到值为 0.0 到 1.0（ 4.04762 被约等于 $255 \div 63$ ）。如果一个组件输出值超出允许的范围，它会自动调整至最接近的允许值。

图像内插法

当一个源图像的分辨率明显高于输出设备，每个源样本涵盖许多设备像素。结果，图像可以出现锯齿形或块状。这些可视化的组件可将减少图像插值算法的渲染。而是涵盖所有像素源具有相同的色样，图像插值尝试产生平稳过渡相邻样本值。图像插值启用通过设置在图像字典真正插补项。它默认是禁用的，因为它可能会增加所需的呈现图像的时间。

备用图像

备用图像（PDF 1.3）提供了一个直接向后兼容的办法包括图像的多个版本在不同用途的 PDF 文件。这些变异的图像可能表示不同，例如，在解析空间或彩色空间。主要目标是减少维护单独版本为低分辨率的屏幕观看和高分辨 PDF 文件印刷的需求。

在 PDF 1.3，基本图像（即解析字典提到的 `XObject` 图像）可以包含一个候补项。此项的值是一个候补数组字典，指定不同的图像库代表。每个图像字典包含一个变量和 `XObject` 图像指定的属性。表 4.41 显示了一个备用图像词典的内容。

表 4.41 备用图像字典的入口

关键字	类型	值
Image	流	(必须) 可选图像的 XObject 图像。
DefaultForPrinting	布尔	(可选) 一个标志说明是否此备用形象是要用于绘制的默认版本。一个最基础的候补图像是如此的明确。如果没有备用的此项设置为 1, 图像库用于绘制。
OC	字典	(可选; PDF 1.5) 可选的内容组 (见第 4.10.1 节, “可选内容组”) 或可选的内容成员词典 (见“任择会员词典”内容在 335 页), 方便于候补图像的选择。

例如 4.29 显示了一个备用的图像。该图像库是一个灰度图像, 是一个存储在 Web 服务器的备用高分辨率 RGB 图像。

例子 4.29

```
10 0 obj % Image XObject
<< /Type /XObject
/Subtype /Image
/Width 100
/Height 200
/ColorSpace /DeviceGray
/BitsPerComponent 8
/Alternates 15 0 R
/Length 2167
/Filter /DCTDecode
>>
stream
...Image data...
endstream
```

```
endobj
15 0 obj % Alternate images array
[ << /Image 16 0 R
/DefaultForPrinting true
>>
]
endobj
16 0 obj % Alternate image
<< /Type /XObject
/Subtype /Image
/Width 1000
/Height 2000
/ColorSpace /DeviceRGB
/BitsPerComponent 8
/Length 0 % This is an external stream
/F << /FS /URL
/F (http://www.myserver.mycorp.com/images/exttest.jpg)
>>
/FFilter /DCTDecode
>>
stream
endstream
endobj
```

在 PDF 1.5，可选的内容（见第 4.10 节）可以用来方便地在可选图像之间进行选择。如果

一个 XObject 图像包含了可选入口和一个 OC 入口，选择要使用的图像取决于如下：

1. 如果图像的 OC 入口指定基本图像是可见的，则该图像显示。
2. 否则，候补项中指定的候补入口将被复查，第一替补包含有一个 OC 入口，其

指明其内容应该可见的显示。 (候补图像没有 OC 入口)。

4.8.5 图像掩码

通常，在不透明的成像模型，图像标记所有他们占用的地区，好象用不透明方式绘制。图像的所有部分，无论是黑色还是白色，灰色，或其他颜色，完全掩盖之前任何可能存在的

页面上同样的地方。然而在页面布局应用程序，覆盖图像的背景然后放置掩码图像于不同的背景色下以至于存在的背景会出现在掩码图像的整个区域，这种做法是很普遍的。

一些 PDF 特性对于获取这种掩码效果是支持的（见附录 H 的说明 53）：

- 在图像字典的 `ImageMask` 入口，支持所有版本的 PDF，指定了图像数据，这些数据对于在当前色彩的绘制方面被当作掩码模具。
- 在图像字典的掩码入口（PDF 1.3）可能指定了分离的 `XObject` 图像，这种图像被当作掩码模具使用，用来指定用来绘制以及去除掩码的图像区域。
- 另外，掩码入口（PDF 1.3）可指定用来掩去的色彩范围，无论他们出现在哪里。这种技术被称为颜色键屏蔽。

注：虽然掩码入口是一个 PDF 1.3 的功能，其一般效果是模拟在早期版本的 PDF 通过定义一个剪辑路径内。然而，如果错误地实施可能会导致是非常复杂的剪辑路径（或者如果有多个剪辑路径）。另一种方法来达到 PDF 1.2 掩码的功能是在图案剪下定义图片作为当前的颜色，然后绘制明确的掩码，其参数 `ImageMask` 入口值为 1。在任何情况下，PDF 1.3 功能让在页面上放置的掩码图像存在，无论多么复杂的剪辑路径。

在透明的成像模型，掩码效果的第四种，软掩码，对于 `SMask` 入口（PDF 1.4）或者 `SMaskInData` 入口（PDF 1.5）是支持的；见 7.5.4 节，“指定的软掩码”有进一步的讨论。

模具掩码

图像面具（ImageMask 入口值为 1 的 XObject 图像）是一个单色影像，其中每个样品是由一个位指定。然而，与被绘制成不透明的黑色和白色不同，图像掩码被当作模具掩码，即部分透明部分不透明。在图像中的样本值不代表黑色和白色像素；它们在页面上指定区域，以至于要么被当前色彩掩码要么被掩去掉（不会产生掩码）。被掩去的区域保留了之前的内容。效果如同通过去除模具的方法在当前色彩执行绘制，以至于绘制图到达一些区域的页面，而在另外的一些图里被掩去掉。

普通图像与掩码图像存在以下不同点：

- 图像词典不包含 ColorSpace 入口，因为样本值代表掩码特性（每个样品 1 位），而不是颜色。
- BitsPerComponent 参数的值必须为 1。
- 解码入口决定样本来源是如何被解析。如果解码数组是 [0 1]（图像掩码默认的），一个 0 的样本值标志着当前色彩的页面，1 表示之前没有改变的内容。如果解码数组是 [1 0]，则反过来解释。

模具掩盖的最重要的用途之一是绘画字形符号用以表示位图。利用如同模具掩码的符号传送它的“黑色”位到页面上，留下没有变化的“白色”位（仅仅只有背景）。其中的原因在 5.5.4 节中有所讨论，“类型 3 字体”，一种图像掩码，而不是图像，应该常常被用于绘制符号位图。

注：如果图像插值（见“图像插值”316 页）在模具掩码过程中有需求，其效果是平滑掩码边缘，而不是插入该画的颜色值。这个效果可以减少锯齿形的外观和低分辨率模具掩码。

明确掩码

在 PDF 1.3，在图像字典的掩码入口可能是图像掩码，如之前“模具掩码”所描

述的，用来跟明确掩码那样服务于基础色。基础色和图像样吗不需要有相同的解决办法（宽和高的值），但是自从所有图像在用户空间区域被定义，界面上的出现分界重合，也就是说，他们将相互叠加。图像掩码显示于页面中将要绘制的地方以及将要被屏蔽掉的地方（余下的不变）。未被掩码的区域通过相应的基色部分来绘制；掩码区域则不是。

主要色彩掩码

在 PDF 1.3，在图像字典的掩码入口可能是一个数组，用于指定被掩掉的色彩范围。图像里的样本超过了这个范围将不会被绘制，只容许存在的背景色予以显示。其效果如同视频技术中有名的色度键技术。

对于色彩掩码，掩码入口的值是一个 $2*n$ 的整数数组， $[min1\ max1\ \dots\ minn\ maxn]$ ，其中 n 是图像色彩空间中的色彩组件数值。每个整数必须在范围 0 至 $2\text{BitsPerComponent} - 1$ 内，在用解码数组解码之前代表颜色值。如果所有的色彩组件在解码之前进入指定区域（即对于所有 $1 \leq i \leq n$, $mini \leq ci \leq maxi$ ），图像样本将被掩去（不是绘制）

注：当色彩掩码被指定，DCTDecode 滤波器的使用是不推荐的。DCTDecode 是一个有损过滤器，这意味着输出一个近似原始输入的数据。因此，该过滤器的使用可导致图像样本的颜色值略有变化，可能导致样本会出现被意外的绘制行为。

4.8.6 内敛图像

正如 4.8.4 描述的可选择 XObjects 图像，“图像字典”，一个可能被内敛图像指定的样本图像。这种类型的图像在内容流里被直接定义，内容流将被绘制而不是作为单独的部件。因为内敛格式让应用程序在管理图像数据更具灵活性，因此它只能被用于小图像里（小于或等于 4KB）。

内敛图像对象在内容流上是被某些参数分隔的，这些参数是：BI（开始图像），ID（图像数据），EI（图像结尾）。这些操作数在表 4.42 予以总结。BI 和 ID 归

类了一系列关键值，这些值指定了图像的特点，比如它的尺寸和色彩空间；图像数据介于 **ID** 和 **EI** 操作数之间。因此格式对于流对象是相似的，比如 **XObject** 图像：

BI

…Key-value 对…

ID

…图像数据…

EI

表 4.42 内嵌图像操作数

操作对象	操作数	描述
—	BI	开始一个内嵌图像对象。
—	ID	开始的内嵌图像对象的图像数据。
—	EI	结束一个内嵌图像对象。

内嵌图像对象可能不能嵌套；也就是说，两个 **BI** 操作数不可能在没有 **EI** 操作数的情况下出现。同样的，**ID** 操作数只有在 **BI** 和它的 **EI** 之间才能出现。除非图像利用 **ASCIIHexDecode** 或 **ASCII85Decode** 之间的一种滤波器，**ID** 操作数应该在单独的空白字符的后面，下一个字符应该解析为图像数据的第一个字节。

关键值对出现在 **BI** 和 **ID** 操作数之间这个现象在 **XObject** 图像的字典部分也出现过（虽然语法不同）。表 4.43 说明了入口对于内嵌图像是有效的，其意义跟字典流（表 3.4 第 38 页）或者图像字典（表 4.39）相同。列表中除了入口，其他的没有被忽略；特别的是，存在于流或者图像字典里的类型、子类型以及长度入口不是必须的。为了方便，表里显示的缩写可能被用于分解完全的主要值。表 4.44 显示了附加的缩写，这些缩写可以用于对色彩空间和滤波器的命名。然而需要注意的是，这些缩写只有在内嵌图像里才是有效的；它们可能不能用于 **XObjects**

图像。还需要注意的是 JBIG2Decode 和 JPXDecode 没有在表 4.44 列出是因为这些滤波器只有在 XObjects 图像里可以被用到。

表 4.43 在内敛图像对象的入口

全名	缩写
BitsPerComponent	BPC
ColorSpace	CS
Decode	D
DecodeParms	DP
Filter	F
Height	H
ImageMask	IM
Intent (PDF1.1)	没有缩写
Interpolate	I
Width	W

表 4.44 内敛图像对象的附加缩写

全名	缩写
----	----

DeviceGray	G
DeviceRGB	CS
DeviceCMYK	CMYK
Indexed	I
ASCIIHexDecode	AHx
ASCII85Decode	A85
LZWDecode	LZW
FlateDecode	FI
RunLengthDecode	RL
CCITTFaxDecode	CCF
DCTDecode	DCT

由色彩空间入口指定的从色彩空间可能是任何标准的设备色彩空间 (DeviceGray, DeviceRGB, 或者 DeviceCMYK)。它可能不是 CIE 色彩库或者特殊的色彩空间，有限类型的色彩空间索引是个例外，这种空间的色彩空间库是设备空间并且其色彩表是被一个字符串指定的（见“色彩空间索引”，在 232 页）。在 PDF1.2 的开头，色彩空间入口的值可能也是当前资源字典中色彩空间子字典里的色彩空间值（见 3.7.2 节，“资源字典”）。在这里，名字可以指定任何色彩空间，这些色彩空间可以被用于 XObject 图像。

注：一些名字，如 DeviceGray, DeviceRGB, 和 DeviceCMYK (缩写为 G, RGB, CMYK)

经常直接指定相应的色彩空间；它们从不指定色彩空间子字典中的资源。

在内敛图像里的图像数据可能被使用任何标准 PDF 滤波器的进行编码。介于 ID 和 EI 操作数的比特被当作流对象数据一样处理（见 3.2.7 节，“流对象”），即使它们并不按标准流语法来。（这对于一般规则来说是个例外，即在内容流的数据根据标准 PDF 语法来解析）。

例子 4.30 显示了内敛图像样本，17 宽，17 高，在 DeviceRGB 色彩空间用 8 比特每组件来表示。图像通过用 LZW 和 ASCII base-85 来编码。参数 CM 用于扫描用户空间中 17 个单位的长度和高度以及定位其在 (298, 388) 位置。操作数 q 和 Q 囊括 cm 操作数用来限制它的作用来打到改变图像的目的。

例子 4.30

```
q % Save graphics state
17 0 0 17 298 388 cm % Scale and translate coordinate space
BI % Begin inline image object
/W 17 % Width in samples
/H 17 % Height in samples
/CS /RGB % Color space
/BPC 8 % Bits per component
/F [/A85 /LZW] % Filters
ID % Begin image data
J1/gKA>. ]AN&J?] -<HW] aRVcg*bb. \eKAdVV%/PcZ
...Omitted data...
R. s (4KE3&d&7hb*7[%Ct2HCqC^>
EI % End inline image object
Q % Restore graphics state
```

4.9、Xobjects 格式

Xobject 格式是 PDF 的内容流，它是任何图像对象序列的独立描述（包括路径对

象，文本对象以及样本图像)。XObject 格式可能是被绘制了多次—要么在几个页上面要么在同页上几个点上面—并且每次产生相同的结果，只根据调用时的图形状态。它不仅是 pdf 文件中共享的定义，并且在适当的环境下，PDF 用户应用程序能够通过获取渲染的结果来优化执行过程。

注：术语格式还指完全不同的对象，一种完全不同的格式（有时被称为 AcroForm），在 8.6 节讨论，“相互作用的格式”，然而 XObjects 格式在这章被描述为 PostScript 语言中相应的格式概念，交互格式是等同于 PDF 中的页面工具。任何字符格式不合格的用法被理解为指定一个交互格式；这里描述的格式类型经常是明确地指 XObject 格式。

XObjects 类型有各种用法：

- 正如其名称所示，XObject 格式可以作为一个完整的模板页面。例如，一个要绘制填充格式的程序可以首先绘制填充的模板如同 XObjects 格式然后在它上面绘制不同种类的信息。
- 任何图形元素是可以重复使用，如公司徽标或电脑辅助设计系统输出的标准组件，可以被定义为一个 XObject 格式。
- 某些文档元素不是一个页面的内容，如部分注释显示（见第 8.4.4 节，“外观流”），是代表 XObjects 格式。
- 一个专门类型的 XObject 格式，称为一 XObject 组（PDF 1.4），可用于组织图形元素作为不同用途的单位（见 4.9.2 节，“XObjects 组”）。特别的是，XObjects 组用来定义用于透明图像模型的透明组以及软掩码（见“软掩码字典”第 520 面和 7.5.5 节，“透明 XObjects 组”）。
- 另一种 XObjects 格式的特殊类型，XObjects 参考型（PDF 1.4），可用于将内容从一个 PDF 文件导入到另一个 PDF 文件（见 4.9.3 节，“参考型 XObjects”）。

使用的形式 XObjects 需要两个步骤：

- 1。界定了 XObject 格式的外观。XObject 格式是 PDF 内容流。流的字典部分（成为字典格式）包含了有关 XObject 格式的描述性信息；流的本身描述了可以产生其外观的图形对象。字典格式的内容描述于 4.9.1 节，“字典格式”。
- 2。绘制 XObject 格式。Do 算子（见 4.7 节，“外部对象”）绘制了 XObject 格式，其名字是作为操作数。（名字在当前资源字典的 XObject 子字典里被定义）。调用这个操作数之前，内容流应该在图形状态里设置合适的参数。特别的是，它应该改变当前转换矩阵来控制位置，大小以及用户空间 XObject 格式的方向。

每个 XObject 格式定义了自己所在的坐标系统，称为空间格式。在格式字典里的 BBOX 参数入口在空间格式里被定义，如同任何 XObject 格式内容流用过的坐标系统，比如路径坐标。

字典格式的矩阵入口指定了由空间格式到当前用户空间的映射。每次 XObject 格式被 Do 操作数绘制，这种矩阵将当前转换矩阵联系起来用于定义由空间格式到设备空间的映射。（这种在模式字典里是不同于矩阵入口，其将模式空间映射到内容流原始的用户空间）。

当你运营商应用到了 XObject 的形式，它执行以下任务：

- 1。保存当前的状态图形，如同通过调用 q 操作数（见第 4.3.3 节，“图形状态操作数”）
- 2。将字典格式矩阵入口的矩阵与当前转换矩阵（CTM）连接起来
- 3。根据字典格式的 BBOX 入口进行裁剪
- 4。绘制由内容流格式指定的图形对象
- 5。恢复已保存的图形状态，如同调用 q 操作数（见第 4.3.3 节，“图形状态操作

数”)

除以上所述，当 `Do` 被调用且图形状态产生作用时格式原始的图形状态被图形状态继承。

4.9.1 字典格式

每个 `XObject` 格式都有一个类型格式，其决定了字典格式入口的类型和意义。绘制时，只有一种类型格式，1 类型，已被定义。`XObject` 字典格式可能包含表 4.45 所示的入口，还包含一般跟所有流相同的入口（见表 3.4 第 38 页）。

表 4.45 类型 1 格式字典的附加入口

关键字	类型	值
Type	名字	(可选) 由这种字典描述的 PDF 对象类型；如果存在，必须是 <code>XObject</code> 。
Subtype	名字	(必须) 由这种字典描述的 PDF 对象类型；必须是 <code>Form</code> 。
FormType	整数	(可选) 代码确定了由字典描述的 <code>XObject</code> 格式类型。唯一的有效值在 <code>publication</code> 的值是 1 的情况下被确定。默认值：1。
BBOX	矩阵	(必须) 在格式坐标系统里的 4 数字数组（见上），提供了各自 <code>XObject</code> 边界框格式的左，右，上边缘的坐标值。这些边界被用于剪辑 <code>XObject</code> 格式以及用来确定它的大小。
矩阵	数组	(可选) 6 个数的数组指定了矩阵格式，其用来将空间映射到用户空间（见 4.2.3 节，“转换矩阵”）。默认值：单位矩阵 [1 0 0 1 0 0]。

Resources	字典	(可选的但强烈建议; PDF 1.2) 一个指定任何资源(比如字体和图像)的字典对于 XObject 格式来说是必须的(见 3.7 节,“内容流和资源”)。
		在 PDF 1.1 和更早版本,所有被命名的用于 XObject 个格式的资源必须包含在每个页对象的资源字典里,无论它们是否在 XObject 格式的资源字典出现过。它常用于指定这些在 XObject 资源字典里的资源,来指定哪种资源是被用于 XObject 格式内部。如果资源在两种字典里都有,它应该在两个位置具有相同的名字。
		在 PDF 1.2 和更晚的版本,XObject 格式可以单独于它们出现过的内容流,这虽然不是必须的,但是强烈建议。在独立的 XObject 格式里,XObject 格式的资源字典是必须的并且它包含了所有被 XObject 格式用到的资源名称。这些资源并不是外部内容流资源字典里的,虽然资源字典流指定了 XObject 格式。
Group	字典	(可选; PDF 1.4) 组的属性字典显示了 XObject 格式的内容,有这种行为的才能称为一个组并能指定组的属性(见 4.9.2 节,“XObject 组”)。
		注:如果一个 Ref 入口(见下)存在,组的属性也适用于被入口导入的外部页,其容许这种导入页在没有进一步修改的情况下作为一个组。
Ref	字典	(可选; PDF 1.4) 参考字典确定了由另一个 PDF 文件导入的页,并作为 XObject 格式如同代理那样服务的对象(见 4.9.3 节,“XObject 参考”)。
Metadata	流	(可选; PDF 1.4) 元数据流包含了 XObject 格式的元数

据（见 10.2.2 节，“元数据流”）。

PieceInfo	字典	(可选; PDF 1.3) 页面字典与 XObject 格式相关联（见 10.4 节，“页字典”）。
LastModified	数据	(如果 PieceInfo 存在，则是必须的；否则可选; PDF 1.3) 日期和时间（见 3.8.3, “日期”）当 XObject 格式内容被修改。如果页字典 (PieceInfo) 存在，日期的修改被用于确定哪种应用程序的日期字典包含了相应的当前格式内容。（见 10.4 节，“页字典”）。
StructParent	整数	(如果 XObject 格式是结构性内容，则是必须的； PDF 1.3) 在结构性父母树里的 XObject 格式入口的整数关键字（见“寻找来自内容项目里的结构性元素”，797 页）。
StructParents	整数	(如果 XObject 格式包含了标志了的内容序列则是必须的，内容序列是结构性内容项目； PDF 1.3) XObject 格式入口的整型关键字是在结构性父母树里的（见“在内容项里寻找结构性元素”在 797 页）。
注：大部分情况下 StructParent 或者 StructParents 入口可能会存在。一个 XObject 格式可以是在它的全部内容项里，也可以是一个标志了内容序列的容器。		
OPI	字典	(可选; PDF 1.2) 对于 XObject 格式来说是一个 OPI 版本的字典（见 10.10.6, “开放式前绘制界面 (OPI)”）。
OC	字典	(可选; PDF 1.5) 一个可选的内容组或可选的内容成员字典（见 4.10 节，“可选内容”）为 XObject 格式指定了可选的内容属性。在格式被处理之前，它的可见度在这个入口的基础上被确定。如果它被确定为不可视，整个

格式会被忽略，如同没有 Do 操作数来调用它。

Name	名字	(在 PDF 1.0 是必须;其他的可选) XObject 格式在当前资源字典里 XObject 子字典引用的名字 (见 3.7.2 章, “资源字典”)。
------	----	--

注：这个入口是过时的，并且它的用法不再推荐（见附录 H 的说明 54）。

例如 4.31 显示了一个简单的 XObject 格式，其正在每个边绘制一个填充的 1000 个单位正方形。

例子 4.31

```
6 0 obj % Form XObject
<< /Type /XObject
/Subtype /Form
/FormType 1
/BBox [0 0 1000 1000]
/Matrix [1 0 0 1 0 0]
/Resources << /ProcSet [/PDF] >>
/Length 58
>>
stream
0 0 m
0 1000 l
1000 1000 l
1000 0 l
f
endstream
endobj
```

4.9.2 Xobjects 组

一个 XObject 组 (PDF 1.4) 是一个特殊的 XObject 格式类型，它可以针对不同的目的用来集中图形要素使其成为一个单位。它是通过在字典格式可选组入口来区分的 (见 4.9.1 节，“字典格式”)。这个入口的值是一个附属组属性字典，用来描述组的属性。

如表 4.46 所示，每个 XObject 组有一个子类型组 (被在组属性字典的 S 入口指定)，其指定了字典剩余入口的格式和含义。只有一个此子类型是当前定义的，一个透明的 XObject 组 (子类型透明) 代表一个透明组，用于透明图像模型 (见 7.3 节，“透明组”)。这个字典类型的剩余内容在 7.5.5 节描述，“XObjects 透明组”。

表 4.46 所有属性字典组相同的入口

关键字	类型	值
Type	名字	(可选) 这个字典所描述的 PDF 对象类型；如果存在，必须是属性字典组里的成员。
S	名字	(必须的) 组的子类型，定义了组的类型，它的属性由字典描述并且决定了字典剩余入口的格式和含义。在 PDF1.4 定义的子类型组是透明的；见 7.5.5 节，“透明的 XObjects 组”，是这种字典类型的剩余内容。其他组的子类型可能在未来中添加。

4.9.3 Xobjects 关联

XObjects 关联 (PDF1.4) 使一个 PDF 文件从另一个文件中导入内容。发生了关联的文件被称为容器文件；内容被导入的文件是目标文件。目标文件可以驻留在容器文件的外部，或者可以被它包含如同嵌入的文件流 (见 3.10.3 节，“嵌入文件流”)。

在容器文件里的 XObject 关联是一个 XObject 格式，它包含了字典格式里的可选 Ref 入口，如下所述。这种 XObject 格式像代理一样提供服务，其可以在导入内容处被显示或者被绘制。代理可能由导入内容的低分辨率图像组成，一些描述性文件会描述它，一个在此处显示的灰色框，或者任何其他类似的位置标识符。PDF 使用者不能认出 Ref 入口很简单地显示或者绘制代理就如同普通的 XObject 格式（见附录 H 的说明 55）。那些执行 XObjects 关联的可以在导入内容的地方使用代理，如果后者是不能得到的。一个应用程序可能也提供用户界面来对导入内容的链接进行编辑和升级。

在目标文件里导入的内容由一个单独的，完整的 PDF 页组成。关联字典可以指定在关联 XObject 格式字典的 Ref 入口值（见 4.9.1 节，“字典格式”）。Ref 入口的存在区分了 XObjects 关联与其他 XObjects 格式类型。表 4.47 表明了关联字典的内容。

表 4.47 关联字典里的入口

关键字	类型	值
F	文件规格	(必须) 文件包含目标文件。
Page	整数或者文档字符串	(必须) 页面索引或者页面标签（见 8.3.1 节，“页面标签”）确定目标文件的页面，目标文件包含了导入的内容。注意的是如果关联页面在目标文件里是改变的或被取代的，当关联创建之后，关联此时就是一个弱项并且可以无意中失效。
ID	数组	(可选) 两个字符的数组构成一个文件标识符（见 10.3 节，“文件标识符”），是为了文件包含目标文件。这个入口的使用改善了应用程序查找之后文件的机会以及当关联被创建后文件已经改变时容许其对用户发出警告。

当输入内容取代了代理，它则根据代理对象的转换矩阵转换以及缩短到它的边界框的边界以内，如被代理表格字典中的矩阵和 BBox 定义的那样（见 4.9.1，“表

格字典”)。代理对象的矩阵和边界框的这种结合隐式定义了输入页的边界框。这个边界框一般与输入页面的提取框或者技能框一样（见 10.10.1，“页边界”），但是它不被要求去相应于任何一个定义的页边界。如果代理对象结构字典包含了一个组项，这个定义的组属性也应用于输入页，其允许输入页可以被作为一个没有进一步修改的组。

XObject 印刷参考

当打印一个含有对 XObject 引用的页面的时候，一个应用可能会产生以一些条目，这依赖于该应用的兼容性，用户的引用和打印工作的性质：

- 由应用 XObject 指定的导入内容。
- 引用 XObject 作为应用的代理。
- 如果存在，会从引用 XObject 的字典中取得 OPI 代理或者替代图片、
- 当生成 OPI 在发送脚本中被注释的时候，导入的内容或者引用的 OPI 可能会安排以替代 OPI 代理。

特别注意事项

当引用的 XObject 和其他的 PDF 特征有交互的时候，应该做一些特殊考虑。

- 当该页面导入的引用 XObject 包含注释的时候，所有那些包含可打印，不可隐藏而且可见的外观流，必须被包含在导入的页面里面。如果该代理是导入页面的一个快照，它也必须包含注释的完整内容。这些外表因此，必须转换成部分代理的内容流，要么作为 XObject 的子项或或者将他们压缩，使他们直接进入内容流。
- 当把一个页面导入另一个引用 XObject 的文档中时，和该页相关联的逻辑信息结构应该被忽略。在一个具有多页的目标文档中，当导入页面时一个文档的大型结构元素的一部分的时候，结构元素就会出现，这中结构元素不能直接被 有意义的纳入到改文档的结构元素中。在一个只有一页或者是由各个相互独立的页组成的目标文档中，导入页面的逻辑结构完全是自足的。在这种情况下，可以将该信息结构纳入到包含文件中。然后，PDF 没有为一个一个文档的逻辑结构目录直接引用另外一个文档的逻辑结构目录提供任何机制。

4.10、可选内容

可选内容是指在一个 PDF 文档中可以被 PDF 的作者或者读者有选择性的阅读或者隐藏的内容。这种功能在如 CAD 制图，分层次艺术作品，地图和多语言文档中都很有用。

接下来的章节将讨论用于完成可选章节的 PDF 结构。

- 4.10.1 节，“可选内容组”，描述用于控制内容能见特性的主要结构。
- 4.10.2 节，“标记图像内容选项”，描述文档中的单个内容如何向属于多个可选的组一样去描述他们自己。
- 4.10.3 节，“可选内容的配置”，描述一个可选内容组的状态时如何设置的。

4.10.1、可选内容组

可选内容组是一个代表图像内容集合的字典，这个图像集合可以被用户或者浏览应用那个动态的设置为可见或者不可见。属于这样一个群组的图像可以位于该文档的任何位置。他们不需要以绘制顺序来连接，甚至不用属于同一个内容流。表 4.48 展示了一个可选内容群组字典中的条目。

表 4.48 可选内容群组字典中的条目

关键域	类型	取值
Type	名称	(必须的) 该字典所描述的 PDF 对象的类型，对于一个可选的内容群组字典来说，必须是 OCG。
Name	文本字符串	(必须的) 可选内容群组的名称，适合在用户应用界面中显示。

关键域	类型	取值
Intent	名称或数组	(可选) 一个单一的意图名称或一个包含所有组合名称的数组。PDF1.5 定义了两个名称，View 和 Design，表示该组中将要使用的图形。以后的版本

可能会定义其他的。一个应用程序只能选择具有特定意图的组而忽略其他的。 默认值: View

Usage	字典	(可选的) 用法字典描述被该组所控制的内容的性质。可以通过它基于外部因素自动控制群组状态的特性来使用它。
-------	----	--

在最简单的格式中，每个字典包含一个 Type 项和 Name 项，用于在用户界面显示。他也可能包含一个 Intent 项用于描述使用意图和一个 Usage 项来描述它的内容的性质。在一个文件中，单独的内容元素指定可选的内容组或者对可是效果有影响的内容组。如果一个内容的可视程度受一个给定的可选内容组的影响就说他属于该组。

每一个组都会被分配一个状态，是 ON、或者是 OFF。状态自身不是 PDF 文档的一部分，但可以通过程序或者用户浏览接口去改变内容的可见性。当一个文档被第一次打开的时候，该组的状态会安装文档内部默认的状态进行初始化。

在通常情况下，当该组处于 ON 状态的时候，该组的内容是可见的，当该组出去 OFF 状态的时候，该组的内容不可见。在更复杂的情况下，内容可以属于不同的组，它们可能会具有相互冲突的配置。这钟情形将下一节的可选择内容成员关系使用字典中描述。

可选择内容程序关系字典

像上文所提到的，一般典型的内容是属于一个可选择的单一的内容组，当这个组处于 ON 状态时他是可视的，当这个组处于 OFF 状态时不可见。为了表达更复杂的能见度策略，内容应申报本身并不直接属于可选内容，而是一组可选的内容成员字典，这些项目将在表 4.49 中展示。

表 4.49 可选内容成员字典的条目

关键域	类型	取值
Type	名称	(必须的) 该字典所描述的 PDF 对象的类型，对于一个可选的内容成员关系字典，他必须是 OCMD
OCGs	字典或数组	(可选的) 指定被该成员关系字典所控制的而且其状态可以决定内容可见性的可选内容组的字典或者数组

字典。

注：空值或者一个对被删除对象的引用将会被忽略。

如果此项不存在，是一个空数组或者或者只包含为空或已删除的对象，该成员关系字典对内容的可见度没有影响。

P	名称	(可选) 一个指定该成员关系字典的内容的可见度策略的名称。合法的取值如下：
---	----	---------------------------------------

- AllOn: 只有所有 OCGs 中的条目处于 ON 状态时是可见的。
- AnyOn: 当 OCGs 中任意一个条目处于 ON 状态就可见。
- AnyOff: 当 OCGs 中任意一个条目处于 OFF 状态时可见。
- AllOff: 当 OCGs 中所有条目都处于 OFF 状态时可见。

默认值: AnyOn

VE	数组	(可选) 一个指定能见度表达方式的数组，它是用来基于一个可选内容群组集合计算内容能见度的；参加下面讨论。
----	----	--

一个可选的成员关系字典可以通过两种方式表达他的成员关系策略。

- P 项目指定一个特殊的布尔表达式来说明可选的内容群组是如何被 OCGs 指定来决定被成员关系字典所控制的内容的能见度的。
- PDF1.6 介绍了 VE 条目，它是一个能见度表达式，它可以指定任意一个布尔表达式来通过可选内容组的状态来计算内容的能见度。

注：由于 VE 项更通用，如果它存在而且被 PDF 客户软件所支持，它应该在引用 OCGs 和 P 的时候使用。然而，出于兼容性目的，PDF 制作者应该在任何可能的地方都使用 OCGs 和 P 项。如果必须使用 VE 项来表达意图行为，OCGS 和 P 也必须被提供，以使他们和老版本的 PDF 客户软件相近。

可视表达式是一个具有下列特征的数组：

- 它的第一个元素师代表布尔操作符的名称
- 随后的元素是可选的内容群组或者其他可视表达式。
- 如果第一个元素师 Not，则后面只能跟一个元素。如果第一个元素是 And 或者 or，则可以有多个后续元素。
- 在评价一个能见度表达式的时候，可选内容组的 On 状态和布尔值 true 是等价的，OFF 和 false 是等价的。

例 4.33 和例 4.44 说明了能见度表达式的用法。

成员关系字典在下列情况中是很有用的：

- 可能有些内容会在组处于 On 状态时选择不可见，而在组处于 Off 状态时选择可见。在这种情况下，内容应该属于一个 OCGs 项由单一可选项群组组成而且 P 项是 AllOff 或者 AllOn 的成员关系字典。

注：一个单一群组组成的 OCG 条目而且他的 P 条目是 AnyOn 或者 AllOn，这是合法的。但是，在这种情况下，最好使用一个可选的内容组，因为他会时候较少的对象。有些内容可能会属于多个群组，当这些群组的状态相互冲突的时候，必须给他们知道策略。在这种情况下，该内容应该属于一个 OCGs 条目由可选的数组内容群组组成而且而且其 P 条目指定了可视策略的成员关系字典，如例 4.32 所示：

例 4.32

```
<< /Type /OCMD % 属于可选内容的内容
                           % 被该状态控制的成员关系字典
/OCGs [12 0 R 13 0 R 14 0 R] % 三个可选的内容组
/P /AllOn % 只有三个内容组的状态都是 On 的时候
>> % 内容才可见，否则就是不可见。
```

例 4.33

```
<< /Type /OCMD
```

```
/VE [/And 12 0 R 13 0 R 14 0 R] % 和例 4.32 等价的可视表达式  
>>
```

例 4.34 展示了一个基于 5 个可选内容群组的更加复杂可视表达式，分别由对象 1 到 5 代表。他和

“OCG 1” OR (NOT “OCG 2”) OR (“OCG 3” AND “OCG 4” AND “OCG 5”)
等价。

例 4.44

```
<< /Type /OCMD  
/VE [/Or % 能见读表达式: OR  
1 0 R % OCG 1  
[/Not 2 0 R] % NOT OCG 2  
[/And 3 0 R 4 0 R 5 0 R] % OCG 3 AND OCG 4 AND OCG 5  
]  
>>
```

意图

表 4.48 中的 Intent 项提供了一种利用可选内容区分不同的意图的方法。例如，很多文档设计应用，如 CAD 包，提供分层特性把图像组收集在一起，为制作者提供有选择的隐藏或者浏览。然而，这种分层可能不同，但对消费者的应用很有用。因此，可以为一个文档内的可选内容组指定不同的意图。一个给的应用可以决定只是用那种指定了意图的组。

PDF1.5 定义了两个意图：Design，它的意图是代表艺术设计者的组织构架。还有 View，其意图是文档使用者的交互功能。

更多的意图会被加到未来的 PDF 版本中，为了和未来版本的 PDF 兼容，PDF 客户应该允许不被识别的意图值。

配置字典也包含一个 Intent 项，如果一个或者多个意图被包含在当前配置意图集合中，这个组将被用于决定能见度。如果没有匹配项，该组对能见度没有影响。

注：如果该配置意图是一个空数组，没有组会被用于决定能见度，因此，所有的内容都是可见的。

4.10.2、制作可选图像内容

可以通过指定一个可选内容组或者可选内容成员关系字典的成员关系使 PDF 文件中的内容可选择。有如下两个可用的重要机制：

- 一些被标记内容提供商分割的内容流可以使其变的可选择，如下面“内容流中的可选内容”所描述的那样。
- 表格、图像 XObject 和注释可以通过字典条目使其整体边的可选择，如 344 页“注释和 XObject 中的可选内容”所描述的一样。

当决定要隐藏一个 PDF 文档中的可选内容的时候，会发生如下事情：

- 内容没有被绘制
- 对图像状态的操作，比如设置颜色，变换矩阵，还有裁剪，都同样适用。此外，由画图产生的图像状态副作用会出现。此外，即使包裹在可选内的当前文本的位置也会被更新。换句话说，不管该可选内容是否可见，从标记内容区的结束到现在，图像状态参数必须相同。例如：隐藏一个可选内容的区域不会改变其他内容区域的对象的颜色。

注：该规则同样适用于那些设置的状态不是严格图形状态的操作，比如 BX 和 EX。

- 像表格 XObject 或者注释之类的对象，如果他们被设置成可选的，他们有可能被完全跳过。因为他们的内容被封装，因而没有对图像状态的改变能超出内容流处理之外。

PDF 客户应用中的一些其他特征，比如搜索和编辑，可能会受有选择的显示和影藏能力的影响。这些特征必须选择要么使用可选择内容组的文档的当前状态或者提供他们自动可选内容组的当期状态去控制他们处理的图像。比如，当移动光标或者点击鼠标的时候，选择和移动注释的工具应该记住当前屏幕上注释的能见度。一个全文本搜索引擎，可能需要处理一个文档中的所有内容，无论他在屏幕上的能见度是多大。导出过滤器可以选择当前屏幕的可见度，整屏内容或者通过 OCGs 的一个选项来控制可见度。

注：所有可选的内容相关的 PDF 格式的结构是未知的，因而忽略由 PDF1.4 的格式和更早的消费者，因此这些画和以其他方式处理

文件中的所有内容。

内容流中的可选内容

内容流中的一些内容可以通过在标记内容操作 **BDC** 和 **EMC** 之间附一个 **OC** 的标记内容标志使其变得可选择。此外，一个 **DP** 内容标记的操作可以被放在一个页面的内容流中来强制引用一个可选的内容群组或者该页的组，即使该页在该层中没有当前内容。与标记内容相关的资源清单指定一个可选的内容组或者可选的内容关系词典，该内容属于它。因为组必须是一种间接对像而一个成员关系字典包含了对间接对象的引用，资源清单必须是一个当前资源字典属性子字典中的名称列表。如例 4.35 和例 4.36 所示。

注：尽管标记内容的标志必须是 **OC**，其他的标记内容的应用也不排除使用 **OC** 作为标记。只有当标记是 **OC** 的时候，标记内容才被认为是针对可选内容的而且字典组数是一个可选的内容群组或者可选的内容关系字典。

为了避免与使用其他标记内容的特征的冲突，建议遵循如下策略：

- 当要给内容打上像其他内容一样的可选内容标记的时候，可选内容标记应该嵌套在其他标记内容里。
- 当可选内容和其他标记会发生重叠而又没有严格控制的时候，可选内容应该被分为两个或者多个 **BDS/EMC** 区，必要的时候将可选内容区嵌套在其他的里面。被分裂的可选内容不会损坏可选内容的能见度特性。但这并不适用于其他用途的内容标记。

注：嵌套在其他的标记内容里面并且被标记为可选内容的标记，只有在表明了能见度的情况下才是可视的。换句话说，如果适用于外部等级的设置表面该内容应该被隐藏，则内部等级也应该是隐藏的，不管其设置如何。

在下面的例子中，**Show Greeting** 可选内容群组的状态直接控制该页文本字符串“Hello”的能见度。当该组处于 **On** 状态的时候，文本可见，当该组处于 **Off** 状态的时候，文本不可见。

例 4.35

% 在内容流内部

...

/OC /oc1 BDC % 下面是可选内容

```
BT
/F1 1 Tf
12 0 0 12 100 600 Tm
(Hello) Tj
ET
EMC % 可选内容结束。
...
```

```
<< % 资源字典里
/Properties << /oc1 5 0 R >> % 该字典将名称OC1映射到一个可选的内
容组
>>
5 0 obj % OCG控制文档的可视度
<<
/Type /OCG
/Name (Show Greeting)
>>
endobj
```

上述例子显示一块与一个可选内容相关的内容小组。还有其他的可能性：

- 在超过一节的内容可以参考同组或成员字典的情况下，在这种情形下，所有小节的可视度都是一样的。
- 相同的，虽然效率空间比较低，不同的区域可以拥有单独的成员关系字典而且拥有相同的 OCGs 和 P 项。本节将具有相同的可视性行为。
- 一个内容的两部分一个应用相同组但是具有不同 P 设置的成员关系字典。比如，有一个部分没有 P 项，而其他的部分拥有一个 AllOff 的 P 项，该内容的两个区域的能见度是相反的。这就是说，当第二个区域是不可见的时候，第一个区域是可见的，反之依然。下面的例子同时展示了通过成员关系字典直接过着间接使用可选内容组的方法。如果被命名

为图像 A 或者图像 B 控制的图像显示出来该内容区域就会被绘制。如果所有的区域都被隐藏了，方框帧也会被隐藏。

例 4.36

% 内容流的内部

...

/OC /OC2 BDC % 画一个黑色的方框帧

0 g

4 w

100 100 412 592 re s

EMC

/OC /OC3 BDC % 画成图像XObject

q

412 0 0 592 100 100 cm

/Im3 Do

Q

EMC

/OC /OC4 BDC % 画作一个图像XObject

q

412 0 0 592 100 100 cm

/Im4 Do

Q

EMC

...

<< % 源字典

/Properties << /OC2 20 0 R /OC3 30 0 R /OC4 40 0 R >>

```
/XObject << /Im3 50 0 R /Im4 /60 0 R >>
>>
20 0 obj
<< % 可选内容关系字典
/Type /OCMD
/OCGs [30 0 R 40 0 R]
/P /AnyOn
>>
endobj
30 0 obj %可选内容群组 “Image A”
<<
/Type /OCG
/Name (Image A)
>>
endobj
40 0 obj % 可选内容群组 “Image B”
<<
/Type /OCG
/Name (Image B)
>>
Endobj
```

XObject 和注释中的可选内容

除了在内容流内部标记的内容，来自 XObject 和图像 XObject 或者注释中的内容可能包含一个 OC 项，它是一个可选的内容群组或者可选的内容关系字典。

一个表格或者图像 XObject 的能见度是由群组的状态或者被成员关系字典连同 P (或者 VE) 项一起引用的组和 XObject 所援引的背景状态的当前可见度所决定。的。

注释用用控制屏幕和打印能见度的各种标志。如果一个注释拥有 OC 项，只有在这些标志拥有适当的设置而且该组或者成员关系字典表明他是可视的的时候，他才对屏幕和打印机是可见的。

4.10.3、可选内容的配置

一个 PDF 文档含有可以指定该文档中可选内容组的默认状态的可选内容，这些可选内容还可以说明什么外部因素可以改变这些状态。接下来的章节描述了 PDF 中那些用来指定这些信息的结构：

- 345 页的“可选内容管理字典”描述了列出文档中所有可选内容组的结构和他们的配置。
- 345 页的“可选内容配置字典”描述了指定初始状态设置和该文档群组有关的其他信息的结构。
- 350 页的“应用使用字典”和 355 页的“可选内容组状态的确定”描述了如何通过外部因素影响组的状态。

可选内容管理字典

文件目录中的可选的 `OCProperties` 项含有可选内容管理字典，它拥有一个该文档中所有可选内容群组清单，还有关于可选内容默认备用配置的信息。如果该文件 含有任何一个可选内容，该字典就是必须的。如果它缺失了，PDF 消费者应该忽略该文档中的所有可选内容结构。

该字典拥有如下项：

表 4.50 可选内容管理字典中的项目

关键域	类型	取值
<code>OCGs</code>	数组	(必须的) 一个记录该文档中对所有可选内容进行间接应用的数组，排列顺序不定。所有的可选内容组都应该包含在该数组中。
<code>D</code>	字典	(必须的) 默认可选内容浏览配置字典。
<code>Configs</code>	数组	(可选的) 一个针对 PDF 处理应用或者特性的备用可选内容配置字典

可选内容配置字典

表 4.50 中的 **D** 和 **Configs** 项是配置字典，他们代表被 PDF 处理应用或者特性所使用的可选内容组的简报。当一个文档第一次被打开的时候，**D** 配置字典指定了可选内容组的初始状态，**Configs** 列出了在特定环境下的其他配置。配置字典中的项如表 4.51 所示；

表 4.51 可选内容配置字典中的项目

关键域	类型	取值
Name	文本字符串	(可选的) 配置名称，适合作为用户界面的介绍
Creator	文本字符串	(可选的) 生成该配置字典的应用的名称或者特性
BaseState	名称	(可选的) 当该配置生效的时候，用于初始化文档中所有可选内容群组的状态。该项的取值必须是下面三个值之一： <ul style="list-style-type: none">● ON: 所有群组的状态必须为开● OFF: 所有群组的状态必须为关闭● Uchanged: 所有群组的状态不能被改变 在初始化之后， ON 和 OFF 数组中的内容将会被处理，覆盖该数组中群组的状态。 默认值: ON 注：如果该文档的默认配置字典中有 BaseState ，其取值必须是 On 。
ON	数组	(可选) 当该配置被启用的时候，一个可选内容组应该被设置为 OFF 的数组。 注：如果 BaseState 项是 OFF ，该项就是多余的。
Intent	名称或数组	(可选的) 一个单一的意图名称或者由意图名称组成的数组。在计算能容的能见度的时候，用来决定哪一个可选内容组的状态该被舍弃或者考虑。 PDF1.5 定义了两个意图名称， View 和 Design 。未来的版本可能会定义一些其他的特性。此外，此外名称 All 表示意图设置，包括未定义的在内。默

认值是 View，对于 PDF 文档的默认设置，该取值也必须是 View。

关键域	类型	取值
AS	数组	(可选的)一个应用方法字典数组，是当他们被使用的时候用来指定浏览应用该采用哪种字典方法策略来根据外部因素来自动设置可选内容群组状态的，比如当前系统语言，放大 浏览。
Order	数组	(可选的)一个指定用户界面可选内容组的说明的推荐顺序的数组。该数组元素可能包含以下对象： <ul style="list-style-type: none">● 可选内容组字典，他们的 Name 选项将在用户界面显示。● 一个运行像大纲结构或者树一样嵌套的可选内容群组数组。每个嵌套数组可以选择作为其第一个元素的文本字符串作为一个在用户界面的非可选标签。

注：嵌套数组中的文本标签应该被用来收集目前相关的可选内容组而不和多层群组内部的实际内容进行通信。为了反映内容群组中的直接嵌套，应该使用嵌套在没有文本标签的组中的数组，比如层和子层之间。

所有的群组中都不能应该有一个明确声明的空数组。

在默认的配置字典中，默认值是一个空数组。在其其他的配置字典中，默认值是来自默认配置字典的 Order 值。

注：任何一个没有在该数组中出现的组也不应该在用户界面中出现。

ListMode 名称 (可选) 一个指定 Order 数组中哪个可选内容群组应该被显示给用户的名称。合法的取值如下：

- **AllPages**: 显示 Order 数组中所有的组
- **VisiblePages**: 只显示 Order 数组中那些被一个或者多个可视页引用的群组。

默认值: Allpages

关键域	类型	取值
RBGroups	数组	(可选的) 一个由多个数组组成的多维数组，他们每一个都代表一组状态遵循单选按钮规范的可选内容群。这就是说，一个数组中每次至多只能有一个可选内容组处于 ON 状态。入股一个数组被置于 ON 状态，所有其他的必须被置于 OFF 状态。然而，把一个群组从 ON 转成 OFF 状态，并不过强制其他的组被置于 ON 状态。

一个明确声明的空数组表明不存在这个样的集合。

在默认的配置字典中，默认的值是空数组，在其

他的配置字典中， 默认值是默认配置字典中的 RBGroups 的值。

Locked	数组	(可选, PDF1.6) 当该配置启动的时候那些应该被锁住的可选内容群组组成的数组, 被锁定的内容群组的状态时不能通过用户空间的浏览应用进行改变的。可以使用这个项目来防止那些依靠该群组的内容的能见度被用户改变。 默认取值: 空数组 注: 该项并不能阻止可选内容群组的状态被通过用户界面之外的途径改变, 比如 java 脚本, 或者配置字典中的 AS 项的条目。
--------	----	--

例 4.37 和例 4.38 说明了使用 Order 项来控制在用户界面显示群组的方法。

例 4.37

给定如下 PDF 对象

```
1 0 obj <</Type /OCG /Name (Skin)>> endobj % 可选内容群组
2 0 obj <</Type /OCG /Name (Bones)>> endobj
3 0 obj <</Type /OCG /Name (Bark)>> endobj
4 0 obj <</Type /OCG /Name (Wood)>> endobj
5 0 obj %配置字典
<< /Order [[(Frog Anatomy) 1 0 R 2 0 R] [(Tree Anatomy) 3 0 R 4 0 R] ]>>
```

一个 PDF 浏览器应该像如下那样显示可选内容群组

整体解剖

皮肤

骨架

树解剖

巴尔克

木

例 4.38

给定如下PDF对象：

```
% 页面内容
/Oc /L1 BDC          % 第一层
/Oc /L1a BDC          % 第一层的子层A
0 0 100 100 re f
EMC
/Oc /L1b BDC          % 第一层的子层B
0 100 100 100 re f
EMC
EMC
...
<< /L1 1 0 R          % 源名称
/L1a 2 0 R
/L1b 3 0 R
>>
... %Optional content groups
1 0 obj <</Type /OCG /Name (Layer 1)>> endobj
2 0 obj <</Type /OCG /Name (Sublayer A)>> endobj
3 0 obj <</Type /OCG /Name (Sublayer B)>> endobj
...
4 0 obj                 %配置字典
<< /Order [1 0 R [2 0 R 3 0 R]] >>
```

一个 PDF 浏览器应该像如下一样显示 OCGs

层 1

子层 A

子层 B

AS 项是一个自动状态数组，它有一个或多个指定浏览应用时如何自动根据外部

因素设置可选内容群组状态的应用用法字典组成，具体如下面所描述。

用法和用法应用字典

可选的内容组经常被构造用来控制在某种程度上相关联对象的能见度。对象可以通过很多途径相关联；比如，一个群组可能包含一些特定语言的内容，或者包含一些适合特别放大浏览的内容。

一个可选内容群组用法字典包含有用于描述该组所控制的内容的特性的信息。该字典可以包含表 4.52 所描述的任意一种复合项目。

表 4.52 可选内容用法字典中的项目

关键域	类型	取值
CreatorInfo	字典	<p>(可选的) 创建应用用来存储和可选内容组相关的应用指定的数据的字典。它包含两个必须的选项：</p> <ul style="list-style-type: none">● Creator: 一个指定创建该组的应用的字符串● Subtype: 定义该组所控制的内容的类型的名称。建议的值包括但是不限制 Artwork，对于图像设计或者出版应用，还有针对原理图设计或者建筑设计之类的技术。 <p>其他条目可能包括提出有关的信息创造应用程序或相关的应用程序。</p>
Language	字典	<p>注：一般情况下，Intent 项包含 Design 的组都会有一个 CreatorInfo 项。</p> <p>(可选) 指定被可选内容组控制的内容的语言的字典。他有两个项：</p> <p>Lang: (必须的) 一个指定语言的字符串，也可能是语言环境。</p> <p>Preferred: (可选的) 一个取值可能是 ON</p>

或者是 OFF 的名称，默认值是 OFF。当有一个特定匹配，但是系统语言和所有使用字典中的字符串都不能准确匹配的时候系统浏览应用会使用它。可以参考 350 页的“用法和使用应用字典”来得到更多的信息。

关键域	类型	取值
Export	字典	(可选的) 一个包含一个 ExportState 项的字典，其取值可能是 ON 或者 OFF。当该文档被一个浏览应用保存为一个不支持可选内容的格式的时候，该值表示该组内容的推荐状态。
Zoom	字典	(可选的) 一个指定在什么放大范围内该可选内容组被更好的浏览的字典。他可能包含一个或者下面所有的两个值： min: 该组处于 ON 状态时，推荐的最小放大因子。默认值是 0 max: 该组处于 ON 状态时，推荐的最大放大因子。默认值：无穷大
Print	字典	(可选的) 一个指定该组中用于打印的内容的字典。它包含有下列选项： ● Subtype: 一个指定被该组所控制的内容的类型的名称对象。 ● PrintState: 一个可能是 ON 或者 OFF 的名称，用于表明文档被一个浏览应用打印时该组应该设置成的状态。
View	字典	(可选的) 指定该可选内容组初始用户的字典。每一个字典拥有如下两个选项： ● Type: 一个可能是 Ind, Ttl 或者 Org 的名称对象。

- **Name:** 一个文本字符串或文字代表的名称
(个人) 字符串数组, 位置或组织。

PageElement 字典 (可选的) 一个声明分页群组的字典。它包含一个选项, 它的值可能是 HF, FG, BG, 或者 L。

尽管使用字典中的数据可被看做文档信息供用户审查, 它也可以被用来供浏览操作通过当前语言设置或者放大级别之类的外部因素来操纵可选内容群组的状态。文档作者可以使用用法字典指定应该使用该用法字典中的哪一个条目来基于这些因素自动设置可选内容组的状态。用法应用字典是在一个可选内容配置字典的 AS 条目中被列出来的。如果没有 AS 条目, 相关状态就不能基于使用信息做自动调整。

使用应用字典指定了哪个使用条目应该被浏览应用使用来自动操作可选内容组的状态, 哪些组应该被影响, 以及在什么情况下的规则。表 4.53 展示了使用应用字典中的选项。

注: 用法应用字典只打算用于互动浏览应用, 不适用与那些将 PDF 作为最终输出结构的应用。

表 4.53 应用用法字典中的选项

关键域	类型	取值
Event	名称	(必须的) 一个定义该用法应用字典在什么情形下被使用的情形的名称。可能是 View, Print 或者 Export
OCGs	数组	(可选的) 一个列出可选内容群组的字典, 该群组应该基于使用字典中的信息自动管理他们的状态。默认值: 一个空数组, 表明没有组会被影响。
Category	数组	(可选的) 一个名称数组, 每一个名称对应一个使用字典。在管理 OCG 数组中可选内容组的

状态的时候，应当考虑该组使用字典中对应的策略。

Event 项决定在浏览，打印或者导出该文档的时候，该使用设置是否被启用。OCGS 项指定了可选内容组的设置。对于 OCGS 中的每一个组，Category 所指定的使用字典中的选项，应该遵循改组的状态。如果所有状态都被推荐位 ON，该组的状态应该被设置为 ON，否则，设置为 OFF。

用法字典中的选项的使用方法如下

- **View:** 该项的推荐状态是 ViewState 项的取值。该项运行文档拥有只有在该文档被交互浏览时才相关的内容，比如如何与文档交互的介绍。
- **Print:** 该项的推荐状态是 Printstate 项的值。如果没有 PrintState，可选内容组的状态保持原来的状态不改变。
- **Export:** 该项的推荐状态是 ExportState 的值
- **Zoom:** 如果该文档的当前放大级别大于等于 min 而且比 max 小，推荐的状态是 ON，否则就是 OFF 状态。
- **User:** 指定一个或者多个与用户的识别符相匹配的 Name 项，Type 项决定了 Name 项的解释方法。如果有多个准确的匹配，推荐的状态是 ON，否则推荐使用 OFF 状态。
- **Language:** 这个分类运行以语言和现场应用为基础的选择。如果在用法应用字典的 OCG 列表中的可选应用群组中有和 Lang 条目准确匹配的语言和现场，所有能准确匹配的组都推荐使用 ON 状态。如果没有准确的匹配，但是有局部的匹配，那些拥有 Preferred 选项而且取值为 ON 的局部匹配的组推荐使用 ON 状态，其他的使用 OFF 状态。

例 4.39 显示了通过用法应用字典使用自动状态数组的方法，默配置字典中的 AS 项是拥有三个用法应用字典的数组，每一个 Event 取值 View、Print 和 Export。

注：虽然这种情况是典型的，有没有多个条目的限制同样价值的事件，它允许使用的申请文件不兼容字典将合并成较大的文档，并保存它们的行为。如果一个给的的可选内容组在不止一个 OCGs 数组中出现，只有在所有用法应用字典中出现的所有类别的推荐值都是 ON 的时候，它的状态才为 ON。

例 4.39

```
/OCProperties % 在文档目录中的 OC 资源
<< /OCGs [1 0 R 2 0 R 3 0 R 4 0 R]
/D << /BaseState /OFF % 默认配置
/ON [1 0 R]
/AS [ % 用法应用字典中的字典状态数组
<< /Event /View /Category [/Zoom] /OCGs [1 0 R 2 0 R 3 0 R 4 0 R] >>
<< /Event /Print /Category [/Print] /OCGs [4 0 R] >>
<< /Event /Export /Category [/Export] /OCGs [3 0 R 4 0 R] >>
]
>>
>>
...
1 0 obj
<< /Type /OCG
/Name (20000 foot view)
/Usage << /Zoom << /max 1.0 >> >>
>>
Endobj

2 0 obj
<< /Type /OCG
/Name (10000 foot view)
/Usage << /Zoom << /min 1.0 /max 2.0 >> >>
>>
endobj

3 0 obj
<< /Type /OCG
/Name (1000 foot view)
/Usage << /Zoom << /min 2.0 /max 20.0 >>
```

```
/Export << /ExportState /OFF >> >>  
>>  
endobj  
4 0 obj  
<< /Type /OCG  
/Name (Copyright notice)  
/Usage << /Print << /PrintState /ON >>  
/Export << /ExportState /ON>> >>  
>>  
endobj
```

在上面的例子中，具有 View 事件类型的用法应用字典指定所有可选应用群组在浏览的时候基于放大级别管理他们的状态。三组（对象 1、2、3）都包含 Zoom 使用信息，对象 4 则没有。因此，它不会被放大级别改变。对象 3 在导出的时候拥有一个推荐的 OFF 状态，在打印或者导出的时候，对象 4 接收到的推荐状态是 ON。

决定可选内容组的状态

该章总结了应用使用配置和应用使用字典来设置可选内容组状态的规则。为了讨论，区分开下列类型的使用时很有用的。

- 浏览应用，比如 Acrobat，它允许用户通过多种方式与文档交互。
- 设计应用，为收集图像的组提供分层功能，用于由选择的浏览或者隐藏内容。

注：以下规则并不意味着适用于设计应用，他们完全可能使用一个不同的管理方式管理他们的状态，如果他们可以选择。

- 聚合应用，将 PDF 文件作为图像导入。
- 打印应用，打印 PDF 文件。

当一个文档第一次被打开的时候，它的可选内容组会被基于 OCProperties 字典 D 配置字典分配一个状态：

1. BaseState 的值适用于所有的组
2. 数组中列出的处于 ON 或者 OFF 状态的群组调整他们的状态。

该状态是一个针对打印或者整合应用的推荐状态，他们不应当申请基于下面所描述的用法应用字典中的改变。然而，为了更高级的功能，他们可能提供用户控制来操纵可选内容群组中的各个不同状态。

注：浏览应用应该给用户提供一个在这种状态下浏览文档的选择。这个选项允许准确，因为它的内容时会出现放置准确预览聚合成一个应用程序或发送到一个独立的印刷系统。

接下来的讨论只适用于浏览应用。该应用应该检查针对具有 **View** 的 **Event** 类型的用法应用字典的 **AS** 项。对于每一个发现的项，**OCGs** 中所列出的组应该像 350 页中的“用法应用字典”所描述的那样进行调整。

通常，文档都是要准备被用户进行交互浏览的。每当有一个因素的变化，与事件类型用法词典应用查看依赖（如缩放级别），相应的字典，应重新申请。

用户可能会人为地通过触发 **SetOCGState** 行动去操纵可选内容群组的状态，比如点击链接或者标签。人为操作会覆盖自动设置的状态。这些组的状态会保持被覆盖，而且在文档被打开的时候，它是不能基于具有 **View** 事件类型的用户应用字典对状态进行调整的。

当文档被浏览应用打印的时候，会基于当前可选内容组的状态提供具有 **Print** 事件类型的用法应用字典。这些改变只能在操作期间保持存在，然后所有的组都会回到他们的初始状态。

注：虽然事件类型 **Print** 和 **Export** 具有相同的名称来对应使用类别，相应的用法应用字的允许指定其他被应用的类别。

第 5 章 文本

本章描述了在 PDF 格式中采取特殊的工具来处理文本 - 尤其是代表字形与字体的字符。一个字形是一个图象的形状而且归属于所有的图形处理，如坐标转换。由于文本在大多数页面说明的重要性，PDF 提供更高级别的工具允许一个应用来描述，选择，和方便、高效的提供字形。

第一部分是对字形如何从字体中提取出来并画到该页面的大致描述。随后的章节中详细涵盖以下主题：

文字状态。关于文本的一个图形状态参数的子集，包括选择字体的参数，规模大小适当的字形，并完成其他图形效果。

文本对象和操作。文本操作指定哪个字形被画上，该字形是由字符串对象的字符编码序列作为解释。文本对象由一系列文本操作和相关的参数围绕。

字体的数据结构。字体字典和相关数据结构提供信息，它给每个用户应用正确提供字形的文本和位置的解释。字形自己的定义被包含在字体程序集里，这有可能是被嵌入在 PDF 文件中，编译进应用程序中或者从外部字体文件中获得。

5.1、组织和字体的使用

一个字符是一个抽象的符号，而一个字形是一个字符的特定图形表现，例如，字形 A, A, A 是 A 字符的抽象表现。历史上这两个术语常常被交替使用在电脑排版上（通过名字选择为一些 PDF 字典键和附录操作提供依据），但是在这方面的进步所造成的的区分更有意义。因此，尽管还存在一些名字上的误差让前后不一致，但本书的还是要区别字符和字形。

字形被组织到字体。一个字体被定义为一个特定的字符字形集；例如，黑体和 Times 字体定义了一套标准的拉丁文字形。一个为用户应用程序中使用的 PDF 字体是以程序的形式组织好的。这样的字体方案是用一种特殊用途的语言，就像类型 1 或 TrueType 字体格式，它由一个专门的字体翻译器来解释。

在 PDF 中，术语字体是指一种字体字典，一个 PDF 对象能够标识的字体方案，而且包含了关于它的额外信息。它有几种不同的字体类型，由字体字典的子类条目来鉴别。

对于大多数的字体类型、字体方案被定义在单独的字体文件中，它可以是嵌

入在一个 PDF 流对象或从外部信源得到。字体方案包含形成字形的字形描述。

一个内容流通过指定一个字体字典和一个字符串对象将字形绘画在页面上，它是作为一个或多个字符编码来识别字体的字形。这个操作叫做显示文本字符串；文本字符串以这种方式被绘制称为显示字符串。这个字形描述包含一系列图形操作，该图形操作为在这个字体中的角色产生特定形状。为了渲染一个字形，应用程序执行字形描述。

这似乎暗示说明有通用图形扫描转换经验的程序员可能担心计算量。然而，这仅仅是字形描述和字体程序的抽象表现，而不是他们如何被实施。事实上，有效的实施可以通过仔细缓存和重用以前渲染的字形。

5.1.1、基本的显示文本

例 5.1 说明了一个最简单的使用字体。该文本 ABC 被放置在距离页面底部的 4 英寸和左边缘 10 英寸的地方，使用 12 号的黑体。

例 5.1

```
BT  
/F13 12 Tf  
288 720 Td  
(ABC) Tj  
ET
```

这个例子的五行执行以下步骤：

- 1。开始一个文本对象。
- 2。设置字体和字体大小来使用，在文本状态中作为参数设置他们。
(该字体的名称为 F13 对外指定字体为黑体字体。)
- 3。指定在页面的起始位置，设置文本对象的参数。
- 4。在这个位置上为一个字符串绘制字形。
- 5。完文本对象。

以下各段更详细地解释这些操作。

为了绘制字形，内容流必须首先确定要使用的字体。**Tf** 操作指定一个字体资源的名称，也就是说，是当前字体资源字典的一个入口。该项目的值是一个字体字典。字体字典通过外部标识辨认字体，如黑体，并提供一些额外的信息，这

样应用程序需要做的就是从字体中绘制出字形。字体字典本身提供了可选的字体方案。

注：字体资源名称提交给 **Tf** 操作是任意的，是对各种资源的命名。它承担对一个实际没有任何关系的字体名称关联，如黑体。

示例 5.2 展示了一个对当前页的资源字典的引用，例 5.2 定义字体词典是参考例 5.1 中的 F13。

例 5.2

```
/Resources
<</Font<</F13 23 0 R>>
>>
23 0 obj
<< /Type/Font
/Subtype/Type1
/BaseFont/Helvetica
>>
Endobj
```

一个字体定义一个标准尺寸的字形。这个标准的安排使文本的行间距为 1 个单位。在默认的用户坐标系统中，这意味着该标准字形在用户空间大小为 1 个单位，或 $1 / 72$ 英寸。（在 PDF 1.6，根据用户组的页面词条这个单位的大小可能被指定为大于 $1 / 72$ 英寸；见表 3.27。）标准大小的字体必须按比例缩小到可用。比例因子被指定为第二个 **Tf** 操作的操作数，因此在图形状态参数中设置文本的字体大小。例 5.1 在图形状态下设置了一个 12 个单位的黑体。

一旦该字体被选中和规模化，它就能够用来绘制字形，**Td** 操作调整现在的文本位置（事实上文本模型的翻译在 5.3.1 部分被描述为“文本位置运算符”）当在 **BT** 之后执行第一次，**Td** 在当前用户坐标系统中设定文本位置，这个位置决定从页面的哪个地方开始绘制字形。

Tj 操作通过当前的字体和其他文字相关的图形状态参数来取走一串操作和绘制相应的字形。在例 5.1，**Tj** 操作对待每一个字符串（在 0 到 255 的整数元素）作为一个字符代码。每个代码在字体中选择一个字形描述，而字形描述被执行绘

制在字形应在的页面的位置。这是 **Tj** 对于简单字体的操作，就像普通的拉丁文字的字体。对于一个字符代码序列解释来说复合字体更复杂，在 5.6 节中所述，“复合字体。”

注意：这些步骤是在页面上产生不是 12 点字形，而是 12 个单位字形，其中单位尺寸是文本空间，字形被提交到页面上，在文本对象中字形的实际尺寸取决于文本矩阵 (**Tm**)，几个文本状态参数，和目前的变换矩阵 (**CTM**) 的图形状态，见 5.3.3 节，“文本空间细节”，如果文本空间缩小后，使单位尺寸变为 1 厘米，从相同的 12 个单位的字体绘制字字形结果会有 12 厘米高。

5.1.2、实现特殊图形特效

正常使用 **Tj** 和其他字形绘制操作导致黑色填充字形被绘制。通过常规字形操作和结合字体操作可以获得其他效果。

在图像状态下的当前颜色被用来绘制字形：要么不画颜色或画颜色（或两者都有），依靠文本渲染模式（见第 5.2.5，“文本渲染模式”）。默认的颜色是黑色，但是其他颜色可以在绘制字形之前通过执行一个适当的颜色设定操作或操作序列表来得到（见第 4.5.7，“颜色操作”）。示例 5.3 使用文本渲染模式 0 和 **g** 操作以 50% 的灰度填充字形，如图 5.1 所示。

例 5.3

BT

/F1348Tf

2040Td

0Tr

0.5g

(ABC) Tj

ET



图 5.1 50%的灰度填充字形

通过处理字形轮廓作为一个路径而不是填充它，其他图形处理也可以达到效果。该文本在图形渲染模式参数状态指定是否字形轮廓被填补，勾勒，作为一个剪辑边界，或将这些影响结合起来使用。（此参数不适用类型 3 字体。）

例如 5.4 作为一个路径被勾勒来对待字形轮廓。Tr 操作设置文本渲染模式为 1（勾勒）。W 操作在用户空间中将线宽设置为两个单位。考虑到这些图形状态元素，Ti 操作勾勒字形轮廓两个厚度（见图 5.2）。

例 5.4

BT

/F1348Tf

2038Td

1Tr

2w

(ABC) Tj

ET

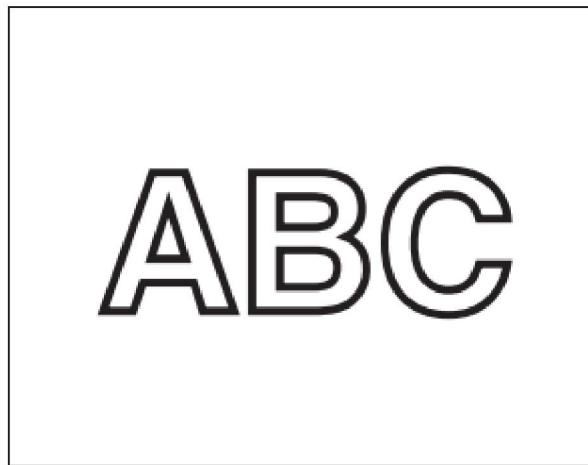


图 5.2 字形轮廓被看做勾勒路径

例 5.5 作为一个剪辑边界来对待字形轮廓。Tr 操作设置的文本渲染模式为 7 (剪辑)，导致随后的 Tj 操作限制字形轮廓作为当前剪辑路径。所有后续绘制操作只在这个路径上标记页面，如图 5.3 所示。这种状态一直持续到一些早期剪辑路径被 Q 操作恢复。

例 5.5

BT

/F1348Tf

2038Td

7Tr

(ABC) Tj

ET

…图形操作来绘制一个星暴…

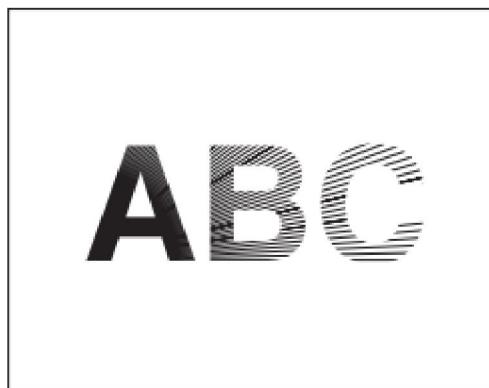


图 5.3 通过字形路径进行图形剪辑

5.1.3、字形定位和度量

一个字形的宽度—正式的，其水平位移—是沿着文本水平基线占有的空间大小。换句话说，它是当前文本位置移动的距离（通过转换文本空间）当字形被绘制。请注意 字形轮廓尺寸的宽度不同

在某些字体中，宽度是恒定的，它不是从一个字形变化到另一个字形。这些字体被称为固定尺寸或等宽字体。它们主要用于打字机式打印。然而，大多数字体用于高品质的排版和关联不同宽度的每个字形。这些字体被称为比例或可变节距字体。在任意一种情况下，`Tj` 根据其宽度操作连续的一系列字形的位置。

对于每一个字形宽度信息都存储在字体字典和字体方案本身。（这两套的宽度必须相同；字体字典这类存储信息，虽然多余，但却能保证使用户不用看里面的字体方案就可以确定字形位置。）操作用来展示文本被设计在字形定位是按照通常的标准宽度的假设上。但是，在某些有限的方式下这绝不是提供给各不相同的定位。例如，`TJ` 操作能够使文本字符串中的文本的位置在任何一对连续的字形之间被调整。有图形状态参数系统的调整字符和文字间距。

除了宽，字形还有几个其他度量能够影响字形位置和绘制。对于大多数的字体类型，这些信息主要是内部的字体方案，在 PDF 字体字典中并没有明确的规定。然而，在 3 号字体中，所有度量都被明确规定（见 5.5.4 节，“3 号字体”）。

字形的坐标系统是被定义的单个字符的空间。所有的路径坐标和度量被解释为字形空间。对于除去 3 号字的所有字体种类，字形空间的单位是一个文本空间单位的千分之一；对一个 3 号的字体，在字体中从字形空间到文本空间的转换是通过一个确定的矩阵。图 5.4 显示了一个典型的字形轮廓和它的度量。

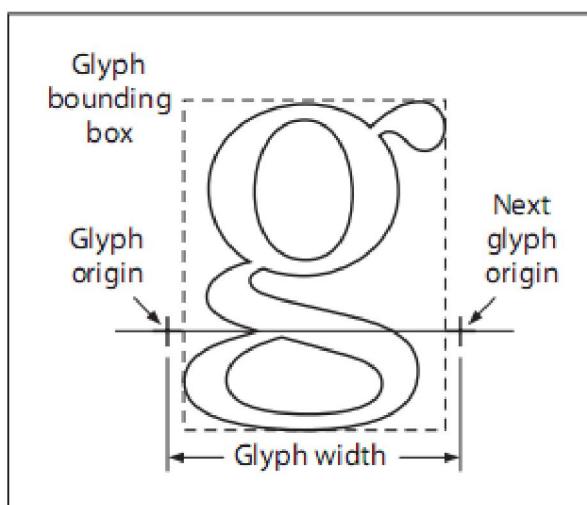


图 5.4 字形度量

在字形坐标系中字形的起始点是点 (0, 0)。**Tj** 和其他的文本位置展示操作将在文本空间的起始处绘制第一个字形的起始位置。例如，下面的代码调整文本空间的起始位置至用户坐标系统 (40, 50)，然后将 A 字形的起始位置放在该点上：

BT

4050Td

(ABC) Tj

ET

在文本行中绘制连续字形时字形位移是从字形的原点到下一个字形应该正确被放置的位置的原点。在字形坐标系统中这个位移是个矢量（称为位移向量）它有横向和纵向的组成部分。（即横向的位移通常称为宽。）对于大多数西方书写系统，包括那些基于拉丁字母的国家都有正的水平位移和零垂直位移，一些亚洲书写系统有一个非零的垂直位移。在所有情况下，文本显示操作将位移矢量转化为文本空间，然后通过这一数额转换文本空间。

字形边界框是最小的矩形（以字形坐标系的坐标轴方向为方向），仅仅环绕整个字形的形状。边界框体根据相对字形坐标系统的字形起始位置来确定它的左侧，底部，右侧，和顶部坐标

在某些书写系统，文本经常在两个不同的方向对齐。例如，常见的日文和中文字形书写是水平或垂直的。为了解决这个问题，字体可以任意为每一个字形包含第二套度量。按写作模式来选择使用哪套标准，其中 0 和 1 分别指定水平和垂直书写。此功能仅适用于复合字体，在 5.6 节讨论，“复合字体。”

当一个字形有两套衡量标准，每套指定一个标志符号的起源和为该书写模式的位移向量。在垂直书写方面，字形的位置被描述为一个从起始横向书写（起始 0）到起始垂直书写（起始 1）的位置向量。图 5.5 说明了这两个写作模式的指标：

- 左边的图说明了字形度量与写作模式 0，横向书写相关。坐标 ll 和 ur 指定相对原点 0 的字形边界框。 $W0$ 是位移向量，指定文本的位置在写作模式 0 下字形绘制之后怎样变化，其垂直分量始终为 0。
- 中间图说明了写作模式 1，垂直书写。 $W1$ 是写作模式 1 的位移向量，它的水平分量始终为 0。

- 右边的图， v 是一个定义相对原点起始位置为 1 的位置向量。

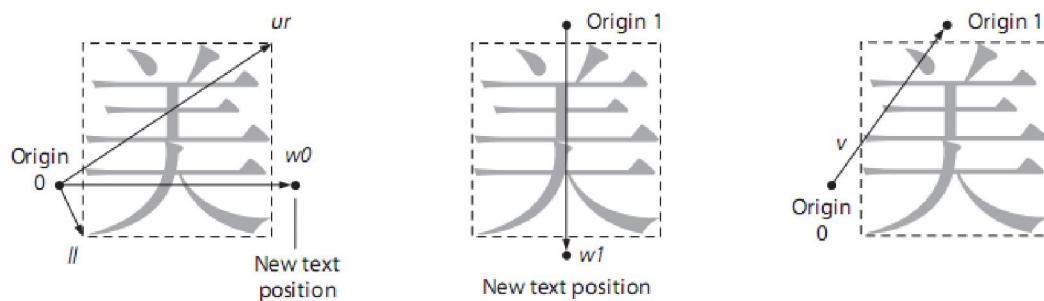


图 5.5 水平度量和垂直写作模式

字形度量信息也可以是单独的 Adobe 字体度量 (AFM) 和 Adobe 复合字体度量 (的 ACFM) 文件的形式。这些文件是为应用程序生成 PDF 页面描述格式，而且必须作出决定，对字形的宽度和其他字形的度量。也可在原子力显微镜和字距的 ACFM 文件是信息，它允许应用程序生成的 PDF 文件，以确定根据上下文之间的间距调整字形。在 AFM 和 ACFM 文件格式规格在 Adobe 提供的技术说明 #5004, Adobe 字体度量文件格式规范，该文件可从 Adobe 解决方案网络网站（见参考文献）。

5.2、文本状态参数和操作

文本状态包含这些只能影响文本的图形状态参数。有 9 个参数在文本状态(见表 5.1)。

表格 5.1 文本状态参数

参数	描述
T_c	字符间距
T_w	字间距
T_h	水平缩放
T_l	行距
T_f	文本字体
T_{fs}	正文字体大小

T_{mode}	文本渲染模式
T_{rise}	文本上升
T_k	文本分离

除了不需要说明的 T_f 和 T_{fs} ，这些参数在以下各节被进一步讨论。（如上文第 5.3 节，“文本对象”，另外三个文本的相关参数仅在一个文本对象中被定义： T_m ，文本矩阵； T_{lm} ，文本行矩阵； T_{rm} ，文本渲染矩阵。）文本状态参数值被查询当文本被放置和显示（使用第 5.3.1 所述的操作，“文本状态操作”和 5.3.2，“文本显示操作”）。特别是，在 5.3.3 节描述的间距和换算参数，“文本空间详细信息。”通过使用表 5.2 中列出的操作文本状态参数可以被设置。

注：文本分离参数 T_k 在图形状态参数字典中通过使用 Gs 操作（见第 4.3.4 节，“图形状态参数词典”）设置 TK 入口。设置这些参数没有特殊的方法。

文本状态参数能够出现在文本对象的外面，它们设置的值在单一的内容流中被保存。和其他图形状态参数一样，这些参数在每一页的开头被初始化为它们的默认值。

表 5.2 文本状态操作		
OPERANDS	OPERATOR	DESCRIPTION
charSpace	Tc	设置字符间距, Tc, 对字符间距来说这是在未缩放的文本单位间距下表示一个数字。字符间距通过 TJ, TJ, 和 ' 操作来使用。 初始化值: 0。
wordSpace	Tw	设置字间距, Tw, 它是一个表示在未缩放文本单元的数字。字间距通过 TJ, TJ, 和 ' 操作来使用。 初始化值: 0。
scale	Tz	设置水平缩放, (规模除以 100) , 规模是一个指定 Th 的正常宽度的百分比的数字。

		初始化值：100（正常宽度）。
leading	TL	设置文本行距， Tl ，它是一个表示在未缩放文本单元的数字。缩放文本仅仅通过 T^* , ' 和 " 操作来实现。 初始化值：0。
font size	Tf	设置文本字体， Tf ，且文本字体大小， Tfs 。 $Font$ 是一个字体资源的名字，当前资源字体字典中；大小是一个代表比例因子的数字。 $font$ 或 $size$ 都没有初始化值；他们必须通过 Tf 被明确的指定，在任何文本显示之前。
render	Tr	设置文本渲染， $Tmode$ ，它是一个整型。初始化值：0。
rise	Ts	设置文本上升， $Trise$ ，它是一个表示在未缩放文本单元的数字。初始化值：0。

请注意，这些参数有些是未缩放文本间距单位。这意味着他们在指定的坐标系统，该坐标系统是指由文字矩阵 Tm 给定，但规模不由字体大小参数给定， Tfs 。

5.2.1、字符间距

字符的间距参数， T_c ，是未缩放的文本间距指定的一个数字（尽管它服从 T_h 参数缩放，当写作模式是横向的时候）。当字符串中的每个字符的字形被呈现，根据写作模式 T_c 被添加到水平线或者垂直分量。（见第 5.1.3，“字形定位和度量，作为字形位移的讨论。）在默认的坐标系统，水平坐标左到右和垂直坐标从底部到顶部增加。因此，对于横向书写，一个正的 T_c 将会影响字形之间展开的距离（见图 5.6），反之对于垂直书写，一个负的 T_c 将会有这个影响。

$T_c = 0$ (default)	Character
$T_c = 0.25$	Character

图 5.6 水平书写模式下的字符间距

5.2.2、字间距

字间距与字符间距工作方式相同，但仅适用于空格字符，代码 32。词间距参数， T_w ，被添加到字形的水平或垂直位移（取决于书写模式）。对于水平书写模式，一个正的 T_w 将增大字与字之间的间距。对于垂直书写模式，一个正的 T_w 将缩小字与字之间的间距（它的一个负增值），因为垂直坐标从底部到顶部增加。

图 5.7 阐明了字间距在的作用

$T_w = 0$ (default)	Word Space
$T_w = 2.5$	Word Space

图 5.7 在水平位置的字间距

注：字间距被用在字符串中的单字节字符编码每个可能出现的地方，当使用一个 32 位的简单字体或者复合字体作为一个单字节编码。它不适应于 32 位多重字节码。

5.2.3、横向缩放

水平缩放参数， T_h ，通过拉伸或压缩调整在他们水平方向的字形宽度。它的值被指定为一个字形正常宽度的百分比，并以 100 为正常宽度。缩放始终适用于文本空间的横向调整，独立其写作模式。这既影响字形的形状和它的水平位移（即

其位移向量)。如果是书写模式是水平的, 它也会影响间距参数 T_c 和 T_w , 以及任何由 TJ 操作进行的位置调整。图 5.8 展示了水平缩放的效果。

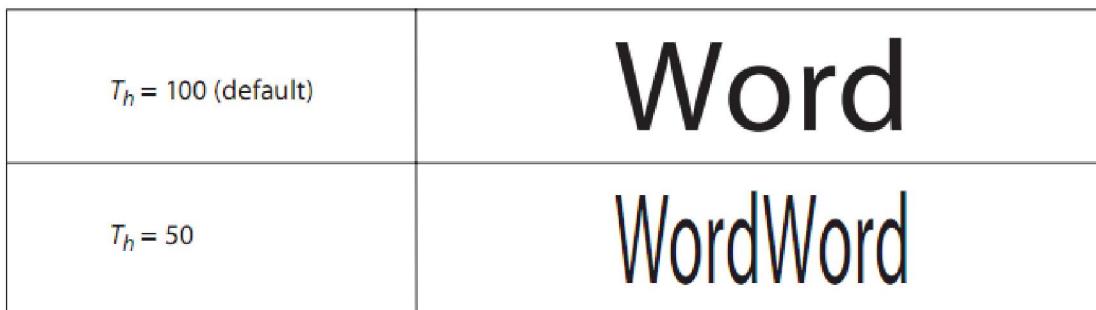


图 5.8 水平缩放

5.2.4、行距

行距参数 T_l , 是用来衡量未缩放文本间距。它指定了两个相邻基线的文本行的垂直距离, 如图 5.9 所示。

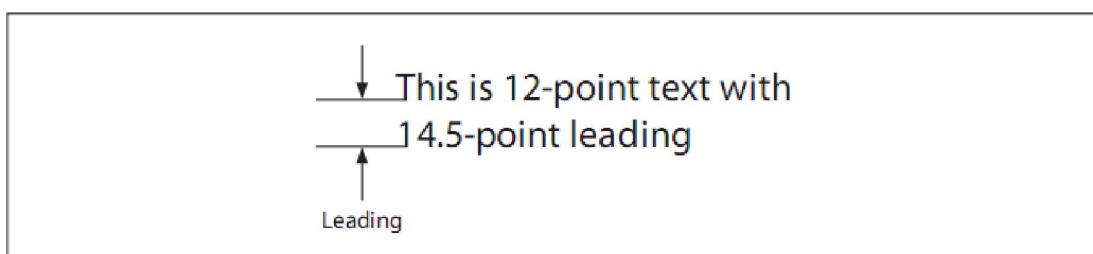


图 5.9 行距

行距的参数有 TD, T*, ', 和 “操作, 其效果精确的描述见 376 页表 5.5。在文本空间这个参数总是适用于纵坐标, 独立于写作模式。

5.2.5、文本渲染模式

文本渲染模式, TMODE, 确定是否显示文本导致字形轮廓被笔触, 填充, 作为一个剪辑边界, 或者将以上三个结合使用。绘制, 填充和剪辑对一个文本对象有相同的效果, 因为他们做了一个路径对象 (见第 4.4.2 节, “路径画法操作”和 4.4.3, “裁剪路径算法”), 虽然他们是在一个完全不同的方式下被指定。图形状态参数影响这些操作, 如线宽, 都是在用户空间被解释, 而不是在文本空间。

注：在 3 号字体下，文本渲染模式并没有对文本表现产生效果（见 5.5.4 节，“3 号字体”）。

该文本渲染模式列于表 5.3。在该例中，黑色是笔触颜色和浅灰色是填充颜色。至于剪辑模式（4 至 7），通过字形显示哪个地方被裁剪，一系列线条会被绘制。

如果文本渲染模式需要填充，在图形状态下当前无笔触颜色，如果它需要笔触，则使用当前笔触颜色。在同时执行填充和笔触模式，其效果就好像每个图形轮廓都被填充然后在几个分开操作下进行笔触。如果有任何重叠的字形，其结果是相当于同一时间笔触和填充它们，产生重叠不透明的字形外观，而不是先填充，然后将他们一下子笔触（见附录 H 执行情况的说明 56）。在透明的成像模型中，这些组合的填充和笔触模式，有进一步的注意事项，见特别路径绘图注意事项“538 页”。

剪辑模式行为需要进一步的解释。字形轮廓开始累积，如果一个 BT 操作被执行当文本渲染模式被设定为剪辑模式或被设定为文本对象内的剪辑模式。字形累积直到文本对象是由 ET 操作所结束；文本渲染模式不能更改指向前一个非笔触模式。

表 5.3 文本渲染模式

MODE	EXAMPLE	DESCRIPTION
0		Fill text.
1		Stroke text.
2		Fill, then stroke text.
3		Neither fill nor stroke text (invisible).
4		Fill text and add to path for clipping (see above).
5		Stroke text and add to path for clipping.
6		Fill, then stroke text and add to path for clipping.
7		Add text to path for clipping.

在文本对象的末尾，如果有累积的字形轮廓的话，会被合并成一个单一的路径，处理个体的显示轮廓作为该路径的子路径和申请非零绕数规则（见“非零绕数规则”202页）。图像状态下的调节当前的剪辑路径与以前的剪辑路径路径相交。对当前路径对象来说，剪辑发生在当前文本状态下的填充和笔触操作之后。剪辑仍起作用，直到它以前的一些剪辑路径被申请的 Q 操作修复。

注：如果没有自形显示或只有字形显示而没有轮廓（例如，如果他们是空格字符），将不发生裁剪。

5.2.6、文本上升

文本上升， T_{rise} ，指定该距离，在未缩放文本空间单位中，从它的默认位置移动基线上升或下降。相对基线来说。对于绘制上标或下标来说调整基准线是有益的。默认的基准位置可以通过设置文本上升为 0。图 5.10 显示了上升的文本效果。文本上升总是适应于空间的垂直坐标，而不是书写模式。

(This text is) Tj 5 Ts (superscripted) Tj	This text is ^{superscripted}
(This text is) Tj -5 Ts (subscripted) Tj	This text is _{subscripted}
(This) Tj -5 Ts (text) Tj 5 Ts (moves) Tj 0 Ts (around) Tj	This text moves around

图 5.10 文本上升

5.2.7、文本脱模

文本脱模参数， T_k （PDF 格式 1.4）是一个布尔标志，决定了哪个文本元素在合成彩色透明成像模型中被认为是基本对象。只有通过 gs 操作穿过 Tk 进入图形状态参数字典它才能够被设置（参见 4.3.4 节，“图形状态参数词典”）。

该文本脱模参数只适用于整个文本对象，在划定的文本对象中它可能无法在 BT 和 ET 之间设定操作。它的初始值是真。如果它的值为假，文本对象中的每一个字形会被视为一个单独的基本对象，当字形重叠，它们也会相互组合。

如果该参数为 true，在文本对象中的所有标志符号被视为一起作为一个单一的基本对象，当字形重叠，在重叠的地区后来字形覆盖（“淘汰”）较早字形。这种行为等同于整个文本对象处理，就好像它是非单独的脱模透明组，见 7.3.5 节，“脱模组。”透明度参数应用到单个字形，而不是作为一个隐含的透明组：

图形状态参数，包括透明度参数，从上下文中出现的文本对象继承。他们不被保存和恢复，在透明组的开始也不重置透明度参数（其实就是一个透明组目标被明确调用）。文本对象下的图形状态参数的改变已超出文本对象的末尾。

文本对象已被完全评估，组结果和背景幕混合，该小组结果是在复合的背景下，使用普通的混合模式和 alpha 和值为 1.0 的软模。

5.3、文本对象

PDF 格式文本对象包括经营者，可以显示文本字符串，移动文本的位置，并设置文本的状态和某些其他参数。此外，三个参数定义仅在一个文本对象，并没有从一个文本对象持续到下一个文本：

- T_m : 文本矩阵
- T_{lm} : 文本行矩阵
- Tr_m : 文本修饰矩阵，这实际上只是一个中间结果，它结合了文本状态参数，文本矩阵 (T_m) 的影响，以及当前的变换矩阵

一个文本对象开始于 BT，结束于 ET，正像如下所述和如图 5.4 描述：

BT

...Zero or more text operators or other allowed operators...

ET

表 5.4 文本对象操作

操作数	操作者	操作描述
—	BT	开始一个文本对象，初始化文本矩阵 T_m 和文本行矩阵 T_{lm} ，为了定义矩阵，文本对象不能嵌套，第二个 BT 不能出现在一个 ET 之前
—	ET	完成一个文本对象，结束一个文本矩阵

这些与操作相文本关联的特殊类型可以出现在一个文本对象中：

- 文本声明操作，在 5.2 节中阐述，“文本声明参数和操作”
- 文本位置操作，在 5.3.1 节中阐述，“文本位置操作”
- 文本显示操作，在 5.3.2 节中阐述，“文本显示操作”

后两节同时也提供了对前面文本对象参数的更多信息，其他可以出现在文本对象中的操作是那些关系到声明的通用图形，颜色和标记的内容，如 167 页图 4.1。

注意：如果内容不包括任何文本，文本的程序集可能会省略（见 10.1 节，“程序集”）。在这种情况下，没有文本操作（包括操作者，只是设置文字状态）可能会在目前的内容流中，因为这些操作在同一程序中有设置定义。

注意：虽然文本对象不能被静态嵌套，文本在使用 3 号字体包括任何图形对象的时候可以显示，包括其他文本对象，同样，当前的颜色模式可能是一个平铺的模式模块包含一个文本对象。

5.3.1、文本位置操作

文本空间是一个可以在其中显示文字的坐标系统。它是由文本矩阵 T_m , 声明参数 T_{fs} , T_h 和 T_{rise} 共同定义，并且他们一起决定了由文本空间到用户空间的转型。具体的说，由文本显示操作的第一个标志符号放置在文本空间的原点，如果文本空间已被翻译，缩放，或旋转，然后再位置，大小，或在用户空间字形方向相应改变。

表 5.5 文本位置操作

操作数	操作者	描述
$t_x \ t_y$	Td	移动到下一行的开始，从当前行的首部开始用 (t_x, t_y) 。 t_x, t_y 是在被缩放的文本空间中表达的数字，更确切的说，这个运算符执行以下操作： $T_m = T_{lm} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix} \times T_{lm}$
$t_x \ t_y$	TD	移动到下一行的开始，从当前行的首部开始用 (t_x, t_y) 。 作为一个副作用，这个操作设置了在文本声明中的首要参数，这个操作有如下代码相同的效果： $\begin{matrix} -t_y & TL \\ t_x & t_y \end{matrix} \text{ TD}$
$a \ b \ c \ d \ e \ f$	Tm	设置文本矩阵 Tm ，和文本行矩阵 Tlm ： $T_m = T_{lm} = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix}$
—	T^*	操作数都是数字，初始值 Tm 和 Tlm 都是已定义的矩阵 [1 0 0 1 0 0]，虽然操作数指定一个矩阵，但他们作为 6 个单独的数字通到 Tm ，而不是一个数组。 矩阵由指定的操作数而不串联到当前文本矩阵，但是可以替代它。
—	T^*	移动到下一行首部，这个操作将有和下面代码相同的作用 $0 \ T_l \ Td$
		在此处， Tl 是这个文本声明中的当前首要参数

在一个文本对象开始， Tm 是一个单位矩阵，因此，文本的空间最初起源是

相同的用户空间，文本定位操作，表 5.5 描述从而改变 T_m 和控制的标识符是后来描述的位置。此外，文本显示操作，表 5.6 描述在下一节中，更新 T_m （通过改变其翻译的 e 和 f 部分）以满足每个字形画水平或垂直位移以及任何字符或字间距在文本状态下的参数。

此外，文本对象保持了一个文本行矩阵 Tlm 的轨迹，这个矩阵获得了文本矩阵 T_m 首行的值，这方便调整均匀分布的文本行。文本的定位和文本显示操作者阅读并设置 Tlm 在表 5.5 和 5.6 中提到的具体情况。

注意：文本定位操作者只能出现文本对象中。

5.3.2、文本显示操作者

文本显示操作者（表 5.6）显示页面上的文本，重新定义文本空间。所有的操作者解释文本字符串，并如下所示应用文本声明参数。

表 5.6 文本显示操作

操作数	操作者	描述
string	Tj	显示字符串
string	,	移动到下一行，并显示字符串，这个操作符与下面的编码具有相同的作用 T* String Tj
$a_w\ a_c$ string	" "	移动到下一行，并显示字符串， a_w 为字节间距， a_c 为字符间距（在文本参数中设置相应的参数）。 a_w 和 a_c 在未被缩放的文本空间单位中表示。这些操作与下面编码具有相同的作用 $a_w\ Tw$ $a_c\ Tc$ string ,
array	TJ	表示一个或更多的字符串，使个别字形定位（见附录 H，说明 57）。每个数组元素可以是一个字符串或一个数字，如果元素是一个字符串，此运算显示字符串，如果它是一个数字，操作者会适当调整文本的位置，也就是说，它的文本转换矩阵的 Tm 。这个数字表示一个千分之一单位的文字空间（参见 5.3.3 节，“文本空间细节”，并在附录 H 对其执行情况的说明 58）。这个数是当前水平或垂直坐标的差，依据来自于写入模式。在默认坐标系中，适当的调整可以影响到字形向左或减少所给定的数值。 图 5.11 为一个偏移效果的例子。

[(AWAY again) TJ]	
[(A) 120 (W) 120 (A) 95 (Y again) TJ]	

图 5.11 通过 T_j 作水平变动

注意: 文本显示操作只能出现在文本对象中。

一个文本字符串操作数可以被解释为一个字符识别的字形编码序列。由于大多数的字符类型的每个字符串的字节被视为一个单独的字符代码，这些字符代码被看作字体的字形编码选择，如 5.5.5 节所述，“字符编码”

从 PDF1.2 开始，为了使多个字节编码来选择某个字形，一个字符串可能被表示一个复合字体。在这种情况下，一个或多个字符串的连续字节都被视为一个单一的字符代码。该代码长度和代码映射到字形定义的一个称为 CMap 的数据结构中，如 5.6 节所述，“复合字体”。

字符串必须复合字符串对象的语法。当一个字符串被括在括号中的数据写入字节的值是作为 ASCII 字符相同的左括号 (40)，右括号 (41) 和反斜杠 (92) 必须由一个反斜杠字符在前面，0 到 255 之间的所有其他字节值可能被用来表示一个字符串对象。这些规则适用于每一个字节字符串对象，字符串是否有单字节或多字节字符代码的文本由操作者决定。

字符串在文本显示操作中可能是任意长度，甚至一个单独自己编码一个字符串，同时也可为页面中的任何顺序。一组字形组成的字符串对文本显示并无意义。在调用一个如 Tj 的文本显示操作显示复合字形会产生一个和分开调用每个字形相同的结果。然而，如果字符串足够长并且按正常顺序排列时文本查找（和其他提取的文本）会更重要。

注意: 同样，提取的文本会在很大程度上依赖于字形组，例如 818 页“反向阶显示字符串”。

5.3.3、文本空间详解

如 5.3.1，“文本定位操作”，文本显示在文本空间中，其中文本空间由文本矩阵 T_m ，文本声明参数 T_{fs} , T_h , 和 T_{rise} 一起定义。这就决定了文本如何转换到用户空间，字形形状和位移（水平和垂直）都将在文本空间中解释。

注意: 字形实际上定义在字形空间中，通过前面 5.1.3 节“自行定义和衡量标准”的讨论可以看出该空间的全面定义。字形在从字形空间中转换出来之前作了如下转化。

从文本转换到设备的整个空间都可称为文本修饰矩阵 T_{rm} .

$$T_{rm} = \begin{bmatrix} T_{fs} \times T_h & 0 & 0 \\ 0 & T_{fs} & 0 \\ 0 & T_{rise} & 1 \end{bmatrix} \times T_m \times CTM$$

T_{rm} 是一个临时矩阵。定义：在做文本显示操作前要对每个字形重新计算。

在作完字形后，文本矩阵根据字形分布和所应用的空间参数对其进行更新。

首先，结合位移计算，水平方向记 t_x ，垂直方向记 t_y （该变量对应其他方式时是 0）

$$t_x = \left(\left(w0 - \frac{T_j}{1000} \right) \times T_{fs} + T_c + T_w \right) \times T_h$$

$$t_y = \left(w1 - \frac{T_j}{1000} \right) \times T_{fs} + T_c + T_w$$

此处

$w0$ 和 $w1$ 是字形水平和垂直方向的位移

T_j 是适应 TJ 数组的位置调整。

T_{fs} 和 T_h 是在图形声明中的当前文本大小和水平分布参数。

T_c 和 T_w 是在图像声明中的当前字节和字节空间参数

文本矩阵更新为：

$$T_m = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix} \times T_m$$

5.4、字体数据结构导论

在 PDF 中，一个字体被表示为一个字典来指定字体的类型、字体的补充名称、编码以及当字体程序不可用时能提供一个其子集的信息。或者，可以嵌入字体程序可作为一个字符串对象内嵌在 PDF 文件中。

在字体字典中，字体的类型是由 `Subtype` 项来区别确定的。表 5.7 列出了 PDF 格式中定义的字体类型。0 型字体被称为复合字体，其他类型字体被称为简单字体。除了字体，PDF 还支持两类与字体有关的对象，称为 `CIDFonts` 的和 `CMaps`，

在第 5.6.1 小节“CID-keyed 字体概览”中有描述。CIDFonts 也被列于表 5.7 中，因为，类似字体，它也是字形的一个集合，但 CIDFont 绝不能直接使用，只作为一个 0 型的字体的一部分。

表 5.7 字体的类型		
类型	Subtype 赋值	描述
Type 0	Type0	(适用于 PDF 1.2) 一个复合字体，从派生的 CIDFont 字形组合（详见第 5.6 节，“复合字体”）。
Type 1	Type1	使用 1 型字体技术定义字形类别的字体（详见第 5.5.1，“”1 型字体）。
	MMType1	一个多主的字体——一个 1 型字体的拓展，允许在单个字体上使用种类繁多的字体样式（详见“多主的字体”，第 386 页）
Type 3	Type3	使用 PDF 图形操作符的字符串来定义字形的字体（详见第 5.5.4 小节，“3 型字体”）
TrueType	TrueType	基于 TrueType 字体格式的字体（详见第 5.5.2 小节，“TrueType 字体”）
CIDFont	CIDFontType0	(适用于 PDF 1.2) 一个 CIDFont，字形基于 1 型字体技术（详见第 5.6.3 小节，“CIDFonts”）
	CIDFontType2	(适用于 PDF 1.2) 一个 CIDFont，字形基于 TrueType 型字体技术（详见第 5.6.3 小节，“CIDFonts”）

对于所有的字体类型，字体字典一词是指一个包含字体信息的 PDF 字典；同样的，一个 CIDFont 字典也包含 CIDFont 的信息。除了第 3 类型，这个字典与定义字体字形的字体程序是不同的。该字体程序可能作为一个流对象被嵌入一个 PDF 文件中或由一些外部资源而获得。

注：此术语在这不同于在它在 PostScript 语言中。在 PostScript 语言中，一个字体字典是 PostScript 数据结构，它的创建是解释字体程序直接结果。在 PDF 中，一个字体程序总是被看作一个单独的文件，即使其内容是被嵌入到 PDF 文件中的。必要时，字体方案是由一个专门的字体翻译器

翻译的，它的内容从来没有变成 PDF 格式的对象。

大多数字体方案（及相关程序，如 CIDFonts 的和 CMaps）符合外在的规范，例如 Adobe 1 型字体格式。这本书并不包括这些规范。查看目录，有更多关于本章中提到的规格资料的信息。

当所有用来显示文本的字体程序都被嵌入在 PDF 文件中，才制造出最可预测和最可靠的效果。以下各节正是描述了如何做到这一点。如果一个与字体程序有关联的 PDF 文件没有被嵌入，其结果就取决于使用者应用程序的环境中的字体可用性。以下各节为外部字体程序确定了一些公约。然而，字体命名的一些细节，字体替换以及字形的选择是必要的依赖条件，并可能在不同应用程序和不同的操作系统环境中有所差异。

5.5、简单的字体

有几种简单的字体，它们有以下的特征：

- ◆ 字体的字形由一个从某字符串获得的单字节字符代码确定，该字符串由文本显示运算符给出。可以推出，这些代码索引形成 256 个字形的表格；从代码到字形的映射就称为字体的译码过程。每个字体程序都有一个内置的编码。在某些情况下，译码过程可以被 5.5.5 节中描述的方法更改，即为“字符编码”。
- ◆ 每个字形有一个单一的指标，包括水平位移或设置，宽度，如 5.1.3 节所述，“字形定位和衡量标准，”也就是说，简单的字体只支持横向书写模式。
- ◆ 除了 0 型字体，在非标记的 PDF 文件的 3 型字体，以及某些标准 1 型字体，每个字体列表中包含一个子列表，字体描述符，包含字体宽度指标以及字体的其他属性，见 5.7 小节，“字体描述符”。在那些属性中的是一个可选的字体文件流，包含着字体构成程序。

5.5.1、关于 1 型字体

一个 1 型字体构成程序是一个程式化的 PostScript 程序，它描述字形。它使用的字形描述是一个紧凑的编码，它包括提示资料，使得即使是在小尺寸和低决议的情况下也可高品质的演示。关于这种格式的详情我们另外在一本的书中提供，Adobe1 型字体格式。另一种方法是，第 1 型字体方案的一种更紧凑但功能相当的替代，被记录在 Adobe 技术说明 #5176，紧凑型字体格式规范。

注：虽然 1 型字体的程序使用 PostScript 语言的语法，但是使用它不要求完全的 PostScript 解释器，一个专门的 1 型字体翻译器就足够了。

一个 1 型字体列表收录了在表 5.8 中列出的项。对于 385 页上“标准 1 型字体”之后列出的标准的 14 种字体，有些条目可选的，但除此以外的是必须的。

表 5.8 在 1 型字体字典中的项目

关键词	类型	用途
Type	名称	(必须的) 该字典描述的 PDF 对象；必须为字体字典中的字体。
Subtype	名称	(必须的) 字体的类型；必须为 1 型字体的类型。
Name	名称	(在 PDF1.0 是必须的；除此之外是可选的) 当前资源下的字典中，该字体于字体子列表里引用的名称。 注：该项正逐渐被废弃，不再推荐使用。(详见附录 H 中执行情况说明第 59 页。)
BaseFont	名称	(必须的) 字体的补充名称。对于 1 型字体，这通常是字体方案中的字体名称的赋值；更多的信息，请参阅第三版 PostScript 语言参考的 5.2 节。在使用着的应用过程中，字体

		的补充名称可用于查找字体的描述。这也是打印到 PostScript 输出设备时,用到的名称。
FirstChar	整数	(除对标准的 14 种字体外, 是必须的) 字体宽度的数组中定义的第一个字符的代码。
LastChar	整数	(除对标准的 14 种字体外, 是必须的) 字体宽度的数组中定义的最后一个字符的代码。
Width	数组	(除对标准的 14 种字体外, 是必须的; 适合间接引用) 一个宽度数组 (等于最后一个字符代码减去第一个字符代码加上 1) 每个字形宽度元素作为字符代码, 等于 FirstChar 加上数组索引。对于 FirstChar 到 LastChar 之外的字符代码, 就用到该字体的字体描述项中 MissingWidth 的赋值。在字形宽度的衡量中, 其 1000 个单位对应 1 个单位的文本宽度单位。这些宽度必须与特定的字体方案的实际宽度保持一致。(详见附录 H 中执行情况说明第 60 页。) 更多有关字形宽度和其他字形数据项, 请参阅 “字形定位和衡量标准” 第 5.1.3 小节。
FontDescriptor	字典	(除对标准的 14 种字体外, 是必须的; 必须要间接引用) 一个字体描述符, 描述字体的默认度量项而不是它的字形宽度(见第 5.7 节, “字体描述”)。注: 对于标准 14 种字体, FirstChar , LastChar , Width , 以及 Font Descriptor 几项必须全部设置或全部不设置。通常, 不设置它们; 使他们指定的标准字体被覆盖(见“标准 1 型字体, 如下所示”)。
Encoding	名称或字典	(可选的) 与内置的译码过程不同时, 对一个字体字符译码过程的详细说明。 Encoding 项的赋值为一个预定义的译码过程 (MacRomanEncoding , MacExpertEncoding ,

		或 WinAnsiEncoding, 见附录 D) 或者为译码过程字典, 指明字体的内置编码的变化或指明一个预定义的译码过程(见 5.5.5 小节,“字符编码”)。
ToUnicode	字符串	(可选的, PDF 1.2 版) 一个字符串包含一个 CMap 文件, 将字符代码映射到 Unicode 值。(详见第 5.9 小节,“文本内容摘要”)。

例 5.6 显示了 Adobe Garamond ® 半粗体字体的字体字典。虽然字体有一个译码字典 (对象 25), 但是无论是译码字典还是字体描述器 (对象 7) 都没有在例子中给出。

Example 5.6

```
14 0 obj
<< /Type /Font
/Subtype /Type1
/BaseFont /AGaramond-Semibold
/FirstChar 0
/LastChar 255
/Widths 21 0 R
/FontDescriptor 7 0 R
/Encoding 25 0 R
>>
endobj

21 0 obj
[ 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 280 438 510 510 868 834 248 320 320 420 510 255 320 255 347
510 510 510 510 510 510 510 510 510 510 255 255 510 510 510 330
781 627 627 694 784 580 533 743 812 354 354 684 560 921 780 792
588 792 656 504 682 744 650 968 648 590 638 320 329 320 510 500
380 420 510 400 513 409 301 464 522 268 259 484 258 798 533 492
516 503 349 346 321 520 434 684 439 448 390 320 255 320 510 255
627 627 694 580 780 792 744 420 420 420 420 420 420 402 409 409
409 409 268 268 268 533 492 492 492 492 492 520 520 520 520
486 400 510 510 506 398 520 555 800 800 1044 360 380 549 846 792
713 510 549 549 510 522 494 713 823 549 274 354 387 768 615 496
330 280 510 549 510 549 612 421 421 1000 255 627 627 792 1016 730
500 1000 438 438 248 248 510 494 448 590 100 510 256 256 539 539
486 255 248 438 1174 627 580 627 580 580 354 354 354 354 792 792
790 792 744 744 268 380 380 380 380 380 380 380 380 380 380 380]
```

]
endobj

标准 1 型字体

14 种 1 型字体的补充名称, 被称为是标准字体, 列出如下:

Times-Roman	Helvetica	Courier	Symbol
Times-Bold	Helvetica-Bold	Courier-Bold	ZapfDingbats
Times-Italic	Helvetica-Oblique	Courier-Oblique	
Times-BoldItalic	Helvetica-BoldOblique	Courier-BoldOblique	

这些字体，或者它们的字体描述项和适当的替代字体，必须可用于顾客应用程序。字符集以及这些字体的译码过程都在附录 D 中。对标准的 14 种字体的 Adobe 字体描述项 (AFM) 文件，在 ASN 网站上提供（详见参考文献）。欲了解更多有关字体描述项的信息，请参阅 Adobe 技术摘记 #5004, Adobe 字体描述项文件格式规范。

一般来说，一个代表某一标准字体的字体词典，应该省略 FirstChar, LastChar, Width, 以及 Font Descriptor 几个描述项。但是，覆盖标准字体是被允许的，只要包括进这些描述项，并在 PDF 文件中嵌入字体方案。（见附录 H 中的执行情况说明，第 61 页。）

多个主的字体

多个主的字体格式是 1 型字体格式的一个拓展，对单个字体程序，它允许使用的字体样式种类繁多。这是通过字体的各种设计方面的存在实现的。设计方面的例子是字体重量（加粗的大小）和宽度（扩展的大小）。这些设计方面涉及到的角度（例如字体粗的程度）由数字来限定。一组特别的数字选定一个多主的字体实例。Adobe 技术摘记 #5015 中，1 型字体格式的补充，详细介绍了多个主的字体。

一个实例多个主的字体实例的字体字典，与 1 型字体字典有相同描述项（详见 383 页，表 5.8），但它们有以下区别：

- ◆ 描述项 Subtype 赋值为 MMTyp1。
- ◆ 如果该实例的 PostScript 名称包含空格，Base Font 项的赋值中空格被替换为下划线。例如，正像例 5.7 中所示，名称“MinionMM 366 465 11”（它就是以空格结尾的）就变为了 / MinionMM_366_465_11_。

Example 5.7

```
7 0 obj
<< /Type /Font
/Subtype /MMType1
/BaseFont /MinionMM_366_465_11_
/FirstChar 32
/LastChar 255
/Widths 190 R
/FontDescriptor 60 R
-Encoding 50 R
>>
endobj

19 0 obj
[ 187 235 317 430 427 717 607 168 326 326 421 619 219 317 219 282 427
...Omitted data...
569 0 569 607 607 239 400 400 400 400 253 400 400 400 400 400
]
endobj
```

这个例子说明了一个包括设计方面的数字值，作为该公约的坐标实例的 **Base Font** 名称的一部分。该公约常被用于，从使用者应用环境的外部资源获取多主的字体实例；这被记录在 Adobe 技术文档注释 #5088，字体命名问题。然而，该公约也未规定为 PDF 格式规范的一部分。特别是，如果此实例的字体程序嵌入在 PDF 文件中，它必须是一个普通的 1 型字体程序，而不是一个多主的字体程序。这个字体方案被称为是多主的字体实例之简介，这个实例包含了设计方面的选择值。

5.5.2、关于 TrueType 型字体

TrueType 字体格式是由苹果电脑公司开发的，并已被接受为微软 Windows 操作系统的标准字体格式。TrueType 字体文件格式的规范，可在苹果 TrueType 参考手册以及微软的 TrueType 1.0 的字体文件的技术规范中获得。

注：一个 TrueType 字体程序作为流对象，可以被直接嵌入 PDF 文件。
第 42 型字体格式是为 PostScript 定义，并不适用于 PDF 格式。

一个 TrueType 字体字典可以与 1 型字体词典（见 383 页，表 5.8）包含同样的描述项，但它们有以下区别：

- ◆ **Subtype** 项的值为 TrueType。
- ◆ **Base Font** 项的赋值不同，如下边所论述。
- ◆ **Encoding** 项的赋值受 5.5.5 小节中描述的约束条件的限制，“符号译码过程”。

Base Font 项赋值的补充名称，由一或两种方式确定：

- ◆ 是否使用的补充名称，在 TrueType 字体的“name”表格中是可选择的。
- ◆ 若在“name”表没有该描述项的情况下，从字体在主机作业系统中的名称来衍生补充名称。在 Windows 系统上，该名称基于 LOGFONT 结构中的 lfFaceName 区域；在 Mac 操作系统中，它基于 FOND 资源的名称。如果名称包含任何空格，空格将被删除。

如果在源文档的字体使用粗体或斜体样式的字体，但没有该风格字体的数据，主机操作系统将自动合成该风格。在这种情况下，一个逗号和样式名称（粗体，斜体，BoldItalic 中选择）会被附加到字体的名称上。例如，对于一个加粗的 TrueType 字体的 New York 变体，描述项 **Base Font** 值被写为写成/NewYork, Bold（如例 5.8 中所示）。

Example 5.8

```
17 0 obj
<< /Type /Font
/Subtype /TrueType
/BaseFont /NewYork,Bold
/FirstChar 0
/LastChar 255
/Widths 23 0 R
/FontDescriptor 7 0 R
/Encoding /MacRomanEncoding
>>
endobj
```

```
23 0 obj
[ 0 333 333 333 333 333 333 333 333 0 333 333 333 333 333 333 333 333 333 333
...Omitted data...
803 790 803 780 780 780 340 636 636 636 636 636 636 636 636 636 636 636 636
]
endobj
```

请注意，对于 CJK（中文 Chinese，日文 Japanese 和韩文 Korean）字体，字体主机系统中的字体名称往往是编译到主机操作系统的脚本中。例如，日本字体中可能含有在日本被使用着，却是某种（无法识别的）日本编码。因此，TrueType 字体名称可能包含多字节字符代码，每一个都需要多个字符去来显示，在一个 PDF 的对象（使用#符号引述需要特殊字符）。

5.5.3、关于字体子集

PDF1.1 中允许文件中包含 1 型字体以及 TrueType 字体的子集。字体和描述字体子集的字体描述，与传统的字体略有不同。这些差异使得应用程序可以识别字体子集，来合并含有相同字体不同子集的文件。（关于字体描述的更多信息，请参阅第 5.7 节，“字体描述。”）

对于字体的子集，字体的补充名称——字体 Base Font 项的赋值和字体描述的 FontName 项——开头为跟在加号 (+) 后的标签。这个标签包含整整 6 个大写字母，字母的选择是任意的，但在相同的 PDF 文件中的不同子集必须具有不同的标记。例如，E00D1A +Poetica 是 Poetica® 的一个子集，是一个 1 型字体。（详见附录 H 中，执行情况说明第 62 页）。

5.5.4、关于 3 型字体

3 型字体不同于 PDF 格式支持的其他字体。一个 3 型字体字典可定义字体；其他字体的字体字典只是包含字体的有关信息，并指出了实际的字形描述所用的单独字体方案。在 3 型字体中，字形是由 PDF 图形运算符的字符串来定义的。这些字符串与字符名称是相联系的。一个独立的

译码输入映射到一个恰当的字形的符号名称。

3型字体比1型字体更灵活，因为它的字形描述可以包含任意的PDF格式的图形运算符。不过，3型字体没有改善小规模，低产出来改进输出的迹象。一个3型字体字典包含的描述项列出在表5.9中。

表5.9 在3型字体字典中的项目		
关键词	类型	用途
Type	名称	(必须的) 该字典描述的PDF对象；必须为字体字典中的字体。
Subtype	名称	(必须的) 字体的类型；必须为3型字体的类型。
Name	名称	(在PDF1.0是必须的；除此之外是可选的) 详见383页中，表5.8。
FontBBox	矩形	(必须的) 一个在字形坐标系统中的矩形区域(见第3.8.4小节，“矩形区域”)，指定了字体的边框。这是包围形状的最小的矩形，当字体的字形全部存入它们的起始坐标时起作用，然后再进行填写。 如果该矩形的4个边界值都是零，框架内限定的字体不存在。如果任何元素不为零，字体边框的准确性是至关重要的。如果任何字形符号超出了限定的矩形，就会导致不正确的情况出现。
FontMatrix	数组	(必须的) 一个6维数组指定字体矩阵，将字形空间映射到文本空间(详见5.1.3小节，“字形定位和衡量标准”)。一个常见的做法是，在有1000个单位的字形坐标系中定义字形，这时候，字体矩阵为[0.001 0.001 0 0 0 0]。
		(必须的) 一个字典，其中每个关键字是一

CharProcs	字典	一个字符名称及与该关键字相关联的值是一个为该字符构建绘制字形的容量字符串。这个字符串必须包含 d0 或者 d1 作为其第一个操作符，其后跟着描述一个或多个图形对象的操作符，其中可能包括路径，文本或图像对象。查看下边内容，有更多关于 3 型字体的字形描述的细节。
Encoding	名称或字典	(必须的) 一个译码字典，其不同的数组为字体指定完整的字符编码(详见 5.5.5 节，“字符编码”，也可参阅附录 H 中，执行情况说明第 63 页)
FirstChar	整数	(必须的) 字体宽度的数组中定义的第一个字符的代码。
LastChar	整数	(必须的) 字体宽度的数组中定义的最后一个字符的代码。
Width	数组	(必须的；适合间接引用) 一个宽度数组(等于最后一个字符代码减去第一个字符代码加上 1) 每个字形宽度元素作为字符代码，等于 FirstChar 加上数组索引。对于 FirstChar 到 LastChar 之外的字符代码，其宽度值为 0。在字形空间中，这些宽度被用来指定字体矩阵(和 1 型字体的宽度描述项不同，其单位是文本空间单位的千分之一)。 注：如果 FontMatrix 项指定了一个转动，只使用转化宽度项的水平分量。也就是说，文本空间中所产生的位移总是水平的，所有简单字体都是这样的。
FontDescriptor	字典	(在标记过的 PDF 文件中是必须的；必须要间接引用) 一个字体描述符，描述字体的默

		认度量项而不是它的字形宽度（见第 5.7 节，“字体描述”）。
Resources	字典	(可选的，但强烈建议选择；适用于 PDF1.2) 已命名资源的列表，例如字体和图像，字体的字形描述符中需要该项。(详见 3.7.2 小节，“资源字典”)。如果任何字形描述符与命名资源相联系了，词典却不存在，就将在字体使用页上的资源字典中查找名称。(详见附录 H，执行情况说明第 64 页)。
ToUnicode	字符串	(可选的，适用于 PDF 1.2 版) 一个字符串包含一个 CMap 文件，将字符代码映射到 Unicode 值。(详见第 5.9 小节，“文本内容摘要”)。

对于使用 3 型字体的文本显示器显示的每个字符编码，使用者的应用过程如下：

1. 在字体编码输入中查找字符代码，正如 5.5.5 小节“字符编码描述”中描述的，以此取得字符的名称。
2. 在字体的 CharProcs 字典中查找字符名称，获得一个字符串对象，其中包含对字形的描述。(如果查找的名称不在 CharProcs 项的关键字里，就不绘制任何字形。)
3. 调用的字形描述，如下所述。图形状态在此次引用前被保存，其后被恢复；因此，字形描述过程中对图形状态做的任何改变，在过程结束后都不会存在了。

当字形描述过程开始执行，当前转换矩阵 (current transformation matrix，即 CTM) 是该字体矩阵 (字体矩阵指的是当前的字体词典) 和文本空间的链接，它实际上是文本显示器被调用的时候(详见第 5.3.3 小节，“文本空间的内容”)。这表明字形坐标系中描述的形状被转化到用户坐标系中，

并以合适的大小和方向出现在该页上。在字形坐标系中，字形描述过程应将该字形描述在绝对坐标中，把原点设定在这个空间的 (0, 0) 位置。对这个文本初始位置应该没有其它的臆断。

除了 CTM 之外，图形状态还继承了文本显示的环境，调用了字形描述过程。为确保可预测的结果，字形描述过程必须初始化图形状态所依赖的所有参数。特别是，如果它调用了 S (Stoke) 操作符，就必须明确地将线宽，线连接，线帽，冲格局设定为适当的值。通常，初始化当前的颜色参数是不必要的和不可取的，因为文本显示操作符设计为，使用当前的颜色绘制字形。

图形描述过程必须要执行表 5.10 中所述的操作符之一，以向字体构成器传递宽度和边框限定的信息。这必须先于任何描述字形的路径建立或路径设定操作符。

注： PDF 中的 3 型字体与 PostScript 中的非常相似。3 型字体字典和字体字形描述过程中的提供一些信息，看似是多余或不必要的，但仍然为必须的，这样才能在 PDF 文件打印到 PostScript 输出设备时，获得正确的结果。这特别适用于 d0、d1 操作符的操作数，它们在 PostScript 中被命名为 setcharwidth (设定字符宽度) 和 setcachedevice (设定缓存设备)。如需进一步说明，请参阅 PostScript 语言参考，第三版，第 5.7 小节。

表 5.10 字体类型 3 操作		
OPERANDS	OPERATOR	DESCRIPTION
wx wy	d0	<p>设置字形的宽度信息且声明指定的字形在形状和颜色上的描述（请注意，此操作符的名字以数字 0 结尾）。wx 指定了在字形坐标系统中的水平位移；它必须符合在字体的宽度数组中相应的宽度。wy 必须是 0（见 5.1.3 节，“字形定位和指标”）。</p> <p>这个操作是合法的，仅在内容流在类型 3</p>

		CharProcs 字典项显示。它特定的被使用，仅仅字形明确地描述操作去设置颜色。
wx wy llx lly urx ury	d1	<p>设置宽度和边界标志符号框中的信息，并声明仅指定的字形描述的形状，它没有颜色。(请注意，这操作名称结尾的数字是1。) wx 指定在字形坐标系的水平位移；它必须与相应字体的宽度数组的宽度一致。 Wy必须是0(见 5.1.3节，“字形定位和度量”).</p> <p>在字形中，llx 和 lly 是左下角的坐标且 urx 和 ury 是右上角的坐标。字形框是最小的矩形，它面向同字形的坐标轴，即完全包围在一个结果页面，它放置所有对执行字形描述的标记。描述的边界框必须正确的，换句话说，它必须大到足以包围整个字形。如果框以外的任何标志，结果是不可预知的。</p> <p>D1项开始的字形描述，操作不应该执行试图去设置颜色（或其他颜色相关的参数）在图形状态；何这操作使用被忽略，字形描述执行完全确定字形的形状。它的颜色由图形状态确定，实际上每个字形被绘制的时间是被一个文字显示操作。出于同样的原因，字形描述可能不包含图片；然而，图像掩码是可以接受的，因为它仅仅定义页面的区域要与当前的色彩。</p> <p>这是只允许操作在字体 3 的 CharProcs 字典内容中出现的流。</p>

一个三号字体的例子

例子展示了只有两个字形的三号字体的定义，——一个为填充的正方形一个为填充的三角形，由字符编码 a 和 b 表示。图 5.12 展示了展示了字体串(ababab)结果。

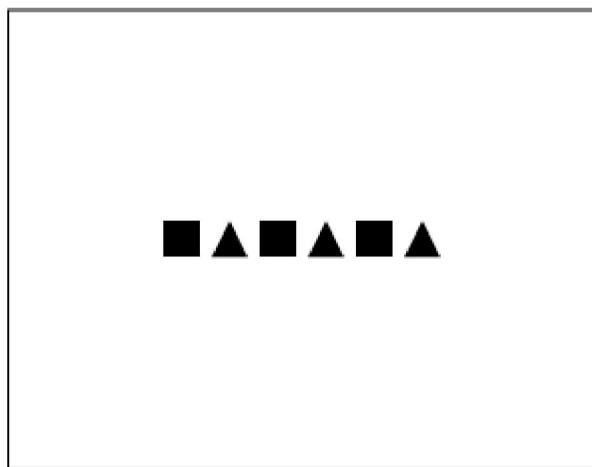


图 5.12 例子输出

```
4 0 obj
<< /Type /Font
/Subtype /Type3
/FontBBox [0 0 750 750]
/FontMatrix [0.001 0 0 0.001 0 0]
/CharProcs 10 0 R
/Encoding 9 0 R
/FirstChar 97
/LastChar 98
/Widths [1000 1000]

>>
endobj

9 0 obj
<< /Type /Encoding
/Differences [97 /square /triangle]
>>
endobj

10 0 obj
<< /square 11 0 R
/triangle 12 0 R
>>
endobj
```

```
11 0 obj
<< /Length 39 >>
stream
1000 0 0 0 750 750 d1
0 0 750 750 re
f
endstream
endobj

12 0 obj
<< /Length 48 >>
stream
1000 0 0 0 750 750 d1
0 0 m
375 750 l
750 0 l
f
endstream
endobj
```

5.5.5、字符编码

字体编码与字符编码（从显示的文本字符串中获得）和字形描述相关。本节描述的字符编码计划以简单的 PDF 字体方式使用。复合字体（类型 0）使用不同的字符映射算法，在 5.6 节“复合字体”讨论。

除第 3 类的字体，每个字体的程序有一个内置的编码。在某些情况下，一个 PDF 字体字典可以更改字体的内置编码来匹配生成的文本应用程序。编码字符的灵活性有以下两个原因：

- 它允许根据现有的各种约定来显示文本编码。例如，微软 Windows 和苹果 Mac 操作系统使用不同编码标准的拉丁文字，而且许多应用程序使用自己的专用编码。
- 它允许应用程序指定字符怎样从大的字符集里选中进行编码。一些字符集

包括超过 256 个字符，包括连字，重音符号，以及高质量排版或非拉丁书写系统所需的其他符号。在相同的字符设置中不同的编码可以选择不同的子集。

拉丁文字的字体程序由 Adobe 公司使用的 Adobe 标准编码产生，通常被称为标准编码。标准编码这个名字在 PDF 格式中没有被明确指定，但这种编码作为默认的编码确实起到很大作用（如表 5.11 所示角色）。有规律的编码用在 Mac OS 和 Windows 操作系统的拉丁文字上，被称为 **MacRomanEncoding** 和 **WinAnsiEncoding**，分开来说。一个名为 **MacExpertEncoding** 编码的“专家”字体，包含其他有用字符来进行复杂的排版。这些编码和字符的完整详细信息在附录 D 中提供的典型字体上。

在 PDF 中，字体归为无符号或符号是根据它的所有字符是否是 Adobe 标准的拉丁字符集的成员。在字体描述中由标记指明，见 5.7.1 节，“字体描述标记。”“符号字体包含其他字符集，那些字符集对上述编码通常并不适用。这种字体方案具有内置的编码，通常是对每种字体来说都是唯一的。该标准的 14 字体包括两个符号字体，符号和印刷符号，其编码和字符集是记录在附录 D 中

一个字体程序的内置编码可以被重写或改变通过在 PDF 中的字体编码字典。可能的修改取决于编码的字体类型，讨论如下。该编码项的值可以是一个名为编码（在预定义的编码一个名字 **MacRomanEncoding**, **MacExpertEncoding**, 或 **WinAnsiEncoding**）或编码字典。编码字典包含在表 5.11 所列的项目中。

表 5.11 编码字典条目

关键字	类型	
Type	name	(可选的)字典描述的 PDF 对象类型，如果存在，必须被编译进编码字典
BaseEncoding	name	(可选的)基编码，也就是说，该编码由该差异项（如果存在）描述的差异，作为一个预定义的编码 MacRomanEncoding , MacExpertEncoding , 或 WinAnsiEncoding (见附录 D 指定名称)。如果此项不存

		<p>在，差异项描述了一个隐含的基础编码的差异。</p> <p>对于嵌入 PDF 文件的字体程序，隐含的基础编码是字体方案的内置编码，如上所述，并进一步在下面特定字体类型进行阐述。否则，一 <code>nonsymbolic</code> 字体，它是 <code>StandardEncoding</code>，并象征性的字体，该字体是内置的编码。</p>
Differences	array	(可选，不推荐使用 TrueType 字体) 数组描述指定的基码差异，或者，如果基码不存在，从一个隐含的基础编码。数组的差异说明如下。

本项值的差异是字符代码和字符名称组织的数组，如下：

```
code1 name1,1 name1,2 ...
code2 name2,1 name2,2 ...
...
coden namen,1 namen,2 ...
```

在一系列改变的字符编码序列中每个代码是第一个索引。第一个字符的名字以代码命名成为与编码相适应的名字。随后的名称取代连续的编码目录，直到下一个编码出现在数组或数组结束。这些序列可以在任何命令中被指定但不应重叠。

例如，在示例 5.9 编码字典中，名称单引号 (') 与 39 个字符代码相关联，音标 (Ä) 与 128 个字符代码，(Å) 与 129 个字符代码，标记 (™) 与 170 个字符代码。

例如 5.9

```
25 0 obj
<< /Type /Encoding
/Differences
[ 39 /quotesingle
  96 /grave
  128 /Adieresis /Aring /Ccedilla /Eacute /Ntilde /Odieresis /Udieresis
    /aacute /agrave /acircumflex /adieresis /atilde /aring /ccedilla
    /eacute /egrave /ecircumflex /adieresis /iacute /igrave /icircumflex
    /idieresis /ntilde /oacute /ograve /ocircumflex /odieresis /otilde
    /uacute /ugrave /ucircumflex /udieresis /dagger /degree /cent
    /sterling /section /bullet /paragraph /germandbls /registered
    /copyright /trademark /acute /dieresis
  174 /AE /Oslash
  177 /plusminus
  180 /yen /mu
  187 /ordfeminine /ordmasculine
  190 /ae /oslash /questiondown /exclamdown /logicalnot
  196 /florin
  199 /guillemotleft /guillemotright /ellipsis
  203 /Agrave /Atilde /Otilde /OE /oe /endash /emdash /quotedblleft
    /quotedblright /quotelleft /quoteright /divide
  216 /ydieresis /Ydieresis /fraction /currency /guilsinglleft /guilsinglright
    /fi /fl /daggerdbl /periodcentered /quotesinglbase /quotedblbase
    /perthousand /Acircumflex /Ecircumflex /Aacute /Edieresis /Egrave
    /iacute /Icircumflex /Idieresis /Igrave /Oacute /Ocircumflex
  241 /Ograve /Uacute /Ucircumflex /Ugrave /dotlessi /circumflex /tilde
    /macron /breve /dotaccent /ring /cedilla /hungarumlaut /ogonek
    /caron
]
>>
endobj
```

按照惯例，`.notdef` 可以用来表示没有字符名称与一个给定的字符代码相关联。

字体类型 1 的编码

一个字体类型 1 方案的字形描述由字符名键入，而不是字符编码。字符的名字是普通的 PDF 格式的名称对象。拉丁文字符的描述通常由单字母组成，如 A 或 a。其他字符都与文字组成的名称有关，如三，符号，或左括号。字体类型

1 的内置编码是指由一个编码数组组成。该数组是字体程序中的一部分，而不是和 PDF 中的字体编码字典混淆在一起。

编码项能够修改字体类型 1 从字符代码到字符名称的映射。不同的排列可以映射一个编码到任何字形描述的名称，该字形存在于字体程序中。而不管字形是否被字体的内置编码或者基码项中特指的编码所引用。

所有字体类型 1 的方案包含一个命名为 `.notdef` 的实际字形。`.notdef` 字形所产生的效果是由字体设计者所决定，在由 Adobe 产生的字体类型 1 的程序集中，它是与空格符相同。如果一个编码字符的名称映射到一个在字体类型 1 程序集中不存在的字符，`.notdef` 字形被替换。

字体类型 3 的编码

字体类型 3 和字体类型 1 一样，包含由字符名键入的字形描述，在这种情况下，字体类型 3 作为字体字典的显示键出现。字体类型 3 从字符编码映射到字符名称是由编码项完全定义，该定义在这种情况下是需要的。

真型字体编码

真型字体方案的内置编码从字符编码直接映射到字形描述，通过称为“`cmap`”（不要与混淆的 CMap 的 5.6.4 节，“CMaps”）的一个内部数据结构。本节介绍如何将 PDF 字体字典的编码项用于“`cmap`”，使得从字符串中的一个字符编码映射到一个真型字体方案。

表“`cmap`”可能包含一个或多个子表，代表了不同的平台，如 Mac OS 和 Windows（用途多个编码）。每个子表由两个数字确定，如 (3, 1)，这代表了一个 ID 平台和一个的特定编码的 ID 平台的组合。在真型字体中字形名称不是强制性的，虽然某些字体方案有一个可选的“后”表中列出字形的名字。如果应用程序需要通过名字来选择字形描述，在字体编码程序表“`cmap`”中，它将字形名转化为编码。如果在表“`cmap`” 没有对应的字形名称，“后”表用来选择一个直接的字形描述。

由于真型字形某些方面的选择是基于用户实现或操作系统，PDF 格式文件使

用真型字体应该遵循一定的准则，以确保所有的应用程序取决于可预测的行为：

- 字体方案应被嵌入。
- 无符号字体应指定 **MacRomanEncoding** 或 **WinAnsiEncoding** 作为没有差异的数组编码项的值。
 - 一个用来显示字形的字体不使用 **MacRomanEncoding** 或 **WinAnsiEncoding**，不应该指定一个编码项。字体描述的象征标志（见表 5.20）应被设置，其字体程序“cmap”表应包含 (1, 0) 子表。它可能还包含一个 (3, 0) 子表，如果存在，此子表应该从 0xF000 到 0xF0FF 范围映射，在 (1, 0) 子表与 0xF0 前面加单字节码和映射到相应的字形描述。

注：一些流行的真型字体方案包含不正确的编码信息。真型字体的实现已演变为启发式处理这些问题，那些启发不在这描述。为了获得最大的可移植性，只有格式良好的真型字体方案形被应用于 PDF 格式的文件。因此，在 PDF 文件中的真型字体程序可能需要进行修改，以符合上述准则。

以下各段是对真型字体编码的描述，开始于 PDF 1.3，在 Acrobat 5.0 和稍后的浏览器。此信息并不一定适用于早期版本或接口。

如果字体有一个名为 **MacRomanEncoding** 或 **WinAnsiEncoding** 的编码，或者如果字体描述的无符号标志（见表 5.20）被设置，查看器创建一个从字符代码映射到字形名字的表：

- 如果编码项是 **MacRomanEncoding** 或 **WinAnsiEncoding**，该表是在附录 D 中描述的初始化映射之一。

如果编码项是一个字典，该表从字典中的基码项初始化（见表 5.11）。记录任何用来更新表格的不同的条目数组。如果存在一个 (3, 1) “cmap”子表（微软的标码）：

- 一个字符编码首先映射到一个字形名称来描述上述表格。
- 字形名然后通过咨询 Adobe 字形列表（见参考书目）映射到标准码。
- 最后，标码值根据 (3, 1) 子表映射到一个字形描述。

如果 (3, 1) 子表不存在，但 (1, 0) 子表 (Macintosh ® Roman) 存在：

- 使用上述表，一个字符编码首先映射到一个字形名。
- 字形名，编码被映射回字形描述，按照标准罗马在微软操作系统中使用的标准（见下文附注）。
- 最后，代码根据 (1, 0) 子表映射到一个字形描述。

在上述情况下，如果字形名称不能被指定映射，字形名可以在字体方案“后”表（如果存在）中查找相关的被使用的字形描述。

表 5.12 MacRomanEncoding 和 Mac OS Roman Encoding 的区别

NAME	CODE (OCTAL)	CODE (DECIMAL)
notequal	255	173
infinity	260	176
lessequal	262	178
greaterequal	263	179
partialdiff	266	182
summation	267	183
product	270	184
pi	271	185
integral	272	186
Omega	275	189
radical	303	195
approxequal	305	197
Delta	306	198
lozenge	327	215
Euro	333	219
apple	360	240

当字体没有编码项，或字体描述的象征标志被设置（在这种情况下，编码项被忽略），发生以下几种情况：

- 如果字体包含一个(3, 0)子表, 字符编码必须是以下范围: 0x0000 – 0x00FF, 0xF000 – 0xF0FF, 0xF100 – 0xF1FF, 或 0xF200 – 0xF2FF 之一。根据编码的范围, 从字符串中的每个比特先从高字节变化, 以形成一个 2 字节的字符, 这是用来从子表选择相关的字形描述。
- 否则, 如果字体包含 (1, 0) 子表, 字符串中的单字节用于从子表中查找相关的字形描述。如果一个字符不能在上述任何一种方式被映射, 结果是相关的实现。

5.6、复合字体

一种复合字体也称为类型 0 的字体, 是一个叫 `CIDFont` 的一个字体对象中得到字形的字体。一种复合字体由字体字典的子类型值是 `Type0` 表示。类型 0 字体被称为根字体, 和其关联的 `CIDFont` 被称为及其子体。

注: 在 PDF 的复合字体是类似于附录中的, 但是有一些限制。特别是, PDF 要求字符内码是由 `CMap` 定义的(下述有描述), 是几种 `PostScript.Also` 中可利用的编码方法的一种, 附言允许类型 0 字体有多个后裔, 也许也是类型 0 字体。PDF 支持只有一个唯一后裔, 这就必须是 `CIDFont`。

当前字体是复合时, 文本显示运算符与简单的字体是不同的。对于简单字体而言, 显示的字符串的每个字节选择一种字形, 而为复合字体时, 可以解码的一个或多个字节序列从子代 `CIDFont` 选择一种字形。这项设施支持大量的字符集的使用, 如中国日、日语以及韩语言。复合字体还简化了有扩音器的编码要求的字体的系统。

此部分首先介绍 `CID` 键字体的结构, 是在 PDF 中支持的唯一的一类复合字体。然后, 它描述了是代表一个 `CID` 键字体的相应命名的组件的 PDF 对象的 `CIDFont` 和 `CMap` 词典。最后, 它描述了相结合一个 `CIDFont` 和一个 `CMap` 生产中要显示的字符串的长度可变字符代码的方式可以访问其字形的字体类型 0 字

体字典。

5.6.1、CID 键字体的概要

CID 键字体提供了方便、高效的方法，来定义多字节字符编码、具有大量的字形的字体和与其他字体中的字形无关的字体。这些功能对大量字符集的书面语言系统的文本表示提供了极大的灵活性，如中国、日本和朝鲜（CJK）的系统。

CID 键的字体结构具体到某些字体节目的外在表示法，称 CMap 和 CIDFont 文件，与一些例子一起为结合和使用那些文件。如前面提到，PDF 不支持整个 CID 键的字体结构，它是独立于 PDF 的；CID 键的字体可以用于其他环境。关于结构的完整文献和文件格式，看 Adobe 技术声明#5092，CID 键的字体技术概要和#5014、Adobe CMap 和 CIDFont 文件规格。这个部分描述代表这些字体节目仅的 PDF 对象。

CID 键字体反映的事实是 CID（字符的标识符）号码用于索引和访问该字体中的字形描述。此方法对于大量字体的情况下，比用于某些简单的字体的字符名称访问的方法更有效。IDs 为符合实施限制的范围从 0 到最大值进行变化（见 920 页上表 C.1）。

字符汇集是字形否认有序集合，用来支持特殊语言的一个或更多普遍的字符集。字符集合的字形顺序决定着每一个字形的 CID 号码。每一个 CID 键字体必须通过 CID 号码具体地对应到字符集合 见第 5.6.2 部分 “**CIDSystemInfo Dictionaries.**”

CMap (character map) 文件指定字符代码和用于标识字形的 CID 号的对应关系。它相当于一个简单字体中的编码的概念。而简单的字体允许最多的一次性编码和同一时间允许的 256 个字形，一个 CMap 可以描述从多字节代码到大量的 CID 键字体中的字形成千上万的映射。例如它可以描述 SHIFT-JIS，几个广泛用于日本编码中的一个。

一个 CMap 可以引用一整个字符集合、一个的子集或多个字符集合。它还可以引用其他字体中的字符的字符代码或字符的名称。CMap 映射产生大量的字体（在 PDF 中，始终是 0）和一个字符选择器（这在 PDF 中始终是一个 CID）。此外，一个 CMap 可以通过引用合并另一个 CMap，而无需重复它。这些特点使字符汇集被结合或被补充并且使所有构成字符通过一个唯一内码进入文本显示的运算符。

CIDFont 文件包含字符集合的标志符号的说明。字形的标志符号描述通常类似于简单的字体如 1 型中所使用的格式。但是，他们由 cid 区分 而不是名称，它们以不同的方式组织的。

在 PDF， CMap 和 CIDFont 由 PDF 对象代表在下文中描述。 CMap 和 CIDFont 编程自己可以名义上参考或被嵌入当在 PDF 文件中有一系列对象时。 如早期陈述，外在文件格式在 Adobe 技术声明#5014、Adobe CMap 和 CIDFont 文件规格被提供。

CID 键字体则与包含标志符号描述的 CIDFont 的 CMap 的组合。它表示为 0 型字体。它包含值为一个 CMap 本词典的编译入口。它包含编码条目的值是一个 CMap 词典，并且其后代字体入口指向包含 CMap 的 CIDFont 词典。

5. 6. 2、CIDSystemInfo 字典

CIDFont 和 CMap 字典包含一个 CIDSystemInfo 入口 即指定 CIDFont 假设的字符收藏同 CMap— CIDFont 使用的 CID 数字的解释联系在一起。字符集由在 CIDSystemInfo 字典的登记，预定和补充词条独特识别，正如表 5.13 所描述。字符集的登记和预定值的是相同的是兼容的。

在一个 CIDFont 中 CIDSystemInfo 条目是指定 CIDFont 的字符集合的一个字典。在 CIDFont 不需要包含在集合中的所有 cid 标志符号描述，它可以包含一个子集。在一个 CMap CIDSystemInfo 条目是单个的字典也可以是字典序列，这取决于是否联系单个字符集合或多个字符集合的字典； 请参阅第 5.6.4，

“CMaps”。

适当行为，一个 CMap 的 CIDSystemInfo 进入可以配合 CIDFont 或 CIDFonts 来使用。如果他们不符，这产生的影响是不可预知的。

表 5.13 在 CIDSystemInfo 字典的词条

关键字	类型	值
注册表	字符串型	(要求)标识的字符集合颁发者的字符串。如 Adobe 关于信息分配注册表标识符，请与 Adobe 解决方案网络或请参阅 ASN Web 站点 (见目录)。
排序	字符串型	(要求)一个字符串，它唯一名称的字符集合内指定的注册表，例如 Japan1。
补充集	整型	(要求)字符集合的补数。原始的字符集合具有大量补 0。每当其他 cid 分配字符集合中，补数增加。补充集不会更改现有 cid 字符集合中的顺序。此值不用来确定字符集合之间的兼容性。

5.6.3、CID 字体

CIDFont 节目包含使用 CID 作为字符选择器的字形的描述。有一下两种 CIDFonts 类型：

- 类型 0 的 CIDFont 包含基于 Adobe 类型 1 字体格式的标志符号明。

注意：当术语“类型 0”时应用于一个 CIDFont 对比“类型 0 型字体”有不同的含义。

- 类型 2 CIDFont 包含基于 TrueType 字体格式的标志符号说明

CIDFont 字典是包含关于 CIDFont 项目的信息的 PDF 对象。虽然它的类型价值是字体， CIDFont 实际上不是字体。它没有一个内码词条，它在字体资源子字典中不可能被列出，并且它不可能使用作为 Tf 运算符的操作数。它仅作为类型 0 字体的后裔使用。在类型 0 字体的 CMap 是定义了内码那地图字符码对在 CIDFont 的 CIDs。表 5.14 列出在 CIDFont 字典的词条。

表 5.14 在 CIDFont 字典的词条

关键字	类型	值
类型	名称	((要求) 这本字典描述 PDF 对象的种类；必须是 CIDFont 字典的字体。)
子类型	名称	(要求) CIDFont 的种类； CIDFontType0 或 CIDFontType2。
基本字体	名称	(要求) 在 CIDFont 的 PostScript 名称。类型 0 CIDFonts 这通常是在程序中 CIDFont CIDFontName 项的值。类型 2 CIDFonts，它派生的简单的 TrueType 字体相同的方式；请参见 5.5.2“TrueType 字体”；在任一情况下名称可以有一个子集，如果适当的前缀；请参阅第 5.5.3，“字体子集合”
CIDSystemInfo	字典	(要求) 包含定义的 CIDFont 的字符集合的项的词典。请参见表 5.13 页 406。
FontDescriptor	字典	(要求；必须是描述 CIDFont's 缺省值的一个间接参考) A 字体描述符除它的字形宽度之外的默认的度量(参见第 5.7 部分，“Font Descriptors”)。
DW	整型	(任意) 在 CIDFont 中，字形默认宽度(参见在第 409 页“CIDFonts 的字形的度量”)。缺省值：1000 年。
W	矩阵	(可选) 在 CIDFont 中字形的宽度的说明。数组的元素具有一种变量的格式，可以为连续的 cid 或一个宽度为一系列的 cid (请参见 409 页的 CIDFonts 字形度量;) 指定个别的宽度。默认值：无 (数据仓库值用于所有字形)。
DW2	矩阵	(可选)；仅适用于用来垂直写字的 CIDFonts) 指定用于垂直写入 (请参见 409 页

		上的“CIDFonts 字形度量”) 的默认度量标准的两个数字的数组。默认值： [880_ 1000]。
W2	矩阵	(可选) ; 仅适用于用来垂直写字的 CIDFonts) 在 CIDFont 中字形的垂直书写的度量标准的说明 (请参见 409 页上的“CIDFonts 字形度量”)。默认值： 无 (DW2 值用于所有字形)。
CIDToGIDMap	流或名称	(可选 ; 只有类型 2 CIDFonts) 一种规范的标志符号指数从 cid 或名称的映射。如果值流中的字节包含标志符号指数从 cid 映射：特定的 CID 值 c 的标志符号索引是一个 2 字节的值存储在字节 2*c 和 2*c+1; 其中第一个字节是高序位字节。如果 CIDToGIDMap 的值的名称，它必须指示 cid 和标志符号指数之间的映射身份映射的身份。默认值： 身份。 这个词条也许仅出现于伴生的字型标准字体节目的在 PDF 文件被埋置的第二类型 CIDFont (参见下个部分)。

在 CIDFonts 的字形选择

类型 0 和类型 2 的 CIDFonts 把柄映射从 CIDs 到字形描述再某些方面是不同的方式。

对于类型 0 CIDFont 程序包含由 cid 标识的标志符号说明。CIDFont 程序标识由一只应将它复制到 PDF CIDFont 字典的 CIDSystemInfo 词典的字符集合。cid 对支持给定的字符集合所有的 CIDFont 程序是否嵌入在 PDF 文件中或从外部源获取的该程序进行统一解释。

当在 CIDFont 中包含一个表示紧凑字体格式 (CFF) 中的嵌入的字体程序时字体描述符中的 FontFile3 词条 (见表 5.23) 可以是 CIDFontType0C 或

OpenType。有两种情况取决于字体程序的内容：

- “CFF”字体程序有一顶字典内使用 `CIDFont` 运算符：这说明用于确定在 CFF 程序中使用字符集表的标志符号过程的 GID 值的 `cid`。GID 值然后用于查找使用 `CharStrings INDEX` 表的标志符号过程。虽然在很多字体中 CID 值和 GID 值是相同的，但是 CID 和 GID 的值可能也会有所不同。
- “CFF”字体程序有不使用 `CIDFont` 操作员的 DICT：IDs 被当做当 GID 价值直接使用，并且字形的产生是通过 `CharStrings` 索引被检索。

对于类型 2，`CIDFont` 程序实际上是 TrueType 字体程序，用在没有地道的 IDs. In 概念的 TrueType 字体程序，由标志符号索引值标识字形说明。字形指数是字体的内部并且不是一致地从一种字体为另一种定义的。相反，TrueType 字体程序包含提供直接从字符代码到字形指数映射为一个或多个预定义编码。

TrueType 字体程序与 CID 字体结构集成以两种方式之一，这取决于是否在 PDF 文件中嵌入字体程序：

- 如果嵌入 TrueType 字体程序，类型 2 的 `CIDFont` 词典必须包含 `CIDToGIDMap` 条目，此条目可以映射到该字体程序中的适当字形描述的字形指数的 `cid`。
- 如果没有嵌入真实类型字体程序，而是参考名称，类型 2 的 `CIDFont` 字典不能包含 `CIDToGIDMap` 词条，因为它提到在一个外在字体程序的字形索引是没有意义的。在这种情况下，IDs 不参加字形选择，并且仅被预定义为 CMapst(参见第 5.6.4 部分，“CMaps”)使用者应用程序选择通过翻译从预定义 CMap 对在 TrueType 字体的“cmap”一个摆放了该编码的编码字符字形手段，这是实现是依赖于实现的。

尽管有时 `cid` 不用在类型 2 `CIDFont` 中选择字形，他们也总是再确定字体的标准中使用，这会在下一节叙述。

每个 CIDFont 必须包含 CID 0，这类似于 .notdef 中 的简单字体中字符的名称（见第 425 页的”未定义字符标志的预处理）。

在 CIDFonts 中的字体标准

正如在 5.1.3 节讨论“字形定位和衡量标准，”一个标志符号的宽度是指之间的标志符号的起源与未来字形横向模式时，创作来源的水平位移。在这种模式下，垂直位移之间的起源始终是 0。为 CIDFont 宽度是指利用数据仓库的 CIDFont 字典和 W 项。这些宽度必须与特定的 CIDFont 方案的实际宽度是一致的。（见执行情况的说明在附录 H 60）

数据仓库项定义默认的宽度，这是所有字形的宽度使用未指定个别。这个项目尤其适合中国，日本和韩文字体，字形，其中的很多相同的宽度。

在 W 阵列允许个别 CIDs 宽度的定义。该数组的元素一两个或三个组成一个组，其中每个组是以下两种格式之一：

$c[w_1 \ w_2 \dots w_n]$

$c_{\text{first}} \ c_{\text{last}} \ w$

第一的格式 c 是一个整数，指定起始的 CID 值；后面的 n 编号为 c 开头的 n 连续 cid 指定宽度的数组。第二个格式定义的范围中的所有 cid 的范围 $c_{\text{first}} \ c_{\text{last}}$ 有相同的宽度.。

下面是一个 W 入口的例子：

```
/W [ 120 [400 325 500]  
      7080 8032 1000  
    ]
```

在此示例中，字形有 cid 120、121 和 122 单元宽度分别为 400、325, 500。在 7080 到 8032 的 所有到 cid 都有 1000 个单位的宽度。

从 CIDFont 可以在垂直书写模式下显示字形。（这是选择由在相关 CMap 的字典 WMode 生效；见第 5.6.4，“CMaps。”），以这种方式使用时，必须界定 CIDFont 垂直解散每个字形布局和位置矢量的横向和有关垂直书写的起源。

默认的位置向量和垂直位移矢量由指定 DW2 入境的 CIDFont 字典。DW2 是两个值：垂直阵列组件的位置向量 v 和垂直分量的位移 w1（见图 5.5 页 366）。该水平分量位置向量总是字形宽度的一半，而该位移向量始终为 0。例如，DW2 矩阵是

/DW2 [880 - 1000]

则字形的位置向量和垂直位移向量

$$v = (w0 \div 2, 880)$$

$$w1 = (0, -1000)$$

其中 对同一个字形，w0 是的宽度（水平位移）。请注意，垂直的组件负值表现的是下一个标志符号的起源，这是因为在一个标准的坐标系中的垂直坐标从底部增加到顶部。

W2 数组允许个别 cid 的垂直度量标准的定义。数组中的元素以两个或五个进行组织，其中每个组是以下两种格式之一：

c [w1_{1y} v_{1x} v_{1y} w1_{2y} v_{2x} v_{2y} ...]
c_{first} c_{last} w1_{1y} v_{1x} v_{1y}

在第一种格式中，c 是起始 CID，跟着包含以三个组为一组的数字的解释的一个数组。每个组由垂直位移矢量 w1 的垂直分量（其水平分量始终是 0）跟水平和垂直的组件组成的，因为连续组位置向量定义垂直度量标准 c 开头的连续 cid 的。第二个格式定义了一系列的 cid 范围从从 cfirat 后到为所有 cid 在此范围内定义垂直度量标准的三个数字的 clast。例如

/W2 [120 [- 1000 250 772]
7080 8032 - 1000 500 900]

此 W2 条目 作为定义为 CID 120 与标志符号的垂直位移矢量 (- 1000, 0) 和作为位置矢量 250 (772)。它还定义了位移矢量的 cid 在范围 [8032 7080] 内作为 (- 1000, 0) 和作为位置向量 (900 500)。

5.6.4、CMaps

从一个 CMap 的指定字符代码映射到字符选择器。在 PDF 格式，字符选择器总是在 CIDFont (如前面提到的，可使用的 PostScript CMaps 名称或代码以及) CIDs。一个 CMap 的服务为简单的字体函数类似于编码字典。该 CMap 的并不是指直接到特定 CIDFont，相反，它是与它相结合的刑事调查的一部分，键入字体，PDF 格式的代表作为一个类型 0 字体词典 (见第 5.6.5，“类型 0 字字典”)。在 CMap 的，字符映射指相关 CIDFont byfont 号码，以 PDF 格式始终为 0。
注： PDF 还使用一种特殊类型的 CMap 字符代码映射到 Unicode 值 (请参见 5.9.2 “ToUnicode CMaps”)。

一个 CMap 还指定了写作模式 —— 水平或垂直 —— 为任何与之相结合的 CMap 和 CIDFont。写作模式确定哪些度量标准字形从该字体绘制时使用。(写作模式被指定为的一部分，CMap 因为在某些情况下编写水平和垂直方向时使用不同的形状。在这种情况下一个 CMap 的水平和垂直变量指定给定的字符代码的不同 cid)。

一个 CMap 所指明的两种方法：

- 作为确定一个预定义的 CMap 消费应用程序
- 已知其定义为其内容是 CMap 文件的流对象的名称对象 (请参阅附录 H 中执行注意 65)

预定义的 CMaps

表 5.15 列出预定义的 CMaps 的名称。这些 CMaps 映射到单个的子代 CIDFont 的 cid 的字符代码。其名称结尾的 H CMaps 指定水平的写作模式；那些结束指定垂直写作模式。

注意： 在 CMaps 的几种定义从 Unicode 编码字符集合到的映射。出现在文本字符串中的 Unicode 值表示大字节序顺序（高序位字节第一次）。包含”UCS2“使用 ucs-2 编码； CMap 名称包含”UTF16“使用 UTF16BE（大字节序）编码。

表 5.15 预定义的中日韩 CMap 名字

(简体) 中文	
GB_ EUC_ H	微软编码 936 页（如果是 0x86），GB 2312-80 字符集，EUC - CN 编码
GB_ EUC_ V	GB_ EUC_ H 竖版
GBpc_ EUC_ H	苹果操作系统，GB 2312-80 字符集，EUC - CN 编码，脚本管理器代码 19
GBpc_ EUC_ V	GBpc_ EUC_ H 竖版
GBK_ EUC_ H	微软编码 936 页（如果是 0x86），GBK 字符集，GBK 编码
GBK_ EUC_ V	GBK_ EUC_ H 竖版
GBKp_ EUC_ H	和 GBK_ EUC_ H 相同但是按比例替换为拉丁文字母的一半，映射字符编码 0x24 为美元标志 (\$) 而不是一元 (¥)
GBKp_ EUC_ V	GBKp_ EUC_ H 竖版
GBK2K_ H	字符设置 GB18030-2000, mixed 1-, 2-, and 4-byte encoding
GBK2K_ V	GBK2K_ H 竖版
UniGB_ UCS2_ H	万国码 (UCS-2) 为 Adobe-GB1 character 字符搜集
UniGB_ UCS2_ V	UniGB_ UCS2_ H 竖版
UniGB_ UTF16_ H	万国码 (UTF-16BE) 为 Adobe-GB1 character 字符搜集
UniGB_ UTF16_ V	UniGB_ UTF16_ H 竖版

中文 (繁体)	
B5pc_ H	苹果公司操作系统，五大字符集，五大编码集，脚本管理代码 2
B5pc_ V	B5pc_ H 竖版
HKscs_ B5_ H	香港统一编码，五大扩展字符集和编码
HKscs_ B5_ V	ETen_ B5_ H 竖版
ETen_ B5_ H	和 ETen_ B5_ H 相同但是按比例替换为拉丁文字母的一半
ETen_ B5_ V	ETenms_ B5_ H 竖版
ETenms_ B5_ H	和 ETen_ B5_ H 相同但是按比例替换为拉丁文字母的一半
ETenms_ B5_ V	ETenms_ B5_ H 竖版
CNS_ EUC_ H	CNS 11643-1992 字符集，繁体中文编码
CNS_ EUC_ V	CNS_ EUC_ H 竖版
UniCNS_ UCS2_ H	为 Adobe-CNS1 字符选择的万国码 (UCS-2)
UniCNS_ UCS2_ V	UniCNS_ UCS2_ H 竖版
UniCNS_ UTF16_ H	为 Adobe-CNS1 字符选择的万国码 (UTF-16BE)，包含 HKSCS-2001 的所有字符和 2-4 字符编码映射
UniCNS_ UTF16_ V	UniCNS_ UTF16_ H 竖版
日语	
83pv_ RKSJ_ H	苹果操作系统，日本汉字的 JIS X 0208 字符集，Shift-JIS 编码，管理脚本编码 1
90ms_ RKSJ_ H	微软编码 932 页 (如果是 0x80)，NEC 和 IBM 公司的 JIS X 0208 字符集
90ms_ RKSJ_ V	90ms_ RKSJ_ H 竖版
90msp_ RKSJ_ H	和 90ms_ RKSJ_ H 相同但是按比例替换为拉丁文字母的一半
90msp_ RKSJ_ V	90msp_ RKSJ_ H 竖版

90pv_ RKSJ_ H	苹果操作系统，日本汉字版本七的 JIS X 0208 字符集，Shift-JIS 编码，管理脚本编码 1
Add_ RKSJ_ H	日本富士通的 JIS X 0208 字符集，Shift-JIS 编码
Add_ RKSJ_ V	Add_ RKSJ_ H 竖版
EUC_ H	JIS X 0208 字符集，EUC-JP 编码
EUC_ V	EUC_ H 编码
Ext_ RKSJ_ H	日本电气公司的 JIS C 6226 (JIS78) 字符集，Shift-JIS 编码
Ext_ RKSJ_ V	Ext_ RKSJ_ H 竖版
H	JIS X 0208 字符集，ISO-2022-JP 编码
V	H 竖版
UniJIS_ UCS2_ H	日本 adobe 公司的万国码(UCS-2) 字符集
UniJIS_ UCS2_ V	UniJIS_ UCS2_ H 竖版
UniJIS_ UCS2_ HW_ H	和 UniJIS_ UCS2_ H 相同但是按比例替换为拉丁文字母的一半
UniJIS_ UCS2_ HW_ V	UniJIS_ UCS2_ HW_ H 竖版
UniJIS_ UTF16-H	为日本 adobe 公司采集的万国码 (UTF-16BE)，包含 JIS X 0213:1000 字符集中的所有字符映射
UniJIS_ UTF16-V	UniJIS_ UTF16-H 竖版
韩语	
KSC_ EUC_ H	KS X 1001:1992 字符集，EUC-KR 编码
KSC_ EUC_ V	KSC_ EUC_ H 竖版
KSCms_ UHC_ H	微软编码 949 页 (lfCharSet 0x81)， KS X 1001:1992 字符集外加 8822 韩语， 统一韩语编码 (UHC)
KSCms_ UHC_ V	KSCms_ UHC_ H 竖版
KSCms_ UHC_ H	和 KSCms_ UHC_ H 相同但是按比例替换

	为拉丁文字符的一半
KSCms_ UHC_ H	KSCms_ UHC_ HW_ H 竖版
KSCpc_ EUC_ H	苹果操作系统，由苹果操作系统延伸的韩国标准 1001:1992 字符集，脚本管理代码 3
UniKS_ UCS2_ H	为韩国 adobe 字符采集的万国码(UCS-2)
UniKS_ UCS2_ V	UniKS_ UCS2_ H 竖版
UniKS_ UTF16_ H	为韩国 adobe 字符采集的万国码(UTF-16BE)
UniKS_ UTF16_ V	UniKS_ UTF16_ H 竖版
Generic	
Identity_ H	两字节的 CIDs 同等映射，可能被用来任何的注册，排序，补充。它映射了从 0-65535 的双字节编码，高字节优先解释
Identity_ V	Identity_ H 竖版，和 Identity_ H 一样映射

当显示一个文本字符串时，H 身份和 V 身份的 CMaps 可以通过 CIDs 直接被用来参考字形。如果当前字体是类型零且编码项是 H 身份或 V 身份，显示的字符串被解释为 CIDs 的一对字节，高字节优先。这适用于任何 CID 字体，独立于它的字符采集。此外，在真型字体程序中，2 字节的 CID 代表字型描述。这只有真型字体方案嵌入在 PDF 文件中时才起作用。

表 5.16 根据不同版本的 PDF，参考 CMaps 列出了字符集。破折号（—）表示该 CMap 在该 PDF 版本中是没有被预定义的。

注：CMaps 为 CID Maps

表 5.16 根据 PDF 版本预定义的 CMaps 字符集

CMAP	PDF 1.2	PDF 1.3	PDF 1.4	PDF 1.5
UniCNS-UCS2-H/V	—	Adobe-CNS1-0	Adobe-CNS1-3	Adobe-CNS1-3
UniCNS-UTF16-H/V	—	—	—	Adobe-CNS1-4
<i>Japanese</i>				
83pv-RKSJ-H	Adobe-Japan1-1	Adobe-Japan1-1	Adobe-Japan1-1	Adobe-Japan1-1
90ms-RKSJ-H/V	Adobe-Japan1-2	Adobe-Japan1-2	Adobe-Japan1-2	Adobe-Japan1-2
90msp-RKSJ-H/V	—	Adobe-Japan1-2	Adobe-Japan1-2	Adobe-Japan1-2
90pv-RKSJ-H	Adobe-Japan1-1	Adobe-Japan1-1	Adobe-Japan1-1	Adobe-Japan1-1
Add-RKSJ-H/V	Adobe-Japan1-1	Adobe-Japan1-1	Adobe-Japan1-1	Adobe-Japan1-1
EUC-H/V	—	Adobe-Japan1-1	Adobe-Japan1-1	Adobe-Japan1-1
Ext-RKSJ-H/V	Adobe-Japan1-2	Adobe-Japan1-2	Adobe-Japan1-2	Adobe-Japan1-2
H/V	Adobe-Japan1-1	Adobe-Japan1-1	Adobe-Japan1-1	Adobe-Japan1-1
UniJIS-UCS2-H/V	—	Adobe-Japan1-2	Adobe-Japan1-4	Adobe-Japan1-4
UniJIS-UCS2-HW-H/V	—	Adobe-Japan1-2	Adobe-Japan1-4	Adobe-Japan1-4
UniJIS-UTF16-H/V	—	—	—	Adobe-Japan1-5

CMap	PDF 1.2	PDF 1.3	PDF 1.4	PDF 1.5
UniCNS-UCS2-H/V	—	Adobe-CNS1-0	Adobe-CNS1-3	Adobe-CNS1-3
UniCNS-UTF16-H/V	—	—	—	Adobe-CNS1-4
<i>Japanese</i>				
83pv-RKSJ-H	Adobe-Japan1-1	Adobe-Japan1-1	Adobe-Japan1-1	Adobe-Japan1-1
90ms-RKSJ-H/V	Adobe-Japan1-2	Adobe-Japan1-2	Adobe-Japan1-2	Adobe-Japan1-2
90msp-RKSJ-H/V	—	Adobe-Japan1-2	Adobe-Japan1-2	Adobe-Japan1-2
90pv-RKSJ-H	Adobe-Japan1-1	Adobe-Japan1-1	Adobe-Japan1-1	Adobe-Japan1-1
Add-RKSJ-H/V	Adobe-Japan1-1	Adobe-Japan1-1	Adobe-Japan1-1	Adobe-Japan1-1
EUC-H/V	—	Adobe-Japan1-1	Adobe-Japan1-1	Adobe-Japan1-1
Ext-RKSJ-H/V	Adobe-Japan1-2	Adobe-Japan1-2	Adobe-Japan1-2	Adobe-Japan1-2
H/V	Adobe-Japan1-1	Adobe-Japan1-1	Adobe-Japan1-1	Adobe-Japan1-1
UniJIS-UCS2-H/V	—	Adobe-Japan1-2	Adobe-Japan1-4	Adobe-Japan1-4
UniJIS-UCS2-HW-H/V	—	Adobe-Japan1-2	Adobe-Japan1-4	Adobe-Japan1-4
UniJIS-UTF16-H/V	—	—	—	Adobe-Japan1-5
<i>Korean</i>				
KSC-EUC-H/V	Adobe-Korea1-0	Adobe-Korea1-0	Adobe-Korea1-0	Adobe-Korea1-0
KSCms-UHC-H/V	Adobe-Korea1-1	Adobe-Korea1-1	Adobe-Korea1-1	Adobe-Korea1-1
KSCms-UHC-HW-H/V	—	Adobe-Korea1-1	Adobe-Korea1-1	Adobe-Korea1-1
KSCpc-EUC-H	Adobe-Korea1-0	Adobe-Korea1-0	Adobe-Korea1-0	Adobe-Korea1-0
UniKS-UCS2-H/V	—	Adobe-Korea1-1	Adobe-Korea1-1	Adobe-Korea1-1
UniKS-UTF16-H/V	—	—	—	Adobe-Korea1-2
<i>Generic</i>				
Identity-H/V	Adobe-Identity-0	Adobe-Identity-0	Adobe-Identity-0	Adobe-Identity-0

如 5.6.2 节指出：“CIDS 系统信息字典”字符集在注册，订购，数量补充，补充累积中被识别，也就是说，一个更高编号的补充，包括较低编号的补充和以及一些额外的 CIDs。因此，文字编码在 CMaps（给定的 PDF 版本中定义的）是有效的，当它由相同或更高的 PDF 版本所支持的用户程序来解释。当由较早的 PDF 版本解释时，如果遇到的 CMap 没有在该 PDF 版本中预定义，该文本会导致错误。如果遇到字符代码，要用一个更高版本的 PDF 而不是一个与之相当的 PDF 版本，没有任何字符显示这些代码，参见第 425 页上“处理未定义”。还有执行情况的说明见附录 H 注解 66。

注意：如果应用程序产生的 PDF 文件遇到文本，该文本使用的 CIDs 来自一个较高的 PDF 版本而不是与之相当的版本，该申请应该被嵌进更高级别的 Cmaps 编号附录而不是嵌进预定义的 CMap（见下一节）。

CMap 方案确定的 CMaps 预定义是可用的，它可以通过 ASN 网站，也可由卡梅隆写的书—《CJKV 信息处理》提供。字符搜集的细节，包括所有的 CIDs 样本字形，可以在一个 Adobe 技术手册上被找到。如果想了解关于这些注解和上面所说的图书信息，请参阅书目。

嵌入式 CMap 的文件

对于没有预定义的字符编码，PDF 文件必须包含一个定义的 CMap 流。除了该流的标准条目（第 38 页表 3.4 列出），CMap 流字典包含表 5.17 所列的项目。流中的数据定义了从字符代码到一个字号码和字符的映射。该数据必须遵循 Adobe 技术规范 #5014，Adobe 公司的 CMap 和 CIDFont 文件规范的语法。

表 5.17 在 CMap 字典中的额外条目

关键字	类型	值
Type	name	(必须的) 该字典描述的 PDF 对象类型，对于 Cmap 字典来说必须是 CMap，(尽管是一个类型为 0 的字体编码，它的值为 CMap。)
CMapName	name	(必须的) CMap 附录的名字。它必须和 CMap 文件中的 CMapName 的值一样
CIDSystemInfo	dictionary	(必须的) 字典(见 5.6.2 节，“CIDS 系统信息字典”) 包含的条目定义了

		<p>字符集，该字符集是 CIDFont 或 CIDFont 相关的 CMap。</p> <p>条目值应该和 CMap 文件中的 CIDSystemInfo 值一样，（然而它确实不需要和 H 身份或 V 身份的 CIDSystemInfo 的值匹配）</p>
WMode	integer	<p>(必须的) C 决定任何 IDFont 的写作模式的编码被组合，可能值为水平的 0 或垂直的 1</p> <p>条目值必须和 CMap 文件中的窗口模式一样</p>
UseCMap	name or stream	<p>(必须的) 预定义的 CMap 名称，或者一个流包含一个 CMap，被用来作为 CMap 的基础。该基础准许 CMap 被定义成不同，指明只有字符映射不同</p>

CMap 实例与操作概要

CMap 文件是完全记录在 Adobe 技术说明 #5014, Adobe 公司的 CMap 和 CIDFont 文件规范上。下面一个 CMap 流对象的例子阐明和部分解释了一个 CMap 的文件的内容。包括以下几个原因：

- 它记录了一些限制条件，这些限制是嵌入在 PDF 文件中的 CMaps 文件。
- 它提供背景以帮助理解以后的材料，特别是 424 页的“CMap 的映射”。
- 这是 PDF 功能的基础，转换为万国码的 CMap，这是一个 CMap 的文件格式扩展。这一扩展在 5.9 节描述，“提取文本内容”。

”范例 5.10 是一个日文 Shift – JIS 编码的 CMap 样本。在这种编码中字符编码可以是 1 或 2 个字节的长度。这种 CMap 可以以 CIDFont 的方式使用，注意：流字典中的几个条目同样被被流数据复制。

例 5.10

```
22 0 obj
<< /Type /CMap
/CMapName /90ms-RKSJ-H
/CIDSystemInfo << /Registry (Adobe)
/Ordering (Japan1)
/Supplement 2
>>
/WMode 0
/Length 23 0 R
>>

stream
%IPS-Adobe-3.0 Resource-CMap
%%DocumentNeededResources: ProcSet (CIDInit)
%%IncludeResource: ProcSet (CIDInit)
%%BeginResource: CMap (90ms-RKSJ-H)
%%Title: (90ms-RKSJ-H Adobe Japan1 2)
%%Version: 10.001
%%Copyright: Copyright 1990–2001 Adobe Systems Inc.
%%Copyright: All Rights Reserved.
%%EndComments

/CIDInit /ProcSet findresource begin
12 dict begin
begincmap
/CIDSystemInfo
3 dict dup begin
/Registry (Adobe) def
/Ordering (Japan1) def
/Supplement 2 def
end def
```

```
/CMapName /90ms-RKSJ-H def
/CMapVersion 10.001 def
/CMapType 1 def
/UIDOffset 950 def
/XUID [1 10 25343] def
/WMode 0 def

4 begincodespacerange
<00> <80>
<8140> <9FFC>
<A0> <DF>
<E040> <FCFC>
endcodespacerange

1 beginnotdefrange
<00> <1F> 231
endnotdefrange

100 begincidrange
<20> <7D> 231
<7E> <7E> 631
<8140> <817E> 633
<8180> <81AC> 696
<81B8> <81BF> 741
<81C8> <81CE> 749
... Additional ranges...
<FB40> <FB7E> 8518
<FB80> <FBFC> 8581
<FC40> <FC4B> 8706
endcidrange
endcmap
CMapName currentdict /CMap defineresource pop
end
end
%%EndResource
%%EOF
endstream
endobj
```

可以从这个例子看到，一个 CMap 的文件符合 PostScript 语言的语法，但是，一个完整的 PostScript 解释器不需要解释它。除了一些需要的样板，CMap 的文件由一个或多个特殊的 CMap 的构造运算，按照特定的顺序调用。以下是这些操作的概述：

开始 cmap 和结束 cmap 包含在 CMap 定义上。

usecmap 含有从另一个 CMap 的文件中的代码映射。在 PDF 中，其他 CMap 的必须同样确定在 CMap 字典中的 UseCMap 项中被鉴定（见表 5.17 419 页）。

开始编码距离变化和结束编码距离变化界定编码距离范围，有效的输入字符代码范围，通过指定一些特定的上，下界给予特别长度；见“424 页上的 CMap 的映射”。

usefont 指定一个字体数字，是一个关于所有字符编码映射操作的，遵循隐含的操作数。在 PDF 格式，字体数必须是 0，因此，**usefont** 通常不会真正出现。

beginbfchar 和 **endbfchar** 明确个体输入字符代码映射到字符代码或在相关的字体字符的名称。**beginbfrange** 和 **endbfrange** 做的输入码范围相同。在 PDF 中，这些运营商可能不会出现在一个被看作是一个类型 0 字型编码输入使用 CMap 的，但是，它们可能出现在 1 ToUnicode CMap 的定义（见 5.9 节，“提取的文本内容”）。

begincidchar 和 **endcidchar** 界定相关 CIDFont 个别输入字符代码映射到 CID。 **begincidrange** 和 **endcidrange** 不相同，但对输入码范围。

beginnotdefchar, **endnotdefchar**, **beginnotdefrange** 和 **endnotdefrange** 定义从字符代码 notdef 映射 CID。正如在第 425 页上的“处理未定义的字符”所描述的，一个 notdef 映射被使用如果正常使用的映射产生 CID，该 CID 没有字形与 CIDFont 关联。

该 **beginrearrangedfont** , **endrearrangedfont** , **beginusematrix** 和 **endusematrix** 操作，在 Adobe 的技术说明注 #5014, Adobe 公司的 CMap 和 CIDFont 文件规范，不能在 CMap 中使用。

5.6.5、字体类型 0 字典

字体类型 0 字典包含表 5.18 所列出的所有内容。例 5.11 显示了一个字体类型 0 引用的一个单一的 CIDFont。该 CMap 的使用是在 412 页表 5.15 中列出的，并按名称引用。

表 5.18 类型零字体字典条目		
关键字	类型	值
Type	name	(必选) PDF 对象字典描述的类型；一定是字体字典的字体
Subtype	name	(必选) 字体类型；一定是字体类型 0 的类型
BaseFont	name	(必选) 字体附录名，原则上这是一个任意的名字，自从没有字体方案与字体类型零直接关联。这里描述的规定确保对已存在的标准产品的最大兼容性。
Encoding	name or stream	(必选) 如果该后代是一个类型零的 CIDFont，该名字由 CIDFont 的基字形名，连字号和 CMap 名称连接而成。如果该后代是一个类型 2 的 CIDFont，这个名字将和 CIDFont 的基字体名称一样。
DescendantFonts	array	(必选) 预先定义的 CMap 名字或流，该流包含的 CMap 从字符编码映射到字体数和 IDs
ToUnicode	stream	(可选) 包含 CMap 文件的流，该流将字符编码映射到万国码的值

例 5.11

```
14 0 obj
<< /Type /Font
/Subtype /Type0
/BaseFont /HeiseiMin-W5-90ms-RKSJ-H
/Encoding /90ms-RKSJ-H
/DescendantFonts [15 0 R]
>>
endobj
```

CMap 映射

一个类型 0 字体字典编码项指定一个 CMap，它确定如何文本显示操作（像 Tj）解释字符串的字节怎样被显示，如果当前字体是字体类型 0。以下各段描述了字符串中的字符如何被解码成字符选择器（在 PDF 中必须始终 CIDs）。

在 CMap（由 **begincodespacerange** 和 **endcodespacerange** 分隔）中的字符间距决定有多少字节从每个连续的字符串编码中提取。一个代码间距范围由某些特定给予的边界上，下长度码对。一个代码被认为是匹配的范围，如果它是作为包围的代码和它的字节长度是相同的每个值之间的上下界的相应字节。该代码的长度不能超过一个整数字节数表示的（见附录 C）。

一个或更多字节的序列从字符串和 CMap 编码间距中提取出来，也就是说，第一个字节和一比特的编码间距相匹配；如果找不到相匹配的，一个两字节的编码被取出，且两字节的编码和两字节的编码间距相匹配。此过程一直持续直到一个匹配被找到或者所有的编码间距都被测试过。这将至多有一个匹配因为编码间距不重叠。

编码从字符串中被提取出来，该字符串在为字符编码长度的映射中被查出（这些是被 **beginbfchar**, **endbfchar**, **begincidchar**, **endcidchar** 和相应的操作变换中所定义的映射）如果没有这个，要在 notdef 映射中查找，就像下节中描述的一样。

CMap 映射结构的结果是一个字体数和字符集。字体数被用来作为字体类型 0 的 **DescendantFonts** 数组索引，该索引用来选择 CIDFont。在 PDF 中，字体数总是 0 字符集总是 CIDFont。如果 CIDFont 在 CID 中不包含字形，notdef 映射被启

用，相关的描述在下一节中

处理不确定的字符

一个 CMap 映射操作可能因为很多原因无法选择一个字形。这个章节描述了这些原因和他们出现时什么将会发生。

如果一个编码映射到一个 CID 且没有这样的字形存在于 CIDFont 的后续，在 CMap 中的 **notdef** 映射被咨询用来获得一个代替的字符选择器。这些映射（所以在简体中文中顺道访问分析. **notdef** 字符机制）通过操作 **beginnotdefchar**, **endnotdefchar**, **beginnotdefrange**, **and endnotdefrange**. 被界定。它们总是映射到一个 CID。如果一个 **notdef** 映射匹配被找到，CID 会在相关的后代中选择一个字形，那个一定是 CIDFont。对于 CID 如果没有字形存在，关于 CID0 (被要求呈现) 的字形会被取代。

对于编码来说，如果 CMap 不包含一个字符映射或一个 **notdef** 映射，descendant 0 被选择且 CID 0 字形从相关的 CIDFont 中被取代。

如果编码是无效的一也就是说，从字符串中提取的被展示字节不匹配 CMap 中的任何编码间距，CMap 是一个被选择用来描述的可替代字形。字符映射结构被重写它在字符串中的初始位置，而且一个改进的映射结构选择部分匹配最好的编码间距：

1. 如果从字符串中被提取出来被展示的第一个字节不匹配任何编码间距的第一个字节，该区间有最小编码被选中。
2. 否则 (也就是说，如果有一个局部的匹配)，对于每一个额外提取的字节，与所有更长编码间距直到最长编码间距 (就像已经被发现的局部匹配) 相比编码累积更长。

在选择的编码间距长度确定整个字节数，消耗的字节误码率从字符串映射到当前操作。

5.7、字体描述符号

一个字体描述符指定了一个简单字体或一个完整的 CIDFont 的度量及其他属性，和单个字形度量不同。当字体程序在难以获得的时候，这些字体度量提供信息可以确保一个用户程序合成一个可替代的字体或选择一个相似的字体。

字体描述以字体类型 0 的方式使用。从 PDF1.5 开始，在标记的 PDF 文档中字体描述器可以以字体类型 3 的方式使用（见 10.7 节，“标记的 PDF”）

一个字体描述符是一个字典，该字典条目指定了不同字体属性。对所有字体描述器来说这个条目是共同的——适用于简单字体和 CIDFonts——在表 5.19 中被列出。CIDFont 中字体描述器附加条目在 5.7.2 节“CIDFonts 字体描述器”中被描述。所有整数值是字形空间单位。从字形空间到文本空间转换在 5.1.3 节“字形位置和属性”中被描述。

表 5.19 所有字体描述条目

key	type	VALUE
Type	name	(必需) 该字典描述的 PDF 对象类型；对于一个字体描述来说一定是一个字体描述符号
FontName	name	(必需) 字体附言名。在字体或 CIDFont 字典中，名字应该和基础字体值一样引用该字体描述
FontFamily	string	(选修); PDF1.5; 强烈推荐字体类型 3 在标记的 PDF 文档中) 字符串指定优先字体家族名字。例如，字体加粗斜体，字体家族是乘以。
FontStretch	name	(选修); PDF1.5; 强烈推荐字体类型 3 在标记的 PDF 文档中) 字体张开值。它一定是下列名字中的一个 (顺序由窄变宽): 极端的, 压缩的, 浓缩的, 半缩的, 正常, 半扩大, 扩大, 极大 注: 这些值的解释不同于字体对字体。例如, 在一种字体中压缩可能出现与另一个正常最大的相似在

FontWeight	number	(选修); PDF1.5; 强烈推荐字体类型 3 在标记的 PDF 文档中) 重量 (厚度) 组成完全质量的字体名或字体说明。可能的值为 100, 200, 300, 400, 500, 600, 700, 800 或 900, 一个数组代表一个加权表明至少和它前面的一样黑。一个 400 的值表明一个正常值; 700 意味着加粗。 提示: 特别说明这些从字体到字体的值是不同的。比如一个字体中的 300 和另一个字体中的 500 可能相似。
Flags	integer	(必需) 字体的不同字形标记采集 (见 5.7.1 节, “字体描述标记”)
FontBBox	rectangle	(必需, 除了字体类型 3) 一个矩形 (见 3.8.4 节, “矩形”), 在字形坐标系统中被表达, 指明其字体边框。如果字体的所有字形在他们的初始处重合然后被填充, 那它将是该形状下的最小矩形
ItalicAngle	number	(必需) 角度, 由垂直到逆时针所呈的度数 (例如, 九点钟位置为 90 度, 三点钟的位置为 -90 度) 字体是负值当斜率朝右时, 也就是所有斜体字的方向。
Ascent	number	(必需, 除了字体类型 3) 字体字形基线上的最大高度为准, 不包括字符重读的字形高度。
Descent	number	(必需, 除了字体类型 3) 字体字形基线上的最大深度为准。该值是个负值。
Leading	number	(可选) 文本连续基线之间的间距。默认值为: 0
CapHeight	number	(字体需要拉丁字母, 除了字体类型 3) 大写字母顶部的垂直坐标, 从基线测量。
XHeight	number	(可选的) 字体 x 的高度: 平面上上升的小写字母的顶部的垂直坐标, 从基线开始测量, 在字体

		中有拉丁字母。默认值为：0
StemV	number	(必需, 除了字体类型 3) 厚度, 水平测量, 字体字型的显示垂直系统。
StemH	number	(可选) 厚度, 垂直测量, 字体字型的显性水平系统。默认值为：0
AvgWidth	number	(可选) 字体字型宽度的平均值, 默认值为：0
MaxWidth	number	(可选) 字体字型宽度的最大值, 默认值为：0
MissingWidth	number	(可选) 字符编码的宽度, 该宽度没有在字体字典的宽度数组中定义。有一个可预测的效果只有当所有的编码映射到字形, 该字形宽度和 MissingWidth 条目值一样。默认值为：0
FontFile	stream	(可选) 流对象包含字体类型 1 方案(见 5.8 节, “嵌入式字体程序”)
FontFile2	stream	(可选; PDF1.1) 流包含一个真型字体程序(5.8 节, “嵌入式字体程序”)
FontFile3	stream	(可选; PDF 格式 1.2) 流包含的字体方案由流字典中的图标类型项指定(见表 5.23 和执行情况的说明见附录 H 注释 67)。至多, 只有字体文件, 字体文件 2 和字体文件 3 中一项可能存在
CharSet	string	(可选, 仅在字体类型 1 有意义, PDF 格式 1.1) 字符串在字体集中列出的字符名。字符串的名称必须以 PDF 格式的语法, 也就是用斜线(前面每个名称/)。该名称可以出现在任何顺序。这个名字。.notdef 应该被省略, 它被假定为存在于字体的子集。如果此项不存在, 只有一种字体的子集, 指标是在字体名的赋值项子集标记(见第 5.5.3 节, “字体子集”)。

5.7.1、字体描述标记

字形描述符号标记项的值是一个无符号 32 位整数, 包含的标记指定字体中

的各种各样的特性。比特位在单词标记的之内从 1 (低一高) 到 32 (高一低)。

表 5.20 显示了标记的意思;所有未定义的标记位被保留而且必须被置零。图 5.13 展示了字体事例的这些特性。

表 5.20 字体标记		
BIT POSITION	NAME	MEANING
1	FixedPitch	所有字形有相同的宽度 (而不是按比例或可变节距的字体, 该字体有不同宽度)
2	Serif	字形有细体, 是字形从上到下角度的短笔触
3	Symbolic	字体包含字形在 Adobe 标准拉丁字符集外。这个标记和无符号标记不可以同时被设置和同时被清除。
4	Script	字形类似手写的草书
6	Nonsymbolic	字体用 Adobe 标准拉丁字符集设置或它的子集 (见下表)
7	Italic	字形有显示的垂直行程, 而且他们是倾斜的
17	AllCap	字体包含小写字母; 经常用来显示目的, 比如标题和头条
18	SmallCap	字体既包含大写的也包含小写的字母。大写字母和那些相同类型家族固定版本相似。小写字母和他们相应的大写字母有

		同样形状
19	ForceBold	见下表

无符号标记（标记项中的比特 6）表明字体字符集是 Adobe 标准拉丁字符集（或是它的一个子集）而且它使用这些字形的标准名字。这个字符集在 D.1 节“拉丁字符集设置和编码”被展示。如果字体包含这个设置之外的任何字形，符号标记应该被设置而且无符号标记应该被清除，换句话说，任何字符集不是 Adobe 标准拉丁字符集的字体被认为是符号的。这个影响字体的隐含基码而且可能影响一个用户方案的字体替换策略。

Fixed-pitch font	The quick brown fox jumped.
Serif font	The quick brown fox jumped.
Sans serif font	The quick brown fox jumped.
Symbolic font	*✿✿ □◆✿✿* ◎□□■✿□✿◆○□✿✿✿
Script font	<i>The quick brown fox jumped.</i>
Italic font	<i>The quick brown fox jumped.</i>
All-cap font	THE QUICK BROWN FOX JUMPED
Small-cap font	THE QUICK BROWN FOX JUMPED.

图 5.13 字体描述符号标记项的代表特性

注：无符号和符号字体分类是 PDF 中特有的。一个字体可能包含额外字符，该字符在拉丁文书写系统中使用但没包含进 Adobe 标准拉丁字符集；PDF 考虑将这类字体作为象征字符。用两个标记代表一个二进制选择是一个历史意外。

加强粗体标记（19 比特）决定是否粗体字形用额外像素绘制在很小的文本字节中。典型的，当字形被以小规格绘制在非常小的方案设备中，就像显示屏，字

形的加粗可能出现一个像素的宽。由于这是像素基本设备的最小特性宽度，普通字形同样以一个像素宽出现，因此不能够与粗体识别。如果加强粗体标记被设置，加粗字体的特征可能被加粗在很小的文本规格上。

例 5.12 阐述了一个字体描述符号，该符号标记的设置有衬线，无特征，和加强加粗标记（比特 2, 6, 19）

例 5.12

```

7 0 obj
<< /Type /FontDescriptor
/FontName /AGaramond-Semibold
/Flags 262178 % Bits 2, 6, and 19
/FontBBox [-177 -269 1123 866]
/MissingWidth 255
/StemV 105
/StemH 45
/CapHeight 660
/XHeight 394
/Ascent 720
/Descent -270
/Leading 83
/MaxWidth 1212
/AvgWidth 478
/ItalicAngle 0
>>
endobj

```

5.7.2、CIDFonts 字体描述

根据 426 页的表 5.19，CIDFonts 字体描述字典可能包含表 5.21 的列出项

表 5.21 CIDFonts 的额外字体描述项		
KEY	TYPE	VALUE
Style	dictionar	(可选的) 一个字典包含描述字体字形样式的条目（见 432 页的“样式”）
Lang	name	(可选) 一个名字指定字体语言，用来处理

		<p>语言不被代码本身默认的编码。可能的编码值被 RFC 3066 所定义，对于语言识别标签（见参考书目）。如果此项不存在，语言被认为是未知的。</p> <p>注：允许的语言代码规范被引入 PDF。以前的版本支持的子集：2 个字符的语言编码被 ISO 639 定义（见参考文献）。</p>
FD	dictionary	<p>（可选）一个字典键确定一个 CIDFont 的标志符号类。每个值是一个字典项包含主要字体描述字典的相应值（见 432 页“FD”）。</p>
CIDSet	stream	<p>（可选）流确定哪些 CIDs 是在 CIDFont 文件中。如果此项存在，CIDFont 只包含由 CIDSystemInfo 字典中定义的字符集的一个子集的标志符号。如果它不存在，一个子集 CIDFont 唯一指标是在 FONTNAME 项中的子集标记（见第 5.5.3 节，“字体子集”）。</p> <p>流的数据被组织成一个 CID 索引表。该位应存储的高序位第一字节。每个位对应一个来 CID。在第一个字节的最高位对应 CID0，下位来电显示 1，依此类推。</p>

风格

字典风格包含的条目有定义风格属性，CIDFont 值。一般的，只有 Panose 条目被定义。Panose 的值是一个 12 比特的字符串由下列元素组成：

- 字体家族类和 ID 子类字节在一个真型字体里考虑到了表“OS / 2”上的家庭类领域。此字段被记录在微软真型字体文件 1.0 技术规范里。

10 字节的字体的 PANOSE 分类。PANOSE 分类系统已记录在惠普公司的 PANOSE 分类指标指南中。

看这些参考的文档来了解更多信息。

下面是一个关于字体描述器的风格条目的例子

/Style << /Panose <01 05 02 02 03 00 00 00 00 00 00 00> >>

FD

一个 CIDFont 可能构造字形的不同类，每个类需要不同的字符集，这些字符集的字体宽度属性出现在字体描述符中。拉丁字形，例如与日本字形相比可能需要不同的属性。在 CIDFont 中，字体描述定义了适应于所有字形的一系列默认属性。FD 条目在字体描述中包含了这些默认的意外情况。

在 FD 字典中的每个辞条的关键是一个字形，也就是一对 CIDFont 的字符集的特定子集类的名称。本项的值是一个字体描述，其内容是覆盖该类的字体全属性只。此字体描述应包含度量信息条目只，它不应该包括字体文件，FontFile2，FontFile3，或在表 5.21 所列的任何项目。

表 5.22 在 CJK 字体中的字形类

CHARACTER COLLECTION	CLASS	GLYPHS IN CLASS
Adobe-GB1	Alphabetic	全角拉丁文，希腊语，和斯拉夫字母字形
	Dingbats	特别符号
	Generic	字体无关字形，就像线图
	Hanzi	全宽汉字（中文）字形
	HRoman	半角拉丁文字形
	HRomanRot	和古罗马相同但使用垂直螺旋书写方式
	Kana	日语假名（片假名和平假名）字形
	Proportional	成比例的拉丁文字形
	ProportionalRot	比例相同，但使用旋转垂直的书写方式
Adobe-CNS1	Alphabetic	全角拉丁文，希腊语，和斯拉夫字母字形

Dingbats	特别符号
Generic	字体无关字形，就像线图
Hanzi	全宽汉字（中文）字形
HRoman	半角拉丁文字形
HRomanRot	和古罗马相同但使用垂直螺旋书写方式
Kana	日语假名（片假名和平假名）字形
Proportional	成比例的拉丁文字形
ProportionalRot	比例相同，但使用旋转垂直的书写方式

	Proportional	成比例的拉丁字形
	ProportionalRot	比例相同，但使用旋转垂直的书写方式
	Ruby	用来设置 ruby 的字形（小的字形用含义和发音来注解其他字形）
Adobe-Japan2	Alphabetic	全角拉丁文，希腊语，和斯拉夫字母字形
	Dingbats	特殊符号
	HojoKanji	全宽日本汉字字形
Adobe-Korea1	Alphabetic	全角拉丁文，希腊语，和斯拉夫字母字形
	Dingbats	特殊符号
	Generic	字体无关字形，就像线图
	Hangul	韩国和 jamo 字形
	Hanja	全宽朝鲜文（汉语）字形
	HRoman	半角拉丁文字形
	HRomanRot	和古罗马相同，但使用旋转垂直的书写方式
	Kana	日语假名（片假名和平假名）字形
	Proportional	比例的拉丁字形
	ProportionalRot	比例相同，但使用旋转垂直的书写方式

例 5.13 阐明一个 FD 字典包含两项

例 5.13

```
/FD << /Proportional 25 0 R
      /HKana 26 0 R
    >>

25 0 obj
<< /Type /FontDescriptor
  /FontName /HeiseiMin-W3-Proportional
  /Flags 2
  /AvgWidth 478
  /MaxWidth 1212
  /MissingWidth 250
  /StemV 105
  /StemH 45
  /CapHeight 660
  /XHeight 394
  /Ascent 720
  /Descent -270
  /Leading 83
>>
endobj

26 0 obj
<< /Type /FontDescriptor
  /FontName /HeiseiMin-W3-HKana
  /Flags 3
  /AvgWidth 500
  /MaxWidth 500
  /MissingWidth 500
  /StemV 50
  /StemH 75
  /Ascent 720
  /Descent 0
  /Leading 83
>>
endobj
```

5.8、字体嵌入程序

字体程序可以嵌入到 PDF 文件的 PDF 流对象中的数据。这些流对象也被称作 **font file** 类似可从外部对消费软件使用。(详情见附录 H 的注 68)。

字体程序如果有版权，则版权所有者可根据自身的要求增加一些使用该程序

的条件。这些权限记录在字体程序或是许可证中的一部分。条件之一可能是在不纳入一个 PDF 文件时该字体程序不被嵌入。一个字体程序可能允许被嵌入后用于查看和打印文档，但不一定是为了创建新的或是修改使用该字体的文本（不管是同一个还是不同的文档）。后者操作需要用户执行操作此字体程序的许可副本，而不是重复摘自 PDF 文件。在缺乏详细信息时，一个 PDF 使用者可能应该假设所有的嵌入字体程序都只使用在浏览和打印文档以及没有除此以外的其他目的用途。

表 5.23 根据对各个字体程序的描述，总结了在 PDF 文件中嵌入字体程序的方法。关键字是使用在字体中用于指定字体文件流使用的名称；如果在字体文件流字符集中存在，则其子类型是 Subtype 关键字的子类型。更详细的字体方案陈述如下。

表 5.23 不同类型的嵌入式字体数据结构

关键字	子类型	描述
FontFi	—	格式 1 字体程序，这是在 Adobe 格式 1 字体格式中描述的最初（非紧密形）的格式。此项可以出现在为 Type1 或是 MMType1 字体字符集中的哦字体描述中。
FontFi	Type1C	(PDF 1.1) 字体程序的真实格式，正如在 TrueType Reference Manual 中描述的那样。此项出现在描述 TrueType 字体字符集或是（在 PDF1.3）用于 CIDFont 字符集的 CIDFontType2。
FontFi	Type1C	(PDF 1.2) 类型字体方案是 Compact Font Format (CFF) 中代表格式 1，正如在 Adobe Technical Note #5176 中描述的，这个 Compact Font Format Specification。此项会出现在 Type1 或是 MMType1 字体字符集中的字体描述中。
CIDFont	Type0C	(PDF 1.3) 正如在 Adobe Technical Note #5176 中描述的 Type 0 CIDFont 程序作为 Compact Font Format (CFF) 的代表，Compact Font Format Specification。此项能偶出现在 CIDFont 字符集中的 CIDFontType0 的字体描述中。
OpenTyp		(PDF 1.6) OpenType 字体方案，正如 OpenType 字体规范

e (见参考) 描述的。OpenType 是一个 TrueType 的扩展类型，允许包含使用 Compact Font Format (CFF) 的字体程序。

此项为以下两种字体字符集中的格式描述的字体：

- 如果嵌入的字体程序包含一个“glyf”表，则为 TrueType 字体字符集或是 CIDFontType2 的 CIDFont 字符集。
 - 如果嵌入字体程序包含 Top DICT 操作（这等价于上面 CIDFontType0C 属性值）“CFF”表。
 - Type1 字体字符集或是 CIDFontType0 的 CIDFont 字符集，如果嵌入字体程序包含“CFF”表，并且脱离 CIDFont 操作。
-

一个字体文件的流字符集包含一个流的正常项，如 Length 和 Filter（在 38 页的表 3.4），加上表 5.24 中列出的额外项。

表 5.24 嵌入字体流字符集的额外项

关键字	格式	值
Length	integer	(对于 Type 1 和 TrueType 字体为必选) 如果有的话，在被流的 Filter 项指定的过滤器解码后，Type 1 字体程序（见下文）的明文在字节长度，或是整体 TrueType 字体程序。
Length	integer	(对于 Type 1 字体是必选) 在被使用流中的 Filter 项指定的过滤器解码后，对 Type 1 字体程序（见下文）进行部分加密时的字节长度。
Length	integer	(对于 Type 1 字体为必选) 对 Type 1 字体程序（见下文）的固定部分的字节长度在被流的 Filter 项使用指定的过滤器解码后。如果此项值为 0，它只是了 512 个 0 和 cleartomark 没有被包含在 FontFile 字体程序中时，必须对其进行添加。
Subtyp	name	(如果参考 FontFile3 则为必选的；PDF 1.2) 指定嵌入字体程序的一个名称格式。这个名称必须是 Type1 字体的 Type 1C 格式，Type0 字体的 CIDFontType0C 格式，或者 OpenType
e		

字体的 OpenType 格式。当额外的字体格式被添加到 PDF,
Subtype 将被定义成更多的值。

Metadata stream (可选; PDF 1.4) 元数据包含嵌入字体程序数据 (见 10.2.2 节, “元数据流”)。

这是一个标准的 Type 1 字体程序, 正如在 Adobe Type 1 Font Format 规范中描述的, 共有三部分: 明文部分 (使用 PostScript 语法写), 加密部分, 固定部分。其中固定部分包含 512 ASCII 0 在 cleartomark 操作, 并且可能在额外数据后面。虽然加密部分可能是使用二进制或是 ASCII 十六进制格式, PDF 只支持二进制格式。然而, 所有的字体程序可使用任何的过滤器进行编码。

例 5.14 显示了一个嵌入 Type 1 字体标准的结构

例 5.14

```
12 0 obj
<< /Filter /ASCII85Decode
/Length 41116
/Length1 2526
/Length2 32393
/Length3 570
>>
stream
, p>`rDKJj'E+LaU0eP. @+AH9dB0u$hFD55nC
...Omitted data...
JJQ&Nt')<=^p&mGf (%: %h1%9c//K /*o=. C>UXkbVGTr~>
endstream
endobj
```

正如在表 5.23 中显示的, 一个 Type 1—相当于字体程序或是一个 Type 0 CIDFont 程序能够代表 Compact Font Format(CFF)。Length1, Length2 和 Length3

项都不能满足这个要求。虽然 CFF 初始化多媒体字体或是 CID 字体程序，使其在一个单一的文件中进行组合操作，一个 PDF 中嵌入的 CFF 字体文件准确来说要由字体或是 CIDFont (作为适当的相关字符集) 组成。

注：根据 Adobe Type 1 Font Format 规范，Type 1 font 程序可能包含 PaintType 项指定外形是否被填充或画。对于嵌入在 PDF 文件中的字体，此项被忽略；决定是否进行填充或画是跟据 PDF 内容播放模式参数（见 5.2.5 节“内容播放模式”）。这适用于 Type 1 型字体和 Type 0 型 CIDFonts。

注：其他表格，就像那些用来为高级路线布局，而不需要被呈现，然而它们的缺席可能会阻止含有字体的文本编辑。

在可缩放字形中发现字形描述的过程有：

- 对于字体类型 1 使用表“CFF”，这个过程在 398 页“字体类型 1 的编码”中被描述。
- 对于真型字体使用表“glyf”，这个过程在 399 页的“真型字体编码”中被描述。由于这个过程有时产生模棱两可的结果，强烈建议 PDF 格式的创造者，不要使用简单的字体，使用一个 H 编码标识的字体类型 0 和使用字形目录作为字符编码，如 412 页表 5.15 所描述。
- 对于 CIDFontType0 字体使用表“CFF”，这个过程由 408 页“CIDFonts 中的字体选择”中被描述，。
- 对于 CIDFontType2 字体使用表“glyf”，这个过程在 408 页“CIDFonts 字型选择”中被描述。

5.5.3 节中所讨论，“字体子集”，嵌入字体程序可能只包含了那些在 PDF 文档中使用的符号集。这可能由字体描述集中的 CharSet 或 CIDSet 项表示。

5.9、文本对象提取

前面的章节描述为了显示文本，和在页面上绘制字形的所有方法。除了显示文本，用户的应用程序有时需要确定文本内容信息，也就是说，它意味着根据一些标准字符识别，和它渲染的外观截然相反。这在操作中需要时出现，如搜索，

索引和文本输出到其他应用程序。

万国码标准定义一个系统，该系统用来给在语言中大量的常用的所有字符编号。它是一种与代表文本内容信息相适应的计划，但不是它的外观，因为万国码值识别字符，而不是字形。有关万国码信息，请参阅由万国码财团编写的万国码标准（见参考文献）。

当提取字符内容时，消费者应用可能容易地转换对编码值的文本，如果字体的字符根据标准被辨认为应用所知的字符集。这字符证明可以进行，如果任一字体使用的是标准编码或由标准字符名字或 CIDs 确定的字体中的字符。5.9.1 段，“映射文字符号代码得到编码值”，详细描述了文字符号代码到编码值的全局算法。

如果字体没有在这些方式之一中被定义，符号仍然可以显示，但是没有附加信息的话，这些字符不能被转换成编码值。

这些信息在字体字典中可以被提供作为备选的编码值输入。它的值是一个流对象，包含一种特殊的 Cmap 文件可以映射文字符号代码得到编码值。

结构元素或标记内容序列的一个实际文本输入可以被用于直接地指定文本内容。

5.9.1、映射文字符号代码得到编码值

消费者应用可以优先使用以下方法映射字符码到编码值，特别是被标记的 PDF 文件，必须至少提供以下方法之中的一种。

- 如果字体字典包含 ToUnicode Cmap，使用 CMap 转换字符码成编码。
- 如果字体是使用一被预定义的内码的一种简单的字体 **MacRomanEncoding**、**MacExpertEncoding** 或者 **WinAnsiEncoding**，或者它拥有编码值，差异数列只包括取自拉丁字符集和符号字体中的字符名（参见附录 D）：
 - 1、根据 924 页表 D.1 和字符的差异数列映射字符编码成字符名。
 - 2、根据 Adobe 字形列表查询字符名得到相应的编码值。

如果字体是一个复合字体，使用的列在一个预定义 CMaps 中，见 412 页表 5.15，其继承者使用 Adobe-gb1, Adobe-CNS1, Adobe-Japan1 或者 Adobe-Korea1 字符集。

1 字符代码根据字体的 CMap 映射到一个字符的标识符 (CID)。

2 从 CIDSystemInfo 字典中获取注册表和所使用的字符集序字体的 CMap。

3 通过连接注册表和第二步骤获取的格式订购顺序 UCS2 (比如 Adobe-Japan-UCS2) 兴建第二个 CMap

4 在第 3 步中获得 CMap (详情可查阅 ASN 网站，见参考书目)。

5 在步骤 1 中通过步骤 4 的 CMap 获得 CID 的位置，产生一个 Unicode 值。

注意：键入 0，其继承者 CIDFonts 使用的 Adobe-GB1, Adobe-CNS1, Adobe-Japan1, 或者 Adobe-Korea1 字符集 (如在指定的 CIDSystemInfo 字典) 必须满足相应的 PDF 版本。如 416 页表 5.16 的列表字符集合对应于一个特定的 PDF 版本。(其他的对字节进行补充。但是如果该补充是一个超出了相应的 PDF 版本，只有后面的才被视作标准的 CIDs。)

如果此方法创建统一码失败，将无法决定字节编码所代表的对象。

5.9.2、ToUnicode CMaps

字体字典中要转化的统一码定义的 CMap 必须遵从 5.6.4 节中 CMaps 句型，“CMaps”和 Adobe CMap Adobe 和 CIDFont 文件规格中的#5014 的全部文档。关于 CMap 的更多的指导请查阅 Adobe 提供的技术说明#5411。要万国码辅助映射文件

• 唯一的 CMap 流字典 (419 页表 5.17) 的相关项是 UseCMap。该相关项可以被使用如果 CMap 是基于另一个 ToUnicode CMaps。

CMap 的文件必须包含 `begin codespace range` 和 `end codespace range` 操作，这样与字体使用编码一致。特别是，对于一个简单字体，`codespace` 必须是 1 个字节长。

- 必须使用 `beginbfchar`, `endbfchar`, `beginbfrange` 和 `endbfrange` 操作来定义从字符编码到 UTF – 16BE 编码中表达的万国码序列的映射。

例 5.15 阐明一个字体类型 0

示例 5.15 演示了一个类型 0 的字体，使用 H 标识的 CMap 的映射从字符代码，其后裔 CIDFont CIDs 使用从 CIDs 身份映射到 TrueType 字形指数。文本字符串显示使用此字体为每一个字形简单地使用一个 2 字节的标志符号索引。在一个 ToUnicode 入境的情况下，不提供有关资料将意味着什么字形。

例 5.15

```
14 0 obj
<< /Type /Font
/Subtype /Type0
/BaseFont /Ryumin-Light
/Encoding /Identity-H
/DescendantFonts [15 0 R]
/ToUnicode 16 0 R
>>
endobj

15 0 obj
<< /Type /Font
/Subtype /CIDFontType2
/BaseFont /Ryumin-Light
/CIDSystemInfo 17 0 R
/FontDescriptor 18 0 R
/CIDToGIDMap /Identity
>>
endobj
```

ToUnicode 项的值是一个流对象，包含了 CMap 的定义，在例 5.16 中被显示

例 5.16

```
16 0 obj
<< /Length 433 >>
stream
/CIDInit /ProcSet findresource begin
12 dict begin
beginicmap
/CIDSystemInfo
<< /Registry (Adobe)
/Ordering (UCS)
/Supplement 0
>> def
/CMapName /Adobe-Identity-UCS def
/CMapType 2 def
1 begincodespacerange
<0000> <FFFF>
endcodespacerange
2 beginbfrange
<0000> <005E> <0020>
<005F> <0061> [<00660066> <00660069> <00660066006C>]
endbfrange
1 beginbfchar
<3A51> <D840DC3E>
endbfchar
endcmap
CMapName currentdict /CMap defineresource pop
end
end
endstream
endobj
```

例 5.16 中的 `begincodespacerange` 和 `endcodespacerange` 操作定义了源字符编码从`<00 00>` 到 `<FF FF>`幅度为两个字节的字符编码。具体映射的几个字符编码被展示。例如 `<00 00>` 到`<00 5E>`被映射为标码值 U+0020 到 U+007E（标码值被写为 U+后面跟着四到六个十六进制数）这是一个映射的定义，其中每个字符编码表示一个以上的标码值：

```
<005F> <0061> [<00660066> <00660069> <00660066006C>]
```

在这种情况下，原始字符编码是 `ff`, `fi`, 和 `ff1` 的字形目录。条目定义了从字符编码`<00 5F>` `<00 60>`, 和 `<00 61>`到标码值字符串，该标码矢量值对于在 U+0066 U+0066 中的每个字符是万国码字符串，字符序列 `ff`, U+0066 U+0069 为 `fi`, 和 U+0066 U+0066 U+006c 为 `ff1`

最后，字符编码<3A 51>被映射到值为 U+2003E 的标码，由在编码 UTF-16BE 中的字节序列<D840DC3E>表达。

例 5.16 说明了几个延伸到终点的值可以被定义。为了支持从源代码映射到一个目的地的代码字符串，在 `beginbfchar` 操作之后下面的延伸定义了范围：

```
n beginbfchar
srcCode dstString
endbfchar
```

在这数据字符串可以成为 512 比特的字符串，同样的在 `beginbfrange` 操作之后的映射可被定义为

```
n beginbfrange
srcCode1 srcCode2 dstString
endbfrange
```

在这种情况下，该字符串的最后一个字节是递增的对于每个源代码范围内的连续代码。当这种类型界定范围时，必须小心以确保在字符串中的最后一个字节的值小于或等于 $255 - (\text{srcCode}_2 - \text{srcCode}_1)$ 。这可以确保字符串的最后一个字节没有增至 255；否则，映射的结果未定义同时发生一个错误。

为了支持更紧凑的表现，从源字符代码范围映射到一个目标码不连续范围，CMaps 用于 ToUnicode 项可能用到以下语法。`beginbfrange` 定义：

```
n beginbfrange
srcCode1 srcCoden [dstString1 dstString2 ... dstStringn]
endbfrange
```

连贯的编码以 srcCode_1 开始和 srcCode_n 为结束被映射到以 srcCode_1 开始和 srcCode_n 终点字符串数组。

第 6 章 渲染

Adobe 图像模型将图形（形状和色彩的规范）渲染（控制渲染输出设备）分开。208 页和 209 页的图 4.12 和 4.13 显示了这种分离。第 4 章描述了以独立于设备的方式指定页面外观的方法。本章描述控制形状和色彩如何在光栅输出设备上渲染的方法。此处讨论的所有方法依赖于特定的输出设备特性。欲独立于设备的 PDF 文档应该将其自身限制到第 4 章中叙述的通用图形功能中。

几乎所有在 PDF 文档控制下的渲染功能都涉及色彩再现。色彩经过下面的多个概要步骤处理来渲染。（依赖于当前色彩空间和设备特性，有些步骤并不总是需要的。）

1. 如果色彩已由一个基于 CIE 的色彩空间所指定（参见 4.5.4 节，“基于 CIE 的色彩空间”），其首先必须被转换为光栅输出设备的本地色彩空间（也称为它的印刷色模型）。
2. 如果一种色彩已由一个对输出设备来说不合适的设备色彩空间所指定（例如：具有 CMYK 的 RGB 色彩或灰度设备），将会调用色彩转换函数。
3. 现在，设备色彩值已通过传递函数得到映射，每个值对应于一个色彩组件。此转换函数对输出设备的特质进行补偿，例如非线性灰度相应。这一步有时被称为伽马校正。
4. 如果设备不能再现连续色调，而仅能再现某些离散的色彩例如黑像素和白像素，则半调函数将被调用，它将通过像素模拟逼近所要的颜色。
5. 最后，执行扫描转换以要求的颜色标记光栅输出设备上适当的像素点。

一旦对页面上所有图形对象执行了这些操作，将使用光栅数据结果来标记物理输出媒体，例如一个显示器上的像素或一个打印页上的墨点。一个 PDF 文档对产生输出的物理媒体属性的规定很少；这些信息可以从下列源头获得：

- 媒体盒与页面字典中的一些其他入口（参见 10.10.1 节，“页面边界”）。
- 当用户请求查看或打印时的一个交互对话框。
- 一个嵌入在 PDF 文件内或单独提供的工作单，指定了向媒体推送 PDF 页面和控制输出设备特殊特性的详细指令。工作单的格式存在多种标准。其中两个：JDF（工作定义格式）和 PJTF（便携工作单格式），分别在 CIP4 文档 JDF 规范和 Adobe 技术

笔记#5620 的便携工作单格式（参见文献）中有描述。

本章中描述的有些渲染方法是由第 182 页表 4.3 中所列的依赖于设备的图形状态参数所控制的。这些参数可以通过使用第 190 页表 4.8 中所示的一个包含入口的参数字典调用 **gs** 运算符来改变。

6.1、基于 CIE 的色彩到设备色彩的转换

为了在一个输出设备上渲染基于 CIE 的色彩，考虑到已知的设备属性，用户应用程序必须从特定的基于 CIE 的色彩空间转换到设备的本地色彩空间（一般是 **DeviceGray**, **DeviceRGB**, 或 **DeviceCMYK**）。正如 4.5.4 节讨论的，“基于 CIE 的色彩空间”，基于 CIE 的色彩是基于一个人类色彩感知模型的。基于 CIE 的色彩渲染的目标是以能准确重现请求的基于 CIE 的色彩值的设备本地色彩空间产生一个人类观察者所见的输出。基于 CIE 的色彩规范和渲染是 PDF 1.1 版 (**CalGray**, **CalRGB**, 和 **Lab**) 和 PDF 1.3 版的一个特性。

这种从基于 CIE 的色彩到设备色彩的转换时复杂的，并且它机遇的理论超出了本书的范围；参见参考文献以获得更进一步的信息。算法有许多参数，包括一个可选的、完全 3 维的色彩查找表。输出的色彩逼真程度取决于这些参数的正确设置，通常包含某周形式上的校正。设备能产生的输出由一个设备概貌所字符化，其通常是由一个与设备关联的 ICC 概貌所指定（并且从一个 **ICCBased** 色彩空间概貌整体分离）。

注意：PDF 没有等价的 *PostScript* 色彩渲染字典。使一个设备概貌与一个用户应用程序的输出设备相关联的方法是不依赖实现的且不能在 PDF 文件中被指定。典型地，这是通过一种由操作系统提供的色彩管理系统 (*CMS*) 来达成的。从 PDF 1.4 版开始，一个 PDF 文档也可以指定一个或多个输出意向，假如有可能被使用的概貌被用来处理文档（参见 10.10.4 节，“输出意向”）。

从基于 CIE 的色彩值转换到设备色彩值，需要两个主要操作：

1. 根据基于 CIE 的色域映射函数调整基于 CIE 的色彩值。色域是指一个在一些色彩空间中所有可能颜色的子集。一个页面描述拥有一个由所有其使用的颜色组成的色域源。一个输出设备有一个由其能重现的所有颜色所组成的色域设备。这一步以企图保留颜色外观、视觉对比或一些其他明确定义的渲染意向（参见第 230 页的“渲染意向”）的方式将颜色从色域源转换到色域设备。

2. 根据一个基于 CIE 的色彩映射函数产生一个相应的设备色彩值。对一个给定的基于 CIE 的色彩值，此函数计算其在设备本地色彩空间的颜色值。

基于 CIE 的色域和色彩映射函数被仅应用于由一个基于 CIE 的色彩空间所呈现的色彩值上。根据定义，设备色彩空间中的色彩值直接控制设备色彩组件（尽管其可以被 **DefaultGray**, **DefaultRGB**, 或 **DefaultCMYK** 色彩空间资源所更改；参见第 227 页“默认色彩空间”）。

色域源由页面描述所指定，当其选择一个基于 CIE 的色彩空间。本规范是设备独立的。输出设备相应的属性在与设备相关联的设备概貌中给出。色域映射和颜色映射函数是用户应用程序实现中的一部分。

6.2、设备色彩空间中的转换

每一个光栅输出设备有一个本地色彩空间，典型地，它是标准设备色彩空间（**DeviceGray**, **DeviceRGB**, 或 **DeviceCMYK**）之一。换句话说，大多数设备支持根据灰度级（单色）、RGB（红-绿-蓝）或 CMYK（青-品红-黄-黑）模型的色彩重现。如果设备支持连续色调输出，重现将直接发生。否则，它将有半调来完成。

一个设备的本地色彩空间也被称为它的印刷色模型。印刷色由一个或多个标准印刷颜色的组合来产生。在任何设备或基于 CIE 的色彩空间中指定的颜色被作为印刷色渲染。（设备也可以支持额外的专色，其仅可由 **Separation** 或 **DeviceN** 色彩空间进行着色。他们不包括在设备或基于 CIE 的色彩空间中，也不服从下列所述的转换。）

注意：有些设备提供一个非上面所述 3 种的本地色彩空间，其由不同色剂的组合组成。这种情况下从标准设备色彩空间到设备本地色彩空间的转换由依赖于设备的方法执行。

知道了设备的本地色彩空间和其他输出能力，用户应用程序就可以自动将一个文档中指定的色彩值转换成适合设备本地色彩空间的色彩值。例如，如果一个文档指定了 **DeviceRGB** 色彩空间中的色彩，但设备支持灰度级（例如单色显示器）或 **CMYK**（例如一个彩色印刷机），用户应用程序会执行必要的转换。如果文档指定的色彩就在设备的本地色彩空间中，就不需要执行转换。

用来进行设备色彩空间之间转换的算法很简单。从以人类观察角度感受，这样的转换仅能对原始颜色产生简陋的近似。对色彩转换更细致的控制能够通过基于 CIE 的色彩规范和渲染达到。此外，设备色彩空间可以被映射到基于 CIE 的色彩空间中（参见第 227 页“默认色

彩空间”）。

6.2.1、DeviceGray 和 DeviceRGB 间的转换

黑、白和中间的灰色阴影可以作为 *RGB* 色彩的特殊情况考虑。灰度值由一个数字描述：0.0 对应于黑、1.0 对应于白，中间值对应于不同的灰度级。

灰度级等价于红绿蓝相等的 *RGB* 值。换句话说，等价于一个特定的灰度值的 *RGB* 色彩值简单地等于：

$$\text{红} = \text{灰}$$

$$\text{绿} = \text{灰}$$

$$\text{蓝} = \text{灰}$$

对于一个给定的 *RGB* 值，其灰度值是根据 NTSC 视频标准来计算的，NTSC 视频标准决定了彩色电视信号是如何在一个黑白电视机上渲染的：

$$\text{灰} = 0.3 \times \text{红} + 0.59 \times \text{绿} + 0.11 \times \text{蓝}$$

6.2.2、DeviceGray 和 DeviceCMYK 间的转换

名义上，灰度级是 CMYK 中黑色组成部分的补充。因此，等价于特定的灰度级的 CMYK 色彩值简单地等于：

$$\text{青} = 0.0$$

$$\text{品红} = 0.0$$

$$\text{黄} = 0.0$$

$$\text{黑} = 1.0 - \text{灰}$$

为获得对于给定的 CMYK 值的等价灰度级，所有组成成分都必须被考虑到：

$$\text{灰} = 1.0 - \min(1.0, 0.3 \times \text{青} + 0.59 \times \text{品红} + 0.11 \times \text{黄} + \text{黑})$$

黑色成分与其他 3 中成分的交互在下面详述。

6.2.3、从 DeviceRGB 到 DeviceCMYK 的转换

从 RGB 到 CMYK 的转换分为两个过程。第一步是将红-绿-蓝值转换成等价的青、品红和黄的组成。第二部是产生黑色组成并变更其他组成来产生对原始色彩更好的近似。

基本的减色，青、品红和黄是加色红、绿、蓝的补充。例如，青色油墨减去白光中的红色成分。理论上，这个转换很简单：

$$\text{青} = 1.0 - \text{红}$$

$$\text{品红} = 1.0 - \text{绿}$$

$$\text{黄} = 1.0 - \text{蓝}$$

例如，由 0.2 红、0.7 绿和 0.4 蓝的颜色也可以由 $1.0 - 0.2 = 0.8$ 青、 $1.0 - 0.7 = 0.3$ 品红和 $1.0 - 0.4 = 0.6$ 黄来表达。

逻辑上，仅有青、品红和黄需要产生一个印刷色。相同水平的青、品红、黄应该产生相同水平的黑。然而，实践中彩色油墨并不能做到完美混合；这中组合常常形成深棕色阴影而不是真正意义上的黑色。为在印刷机上获得更真实的色彩渲染，纯黑油墨经常被颜色的混合黑色部分所替代。大多数彩色印刷机支持黑色成分（CMYK 组成中的 K）。计算此成分的数量需要一些附加步骤：

1. 黑色生成——计算当试图重现一种特定色彩时黑色的用量。
2. 底色消除——减少青、品红和黄色成分的量来补偿由黑色生成步骤中得出的黑色用量。

完整的从 RGB 到 CMYK 的转换如下：BG (k) 和 UCR (k) 分别是对黑色生成和底色消除函数的调用：

$$c = 1.0 - \text{红}$$

$$m = 1.0 - \text{绿}$$

$$y = 1.0 - \text{蓝}$$

$$k = \min(c, m, y)$$

$$\text{青} = \min(1.0, \max(0.0, c - \text{UCR}(k)))$$

品红 = $\min(1.0, \max(0.0, m - UCR(k)))$

黄 = $\min(1.0, \max(0.0, y - UCR(k)))$

黑 = $\min(1.0, \max(0.0, BG(k)))$

在 PDF 1.2 版中，黑色生成和底色消除函数是被作为 PDF 图形状态中的参数函数字典来定义的。他们被指定为图形状态参数字典（参见第 190 页表 4.8）中 BG 和 UCR（或 BG2 和 UCR2）入口的值。每个函数都用一个单一数值操作数调用，且将会返回一个单一数值结果。

黑色生成与底色消除函数的输入都是 k , c 、 m 和 y 值中的最小值分别由 1.0 减去他们原始的红、绿、蓝成分。名义上， k 是黑色量，其可以从青、品红和黄成分中去除和被独立黑色成取代。

黑色生成函数计算的黑色成分，这是名义上 k 值的功能。它简单返回其未改变的 k 操作数，或者对于更深的黑色返回一个更大的值，更浅的黑色返回一个更小的值，或者对于完全没有黑色返回 0.0。

底色消除函数计算从每个 c 、 m 和 y 值中需要减去的量，以产生最终的青、品红和黄色成分。它可以简单地返回未改变的 k 操作数，或者可以返回 0.0（因此没有颜色被消除）、少数黑色量，或者甚至是一个负的量，由此添加到颜色总量中。

应用黑色生成和底色消除后得到的最终成分值将会在 0.0 到 1.0 间。如果一个值在此范围之外，则自动用最近的有效值来替代，而不提示出错。这种替代显式地由上面公式中的 \min 和 \max 操作给出。

黑色生成和底色消除函数的正确选择依赖于输出设备的特性——例如，油墨是如何混合的。每个设备都由适合于该设备的默认值进行配置。

参见第 543 页 7.6.4 节“渲染参数和透明”，并且特别地，“渲染意向和色彩变换”，以获得更进一步关于黑色生成和底色消除在透明模型中的讨论。

6.2.4、从 DeviceCMYK 到 DeviceRGB 的转换

从 CMYK 到 RGB 的颜色值转换是一个简单的操作，不需调用黑色生成和底色消除：

红 = $1.0 - \min(1.0, 青 + 黑)$

绿 = $1.0 - \min(1.0, 品红 + 黑)$

$$\text{蓝} = 1.0 - \min(1.0, \text{黄} + \text{黑})$$

换句话说，黑色成分被简单地添加到每个其他成分中，然后通过对其每个都用 1.0 减去其本身来转换成他们的补色。

6.3、传递函数

在 PDF 1.2 中，传递函数调整颜色成分的值来补偿输出设备和人眼的非线性响应。设备色彩空间的每种成分——例如：**DeviceRGB** 空间的红色成分——用来呈现与成分数值相称的光感或色彩成分的强烈程度。然而，许多设备实际上并不具有如此行为；传递函数的目的就是补偿设备的实际行为。此操作有时被称为伽马校正（不要与作为基于 CIE 色彩渲染一部分执行的基于 *CIE* 的色域映射函数相混淆）。

在处理颜色的一系列步骤中，如果需要的话，用户应用程序应在执行任何必要的色彩空间之间的转换后，但在应用半调函数前，应用此传递函数。每个颜色成分都有其自有的独立传递函数；在各成分间是没有交互的。

传递函数总是操作输出设备的本地色彩空间，无论原始色彩空间是如何指定的。（例如，对一个 CMYK 设备，传递函数应用于设备的青、品红、黄和黑色成分，即使色彩原始定义在，例如，一个 **DeviceRGB** 或 **CaIRGB** 色彩空间中。）传递函数被一个范围在 0.0 到 1.0 间的数值操作数调用，且必须返回一个相同范围的数字。函数的输入是一个在此设备本地色彩空间中的颜色成分值，其或是直接指定的或由从其他色彩空间转换而来。函数的输出是经过转换的要传送到设备（在经过半调处理后，如果需要的话）的成分的值。

传递函数输入和输出都总是按如同相应的颜色成分是加色的来转译（红、绿、蓝或者灰）：数字值越大，色彩越淡。如果成分是减色的（青、品红、黄、黑，或者专色），它将被 1.0 减去，转换为加色后它才被传递给传递函数。此函数的输出总是加色形式，且以此形式被传递到半调函数。

在 PDF 1.2 版中传递函数被定义为 PDF 函数对象（参见 3.9 节“函数”）。有两种方式来指定传递函数：

- 图形状态中的通用传递函数的参数由一个单一传递函数或一个分别对应红绿蓝和灰或其补色青、品红、黄和黑的四个独立传递函数组成的阵列组成。（如果仅指定了一个函数，它将用于所有成分。）RGB 设备使用前三个，而单色设备仅使用灰传

递函数，CMYK 设备四个都使用。通用传递函数可以被指定为一个图形状态参数中 TR 或 TR2 入口的值；参见第 190 页表 4.8。

- 图形状态中的通用半调参数可以指定传递函数作为半调字典（参见 6.4.4 节“半调字典”）中的可选入口。这是唯一为非主要颜色成分或为本地色彩空间使用非上面所列成分的设备中的任何成分设置传递函数的方法。在半调字典中指定的传递函数覆盖在由图形状态中的通用传递函数参数指定的对应传递函数。

除了伽马校正这一用途，传递函数可以被用来产生多种特殊的、依赖于设备的效果。例如，在一个单色设备上，PostScript 计算函数：

```
{1 exch sub}
```

翻转输出的颜色，产生页面的负呈现。一般而言，此方法对于彩色设备不起作用；这种翻转可以更复杂一些，而不仅仅翻转每种成分。因为传递函数产生依赖与设备的效果，打算独立于设备的页面描述不应该变更他们。

注意：若当前色彩空间是 *DeviceGray* 且输出设备的本地色彩空间是 *DeviceCMYK*，转换者仅使用会传递函数。从 *DeviceGray* 到 *DeviceCMYK* 的正常转换为青、品红和黄色成分产生 0.0。这些成分没有被传递到他们各自的传递函数，而是被直接渲染，生成的输出包含无色油墨。这一特殊情况是为了保持与现有使用传递函数来在单色设备上获得特殊效果的应用程序的兼容性而存在的，且仅应用在 *DeviceGray* 色彩空间中指定的颜色。

参见第 542 页 7.6.4 节，“渲染参数和透明”与“半调和传递函数”，以获得关于传递函数在透明图像模型中更进一步的讨论。

6.4、半调

半调是一个过程，在一个仅能实现有限数量离散颜色的输出设备上近似得到连续色调。通过使用可用颜色的像素模型来仿真设备所不能直接生成的颜色。可能最熟悉的例子是在报纸的图片中使用黑和白像素渲染灰色调。

有些输出设备能直接产生连续色调。对这种设备不需要半调；在传递函数做了伽马校正后，颜色成分被直接传送到设备。对于确实需要进行半调化的设备，半调化将在所有颜色组件都已经被适当的传递函数转换后进行。半调函数的输入又连续色调、设备本地色彩空间中经过伽马校正过的颜色成分。其输出由设备能重现色彩的像素组成。

PDF 对半调处理的细节提供高度的控制。例如，在彩色印刷中，可以对每种颜色指定独立半调屏幕。当在低分辨率的显示器上渲染时，对半调样式的细化控制是有必要的，以达到对灰度或颜色的最佳近似并把视觉上的人工影响降至最低。

注意：根据定义，记得属于半调的每样东西都是依赖于设备的。一般来说，当一个 PDF 文档提供了其自有的半调规范，那么它也就牺牲了可移植性。与每个输出设备关联的是一个适合于大多数目的默认半调定义。只有相对较高端的文档才需要定义它们自己的半调来达到特殊效果。

所有的半调都在设备空间定义，不受当前转换矩阵的影响。对于正确的结果，一个定义了一个新半调的 PDF 文档必须假定设备空间的分辨率和朝向。半调参数的最佳的选择常常依赖于输出设备特定的物理属性，比如像素的形状、像素间重叠和电子效果或机械噪声。

6.4.1、半调屏幕

一般而言，半调方法基于半调屏幕的概念，半调屏幕将设备像素阵列分成能被修改的单元格来产生想要的半调效果。在概念上，屏是由放置一块相同的矩形网格在设备像素阵列上方来定义的。每个像素属于网格中的一个单元格；一个单一的单元格典型地包括许多像素。屏幕网格是在设备空间整体定义的，且不受对当前矩阵变换修改的影响。此属性是必不可少的，以确保临近的半调着色区域被正确拼合而看不见接缝。

在一个二层（黑白）设备上，屏的每个单元格可以用描绘一些单元格像素的黑和白来近似一个灰色阴影。数值上，在一个单元格中产生的灰度级是在此单元格中白像素与全部像素的个数之比。一个包括 n 个像素的单元格能渲染 $n+1$ 种不同的灰度级，从全黑像素到全白像素。范围在 0.0 到 1.0 间灰度值 g 是由 i 个白像素所产生的，其中 $i=\text{floor}(g \times n)$ 。

先前的描述同样应用到像素由全开或全关的主颜色组成的彩色输出设备上。大多数彩色印刷机，但不是彩色显示器，以这种方式工作。半调被独立应用到每种颜色成分，产生那种颜色的阴影。

颜色成分以加色的形式展现给半调化机构，无论他们原本是指定为加色（RGB 或灰）或减色（CMYK 或淡色）。较大的颜色成分值代表较淡的色彩——在一个加色设备上，例如显示器或无幽默的减色设备例如打印机，将更浓。传递函数以加色形式产生颜色值；参见 6.3 节“传递函数”。

6.4.2、专函数

一个通用定义一个半调屏幕的方式是指定一个频率、角度和专函数。频率是每英寸半调单元格的个数；角度指出了相对设备坐标系统的网格线朝向。当单元格的灰度从黑变到白，单元格中的独立像素以一种良好定义的顺序从黑变到白：如果一个特定的灰度级包括某个白像素，较淡的灰度将包括相同的白像素与一些额外像素。为增加灰度级，像素从黑变成白的顺序是由一个专函数决定的，其以一种尽量减少屏的频率和角度交互的非直接的方式指定了此顺序。

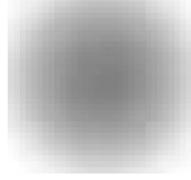
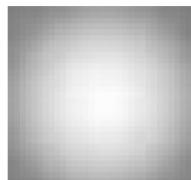
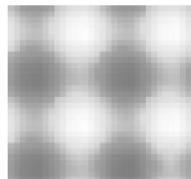
考虑一个有其自有坐标系统的半调单元格：单元格中心是起点，四角在坐标的水平和垂直 ± 1.0 处。单元格中每个像素都在范围均在-1.0 到+1.0 间的水平和垂直坐标的中心。对每个像素，像素坐标作为参数调用专函数，且必须返回范围在-1.0 到+1.0 间的单一数字，按变白的顺序定义了像素的位置。

专函数返回的特定值并不重要；重要的是对不同像素返回的相对值。当一个单元格的灰度级从黑变到白，第一个变白的像素是专函数返回最低值的一个，下一个像素是专函数返回值次低的一个，以此类推。如果两个像素具有相同专函数值，其相对顺序任意决定的。

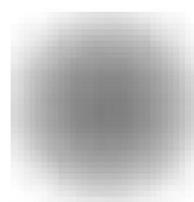
PDF 为许多最常用的专函数提供内建定义。一个半调可以简单地以名称指定任何这些预定义的专函数而不用显式地给出函数定义。例如，名称 **SimpleDot** 标明了一个专函数，其值是与一个像素到半调单元格中心距离的反相关的。这产生一个黑像素簇生在一个圆圈中且其区域是与灰度级成反比的“点屏”。预定义的函数 **Line** 是一个专函数，其值是从一个给定的像素到过单元格中心的一条线的距离，产生一个生长在远离那条线的白像素“线屏”。

表 6.1 显示了预定义的专函数。此表给出了每个函数与相应的在一个 PostScript 计算器函数中定义的 PostScript 语言码在数学上的定义（参见 3.9.4 节“类型 4 (PostScript 计算器) 函数”）。图像连同每个函数显示了函数的相对值是如何在半调单元格分布的，指出了像素变白的近似顺序。图中相应于较深点的像素是较那些较白点晚些变白的。（参见附录 H 中的实现笔记 69。）

表 6.1 预定义的专函数

名称	外观	定义
SimpleDot		$1 - (x^2 + y^2)$ { dup mul exch dup mul add 1 exch sub }
InvertedSimpleDot		$x^2 + y^2 - 1$ { dup mul exch dup mul add 1 sub }
DoubleDot		$\frac{\sin(360 \times x)}{2} + \frac{\sin(360 \times y)}{2}$ { 360 mul sin 2 div exch 360 mul sin 2 div add }
InvertedDoubleDot		$-\left(\frac{\sin(360 \times x)}{2} + \frac{\sin(360 \times y)}{2}\right)$ { 360 mul sin 2 div exch 360 mul sin 2 div add neg }

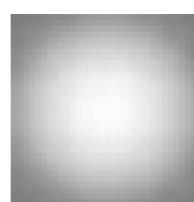
EllipseA



$$1 - (x^2 + 0.9 \times y^2)$$

```
{ dup mul 0.9 mul exch dup mul add 1 exch sub }
```

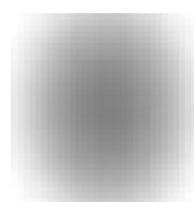
InvertedEllipseA



$$x^2 + 0.9 \times y^2 - 1$$

```
{ dup mul 0.9 mul exch dup mul add 1 sub }
```

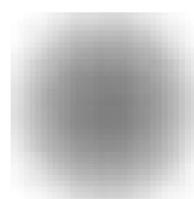
EllipseB



$$1 - \sqrt{x^2 + \frac{5}{8} \times y^2}$$

```
{ dup 5 mul 8 div mul exch dup mul exch add sqrt  
1 exch sub }
```

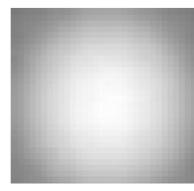
EllipseC



$$1 - (0.9 \times x^2 + y^2)$$

```
{ dup mul exch dup mul 0.9 mul add 1 exch sub }
```

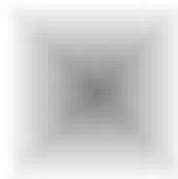
InvertedEllipseC



$$0.9 \times x^2 + y^2 - 1$$

```
{ dup mul exch dup mul 0.9 mul add 1 sub }
```

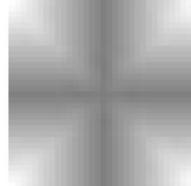
Square



$-\max(|x|, |y|)$

```
{ abs exch abs 2 copy lt
    { exch }
    if
    pop neg }
```

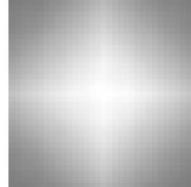
Cross



$-\min(|x|, |y|)$

```
{ abs exch abs 2 copy gt
    { exch }
    if
    pop neg }
```

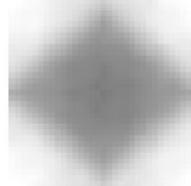
Rhombooid



$\frac{0.9 \times |x| + |y|}{2}$

```
{ abs exch abs 0.9 mul add 2 div }
```

Diamond



if $|x| + |y| \leq 0.75$ *then* $1 - (x^2 + y^2)$

else if $|x| + |y| \leq 1.23$ *then* $1 - (0.85 \times |x| + |y|)$

else $(|x| - 1)^2 + (|y| - 1)^2 - 1$

```
{ abs exch abs 2 copy add 0.75 le
    { dup mul exch dup mul add 1 exch sub }
    { 2 copy add 1.23 le
        { 0.85 mul add 1 exch sub }
        { 1 sub dup mul exch 1 sub dup mul add 1 sub }
        ifelse }
    ifelse }
```

图 6.1 显示了一些预定义专函数的效果。

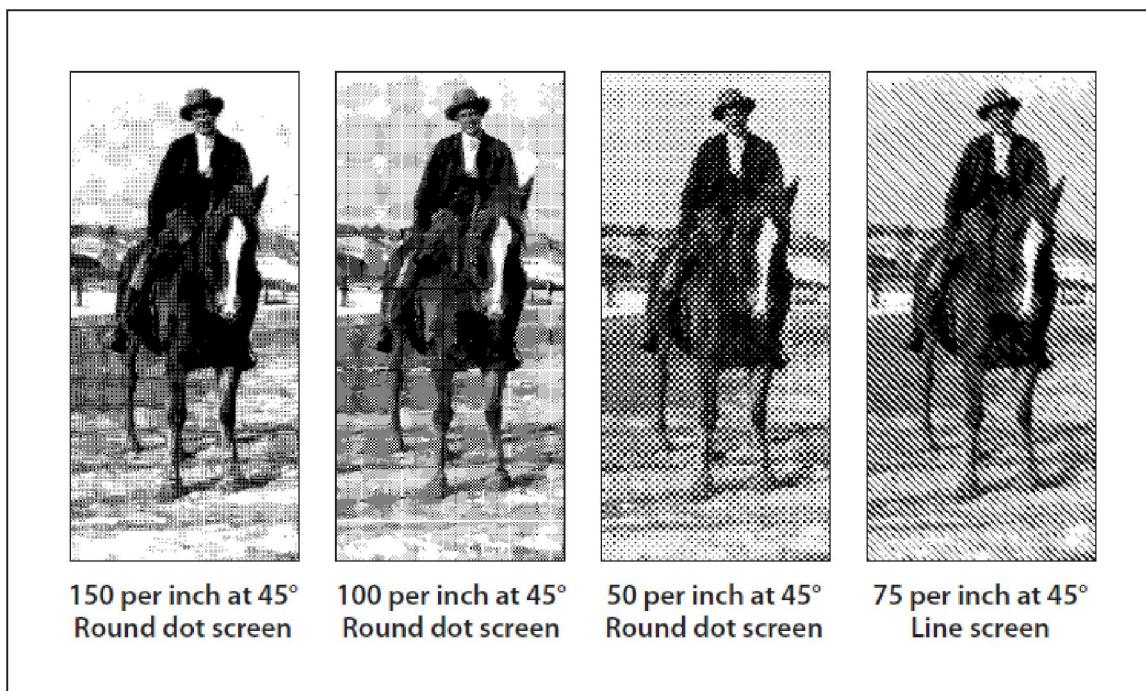


图 6.1 多种半调效果

6.4.3、阈值阵列

另一种定义一个半调屏幕的途径是通过一个直接控制独立设备半调单元格中像素的阈值阵列。这种技术提供一种对半调渲染的高度控制。它也允许半调单元格是任意矩形，而那些由一个专函数控制的半调单元格总是正方形。

阈值阵列更像一个被采样的图像——一个像素值的矩阵阵列——但是整体在设备空间定义的。取决于半调类型，每个阈值占用 8 或 16 位。阈值以常见的方式在名义上代表灰度级，从黑取值 0 到最大白取值（255 或 65535）。阈值阵列被复制以覆盖整个设备空间：设备空间中每个像素都被映射到此阈值阵列中的一个特定采样。在一个二层设备上，每个像素不是黑就是白，以一个阈值阵列来半调化的过程如下：

1. 对每个将被某个灰度级描绘的设备像素，从阈值阵列中查找相应的阈值。
2. 如果请求的灰度级少于阈值，着色设备像素为黑色；否则着色为白色。在范围 0.0 到 1.0 间的灰度级对应于阈值从 0 到最大可用值（255 或 65535）

注意：一个阈值为 0 的值被当作 1 对待；因此灰度级 0.0 会着色所有像素点为黑，不管在阈值阵列中的值是多少。

这种方法容易推广到每像素多位的单色设备。例如，如果每像素有 2 位，每个像素可以直接呈现 4 中灰度级：黑、深灰、浅灰或白，分别编码为 0、1、2 和 3。对任何由某些中间灰度级指定的设备像素，半调算法在阈值阵列中查找相应的值，来决定是否使用次低或次高的可呈现的灰度级。这种情况下，阈值不代表绝对灰度级，而是代表任意两个相邻可呈现灰度级间的梯度。

用这种方式定义的半调也可以被用在对于每个颜色成分拥有有限数量个值的彩色显示器上。红、绿和蓝色成分简单地被作为灰度级独立看待，对每个成分应用适合的阈值阵列。

(这种技术对定义为专函数的屏同样适用，因为专函数被用来内部地计算一个阈值阵列。)

6.4.4、半调字典

在 PDF 1.2 版中，图形状态包括一个通用半调参数，决定了由描绘操作者所使用的半调化过程。通用半调可以指定为一个图形状态参数字典中的 **HT** 入口值；参见第 190 页表 4.8。其可以由一个字典或者一个流来定义，取决于半调的类型；术语半调字典的使用贯穿本节，指的是一个字典对象或一个流对象的部分字典。(由流定义的半调被特别定义为特定半调类型的描述；除非另作申明，它们应被理解为由简单字典定义。)

每个半调字典必须有一个 **HalftoneType** 入口，其值是一个指定了半调定义的全局类型的整型值。字典中余下的入口被依据此类型来转译。PDF 支持的半调类型见表 6.2。

表 6.2 PDF 半调类型

类型	意义
1	用频率、角度和专函数定义一个单半调屏幕
5	定义任意数量的半调屏幕，每个着色剂或颜色成分（包括主色和专色）。此字典中的键是着色剂的名称；值是其他类型的半调字典，每个都为一个单着色剂定义了半调屏幕
6	用一个包含 8 位采样值的阈值阵列定义一个单半调屏幕
10	用一个包含 8 位采样值的阈值阵列定义一个单半调屏幕，代表一个可能具有一个非零屏幕角度的半调单元格。

16

(PDF 1.3) 用一个包含 16 位采样值的阈值阵列定义一个单半调屏幕，代表一个可能具有一个非零屏幕角度的半调单元格。

代表这些半调类型的字典包含相同的入口作为相应的 PostScript 语言半调字典(如 7.4 节中叙述的 *PostScript 语言参考*, 第三版), 但除下列例外:

- PDF 字典可以包含一个值为 **Halftone** 的 **Type** 入口, 鉴别了字典描述的 PDF 对象的类型。
- 专函数和传递函数由函数对象代表, 替代了 PostScript 过程。
- 阈值阵列被指定作为流而不是文件。
- 在半调字典的类型 5 中, 着色剂的键必须是名称对象; 它们不可以是 PostScript 中的那种字符串。

半调字典有一个可选的入口, **HalftoneName**, 用来靠名称识别半调。在 PDF 1.3 中, 如果此入口存在, 所有其他入口, 包括 **HalftoneType**, 都是可选的。在渲染时, 试过输出设备有一个具有特定名称的半调, 那么就使用哪个半调, 而覆写了任何其他在字典中指定的半调参数。这为 PDF 文档提供了一种方式, 来选择被某些设备制造商支持的私有半调, 如果不这样做其将不能被访问, 因为其并没有在 PDF 中显式定义。

如果不存在 **HalftoneName** 入口, 或者如果请求的半调名称在设备上不存在, 半调的参数将被字典中的其他入口定义, 如果有的话。如果不存在其他入口, 就会使用默认半调。

参见第 542 页 7.6.4 节, “渲染参数和透明”并且特别地, “半调和传递函数”, 以获得半调在透明模型中扮演的角色的进一步讨论。

半调类型 1

表 6.3 描述了一个第 1 类半调字典的内容, 其定义了一个依据其频率、角度和专函数的半调屏幕。

表 6.3 半调类型为 1 的字典入口

键	类型	值
---	----	---

Type	名称	(可选的) 此字典所描述的 PDF 对象的类型; 如果存在, 对一个半调字典来说必须是 Halftone 。
HalftoneType	整型	(必须的) 一个代码用来鉴别此字典描述的半调类型; 对此半调类型必须是 1。
HalftoneName	字符串	(可选的) 此半调字典的名称
Frequency	数字	(必须的) 屏幕频率, 以设备空间中每英寸半调单元格个数量度
Angle	数字	(必须的) 屏幕角度, 以逆时针方向相对于设备坐标系统旋转的度数。(大多数输出设备具有左手习惯设备空间。在此种设备上, 设备空间中逆时钟角度对应于一个用户空间中和物理媒介中的顺时针角度。)
SpotFunction	函数或名称	(必须的) 一个定义了一个屏幕单元格中设备像素被调整的顺序的函数对象, 或一个预定义的专函数的名称(参见第 459 页表 6.1)。
AccurateScreens	布尔	(可选的) 一个标志指定了是否调用一个特殊的极度精确但消耗计算资源的半调算法; 参见下面以获得进一步讨论。默认值: false 。
TransferFunction	函数或名称	(可选的) 一个传递函数, 为同一个组件覆写当前图形状态中的传递函数。如果字典是半调类型 5(参见第 475 页“半调类型 5”)的一个组件, 此入口是必需的, 且代表一个非主要或非标准主要颜色成分(参见 6.3 节“传递函数”)。名称 Identity 可以被用来指定鉴别函数。

如果 **AccurateScreens** 入口有一个 **true** 值, 一个高精度半调算法将替代标准算法。如

果 **AccurateScreens** 是 **false** 或不存在，将会使用普通半调。精确半调以非常高的精确性获得请求的屏幕频率和角度，而不同半调仅调整它们使一个单一屏幕单元格量子化为设备像素。在高分辨率设备上做色彩分离，高精度是重要的。然而它有可能消耗大量计算资源，因此其通常是关闭的。

原则上，PDF 允许使用任意大单元格的——换句话说，任意低频率的半调屏幕。然而，相对于设备分辨率非常大或朝向不如意的角度的单元格，可能超出可用内存的容量。如果发生了这种情况，将出现一个错误。**AccurateScreens** 特定常常需要非常大量的内存来完成最高的精确度。

示例 6.1 显示了一个半调类型为 1 的半调字典。

示例 6.1

```
28 0 obj
```

```
<< /Type /Halftone
      /HalftoneType 1
      /Frequency 120
      /Angle 30
      /SpotFunction /CosineDot
      /TransferFunction /Identity
>>
endobj
```

半调类型 6

一个半调类型为 6 的半调用一个阈值阵列定义一个半调屏幕。半调被呈现为一个包含阈值阵列的流；定义半调的参数是由流字典中的入口指定的。此字典可以包含表 6.4 中所示的入口，除了对所有流常见的通常入口（参见第 38 页表 3.4）。**Width** 和 **Height** 入口指定了设备像素中阈值阵列的维度；流必须包含 **Width**×**Height** 字节，每个字节代表一个单一的阈值阵列值。阈值在设备空间中按与图像空间中图像采样相同的顺序定义（参见第 308 页图 4.26），其第一个值位于设备坐标 (0, 0) 且水平坐标比垂直坐标变化要快。

半调类型 10

尽管半调类型 6 能被用来指定一个具有 0 屏幕角度的阈值阵列，但是它们并没有提供其他角度的支持。半调类型 10 移除了此限制，允许对非零屏幕角度的阈值阵列半调的使用。

表 6.4 半调类型为 6 的字典指定的额外入口

键	类型	值
Type	名称	(可选的) 此字典所描述的 PDF 对象的类型；如果存在，对一个半调字典来说必须是 Halftone 。
HalftoneType	整型	(必须的) 一个代码用来鉴别此字典描述的半调类型；对此半调类型必须是 6。
HalftoneName	字符串	(可选的) 此半调字典的名称
Width	整型	(必须的) 阈值阵列的宽度，以设备像素计
Height	整型	(必须的) 阈值阵列的高度，以设备像素计
TransferFunction	函数或名称	(可选的) 一个传递函数，为同一个组件覆写当前图形状态中的传递函数。如果字典是半调类型 5 (参见第 475 页“半调类型 5”) 的一个组件，此入口是必需的，且代表一个非主要或非标准主要颜色成分 (参见 6.3 节“传递函数”)。名称 Identity 可以被用来指定鉴别函数。

非零角度的半调单元格是难以指定的，因为他们可能不能良好地与扫描线成直线排列，也因为其难以决定一个给定的采样点的去向。半调类型为 10 的半调以分割半调单元格为一对以零角度与输出设备的像素网格直线排列的正方形的方式来解决这一困难。此正方形包含与原始单元格相同的信息，但更容易存储和操作。此外，它们可以被映射到所有渲染所使用的内部呈现中。

图 6.2 显示了一个频率为 38.4 每英寸和角度为 50.2 度的半调单元格，以每英寸 300 点的分辨率在设备空间中图形化呈现。图中每个星号代表设备空间中的一个映射到阈值阵列

中特定位置的位置。

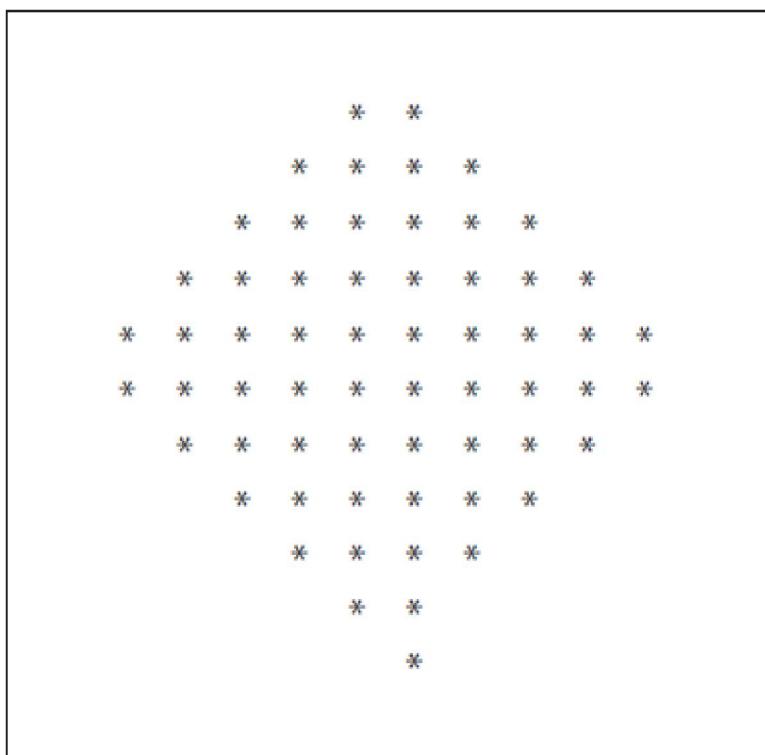


图 6.2 具有非零角度的半调单元格

图 6.3 显示了半调单元格是如何被分割成两个正方形的。如果这两个正方形和原始单元格覆盖了设备空间，正方形右上方区域正好映射到较低正方形的空区域中，反之亦然（参见图 6.4）。第一个正方形中的最后一行与第二个正方形中的第一行相毗邻，起始于相同的列。

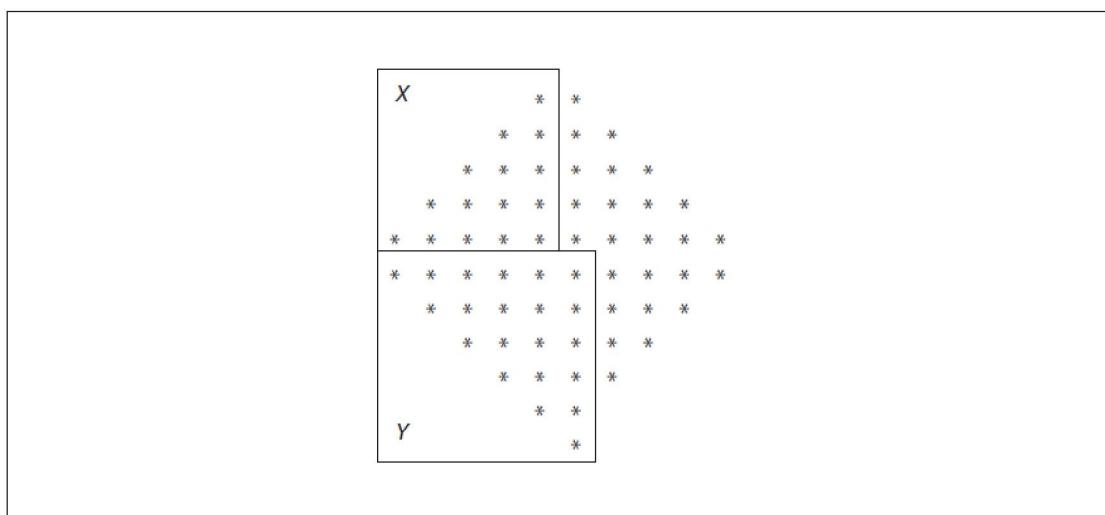


图 6.3 有角度的半调单元格分割为两个正方形

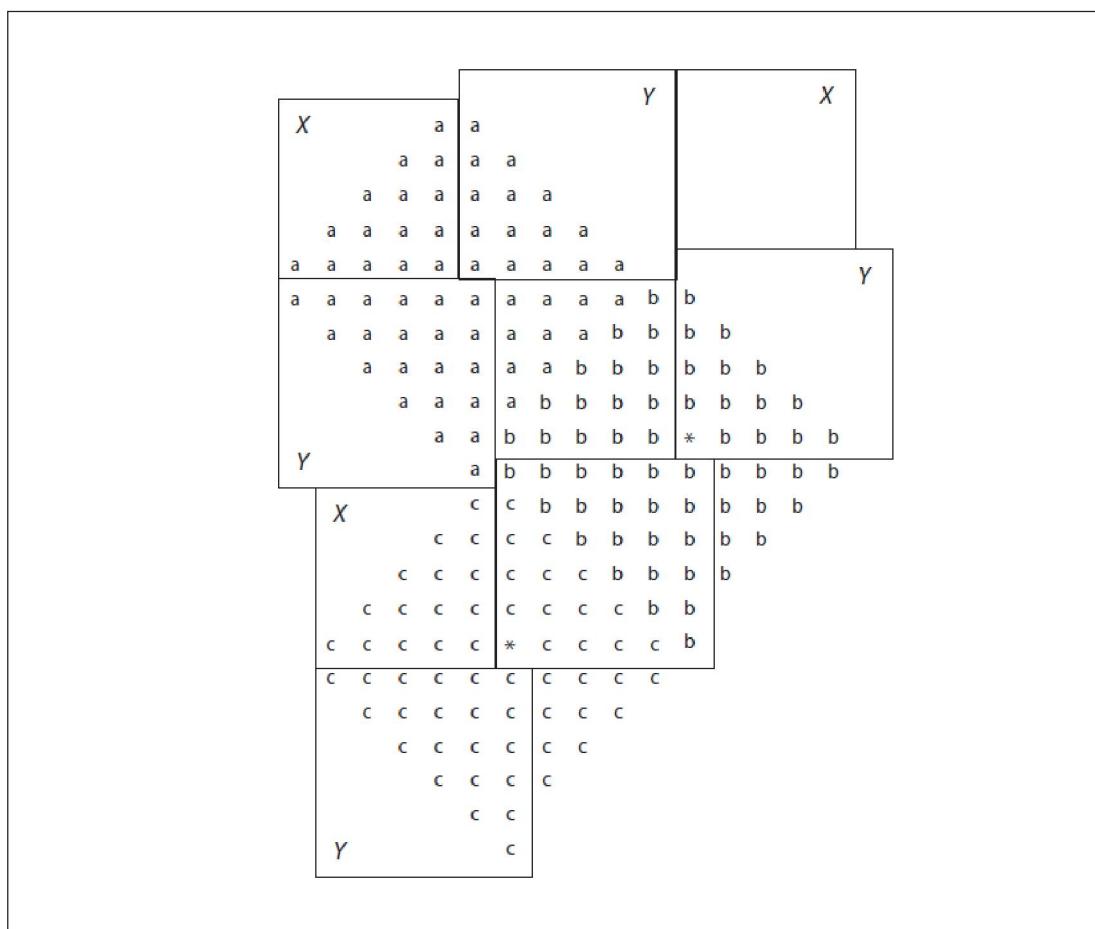


图 6.4 覆盖设备空间的半调单元格与两个正方形

任何半调单元格都可以用这种方法分割。上方正方形 (X) 的边等于从一个半调单元格中的一个点到毗邻单元格的相应点的水平取代，例如图 6.4 中那些用星号标记的。下方正方形 (Y) 的边是相同这样两个点的垂直取代。一个半调屏幕从具有边 X 和 Y 的正方形构造而来的频率是这样给定的：

$$\text{频率} = \frac{\text{分辨率}}{\sqrt{X^2 + Y^2}}$$

角度是由：

$$\text{角度} = \text{atan}\left(\frac{Y}{X}\right)$$

如同半调类型 6，半调类型 10 被作为一个包含阈值的流呈现，定义半调的参数由流字典中的入口指定。除了所有流常见的通常入口（参见第 38 页表 3.4），此字典可以包含表 6.5 中所示的入口。**Xsquare** 和 **Ysquare** 入口替代了半调类型 6 的 **Width** 和 **Height** 入口。

表 6.5 半调类型为 10 的字典指定的额外入口

键	类型	值
Type	名称	(可选的) 此字典所描述的 PDF 对象的类型; 如果存在, 对一个半调字典来说必须是 Halftone 。
HalftoneType	整型	(必须的) 一个代码用来鉴别此字典描述的半调类型; 对此半调类型必须是 10。
HalftoneName	字符串	(可选的) 此半调字典的名称
Xsquare	整型	(必须的) 正方形 X 的边, 以设备像素计
Ysquare	整型	(必须的) 正方形 Y 的边, 以设备像素计
TransferFunction	函数或名称	(可选的) 一个传递函数, 为同一个组件覆写当前图形状态中的传递函数。如果字典是半调类型 5 (参见第 475 页“半调类型 5”) 的一个组件, 此入口是必需的, 且代表一个非主要或非标准主要颜色成分 (参见 6.3 节“传递函数”)。名称 Identity 可以被用来指定鉴别函数。

Xsquare 和 **Ysquare** 入口指定了设备像素中两个正方形的维度。流必须包含 **Xsquare**² + **Ysquare**² 字节, 每个代表一个单一阈值。正方形 X 的内容首先被决定, 然后是正方形 Y。每个正方形中的阈值在设备空间中以与图像空间中的图像采样 (参见第 308 页图 4.26) 相同的顺序被定义, 第一个值位于设备坐标 (0, 0) 且水平坐标比垂直坐标变化要快。

半调类型 16

如同半调类型 10, 半调类型 16 (PDF 1.3) 用一个阈值阵列定义一个半调屏幕, 并允许非零屏幕角度。然而, 在半调类型 16 中, 阈值阵列的每个元素是 16 位宽的而不是 8 位。这使得阈值阵列能区分 65536 级色彩而不仅是 256 级。阈值阵列可以由一个或两个矩形组成。如果指定了两个矩形, 他们像图 6.5 中所示的那样覆盖设备空间。第一个矩形的最后一行与第二个矩形的第一行是毗邻的, 且起始于相同的列。

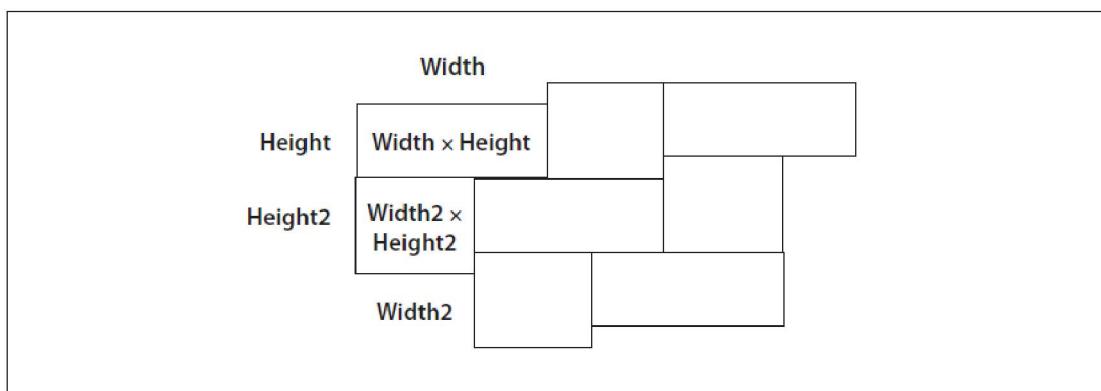


图 6.5 半调类型为 16 的设备空间覆盖

半调类型 16，如同半调类型 6 和半调类型 10，是被作为一个包含阈值的流呈现的，定义半调的参数由流字典中的入口指定。除了所有流常见的通常入口（参见第 38 页表 3.4），此字典可以包含表 6.6 中所示的入口。此字典的 **Width** 和 **Height** 入口定义了第一个（或仅这一个）矩形的维度。第二个可选矩形的维度由可选入口 **Width2** 和 **Height2** 定义。每个阈值由 2 字节呈现，高位字节在前。因此，如果只有一个矩形，流必须包含 **2×Width×Height** 字节，如果有两个矩形，流必须包含 **2×(Width×Height + Width2×Height2)** 字节。第一个矩形的内容首先被定义，然后是第二个。每个矩形中的阈值在设备空间中以与图像空间中的图像采样（参见第 308 页图 4.26）相同的顺序被定义，第一个值位于设备坐标 (0,0) 且水平坐标比垂直坐标变化要快。

表 6.6 半调类型为 16 的字典指定的额外入口

键	类型	值
Type	名称	(可选的) 此字典所描述的 PDF 对象的类型；如果存在，对一个半调字典来说必须是 Halftone 。
HalftoneType	整型	(必须的) 一个代码用来鉴别此字典描述的半调类型；对此半调类型必须是 16。
HalftoneName	字符串	(可选的) 此半调字典的名称
Width	整型	(必须的) 阈值阵列中第一个（或仅这一个）矩形的宽度，以设备像素计

Height	整型	(必须的) 阈值阵列中第一个(或仅这一个)矩形的高度, 以设备像素计
Width2	整型	(可选的) 阈值阵列中可选的第二个矩形的宽度, 以设备像素计。如果此入口存在, Height2 入口也必须存在。如果没有此入口, Height2 入口也必须不存在, 且阈值阵列仅有一个矩形。
Height2	整型	(可选的) 阈值阵列中可选的第二个矩形的高度, 以设备像素计。
TransferFunction	函数或名称	(可选的) 一个传递函数, 为同一个组件覆写当前图形状态中的传递函数。如果字典是半调类型 5 (参见第 475 页“半调类型 5”) 的一个组件, 此入口是必需的, 且代表一个非主要或非标准主要颜色成分 (参见 6.3 节“传递函数”)。名称 Identity 可以被用来指定鉴别函数。

半调类型 5

某些设备, 特别是彩色印刷机, 需要为每个独立的着色剂配以不同的半调。而且, 能产生指定分离操作的设备可能需要为每种分离操作配以单独的半调。半调字典类型 5 允许为一个任意数量的着色剂或颜色成分指定单独的半调。

半调字典类型 5 (表 6.7) 是一个混合字典, 包括对应多种着色剂的独立半调定义。它的键是名称对象, 代表单独周色剂或颜色成分的名称。与这些键关联的值是其他半调字典, 每个半调字典为一个单一着色剂或颜色成分定义了半调屏幕和传递函数。成分半调字典可以是除了类型 5 以外的任何受支持的类型。

表 6.7 半调类型为 5 的字典的入口

键	类型	值
---	----	---

Type	名称	(可选的) 此字典所描述的 PDF 对象的类型; 如果存在, 对一个半调字典来说必须是 Halftone 。
HalftoneType	整型	(必须的) 一个代码用来鉴别此字典描述的半调类型; 对此半调类型必须是 5。
HalftoneName	字符串	(可选的) 此半调字典的名称
任何着色剂的名称	字典或流	(必须的, 每个着色剂对应一个) 相应于着色剂或颜色成分的半调, 以键命名。半调可以是除类型 5 以外的任何类型。注意, 键必须是一个名称对象; 字符串是不允许的, 不像其在 PostScript 半调字典中的类型 5 中那样。
Default	字典或流	(必须的) 一个用于没有其自己的入口的任一着色剂或颜色成分的半调,。值不可以是半调类型 5。如果有任何非主着色剂, 默认半调必须有一个传递函数。

呈现于一个半调字典类型 5 中的着色剂或颜色成分分为两种:

- 对应标准本地设备色彩空间 (灰对应于**DeviceGray**; 红、绿和蓝对应于**DeviceRGB**; 青、品红、黄和黑对应于**DeviceCMYK**;) 的主颜色成分。
- 对应在**Separation**和**DeviceN**色彩空间中被用作专色的非标准颜色成分。如果本地色彩空间是非标准的, 其中一些也可以被用作印刷色。

字典必须同时包含一个键为**Default**的入口。此入口的值是一个被用作任何颜色成分的、没有其自己入口的半调字典。

当一个其他类型的半调字典作为一个半调字典类型5的一个入口出现, 其仅应用于由入口键命名的单一着色剂或颜色成分。这是与一个被用作图形状态中通用半调参数相比的, 其应用于所有颜色成分。如果在当通用半调是由除半调字典类型5以外的任何方法定义时, 请求了非主着色剂, 将会为所有这样的着色剂使用灰半调屏幕和传递函数。

示例6.2显示了一个具有一个CMYK设备主色成分的半调字典类型5。在此例子中, 对应颜色成分和默认的半调字典全部使用相同的专函数。

示例6.2

```
27 0 obj
<< /Type /Halftone
    /HalftoneType 5
    /Cyan 31 0 R
    /Magenta 32 0 R
    /Yellow 33 0 R
    /Black 34 0 R
    /Default 35 0 R
>>
endobj

31 0 obj
<< /Type /Halftone
    /HalftoneType 1
    /Frequency 89.827
    /Angle 15
    /SpotFunction /Round
    /AccurateScreens true
>>
endobj

32 0 obj
<< /Type /Halftone
    /HalftoneType 1
    /Frequency 89.827
    /Angle 75
    /SpotFunction /Round
    /AccurateScreens true
>>
```

endobj

33 0 obj

```
<< /Type /Halftone
    /HalftoneType 1
    /Frequency 90.714
    /Angle 0
    /SpotFunction /Round
    /AccurateScreens true
```

>>

endobj

34 0 obj

```
<< /Type /Halftone
    /HalftoneType 1
    /Frequency 89.803
    /Angle 45
    /SpotFunction /Round
    /AccurateScreens true
```

>>

endobj

35 0 obj

```
<< /Type /Halftone
    /HalftoneType 1
    /Frequency 90.000
    /Angle 45
    /SpotFunction /Round
    /AccurateScreens true
```

>>

endobj

6.5、扫描转换的细节

渲染的最后一步是扫描转换。如同 2.1.4 节中讨论的，“扫描转换，”应用程序执行一个扫描算法来在输出设备的光栅内存中绘制图形、文本和图像。

扫描转换算法的规范没有定义为 PDF 的一部分。不同的实现可以以不同的方式执行扫描转换；对一个设备适合的技术可能对另一个并不适合。但是，有一个对扫描函数是如何工作的通用理解是有用的，特别是当创建期望在显示器上查看的 PDF 文档时。在典型的低分辨率显示器上，一个像素宽度的变化都可能会对所描绘的形状的外观带来一个显著的影响。

下列各节描述了 Acrobat 产品中典型的扫描转换算法。（这些细节同样应用于 PostScript 产品，当一个应用程序在一个 PostScript 打印机上描绘一个文档，其可产生一贯相同的结果。）大多数扫描转换的细节不受程序的控制，但有一些是；用来控制它们的参数在此作了描述。

6.5.1、平面公差

平面公差控制数学上正确的路径与一个由直线段构造的近似之间设备像素间距允许的最大值，如图 6.6 所示。平面可以被指定为 `i` 操作符的操作数（参见第 189 页表 4.7）或者被指定为在一个特性状态参数字典中 `FL` 入口的值（参见第 190 页表 4.8）。它必须是一个正数；较小的值以需要更多计算的代价换来更高的精度。

注意：尽管图中为了看得清晰，夸大了弧线与取直路径间的差别，但平面公差的目的是为了控制曲线渲染时的精度，而不是画内接多边形。如果参数的值足够大，引起可见的直线段出现，那么结果是不可预测的。

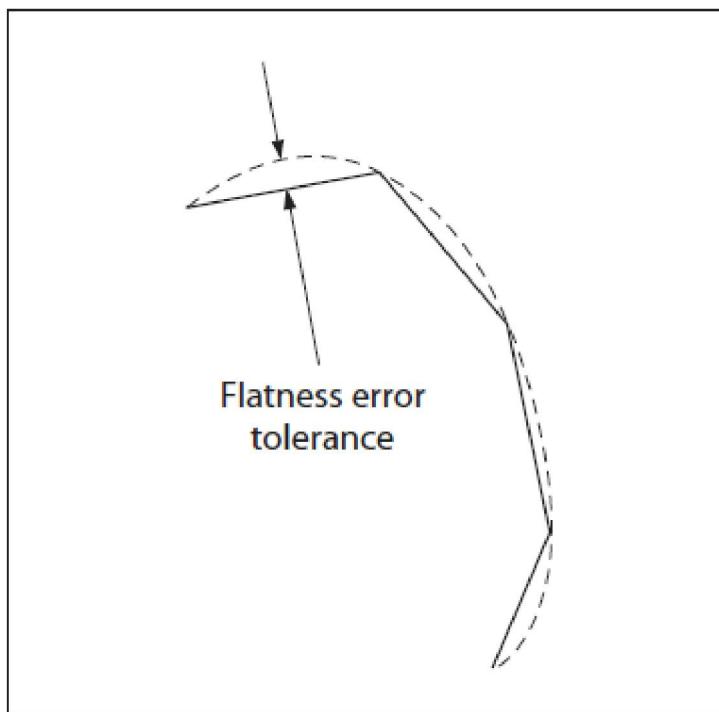


图 6.6 平面公差

6.5.2、平滑度公差

平滑度公差 (PDF 1.3) 控制平滑造影 (类型 2 样式和 **sh** 操作符) 的质量，从而间接控制渲染性能。平滑度是一个由分段线性插值近似的造影和一个造影函数真实值 (可能是非线性的) 间可允许的颜色误差。误差是对每个颜色成分进行测量，并取最大误差值。可允许的误差 (或公差) 是作为一个颜色成分范围的分数来表达的，从 0.0 到 1.0。这样，一个 0.1 的平滑度公差代表在每个颜色成分中有 10% 的公差。平滑度可以被定义为一个图形状态参数字典 (参见第 190 页表 4.8) 中 **SM** 入口的值。

每个输出设备可以有内部最大和最小可达公差的限制。例如：设置平滑度为 1.0 可在一个高质量彩色设备上导致一个内部平滑度 0.5，而在相同的设备上设置其为 0.0 可能导致一个内部平滑度 0.01，如果那种误差在设备上是觉察不到的话。

平滑度公差也可能对色彩转换的精确度有影响。如果一种色彩转换是由一个采样函数定义，那么转换函数是未知的。这样误差可能被以一个很低的频率采样，因而由平滑度公差定义的精确度就不能得到保证。然而在大多数情况下，当转换函数是平滑且连续的，精确度应该会在指定的公差范围内。

平滑度公差的效果与平面公差的效果相似。然而，要注意平面公差是以依赖于设备的像素单元宽度量度的，而平滑度公差是以一个颜色成分范围的一小部分做量度的。

6.5.3、扫描转换规则

下列规则决定了描绘操作影响哪一个设备像素。所有对坐标和像素的引用都是在设备空间中。形状是一个以当前颜色或以一幅图像描绘的路径。它的坐标被映射到设备空间中而不是围在设备像素边界周围。在这一级别，曲线被拉直为一系列直线，所有“内部”计算都被执行。

像素边界总是落在整数设备空间坐标上。像素是一个由具有最小水平和垂直坐标角的位置识别的方形区域。此区域是半开的，意味着它包括其较低而不是较高边界。更精确点，对任何实数坐标为 (x, y) 的点，令 $i = \text{floor}(x)$ 且 $j = \text{floor}(y)$ 。包含此点的像素被看作 (i, j) 。属于该像素的区域被定义为点集 (x', y') ， $i \leq x' < i + 1$ 且 $j \leq y' < j + 1$ 。像被填充和加横运算描绘的像素、形状，也被当作在其“底”边而不是“顶”边包含边界的半开区域。

形状是由描绘与形状相交的方形区域的任意像素进行扫描转换的，无论此相交有多么小。这确保了没有形状会以一种相对于设备像素网格不可接受的放置而消失。由描绘的像素所覆盖的此区域总是至少与原始形状的区域一样大。此规则应用于填充操作和以非零宽度加横。零宽度的加横是以一种依赖于设备、可能包含比规则所暗示的较少的像素的方式完成的。

注意：通常上，两个区域的相交被定义为其内部的相交。然而，对于扫描转换的目的来说，填充区域被认为是使每一个其边界通过路径上的像素相交，即使填充区域的内部是空的。这样，举个例子，一个零宽度或零高度的矩形会描绘一个 1 像素宽的直线。

由一个采样的图像描绘的设备空间区域是由类似于一个填充过的形状的区域决定的，虽然不是完全等同的。应用程序转换图像的源矩形到设备空间并定义一个半开区域，如同填充操作那样。然而，只有中心落在区域内的那些像素会被描绘。这种像素的中心的位置——换句话说，有二分之一的一部分的坐标值的点，被映射回到源空间以决定如何着色此像素。像素区域没有平均数；如果源图像的分辨率高于设备空间，某些采样将不被使用。

对于剪切，剪切区域由被一个填充操作所包含的一像素集组成。随后的描绘操作影响一个区域，此区域是由具有该区域要描绘的像素集的剪切区域定义的这个像素集的相交。

文字字符的扫描转换是由一个不同于上面所述的算法执行的。字体渲染算法使用字符描述中的提示和指定的字符光栅化技术。

6.5.4、自动加横调整

当一个加横顺着一条路径画出，扫描转换算法可能因为光栅化的影响，产生非均匀厚度的直线。一般而言，被转换到设备空间中的线宽和端点的坐标，是没有量化为设备像素的任意实数。一个给定的线宽可以与不同数量的设备像素相交，取决于其位置在何处。图 6.7 指明了这种影响。

为获得最佳结果，对光栅化的影响做补偿以产生厚度均匀的加横是重要的。这在低分辨率显示器应用程序上尤其重要。为达这一需求，PDF 1.2 版提供了一个可选的自动加横调整特性。当加横调整特性启用，线宽和一个加横的坐标会被自动调整，以产生均匀厚度。厚度尽可能接近请求的线宽——误差不超过半个像素。

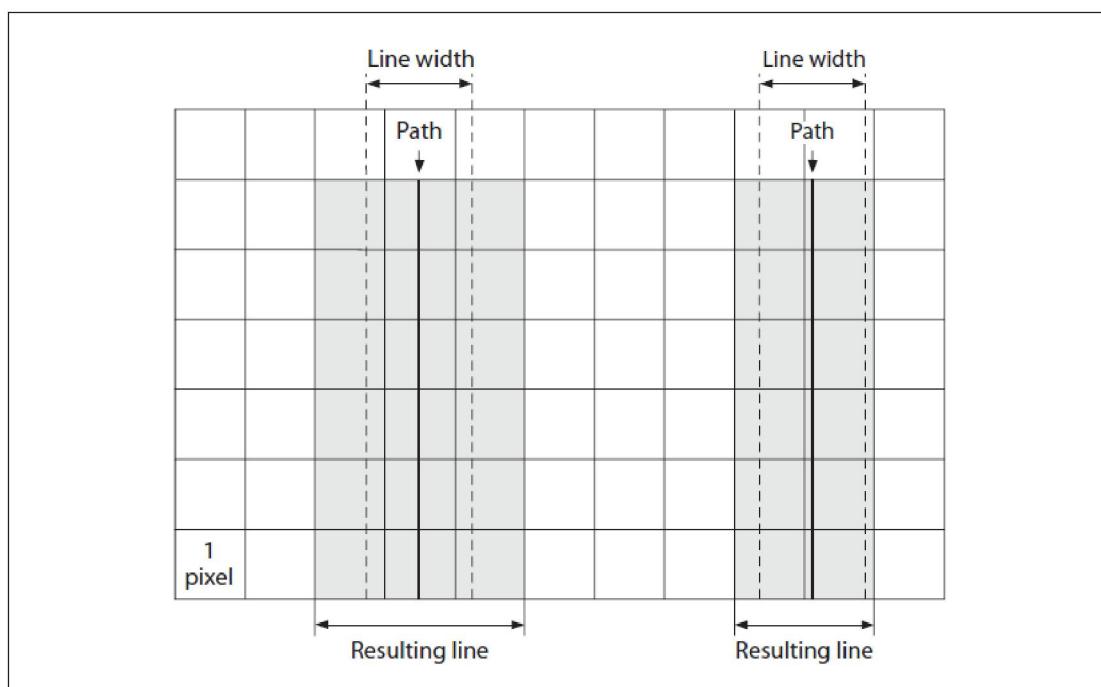


图 6.7 不使用加横调整的光栅化

注意：如果启用了加横调整，且转换到设备空间中的要求的线宽小于半个像素，那么加横会被作为一个像素直线渲染。这是在设备分辨率下能够渲染的最薄直线。其与设置线宽为 0 的效果是一致的（参见 6.5.3 节，“扫描转换规则”）。

因为自动加横调整可能对直线的外观产生实质上的影响，一个 PDF 文档必须能够控制是

否执行此调整。这可通过图形状态中的加横调整参数指定，依靠设置图形状态参数字典中的 SA 入口达到（参见 4.3.4 节，“图形状态参数字典”）；参见附录 H 中的实现笔记 70。

第 7 章 透明度

PDF1.4 扩展了 Adobe 成像模型，包括透明度的概念。透明物体不是必须遵从一个严格的不透明画板模型，但是可以以一种有趣的方式和其他重叠的对象混合（复合）。本章描述通常的透明模型但是不涵盖它们是如何实现的。为了阐明该模型的行为，类似实现的描述被用于不同的地方去描述事物如何工作的。实际上的实现几乎将确定与这些描述所暗示的有很大的不同。

本章将按以下组织：

- 7.1 小节，“透明度的概述”，介绍透明模型的基本概念和它的相关术语。
- 7.2 小节，“基本合成计算”，描述一个物体与它背景合成所包含的数学关系。
- 7.3 小节，“透明组”，介绍透明组的概念和描述它们的特性和行为。
- 7.4 小节，“软掩码”，包括建立和使用掩码去指定位置依赖的形状和不透明性。
- 7.5 小节，“指定 PDF 的透明度”，描述透明度属性如何在 PDF 文档上被呈现的。
- 7.6 小节，“颜色空间和渲染的问题”，处理一些特殊的，关于透明度与颜色编码、渲染其他方面的交互。

7.1、透明度的概述

原始 Adobe 成像模型描述对象（填充，划线，文本和图像），很可能按照路径修剪，不透明地放进一个页中。该页的任何点的颜色就是最上面的附着对象，忽视任何之前的对象它也许会重叠。这个影响可以是一并且经常是一简单地使用渲染对象以它们指定的顺序嵌入页面来实现，并且每个对象完全遮住了其他任何被它所覆盖的对象。

根据透明成像模型，页面的所有对象可以潜在地影响这个结果。在给定的点的对象可以被认为是以一种透明度栈方式组组织起来的（或者短堆栈）。这些对象分布从底部到顶部以一种它们特定的顺序排列。该页在每个点的颜色由所有栈内的附着对象的颜色共同决定。它是根据透明度模型的合成规则来合成。

注意：对象指定的顺序决定了栈的顺序，但是不是对象实际被画在页面中的顺序。特别的，透明度模型在整个栈渲染进页面之前，不需要一个消费应用程序立即去光栅化处理这些对象或者任何时候执行一个光栅呈现。这个很重要，因为光栅化经常引起重要信息和细节的丢失，所以最好在透明化计算的中间步骤中避免光栅化处理。

一个给定的对象是和背景复合的。一般来说，背景包含所有对象所在的栈，这个栈之前已经指定了。复合的结果被当做下一个对象要复合时候的背景。然而，在特定种类的透明组（看下面）中，我们选择一种不同的背景。

当一个对象与它的背景复合时，各点的颜色使用一个特定的混合模型来计算，这个模型是一个对象颜色和背景颜色的函数。这混合模型决定了颜色之间如何相互作用；不同的混合模型可以被用于取得各种不同的有用的效果。一个单一的混合模型实际上是为了将所有给定的对象的混合，但是不同的混合模型可以应用于不同的对象。

一个对象和它的背景的混合是由两个标量综合决定的，这两个标量叫做：形状和不透明性。从概念上说，对于每个对象，这些量是由平面上每个点决定的，就好像它们是附加的颜色组件一样。（在实际的运作中，它们经常从辅助源取得而不是从对象内部获得。）

形状和不透明性都是从 0.0 (无贡献) 到 1.0 (最大贡献)。在任何形状或者不透明性都是 0.0 的物体上，它的颜色是未定义的。在任何形状是 0.0 的点上，它的不透明性也是未定义的。形状和不透明性是主要的混合规则；因此，栈作为一个整体在任何点也有形状和不透明性。

一个对象的不透明性，结合背景的不透明性，决定了对背景颜色，对象颜色以及由此产生的混合色、复合色的相对贡献。该对象的形状决定了混合色在多大程度上取代背景色。0.0 和 1.0 的形状值标识了这些点是落在一个严格约定边界的物体内还是物体外；0.0 和 1.0 中间的值在定义软边界对象时很有用。

形状和不透明性在概念上是非常相识的。实际上，它们通常可以联合成一个单一的数值，叫做 `alpha`，这个量控制颜色合成计算以及一个对象和它背景之间色衰计算。然而，有一些情况必须被区分开对待；看 7.3.5 小节，“淘汰组。”此外，基于光栅的实现必须维持一个独立的形状参数来实现反锯齿特性；因此，这很方便地使之成为该模型的一个明确部分。

在一个栈中，一个或者多个连续的对象可以被一起收集起来放进一个透明组中（我们一下简称为一个组）。这个组作为一个整体可以有各种特性，包括修改组内对象复合的行为和它们和它们的背景之间的交互。另一个混合模型，形状和不透明性也可以与这个组联系在一起成为一个整体，并且当组和它的背景混合时被使用。组可以被嵌套进其他组，形成一个树形的层次结构。

注意：一个透明组的概念是独立于已经存在的组的概念，或者类似 Adobe Illustrator 中在应用程序中的层。这些组反映了对象之间的逻辑关系，这些关系在编辑它们的时候是很有意义的，但是它们不是成型模型的一部分。

版面 16 说明了透明组的影响。在上面两个图表中，三色圆被作为独立的对象画出来，并没有组。在左上角，三个对象被不透明的画出来（不透明性=1.0）；各个对象

用它自己的颜色完全代替了它的背景（包括之前画的对象）。在右上角，同样的三个独立的对象以不透明性为 0.5 画出，引起它们相互之间，与灰白色背景之间的复合。在下面的两张图标中，三个对象被联合成一个透明组。在左下角，组内那些独立的对象拥有的不透明性为 1.0，但是组作为一个整体以不透明性为 0.5 常规的混合模型画出来。这些对象因此在组内完全覆盖了各自，但是组最后的结构透明地与灰白的背景相混合。在右下角，对象在组内有 0.5 的不透明性，因此相互之间混合。这个组作为一个整体以不透明性为 1.0 画在背景之上，但以一种不同的混合模型 (`HardLight`)，创造一种不同的虚拟效果。

一个组复合的颜色可以被转化成一个单组件的亮度值，并称之为软掩码。这个掩码可以被用作一个形状和不透明性数值的附加源，以便进行随后的复合操作。当掩码被当作一个形状使用时，这个技术就是大家熟知的软切割；这是在不

透明成像模型中一种当前剪切路径的广义概述（看 4.4.3 节，“截取路径操作”）。

当前页的概念广义指的是包含整个对象的栈的透明组所在的页，它与纯白的完全不透明的背景相复合。逻辑上，这整个栈被光栅化来决定实际的传给输出设备的像素值。

注意：在一个 PDF 页被当作一个艺术品摆放在其他一些页面——例如一个 Illustrator 艺术板或者一个 Encapsulated PostScript (EPS) 文件——它将被视作一个组而不是一个页面，这个组的背景也许被定义成与该页不同的。

7.2、基本的复合计算

本节描述了一个单一的对象与它的背景之间复合的基本计算。这些计算将在 7.3 节中展开，“透明组，”来代表含有多个对象的组。

7.2.1 复合计算的基本符号

通常来说，在本章中，变量名含有一个小写字母来表示标量，例如一个不透明度。大写字母表示一个有多重标量的组件值，例如一个颜色。在基本颜色复合计算的描述中，颜色值通常由字母 C 表示，并使用一个助记的下标来表示指的是哪一种颜色；例如， C_s 代表“光源颜色。”形状和不透明性的数值分别以字母 f（代表“form facotr”）和 q（代表“opaqueness”）——同样我们采取下标助记符，例如 q_s 代表“光源不透明性。”符号 a (alpha) 代表形状和不透明性数值的一个乘积。

在特定的计算中，一个或者多个变量也许有未定义的数值；例如，当不透明性数值为 0 时，相应的颜色是未定义的。一个量也可以未定义，如果它是从除以 0 的运算获得结果。在任何使用这些未定义的量的公式中，这个量对最终的机构没有影响，因为它随后被 0 乘或者相反取消掉。这个的意义在于，当任何数值可以被选择成一个未定义的量，计算就必须不能由于溢出或者除以 0 引起的异常而中断。这将方便地进一步得到 $0 \div 0 = 0$.

7.2.2 基本的复合公式

先前为了适应透明度在成像模型上所做的改变是在颜色如何被画上去的。在

透明模型里，画的结果（最后的颜色）是一个函数，它是画上去的颜色（光源颜色）和将被画掉的颜色（背景颜色）的函数。这些颜色也都随着所处页面的不同位置而不同；然而，本节主要集中在页面的某些固定点并假设一个固定的背景颜色和光源颜色。

在这个计算中其他参数是 **alpha**，它控制背景色和光源颜色的相对贡献，以及混合函数，它明确说明它们在画操作过程中是如何联系的。由此产生的基本颜色复合公式

（或者短的基本颜色复合公式）决定了画操作产生的最后颜色：

$$C_r = \left(1 - \frac{\alpha_s}{\alpha_r}\right) \times C_b + \frac{\alpha_s}{\alpha_r} \times [(1 - \alpha_b) \times C_s + \alpha_b \times B(C_b, C_s)]$$

这些变量的意义将在表 7.1 中说明。

表 7.1 基本复合公式所用到的变量

变量	意义
C_b	背景色
C_s	光源颜色
C_r	结果色
α_b	背景 alpha
α_s	光源 alpha
α_r	结果 alpha
$B(C_b, C_s)$	混合函数

这个公式实际上是复合公式的简化，原始公式中性状和不透明性数值是联系起来的，并使用一个单一的 **alpha** 表示；更多的一般形式稍后将展示出来。这个函数是基于过度操作，过度操作是由波特和达芙在文章“复合数字图像”中定义（看自传），扩展到包括一个覆盖边界区域的混合模型。下面的小节详细说明它的意义和这个公式暗含的意义。

7.2.3 混合颜色空间

上面展示的复合公式实际上是一个向量函数：它操作的颜色被表示成含有 n 个元素的向量的形式，其中 n 是复合进行时颜色空间需要的组件数量。结果色 Cr 的第 i 个组件是应用复合公式到原色 Cb, Cs 的第 i 个组件结合 B(Cb, Cs) 来获得的。

这个计算结果依据颜色所代表的颜色空间。基于这个原因，用于复合的颜色空间被成为混合颜色空间，很明显是透明成像模型的一部分。当需要时，背景和光源颜色在复合计算之前转化为混合颜色空间。

PDF 颜色空间在 4.5 小节，“颜色空间”有描述，下面这些被支持成为混合颜色空间：

- **DeviceGray**
- **DeviceRGB**
- **DeviceCMYK**
- **CalGray**
- **CalRGB**
- **ICCBased** 颜色空间相当于以上这些（包括校准的 CMYK）

Lab 空间和 ICCBased 空间分别代表光线亮度和色度（例如 L*a*b*, L*u*v*，以及 HSV），它们不被允许作为混合颜色空间，因为当分别应用于各个组件时，这些空间的复合计算得不出有意义的结果。另外，一个作为混合颜色空间的 ICCBased 空间必须是双向作用的：也就是说 ICC 轮廓必须同时包含 AToB 和 BToA 转换。

混合颜色空间仅对印刷色有参考意义。虽然混合可以在一个分离的或者 DeviceN 颜色空间指定的独立颜色点上完成，但是这些颜色从不被转化成混合颜色空间（除了一种情况是它们首先恢复其可选的颜色空间，正如在 234 页“分离颜色空间”所描述的和 238 页“DeviceN 颜色空间”所描述的）。相反，指定的颜色组件是与其相应的背景混合的。

各种混合模型的混合函数假设各颜色组件从 0.0 到 1.0，并且颜色空间是附加的。前面的条件是适合所有被允许的混合颜色空间的，但是后面的条件并不是

这样。特别地，DeviceCMYK，分离的，以及 DeviceN 空间是可负的。当进行有负的颜色空间的混合操作时，假设颜色组件数值在混合函数被应用之前是有多余的（从 1.0 开始减），那么函数的结果将在被使用前补充回去。这个调整有效地使得所有的颜色空间在数值上的一致。然而，一个给定的混合模型的实际虚拟效果是依赖于颜色空间。在一个图案颜色空间的混合产生图案依赖的结果，然而在基于 CIE 的空间，它产生的结果是对所有图案一致的。看 7.6 小节，“颜色空间和渲染的问题”，将有更多的关于颜色空间的具体细节。

7.2.4 混合模型

原则来说，用在复合公式中去指定混合操作的混合函数 $B(c_b, c_s)$ ，可以是背景和光源颜色的任何函数，并从结果上来说，产生另一种颜色， c_r 。PDF 定义了一个标准的命名的混合函数集，或者混合模型，在表 7.2 和 7.3 中列出。图板 18 和 19 分别展示了 RGB 和 CMYK 的虚拟效果。

一个混合模型被称为可分离的，如果结果颜色的每个组件是完全由相应构成的背景组件和光源颜色决定的—也就是说，如果混合模型函数 B 独立应用到每个相应组件：

$$c_r = B(c_b, c_s)$$

小写变量 cb , cs 以及 cr 代表相应的颜色 Cb , Cr , Cs 组件，这些变量一附件的形式表现出来。（理论上，一个混合模型可以为每个颜色组件设置一个不同的函数并且该模型仍然可以是可分离的；然而，标准的 PDF 混合函数并没有这个特性。）一个可分离的混合模型可以和任何颜色空间一起使用，因为它可独立地应用于任何数目的组件。只有可分离的混合模型可以被用于专色的混合。

表 7.2 列举出了 PDF 中标准的可分离的混合模型。这些模型中的一些被实际的数学公式定义的；其他的只是用他们预期的效果的常规描述来定性。

表 7.2 标准可分离的混合模型

名字	结果
Normal	选择光源颜色，忽视背景： $B(c_b, c_s) = c_s$

Multiply

背景和光源颜色数值的乘

$$B(c_b, c_s) = c_s \times c_s$$

结果颜色总是至少和两种构成的颜色中的任意一种一样深。任何颜色和黑色相乘产生黑色；和白色相乘保持原始的颜色不改变。用一种不是黑色或者白色的颜色连续地画重叠对象将逐渐地产生越来越暗的颜色。

Screen

背景和光源颜色数值补充值的相乘， 补足的结果是：

$$\begin{aligned} B(c_b, c_s) &= 1 - [(1 - c_b) \times (1 - c_s)] \\ &= c_b + c_s - (c_b \times c_s) \end{aligned}$$

结果颜色总是至少和两种构成的颜色中的任意一种一样浅。用白色遮蔽任何颜色都产生白色；用黑色遮蔽任何颜色都是对象保持原始的颜色不改变。这效果类似于多个摄像幻灯片同时投影到一个屏幕上。

Overlay

乘或者遮蔽这些颜色，依靠的是背景色。光源颜色覆盖背景却保持它的亮度和阴影。背景颜色并不被取代但是被光源颜色所混合，反映了背景的亮度和灰度。

Darken

选择背景和光源颜色中更黑的那个：

$$B(c_b, c_s) = \min(c_b, c_s)$$

如果光源颜色更深， 背景被光源颜色取代；反之， 不改变。

Lighten

选择背景和光源颜色中更亮的那个：

$$B(c_b, c_s) = \max(c_b, c_s)$$

如果光源颜色更亮， 背景被光源颜色取代；反之， 不改变。

Color Dodge

加亮背景色来反映光源颜色。用黑色描绘不产生改变。

Color Burn

加灰背景色来反映光源的颜色。用白色描绘不产生改变。

HardLight	依据广域颜色的数值，乘或者遮蔽这些颜色。如果光源颜色比 0.5 还要浅，背景将变浅就像被遮蔽了一样；这在给场景加高亮的时候是很有用处的。如果光源颜色比 0.5 深，背景将变深就像被乘了一样。这在给场景添加阴影的时候很有用处的。加亮或者加灰的程度与光源颜色和 0.5 的差值是成比例的。如果等于 0.5，背景将不改变。用纯黑或者纯白来描绘，产生纯黑或者纯白。这效果类似于将一个粗糙的聚光灯找到背景上。
SoftLight	加灰或者加亮颜色，依据的是光源颜色数值。如果光源颜色比 0.5 浅，背景将加亮，就像它躲开了一样；这在给场景加亮是很有用处的。如果光源颜色比 0.5 深，背景色将变深就好像被烧了一样。加亮或者加深的程度与光源颜色和 0.5 的差值成比例；如果光源颜色等于 0.5，背景将不改变。用纯黑或者纯白描绘，产生分明的更深或者更浅区域，但是不产生纯白或者纯黑。这效果就类似于将散开的聚光灯照在背景上。
Difference	将两种构成的颜色中较深的颜色减去较浅的颜色： $B(c_b, c_s) = c_b - c_s $ 用白色描绘将是背景色变成相反的；用黑色描绘将不产生改变。
Exclusion	产生的效果类似于 Difference 模型，但是相比之下更低。用白色描绘将是背景色变成相反的；用黑色描绘将不产生改变。

表 7.3 列举了标准的不可分离的混合模型。它们的效果被描述出来了，但是不是以数学公式给出。这些模型都必须转化到或者从中间的 HSL 表示（色调-饱和-亮度）转化而来。因为不可分离的混合模型考虑了组合内的所有的颜色组件，它们的计算根据混合颜色空间，这混合颜色空间内组件将被解析。

表 7.3 标准不可分离混合模型

名字	结果
----	----

Hue	用光源颜色的色调，背景色的饱和度以及亮度来创造一种颜色
Saturation	用光源颜色的饱和度，色调，背景色的亮度创造一种颜色。用这种模型在背景内一个纯灰色的区域内描绘将不产生改变。
Color	用光源颜色的色调和饱和度，背景色的亮度创造一种颜色。它保持背景色的灰度，并且在给单色图像着色或者给彩色图像着色的时候是很有效的。
Luminosity	用光源颜色的亮度，背景颜色的色调和饱和度创造一种颜色。这产生与颜色模型相反的效果。

注意：一个附加的标准混合模型，*Compatible*，是早先设计的痕迹，并且不再需要的，但仍然为了兼容性而保留的。它的效果就相当于正常混合模型。看 536 页“不透明叠加的兼容性”将做进一步讨论。

7.2.5 Alpha 的阐述

颜色复合公式

$$C_r = \left(1 - \frac{\alpha_s}{\alpha_r}\right) \times C_b + \frac{\alpha_s}{\alpha_r} \times [(1 - \alpha_b) \times C_s + \alpha_b \times B(C_b, C_s)]$$

产生一个结果色，改色是背景色，光源颜色以及混合 $B(C_b, C_s)$ 的加权平均。权重是由背景色的 α_b 和 α_s 决定的。对于最简单的混合模型，正常型，由以定义：

$$B(C_b, C_s) = C_s$$

复合公式退化成一种简单的背景色于光源颜色的加权平均，这是由背景和光源的 α 数值控制的。对于更多的有趣的混合函数，背景和光源 α 数值控制着是否混合模型的效果完全被呈现或者被背景色与光源颜色的混合结果所淡化。

结果的 α_{r} , α_{s} , 实际上是一个计算后的结果, 在下面 7.2.6 小节“形状和不透明性计算”中描述。结果颜色被结果的 α_{r} 正常化处理, 将确保当这颜色和 α_{r} 是接连用起来, 在另一个复合计算中使用, 颜色的贡献值是正确反映的。请注意如果 α_{r} 是 0, 那么结果是未定义的。

上面所示的公式是下面公式的简化, 下面的公式以更直接的方式代表了背景, 光源和混合颜色的相对贡献度:

$$\alpha_{\text{r}} \times C_{\text{r}} = [(1 - \alpha_{\text{s}}) \times \alpha_{\text{b}} \times C_{\text{b}}] + [(1 - \alpha_{\text{b}}) \times \alpha_{\text{s}} \times C_{\text{s}}] + [\alpha_{\text{b}} \times \alpha_{\text{s}} \times B(C_{\text{b}}, C_{\text{s}})]$$

(这简化需要进行一个根据 α_{r} 复合公式的替代, 我们将在随后的小节展示出来。) 因此, 很数学化地, 当他们对混合函数的影响产生控制的时候, 背景和光源 α_{r} 分别控制了背景色和光源颜色的影响。一个 α_{r} 数值 $\alpha_{\text{s}} = 0.0$ 或者 $\alpha_{\text{b}} = 0.0$ 将导致没有混合模型的效果; 设 $\alpha_{\text{s}} = 1.0$ 且 $\alpha_{\text{b}} = 1.0$ 将导致混合模型的最大效果。

7.2.6 形状和不透明性计算

就像前面说的那样, 控制复合过程的 α_{r} 数值定义为形状和不透明性乘积:

$$\alpha_{\text{b}} = f_{\text{b}} \times q_{\text{b}}$$

$$\alpha_{\text{r}} = f_{\text{r}} \times q_{\text{r}}$$

$$\alpha_{\text{s}} = f_{\text{s}} \times q_{\text{s}}$$

本节独立检测了各种形状和不透明性数值。请再一次记住, 在概念上这些数值是为了页内的每个点而计算的。

光源形状和不透明性

形状和不透明性数值可以来自许多光源。透明模型为每个提供了独立的三个光源。然而, PDF 表象在区分所有独立光源的能力上有一定的限制性 (看 7.5.3 小节, “区分形状和不透明性”)。

- 对象形状。初级的形状如线条, 填充以及文本有一个固有的形状, 1.0 表示点在对象内, 0.0 表示点在对象外。类似的, 一个有明确掩码的图像 (看 321 页

的“明掩码”）有一个形状，1.0 表示在未盖住的部分，0.0 表示在盖住的部分。一个组对象的形状是它所含有的对象的形状的联合。

注意：精确地说，初级对象有“实”边，每个点有一个 0.0 或者 1.0 的形状数值。然而，当这些对象被光栅化成图像像素，边界的形状数值也许进行反锯齿化，分数值代表了这些像素的覆盖比例。当这样的反锯齿化进行时，将覆盖度当做形状对待而不是不透明性，这点非常重要。

- 掩码形状。用于复合一个对象的形状数值可以从独立于它自身的一个附加源或者从软掩码获得。（看 7.4 小节，“软掩码”将讨论这些掩码是如何产生的。）应用一个软掩码去修改对象或者组的形状，称之为软裁剪，可以产生在小场景中如同对象和它背景之间逐渐过渡的效果。
- 常量形状。光源的形状的每一个点都可以使用一个标量的形状常数去修改。这仅仅是为了方便，因此相同的效果可以使用一个形状掩码获得，这些形状掩码的数值处处相同。
- 对象不透明性。初级的对象处处的不透明性都是 1.0。一个组对象的不透明性是它所含有的所有对象的不透明性的计算的结果。
- 掩码不透明性。不透明性数值，如同形状数值一样，可以有独立于将要复合的对象之外的一个软掩码提供。
- 常量不透明性。光源的不透明性可以在每个点修改，修改使用的是一个标量的不透明性常数。考虑到这数值作为“当前不透明性”，类似于先前描绘初级对象使用过的当前颜色一样，是很有用处的。

所有这些形状和不透明性输入从 0.0 一直到 1.0，默认值是 1.0。这样做的意图是当它趋向于 0.0 时，任何的输入可以使得描述的操作更加透明。如果超过 1 个输入趋向于 0.0，那么效果也是复合的。从数学上说，这是简单地将每个类型的三个输入相乘取得的，产生的中间值被称为光源形状和光源不透明性：

$$fs = f_j \times f_m \times f_k$$

$$qs = q_j \times q_m \times q_k$$

这些变量的意义在表 7.4 给出。

表 7.4 在光源形状和不透明性公式中使用的变量

变量	意义
f_s	光源形状
f_j	对象形状
f_m	掩码形状
f_k	常量形状
q_s	光源不透明性
q_j	对象不透明性
q_m	掩码不透明性
q_k	常量不透明性

注意：当一个对象以一种瓦片形式描绘出来时，对象形状和对象不透明性对于对象内部的点来说是由在这种形式内相应的点决定的，而不是处处是 1.0（看 7.5.6 小节，“形式与透明度”）。

结果形状和不透明性

除了一个结果颜色，描绘操作也计算一个相关结果形状和结果不透明性。这些计算式基于联合函数

$$\begin{aligned}\text{Union}(b, s) &= 1 - [(1 - b) \times (1 - s)] \\ &= b + s - (b \times s)\end{aligned}$$

这里， b 和 s 是将要复合的背景和光源的数值。这是不透明形状联合的普遍的传统概念，并且这个概念可以被认为是一个“反相乘法运算”——一个输入和输出互补的乘法运算。该结果趋向于 1.0：如果任意一个输入时 1.0，那么结果就是 1.0。

结果形状和不透明性在下面给出

$$f_r = \text{Union}(f_b, f_s)$$

$$q_r = \frac{\text{Union}(f_b \times q_b, f_s \times q_s)}{f_r}$$

这里的变量的意义在表 7.5 给出。

表 7.5 结果形状和不透明性公式中的变量

变量	意义
f_r	结果形状
f_b	背景形状
f_s	光源形状
q_r	结果不透明性
q_b	背景不透明性
q_s	光源不透明性

这些公式可以由下面来解释：

- 结果形状仅是背景形状和光源形状的联合体。
- 结果不透明性是背景不透明性和光源不透明性的联合体，由它们各自形状进行加权。结果随后由结果形状规范化，确保当这个形状和不透明性接下去在其他复合操作中应用时，它的不透明性的贡献度被正确体现出来。

由于 α 仅是形状和不透明性的乘积，它可以很容易地表示成

$$\alpha_r = \text{Union}(a_b, a_s)$$

无论何时独立的形状和不透明度结果不被需要，这个公式都可以用。

7.2.7 基本复合计算的总结

下面是有所本节所展示的计算的总结。在计算前，它们以一定的顺序给出，并且没有变量被使用到；同样，公式中的一些已经重新排列来简化它们。看上面的表 7.1, 7.4 和 7.5，寻找这些公式中用到的变量的意义。

$$\begin{aligned}\text{Union}(b, s) &= 1 - [(1 - b) \times (1 - s)] \\ &= b + s - (b \times s)\end{aligned}$$

$$\begin{aligned}f_s &= f_j \times f_m \times f_k \\ q_s &= q_j \times q_m \times q_k \\ f_r &= \text{Union}(f_b, f_s)\end{aligned}$$

$$\begin{aligned}\alpha_b &= f_b \times q_b \\ \alpha_s &= f_s \times q_s \\ \alpha_r &= \text{Union}(\alpha_b, \alpha_s)\end{aligned}$$

$$q_r = \frac{\alpha_r}{f_r}$$

$$C_r = \left(1 - \frac{\alpha_s}{\alpha_r}\right) \times C_b + \frac{\alpha_s}{\alpha_r} \times [(1 - \alpha_b) \times C_s + \alpha_b \times B(C_b, C_s)]$$

7.3、透明组

一个透明组是透明栈中一串连续的对象，这些对象收集起来并复合在每个点产生一个单一的颜色，形状和不透明性。这结果被看做是好像它是一个单一的进行连续复合操作的对象。这有利于创造独立的艺术品，每个艺术品是由多个对象创造的，然后将它们联合起来，也许在联合期间有附加的透明效果的应用。组可以嵌套进其他组去形成一个树形层次组结构。

一个组内含有的对象被当做可分离的透明栈，称之为组栈。栈中的组于一些初始化的背景复合（稍后讨论），产生一个复合颜色，形状和不透明性，并把该组作为一个整体。结果是一个对象，它的形状是它内部对象的形状的联合，并且它的颜色和不透明性也是复合操作的结果。这对象然后与组的背景以一种常规的方式复合。

除了它的计算的颜色，形状以及不透明性，组作为一个整体也有许多特性：

- 当组和它的背景复合时，所有影响独立对象的复合计算的输入变量也可以被应用。这些变量包括掩码和常量形状，掩码和常量不透明性，以及混合模式。
- 组可以被孤立或者不被孤立来决定它的哪个栈和初始的背景复合。

- 组可以被推倒或者不被推倒来决定是否它栈内的对象和另一复合或者仅仅和组的背景复合。
- 一个孤立的组可以指定它自己的混合颜色空间，独立于这个组的背景。
- 与被复合进当前页不同的是，一个组的结果可以被用作一个光源的形状或者不透明性值来创造一个软掩码（看 7.4 小节，“软掩码”）下一个小节将介绍一些处理组复合的符号。随后的章节描述一个非孤立，非推倒组的复合公式以及孤立和推倒组的特殊性质。

7.3.1 组复合计算的符号

因为我们同时处理多个对象，所有用一些符号区分它们是很有用处的。相应地，前面介绍的变量改变为包含一个两层下标，这些下标决定一个对象的再透明栈的位置。因此，例如， C_{si} 代表地栈中第 i 个对象的光源颜色。下标 0 表示初始的背景。下标从 1 到 n 表示在一个 n 个元素的栈中最底到最顶的对象。另外，下标 b 和 r 是从变量 $Cb, fb, qb, ab, Cr, fr, qr$ 以及 ar 上丢掉；其他变量保留它们的数字下标。

这些约定允许复合公式将栈内的元素看成可递推的。例如，对象 i 的颜色复合计算的结果由 C_i 表示（以前是 Cr ）。这个计算式采用它一个投入直接背景的颜色，这个颜色对象 $i-1$ 的颜色复合计算的结果；这个结果由 C_{i-1} 表示（以前是 Cb ）。

为了一个简单的 n 元素栈而修改的公式（不包括任何组）是，从 $i=1, \dots, n$ ：

$$f_{s_i} = f_{j_i} \times f_{m_i} \times f_{k_i}$$
$$q_{s_i} = q_{j_i} \times q_{m_i} \times q_{k_i}$$

$$\alpha_{s_i} = f_{s_i} \times q_{s_i}$$
$$\alpha_i = \text{Union}(\alpha_{i-1}, \alpha_{s_i})$$

$$f_i = \text{Union}(f_{i-1}, f_{s_i})$$
$$q_i = \frac{\alpha_i}{f_i}$$

$$C_i = \left(1 - \frac{\alpha_{s_i}}{\alpha_i}\right) \times C_{i-1} + \frac{\alpha_{s_i}}{\alpha_i} \times [(1 - \alpha_{i-1}) \times C_{s_i} + \alpha_{i-1} \times B_i(C_{i-1}, C_{s_i})]$$

这里的变量的意义将在表 7.6 中给出。将这些公式于 7.2.7 小节“基本复合计算的总结“所示的公式进行比较。

表 7.6 基本复合公式修改的变量

变量	意义
fsi	对象 i 的光源形状
fji	对象 i 的对象形状
fmi	对象 i 的掩码形状
fki	对象 i 的常量形状
fi	对象 i 的复合后的结果形状
qsi	对象 i 的光源不透明性
qji	对象 i 的对象不透明性
qmi	对象 i 的掩码不透明性
qli	对象 i 的常量不透明性
qi	对象 i 复合计算后的结果不透明性
asi	对象 i 的光源 alpha
ai	对象 i 复合计算后的结果 alpha
Csi	对象 i 的光源颜色
Ci	对象 i 复合运算后的结构颜色
Bi(Ci-1, Csi)	对象 i 的混合函数

7.3.2 组结构和术语

正如前面所说的，一个组的元素被当做一个分离的透明栈，也就是组栈。这些对象与一个选定的初始背景复合（下面将描述这个背景），并且复合的结果颜

色, 形状以及不透明性被当做属于一个单个对象。得到的结果对象和组的背景以常规的方式复合。

这计算将栈解释成一棵树型。对于栈内从 i 开始的一个 n 个元素的组, 它将随后的 n 个对象看成一个 n 元素的子栈, 这个子栈的元素以独立的数字 1 到 n 给出。这些对象随后从父栈的对象编号中移除并被组对象替换, 从 i 开始编号, 随后剩下的对象描绘在组的上面, 从 $i+1$ 开始编号。这个操作递归地应用到任何嵌套的子组中。因此, 该元素 (用 E_i 表示) 指的是一些组中的一个成员。

从一个特定元素在一个嵌套组的观点来说, 有三种不同的有趣的背景:

- 组背景是所有元素复合的结果, 不包括组内第一个元素。(这个定义将被改变, 如果父组是一个推倒组; 看 7.3.5 小节, “推倒组”)
- 初始背景是一个背景, 这种背景是选择来为了与组的第一个元素进行复合的。这也和组背景一样 (对于非独立组) 或者一个全透明背景 (对于一个孤立组)。
- 中间背景是当前元素之前, 前面所有元素的复合的结果。

当组内的所有元素都已经复合之后, 得出的结果将被看出类似组是一个单一的对象一样, 这个结果随后将和组的背景复合。(这个操作都会出现, 无论为了复合组的元素所选的初始化背景是一个组背景还是一个透明背景。这里有一个特别的纠正, 就是背景对整体的贡献度只被应用一次。)

7.3.3 组复合计算

一个组的颜色和不透明性是由组复合函数定义的:

$$\langle C, f, \alpha \rangle = \text{Composite}(C_0, \alpha_0, G)$$

这些变量的意义在表 7.7 中给出。

表 7.7 组复合函数的参数和结果

变量	意义
G	透明组: 一个由组内所有元素 E_1, \dots, E_n 组成的复合对象—— n 个所构成对象的颜色, 形状, 不透明性和混合模型。
C_0	组的背景颜色

c	组的计算的颜色，当组被看做一个对象时将被用做光源颜色
f	组的计算的形状，当组被看做一个对象时将被用做对象形状
a _o	组的背景的 alpha
a	组的计算的 alpha，当组被看做一个对象时将被用做对象 alpha

请注意不透明性不是直接作为一个参数或者这个函数的结果来给出的。几乎所有的计算使用形状和不透明性(alpha)的乘积。, 而不是单独使用不透明性；因此，通常很方便的直接用形状和 alpha 来工作，而不是形状和不透明性。当需要时，不透明性可以由 uige 相关的形状除以 alpha 来计算得到。

应用组复合函数的结果将被看成好像这个结果是一个单一的对象，这个对象接着根据常规与组的背景复合。在这些公式中，颜色，形状和 alpha(C, f 和 a)是由正在使用的组复合函数计算得到的，C_s, f_j, a_j 分别表示光源颜色，对象形状和对象 alpha。

非孤立，非推倒的组的复合公式如下所定义：

- 初始化：

$$f_{g_0} = \alpha_{g_0} = 0.0$$

- 每个组元素 E_i (i = 1, ..., n) :

$$\langle C_{s_i}, f_{j_i}, \alpha_{j_i} \rangle = \begin{cases} \text{Composite}(C_{i-1}, \alpha_{i-1}, E_i) & \text{if } E_i \text{ is a group} \\ \text{intrinsic color, shape, and (shape} \times \text{opacity) of } E_i & \text{otherwise} \end{cases}$$

$$f_{s_i} = f_{j_i} \times f_{m_i} \times f_{k_i}$$

$$\alpha_{s_i} = \alpha_{j_i} \times (f_{m_i} \times q_{m_i}) \times (f_{k_i} \times q_{k_i})$$

$$f_{g_i} = \text{Union}(f_{g_{i-1}}, f_{s_i})$$

$$\alpha_{g_i} = \text{Union}(\alpha_{g_{i-1}}, \alpha_{s_i})$$

$$\alpha_i = \text{Union}(\alpha_0, \alpha_{g_i})$$

$$C_i = \left(1 - \frac{\alpha_{s_i}}{\alpha_i}\right) \times C_{i-1} + \frac{\alpha_{s_i}}{\alpha_i} \times ((1 - \alpha_{i-1}) \times C_{s_i} + \alpha_{i-1} \times B_i(C_{i-1}, C_{s_i}))$$

- 每个组：

$$C = C_n + (C_n - C_0) \times \left(\frac{\alpha_0}{\alpha_{g_n}} - \alpha_0 \right)$$

$$f = f_{g_n}$$

$$\alpha = \alpha_{g_n}$$

这些变量的意义在表 7.8 展示出来（除了这些变量，其他的在上面的 7.7）

一个元素 E_i 是一个初级对象，颜色，形状和 alpha 值 C_{si}, f_{ji}, a_{ji} 是这个对象的内在属性。对于一个元素就是一个组的情况，那么组的复合函数将递归地应用到子组中，并且得到的 C, f 和 a 值被当做在父组计算中的代表它自身的 C_{si}, f_{ji}, a_{ji} 。

表 7.8 在组复合公式中用到的变量

变量	意义
E_i	组内的元素 i :一个复合变量，代表元素的颜色，形状，不透明性以及混合模型
f_{si}	元素 E_i 的光源形状
f_{ji}	元素 E_i 的对象形状
f_{mi}	元素 E_i 的掩码形状
f_{ki}	元素 E_i 的常量形状
f_{gi}	组形状：组元素 E_1 到 E_i 累计的光源形状，除了初始背景
q_{mi}	元素 E_i 的掩码不透明性
q_{ki}	元素 E_i 的常量不透明性
a_{si}	元素 E_i 的光源 alpha
a_{ji}	元素 E_i 的对象 alpha: 它的对象形状和对象不透明性的乘积
a_{gi}	组 alpha: 组元素 E_1 到 E_i 累计的光源 alpha, 除了初始背景
a_i	在复合了元素 E_i 之后累计的 alpha, 包括初始背景
C_{si}	元素 E_i 光源颜色
C_i	在复合了元素 E_i 之后累计的颜色，包括初始背景
$B_i(C_{i-1}, C_{si})$	元素 E_i 的混合函数

请注意，一个组的元素是复合到一个背景中，这背景包括了组的初始背景。当它完成后得到正确的混合模型效果，这些效果中的大部分都独立于参与混合的背景和光源颜色（这个特性是区别非孤立组合孤立组的，将在后面章节讨论。）最后这些公式特别注意，它们计算组复合函数的 C, f, a 最终结果。本质上，这些公式从计算结果中移除了组背景的贡献度。这确保了当组和背景连续复合时（很可能有附加形状或者不透明性输入或者一个不同的混合模型），背景的贡献度仅被包含一次。

对于颜色，背景移动由明确的计算完成，这背景的效果实际上是正常混合模型复合的逆效果。这公式是下面公式的简化，这些公式更直观地代表了这个操作：

$$\phi_b = \frac{(1 - \alpha_{g_n}) \times \alpha_0}{\text{Union}(\alpha_0, \alpha_{g_n})}$$
$$C = \frac{C_n - \phi_b \times C_0}{1 - \phi_b}$$

其中 ϕ_b 是背景分数，是背景颜色对于整体颜色的相对贡献度。

对于形状和 alpha，背景移动式由维持两个变量集来保持累计值而完成的。组的形状和 alpha, fgi 和 agi，仅累计组元素的形状和 alpha，不包括组背景。它们的最终值成为组结果，由组复合函数返回。完整的 alpha, ai，也包括背景贡献度；它的数值被用在颜色复合计算中。（从没有任何需要计算相应的完整形状，fi，包括背景贡献度。）

作为这些更正的结果，复合对象作为一个组的效果就和单独复合它们（没有组）一样，如果下面的条件成立：

- 该组是一个非孤立组，并且和它的父组一样有推倒属性（看 7.3.4 小节，“孤立组”以及 7.3.5 “推倒组”）。

- 当组的结果和组背景复合，正常混合模型将被使用，并且形状和不透明性输入总是 1.0。

7.3.4 孤立组

一个孤立组，它的元素是复合到一个完全透明的初始背景中而不是复合到组的背景中的。组的结果的光源颜色，对象形状，和对象 alpha 因此是独立于组的背景的。它和组背景的唯一交互发生在当组的计算的颜色，形状，alpha 和它复合的时候。特别地，组内的对象的混合模型产生的特殊效果不仅仅顾及到这些对象的内在颜色，不透明性；它们不被组的背景所影响。举个例子，应用乘混合模型到一个组内对象，将在组的栈中更低的其他对象上产生一个更深的效果，当这个效果不是在组的背景上。

板 17 展示了一个包含 4 个浅灰色重叠圆的组的效果 ($C=M=Y=0.0; K=0.15$)。不透明性为 1.0 的组内这些圆用乘混合模型描绘；这个组自身应用正常混合模型与它的背景描绘出来。在定行，组是孤立的，这样不会与彩虹背景相互作用。在底行，组是非孤立的，并且和背景相复合。该板也展示了推倒和非推倒组的不同（看 7.3.5 小节，“推倒组”）。

一个孤立组的效果可以用一个简单对象来呈现，这个简单对象直接地指定每个点的一个颜色，形状和不透明性。一个孤立组的去光泽化有时候对于应用程序中输入和输出完整的复合艺术品是很有用的。

对于一个孤立组，组的复合公式被简单地加入一个声明道初始化中而发生改变：

$a_0 = 0.0$ 如果组是孤立的

也就是说，组的元素所复合的初始背景是透明的，而不是从组的背景继承而来。这种替代也使得 C_0 成为未定义的，但是正常的复合公式考虑了这个因素。同时， C 复合的结果自动地简化成 $C = C_n$ ，因此没有任何背景贡献度将被分解出来。

7.3.5 推倒组

在一个推倒组中，每个独立元素是和组的初始背景复合的，而不是和正在使用元素的组所在的栈。当对象有二进制形状（1.0 代表在内部，0.0 代表在外部），每个对象重写（推倒）那些组内的更早之前的它重叠的效果。在任意给定的点，是有最顶的对象将点贡献到结果颜色和不透明性考虑在内作为一个整体。

图板 17，已经在上面 7.3.4 章节“孤立组”讨论的，展示了推倒组和非推倒组的不同。在左边栏，这四个重叠圆被定义成一个推倒组，因此不需要和各自组内复合。在右边栏，这些圆形成一个非推倒组，因此，不合各自复合。在每个栏，更高和更低的图标分别表述了一个孤立组合一个非孤立组。

这个模型类似于不透明成像模型，除了“最顶对象胜利”原则应用颜色和不透明性。推动啊组在从一个重叠对象中收集一个艺术品并进行复合时时很有用的，组内的对上面的任何重叠对象完全模糊了那些在它之下的对象。同时，最顶的对象和组的初始背景交互是以一种常规的方式，交互时恰当地将不透明性和混合模型应用起来。

推倒的概念将推广到调节微量的形状值。在这种情况下，中间背景仅是部分推倒的，被对象和初始背景复合的结果的一部分取代。重申的组复合公式处理推倒组，采用的是引进一个新变量， b ，这个变量时一个下标，它指定了之前的哪个结果在复合计算中使用背景：0 在一个推倒组中或者 $i-1$ 在一个非推倒组中。当 $b=i-1$ 的时候，公式简化成 7.3.3 小节，“组复合计算”所给出的。

通常情况，计算需要进行两个阶段：

将对象与组的初始背景进行复合，不论背景的形状，并且使用除处处一个 1.0 的

光源形状值。这将产生一个暂时未规范化的 α 和颜色结果， at 和 Ct 。（对于颜色，这个计算本质上是和 7.2.5 小节“ α 的解析”中非简化的颜色复合公式

是一样的，但是使用的光源形状是 1.0 的）

$$\alpha_t = \text{Union}(\alpha_{g_b}, q_{s_i})$$

$$C_t = (1 - q_{s_i}) \times \alpha_b \times C_b + q_{s_i} \times ((1 - \alpha_b) \times C_{s_i} + \alpha_b \times B_i(C_b, C_{s_i}))$$

将上面的结果和对象的中间背景来计算一个加权平均，使用光源形状作为权

重因子。然后用结果 α_{alpha} 规范化结果颜色：

$$\begin{aligned}\alpha_{g_i} &= (1 - f_{s_i}) \times \alpha_{g_{i-1}} + f_{s_i} \times \alpha_t \\ \alpha_i &= \text{Union}(\alpha_0, \alpha_{g_i}) \\ C_i &= \frac{(1 - f_{s_i}) \times \alpha_{i-1} \times C_{i-1} + f_{s_i} \times C_t}{\alpha_i}\end{aligned}$$

进行这个平均计算是为了颜色和 α_{alpha} 。上面展示的这些公式直接说明了这个平均。7.3.7 小节，“组复合计算的总结”中的公式，将被稍微的修改，以便使用光源形状和 α_{alpha} ，而不是光源形状和不透明性。这避免了计算光源不透明性值的需求。（请注意 C_t ，这里稍微喝上面 C_t 的不同：它是被 f_{s_i} 预乘的。）

光源形状的极限值产生了直接推倒的效果。也就是说，一个 1.0 的形状值（在内部）产生了颜色和不透明性，这颜色和不透明性是对象和初始背景复合而来的。在一个 0.0 形状值（外部）保持前面的组的结果不改变。推倒特性的存在是保持一个分离的形状值而不是仅仅一个单一的联合形状和不透明性的 α_{alpha} 主要原因。这分离的形状值在任何组都必须被计算，然后接下来呗当做一个推倒组的元素使用。

一个推倒组可以是孤立的或者非孤立的。也就是说，孤立和推倒是独立属性。一个非孤立且推倒组复合组背景与它的最顶的元素。一个孤立且推倒组与透明背景复合。

请注意：当一个非孤立组嵌套进一个推倒组时，在内的组的初始背景和在外的组的初始背景是一样的；这不是在内组的中间背景。这种行为，虽然也许是不希望看到的，但是是当 $b=0$ 时组复合公式的一种后果。

7.3.6 页组

所有被直接描绘在一个页的元素—顶层组合顶层对象都不是任何组的一部分—将被看成是好像它们被涵盖在一个透明组 P 中，这个透明组反过来和一个内容独立的背景复合。这个组称之为页组。

页组可以被看成两种截然不同的方式：

- 一般来说，该页直接强加于一个输出介质，例如纸或者显示屏。页组可以看成一个孤立组，这个孤立组然后与介质的一个背景色相复合。背景色是正常的白色，虽然根据实际介质会有所不同。然而，某些应用程序也许会选择区提供一个不同的背景，例如一个检查板或者珊格，来帮助虚拟化这些艺术品上的透明效果。
- 一个 PDF 文件的“页”可以被看成一个图像对象，这个对象被当成其他文档的一个页的元素使用。这种情况的出现，打个比方，当使用 Illustrator 将一个包含艺术品的 PDF 文件放进一个由 InDesign 产生的页面底板中。在这种情况下，PDF “页”不和媒介颜色复合；相反，它被当成一个普通透明组，这个透明组可以是孤立的也可以是非孤立的，并且和它的背景以正常的方式复合。

本章节剩下部分仅属于第一次页组的使用，它可以直接强加于媒介中、页给定的点的颜色 C 由一个简化的常规的组复合公式所定义：

$$\langle C_g, f_g, \alpha_g \rangle = \text{Composite}(U, 0, P)$$

$$C = (1 - \alpha_g) \times W + \alpha_g \times C_g$$

这些变量的意思在表 7.9 中给出。第一个公式计算组和一个给定的透明背景的颜色和 alpha—实际上，将 P 看成一个孤立组。第二个公式将内容独立的背景相复合（使用和正常混合模型相同）。

表 7.9 页组复合公式中的变量

变量	意义
P	页组，在页的顶层栈包含所有的元素 E_1, \dots, E_n
C_g	页组的计算的颜色
f_g	页组的计算的形状
α_g	页组的计算的 alpha
C	页的计算的颜色
W	页的初始颜色（通常白色，但是也会根据应用程序所需的媒介的不同而改变）
U	一种未定义的颜色（它不被使用，因为 a_0 参数的复合是 0）

如果没有特别的说明，页组的颜色空间是从输出设备的本地颜色空间中继承的——也就是说，一个设备的颜色空间，例如 DeviceRGBhuozhe DeviceCMYK。通常更好的做法是指定一个明确的颜色空间，特别是基于 CIE 的空间，以确保页组内复合计算有更多可预料的结果。在这种情况下，所有页级的复合将在特定的颜色空间内复合，然后整个结果在于内容独立的背景复合之前转换为输出设备的本地颜色空间。

在类似 PostScript 的不透明成像模型中，当页实际上不是被渲染的时候，但是被转换成一个去光泽化的表象这种情况也可以遇到。

7.3.7、组复合计算的总结

下面对组复合公式的重述也可以将孤立组和推倒组计算在内。看 503 页和 505 页的表格 7.7 和 7.8，查找这些变量的意义。

$$\langle C, f, \alpha \rangle = \text{Composite}(C_0, \alpha_0, G)$$

- 初始化：

$$\begin{aligned} fg0 &= ag0 = 0 \\ a0 &= 0 \quad \text{如果组是孤立的} \end{aligned}$$

- 对于每个组元素 $E_i \in G (i = 1, \dots, n)$ ：

$$\begin{aligned} b &= \begin{cases} 0 & \text{if the group is knockout} \\ i-1 & \text{otherwise} \end{cases} \\ \langle C_{s_i}, f_{j_i}, \alpha_{j_i} \rangle &= \begin{cases} \text{Composite}(C_b, \alpha_b, E_i) & \text{if } E_i \text{ is a group} \\ \text{intrinsic color, shape, and (shape} \times \text{opacity) of } E_i & \text{otherwise} \end{cases} \\ f_{s_i} &= f_{j_i} \times f_{m_i} \times f_{k_i} \\ \alpha_{s_i} &= \alpha_{j_i} \times (f_{m_i} \times q_{m_i}) \times (f_{k_i} \times q_{k_i}) \\ f_{g_i} &= \text{Union}(f_{g_{i-1}}, f_{s_i}) \\ \alpha_{g_i} &= (1 - f_{s_i}) \times \alpha_{g_{i-1}} + (f_{s_i} - \alpha_{s_i}) \times \alpha_{g_b} + \alpha_{s_i} \\ \alpha_i &= \text{Union}(\alpha_0, \alpha_{g_i}) \\ C_t &= (f_{s_i} - \alpha_{s_i}) \times \alpha_b \times C_b + \alpha_{s_i} \times ((1 - \alpha_b) \times C_{s_i} + \alpha_b \times B_i(C_b, C_{s_i})) \end{aligned}$$

$$C_i = \frac{(1 - f_{s_i}) \times \alpha_{i-1} \times C_{i-1} + C_t}{\alpha_i}$$

- 结果：

$$\begin{aligned}C &= C_n + (C_n - C_0) \times \left(\frac{\alpha_0}{\alpha_{g_n}} - \alpha_0 \right) \\f &= f_{g_n} \\ \alpha &= \alpha_{g_n}\end{aligned}$$

注意：再一次，记住这些公式是它们最常见的形式。当某些光源的形状和不透明性不被呈现或者当形状和不透明性不需要分别维持的时候，它们可以简化。更进一步说，在每个指定类型的组（孤立或非孤立，推倒或非推倒），这些公式的某些项可以去掉或者扔掉。一个有效的实现应使用简化的设备公式。

7.4、软掩码

正如之前所说的，在复合公式中的形状和不透明性值可以包含一些叫做掩码形状(fm)的组件以及掩码不透明性(qm)，这些组件是从对象的原独立性中来的。这样一个独立源，称之为软掩码。它定义一些可以在页面不同点更改的数值。软这个词强调了掩码的值在给出的特点点上是不受0.0或者1.0限制的，但是也可以取这中间分数值。这样一个掩码是唯一提供位置独立的不透明性值方法，因为初级对象在内部不含有不透明性。

一个掩码看成一个源的形状值，也可以被称为软裁剪，类似的不透明成像模型路径的“硬”裁剪（看4.4.3小节，“裁剪路径操作”）。软裁剪是和硬裁剪相对的：一个硬裁剪可以看成软裁剪，这软裁剪的形状值是1.0内部和0.0外部的裁剪路径。在硬裁剪路径内的每处，光源对象的颜色代替了背景色。外部的每处，背景不改变。对于软裁剪，对比来看，一个对象和背景之间创造一个改变，正如在插图内。

一个掩码可以用创造一个透明组并将对象描绘进去来定义，接着以常规方

定义颜色，形状以及不透明性。结果组然后可以被用于两种派生掩码的方式：正如后面章节描述的那样。

7.4.1 从组 α 派生一个软掩码

在定义一个软掩码的第一种方法中，一个透明组 G 的颜色，形状以及不透明性首先是由常规的公式计算

$$\langle C, f, \alpha \rangle = \text{Composite}(C_0, \alpha_0, G)$$

这里， C_0 和 α_0 代表一个任意的背景，这个背景的值对最后的结果没有贡献。 C , f 和 α 结果是组的颜色，形状以及 α ，背景不考虑在内。每个点的掩码的值是从组的 α 中来的。因为组的颜色不被用于这种情况，那么就不需要去计算它。 α 值被传递给一个分离的指定的传输函数，允许出现掩盖效果。

7.4.2 从组亮度得到软掩码

从透明组中获得软掩码的第二种方法一组和一个完全不透明的背景复合开始。掩码值在任何给定的点呗定义成结果颜色的亮度。这允许掩码从一个任意艺术品的形状和颜色中获取。

颜色 C 被用于从一个组 G 创造掩码，这个过程被定义为

$$\begin{aligned}\langle C_g, f_g, \alpha_g \rangle &= \text{Composite}(\dot{C}_0, 1, G) \\ C &= (1 - \alpha_g) \times C_0 + \alpha_g \times C_g\end{aligned}$$

这里 C_0 是选的的背景颜色。 G 可以是任何类型的组—孤立或者非孤立，推倒或者非推倒—在 C 产生各种效果。颜色 C 然后用以下的方式之一被转换成亮度，转换根据组的颜色空间：

- 对于基于 CIE 的空间，转换成 CIE1931XYZ 空间以及使用 Y 组件作为亮度。这产生一个色度地纠正亮度。在一个 PDF CaIRGB 空间的例子中，公式是

$$Y = Y_A \times A^{G_R} + Y_B \times B^{G_G} + Y_C \times C^{G_B}$$

使用颜色空间词典（看 218 页表 4.14）Gamma 和 Matrix 条目的组件。一个类似的计算运用到其他基于 CIE 的颜色空间。

- 对于设备颜色空间，使用设备独立性方法和使用结果灰度值当做亮度，以此来转换颜色到 DeviceGray，并且没有其他颜色校正或者补偿。这个方法不依赖于比

色分析的正确性；它仅仅提供一个数字上的简单的方法，来产生续音掩码值。这里有一些推荐的公式用于从 DeviceRGB 和 DeviceCMYK 的转换：

$$Y = 0.30 \times R + 0.59 \times G + 0.11 \times B$$

$$\begin{aligned} Y &= 0.30 \times (1 - C) \times (1 - K) \\ &\quad + 0.59 \times (1 - M) \times (1 - K) \\ &\quad + 0.11 \times (1 - Y) \times (1 - K) \end{aligned}$$

随着这些转换，结果被传递给一个分离的指定的转换函数，然后结果按照掩码效果被指定。

大部分有用的背景色可能是黑色，黑色引起组的形状外的任意区域在结果掩码上有一个 0 的亮度值。如果组的内容被看成一个正掩码，这产生的结果将被预期成关于形状外的点。

7.5 在 PDF 内指定透明度

本章节已经在抽象层面呈现了透明成像模型，但是几乎没有提到它在 PDF 的表象。本小节描述了可用的设备来指定 PDF1.4 中的透明度。

7.5.1 指定源和背景颜色

单一的图像对象，正如 4.1 小节，“图像对象”定义的，将被当成初级对象用于透明度复合应用（文本对象特殊对待，正如 5.2.7 小节，“文本推倒”中描述的）。也就是说，夜歌给定的对象的所有东西将被认为是一个透明栈的一个成员。一个对象的遗留部分将不和另一个复合，即使它们以一种看起来会引起重叠

的方式描述（例如一个自相交路径，联合填充以及一个划线的路径，或者一个包含一个重叠或者重影的避光模式）。一个对象的光源颜色 C_s 在颜色复合公式中使用， C_s 以相同的方式指定，就像在不透明成像模型中；依靠图像中当前颜色或者一个图像中光源例子的方法。背景色 C_b 是之前描绘操作的结果。

7.5.2 指定混合颜色空间和混合模型

混合颜色空间是一个透明组的属性，在这组内一个对象被描绘出来；它的规范在 7.5.5 小结中描述。页作为一个整体，也可以被看成是一个组，页组（看 7.3.6 小节，“页组”），页组的它自己的颜色空间属性。如果没有特别说明，页组的颜色空间是从输出设备的本地颜色空间继承的。

混合模型 $B(C_b, C_s)$ 是由图形状态中的当前混合模型的参数决定的（看 4.3 小节，“图形状态”），它是由图形状态参数字典中的 BM 条目指定的（看 4.3.4，“图形状态参数词典”）。它的值可以是一个名字对象，该对象指定了 491 和 493 页表 7.2 和 7.3 中列出的一个标准混合模型，也可以是一个这类名字的数组。在后面这种情况时，应用程序需要使用在数组内第一个混合模型（或者正常模型，如果它都不能识别后面的）。因此，新的混合模型可以在后续被引进来，并且不能识别它们的应用程序将有合理的后备行为。（看在附录 H 中提到的实现。）

清注意：当前混合模型总是应用于处理颜色组件，但是有时候做点的着色；具体细节看 534 “混合模型和重叠”。

7.5.3 指定形状和不透明性

正如在 495 页“光源形状和不透明性”下讨论的一样，在复合计算中使用的形状 (f) 和不透明性 (q) 值来自各种各样的源：

- 对象内在形状 (f_j) 和不透明性 (q_j) 被复合
- 一个分离形状 (f_m) 或者不透明性 (q_m) 掩码对象自身的独立性
- 一个标量形状 (f_k) 或者不透明性 (q_k) 常量将被加进每个点

下面的小节描述了这些形状和不透明性源中的每个市如何在 PDF 中被指定的。

对象形状和不透明性

一个使用 PDF 描绘操作来描绘的对象的形状值 `fj` 被定义为：

- 对于由一个路径或者一个象形定位的对象，以及以唯一的颜色采用路径描绘或者文本展示操作描绘(4.4.2 小节，“路径描绘操作”，和 5.3.2，“文本展示操作”)，形状总是 1.0 表示内部，0.0 表示外部。
- 对于图像 (4.8 小节，“图像”)，形状总是 1.0 在图像边框内，0.0 在外。这可以在后续被修改成明确的或者颜色关键掩码 (321 页“明确的遮蔽”和 321 页的“颜色关键遮蔽”)。
- 对于图像掩码 (320 页“模板遮蔽”)，形状 1.0 是将被描绘的区域，0.0 是被遮蔽的区域。
- 对于使用瓦片形式描绘的对象 (看 4.6.2，“瓦片形式”) 或者一个阴影模式 (栏 4.6.3，“阴影模型”)，形状将进一步被所定义形式的对象抑制 (看 7.5.6，“形式和透明性”)。
- 对于使用 `sh` 操作来描绘的对象 (273 页“阴影操作”)，形状是 1.0 在阴影描绘的几何学边界内部，0.0 在外部，忽略了阴影词典的背景条目 (看 274 页“阴影词典”)。

所有初级对象有一个内在的处处为 1.0 的不透明性 `qj`。任何低于 1.0 的必须使用一个不透明掩码或者常量来应用，这将在下面小节来描述。

掩码形状和不透明性

大多数掩码输入—称为软掩码，或者 `alpha` 掩码—可以被提供给任何 PDF 复合操作。掩码可以看成形状 (`fm`) 或者不透明性 (`qm`) 的一种源，依据的是图形状态中的 `alpha` 源的设置 (看 4.3 小节，“图形状态”)。这是一个布尔标志位，用图形状态参数词典中的 `AIS` (“`alpha in shape`”) 来设置 (看 4.3.4，“图形状态参数词典”)：`true` 如果软掩码包含形状值，`false` 表示不透明性。

软掩码可以用下面几种方式指定：

- 图形状态的当前软掩码参数，使用图形状态参数词典的 `Smask` 条目，包含一个

软掩码词典（看 520 页“软掩码词典”）定义了掩码的内容。名字 **None** 也可以被指定来代替一个软掩码词典，表示一个软掩码的缺失。在这种情况下，掩码形状和不透明性默认处处为 1.0。（看附录 H 中 71 小点的实现。）

- 一个图像 **XOBJECT** 可以包含它自身的软掩码图片，这是以替代图像 **XOBJECT** 在图像词典中 **Smask** 条目形式的（看 4.8.4，“图像词典”）。这个掩码，如果呈现出来，将忽视任何由图像词典的 **Mask** 条目明确指定或者颜色关键掩码。在图像词典的掩码形式忽视了在图形状态的当前软掩码。（看附录 H 的 72 点的实现）
- 一个 **XOBJECT**，它有一个 **JPXDecode** 过滤器，就像它的数据源可以被指定成一个 **SmaskUnData** 条目，它指示了软掩码是嵌入到数据流中（看 3.3.8 小节，“**JPXDecode** 过滤器”）。

请注意：在图形状态上的当前软掩码是为了被使用来再一个时间裁剪一个单一对象（或者一个初级对象或者一个透明组）。当描绘两个或者更多重叠对象时，如果一个软掩码被应用了，掩码和自身重叠区域的乘积（除了一个推倒组），产生一个结果形状和不透明性，这很可能不是我们所希望得到的，为了应用一个软掩码去乘对象，通常最好去定义对象为一个透明组并且将掩码应用到组，使之成一个整体。这些考虑也可以应用到目前 **alpha** 常量（看随后的小节）。

常量形状和不透明性

在图形状态的当前 **alpha** 常量参数（看 4.3 小节，“图形状态”）指定了两个标量值——一个是为了划线，一个为了所有的其他描绘操作——将被应用为常量形状 (**fk**) 或者常量不透明性 (**qk**) 组件在颜色复合公式中。这个参数也可以被认为是一个和目前颜色相似的，在描绘初级对象时使用的。（注意，无论如何，非划线 **alpha** 常量也是当描绘一个透明组的结果到背景时被应用的；也可以看附录 H 的 71 点的实现）

划线的和非划线的 **alpha** 常量是分别地使用图形状态参数词典（看 4.3.4 小节，“图形状态参数词典”）中的 **CA** 和 **ca** 条目来设置的。正如上面对软掩码的描述一样，在图像状态中的 **alpha** 源标志位决定了是否 **alpha** 常量是被解释成形状值 (**true**) 或者不透明性值 (**false**)。

请注意：在上面“掩码形状和不透明性”后面的提示中，应用当前 *alpha* 常量常数和当前软掩码。

7.5.4 指定软掩码

正如在 518 页“掩码形状和不透明性”下所说的一样，用于复合计算的软掩码可以被以下几种方式之一来指定：

- 在图形状态的当前软掩码参数中的一个软掩码词典；看后面的“软掩码词典”来查看具体细节
- 与一个简单图像相联系的一个软掩码图像；更多细节看 522 页“软掩码图像”
- 一个掩码通道嵌入到 JPEG2000 编码数据 (PDF1.5)；更多细节看 3.3.8 小节，“JPXDecode 过滤器”，以及表 4.39 中的 *SmaskInData* 条目。

软掩码词典

最常见的指定一个软掩码的方式是使用一个软掩码词典来指定，就像图像状态中的当前软掩码（看 4.3 小节，“图形状态”）。表 7.10 展示了这种词典的内容。（看附录 H 的 71 小点的实现）

掩码值是从一个透明组得到的，使用 7.4.1 “从组 *alpha* 中获得一个软掩码”小节和 7.4.2 “从组亮度中获取一个软掩码”中描述的两种方式中的一种。这组是一个透明组 XOBJECT 看定义（看 7.5.5 小节，“透明组 XOBJECT”），这些 XOBJECT 由软掩码词典的 G 条目指定。S (subtype) 条目指定两种方式中的哪一种被使用：

- 如果子类型是 Alpha，那么透明组 XOBJECTG 就等同于仅计算一个组 *alpha*。后续对象的颜色将被忽视，并且对象复合计算将不被进行。转换函数 TR 将被应用于计算组 *alpha* 来产生掩码值。在透明组的边界之外，掩码值是将输入值 0.0 放入转换函数的结果。
- 如果子类型是亮度，透明组 C 对象 G 使用一个完全不透明背景来复合，这背景的颜色处处都是由软掩码词典的 BC 条目定义的。计算的结果颜色将转换成一个单一组件的亮度值，并且转换函数 TR 被应用于这亮度来产生掩码值。在透明

组的边界之外，掩码值是由转换 BC 颜色到亮度和应用转换函数得到的结果所得
到的。

掩码的坐标系是由连结转换矩阵定义的，这矩阵式由透明组形式的 **Matrix** 条目
指定的（看 4.9.1 小节，“组成词典”），目前转换矩阵在当前软掩码被建立在图
形状态上，使用 **gs** 操作。

请注意：在一个透明组 **XOBJECT**，该对象指定一个软掩码，点颜色组件将不再出现，即使在
软掩码曾被使用的组或者页中。如果组 **XOBJECT** 的内容流指定一个分离或者 **DeviceN** 颜色空
间，它将使用点颜色组件，其他颜色空间将被替换（看 234 页的“分离颜色空间”和 238
页的“**DeviceN** 颜色空间”）。

表 7.10 在一个软掩码词典的条目

关键词	类型	值
Type	名字	该词典描述的 PDF 对象的类型；如果呈现出来，必须 为了一个软掩码词典而被覆盖。（可选）
S	名字	一个子类型指定方法来被用于从 G 条目指定的透明组 获取掩码值（必须的）
	Alpha	使用组计算的 alpha，忽视它的颜色（看 7.4.1 小节，“从组 alpha 获得一个软掩 码”）。
	亮度	转换组计算结果到一个单一组件的亮度 值（看 7.4.2 小节，“从组亮度获得一个 软掩码”）
G	流	一个透明组 XOBJECT （看 7.5.5 小节，“透明组 XOBJECT ”） 将被当成源的 alpha 或者颜色值，来获取掩码。如果子 类型 S 是亮度，那么组的属性词典必须包含一个 CS 条 目来定义复合计算使用的颜色空间。（必须的）
BC	数组	一个组件值的数组指定了将被当成背景的颜色，它将与 透明组 XOBJECTG 相复合。这条目仅当子类型 S 是亮度

时被用到。该数组包含 n 个数，当 n 是由组属性词典中 **CS** 条目指定的颜色空间的组件数目（看 7.5.5 小节，“透明组 XOBJECT”）。默认值：颜色空间的初始值，黑色。

TR	函数或者名字	一个函数对象（看 3.9，“函数”）指定转换函数，这函数将被用于获取掩码值。函数接收一个输入，计算组 alpha 或者亮度（根据子类型 S 的值），返回一个输出，得到掩码值。输入和输出都是 0.0 到 1.0；乳沟计算的输出在范围之外，强制到最近的有效点。
-----------	--------	--

软掩码图像

第二种方式定义一个软掩码，是使用一个软掩码图像和一个图像 XOBJECT 联系起来。这是一个辅助的图像 XOBJECT，使用父 XOBJECT 的图像词典中欧冠的 **Smask** 条目来指定（看 4.8.4 小节，“图像词典”；也可以看附录 H 中 72 点的实现）。在辅助图像词典的条目中，一个软掩码图像有相同的格式和一样，就像在普通图像 XOBJECT 中一样（正如 310 页 4.39 表格所描述的），在表 7.11 列出。这种类型的图像词典也可以包含一个附件的条目，**Matte**，稍后讨论。

当一个图像伴随的是一个软掩码图像，它有时在图像数据与背景预混合很有优势，称为雾的颜色。每个图像样本代表和次加权平均原始来源的颜色。（这是一种通常称为预乘法 **alpha** 的技术。）

如果图像数据预混合的，雾色的，必须指定一个图像字典码特进入了软掩码（见表 7.12）。在预均化的计算，表现为每个组件独立：

$$c' = m + \alpha \times (c - m)$$

这里

c' 的是将在图像数据源所提供的值

c 是原始图像组件的值

m 是雾色组成部分的值

α 是相应的掩码采样

请注意：计算使用的是实际颜色成分值，与解码的影响进行了筛选和转换，该计算是一样的，对于颜色的加法或减法。

表 7.11 在软掩码图像词典中的条目的限制

关键词	限制
Type	如果呈现，必须是 XOBJECT。
Subtype	必须是图像。
Width	如果一个雾色条目（表 7.12）被呈现，必须是和父图的宽度值一样；否则将独立于它。所有的图像映射到唯一方形的用户空间（好像是所有图像），无论是否独立一致。
Height	和宽度一样考虑
ColorSpace	必须；必须是 DeviceGray
BitsPerComponent	必须
Intent	忽视
ImageMask	必须是 false 或者缺失
SMask	必须缺失
Smask	必须缺失
Decode	默认值 [01]
Interpolate	可选的
Alternates	忽视
Name	忽视
StructParent	忽视
ID	忽视
OPI	忽视

表 7.12 一个软掩码图像词典的附件条目

关键词	类型	值
雾色	数组	（可选； PDF1.4）一个预混合阵列元件值已被指

定的雾色图像数据与该父形象。该数组由 n 个数，其中 n 是 p 的元件数目入境的色彩在指定的色彩空间

当预混合图像数据的透明度和合成用于混合计算，结果都是一样，如果指定的原始，单料图像数据被用来进行无雾色。特别是，当投入到混合函数的时候。

预混合计算在有父图像的颜色空间条目特别指定的颜色空间中完成。这是组颜色空间独立描绘进入图像的。如果一个颜色的转换被需求，预混合反转必须进行颜色转换。如果图像的颜色空间是一个索引空间（看 232 页“索引颜色空间”），颜色表中的颜色值被预混合（不是索引值自身）。

7.5.5 透明组 XOBJECT

一个透明组被呈现在 PDF 中作为一个特殊类型的组 XOBJECT（看 4.9.2，“组对象”），称之为一个透明组 XOBJECT。一个组 XOBJECT 反过来是 XOBJECT 的类型，不同意于它形式词典中的组条目（看 4.9.1，“形式词典”）。这条目的值是一个辅助组属性词典，定义了组的特性。词典内容的格式和意义决定它的组子类型，在词典的 S 条目指定。透明组的条目（子类型透明度）在表 7.13 中展示。

请注意：一个页面对象（见“页对象”页 119）也有一组项，其值是一组属性字典指定页面组的属性了（见 7.3.6 节，“页组”）。字典的解释有些项目略有不同的一比一的透明度组了 XObject 页面组，看到表上的详细描述

表 7.13 一个透明组属性词典的附加条目

关键词	类型	值
S	名字	（必须）组子类型，标识了组的类型，在词典中描述，对于一个透明组必须是透明。
CS	名字或数组	（有时需要，下文讨论）专案小组色彩空间，它是用于下列用途： <ul style="list-style-type: none"> • 由于色彩空间的颜色转换成组画

- 混合颜色空间的对象是在复合组（见 7.2.3 节，“混合颜色空间”）
- 其背景颜色空间上的群体作为一个整体，当它与背景一起描绘

该组色彩空间可以是任何设备或智库为基础的色彩空间，对待自己的组件作为独立的添加剂或消减值在 1.0 至 0.0 的范围，但对色彩空间的限制 7.2.3 节所述。这些限制排除实验室和亮度，色度 **ICCBased** 颜色，上课，以及特殊色彩空间模式，索引，分离，和 **DeviceN**。设备色彩空间受到重新映射根据 **DefaultGray**，**DefaultRGB** 和资源字典 **DefaultCMYK** 项当前的色彩 **subdictionary** 的（见“默认颜色空间”227 页）。

通常，在 **CS** 条目只被允许用于孤立透明组，并且是可选的。然而，这条目在组属性词典中是必须的。特别地，一个软掩码词典中的 **G** 条目的值是子类型亮度（看 520 页“软掩码词典”）

另外，在组属性词典中指定 **CS** 值是常见的，即使 **I** 是错误的或者缺少了。正常情况下，页直接从输出媒介获取，页组是孤立的，不论 **I** 值。但是如果页被当作其他页的一个元素使用，并且如果组是非孤立的，**CS** 可以忽略并且颜色空间从实际背景继承。（看 7.3.6 “页组”默认值：透明组所描绘的父组或者页的颜色空间。（父的颜色空间也可以是指定的或者继承的。）

请注意：对于一个透明的外观诠释了 **XObject** 组作为一个（见第 8.4.4，“外观流”），默认的色彩空间似乎继承了网页上的注释

|

布尔

(可选的) 标志位指定是否是孤立的透明度组(见小洁

7.3.4, “孤立组”)。如果此标志是真实的, 集团内的对象对一个完全透明的复合最初的背景下, 如果假的, 他们是假定对集团的背景。默认值: 假的。

在组属性字典的页面, 这个词条的解释是稍有改变。在正常情况下在该网页上直接征收输出介质, 有效地孤立页面组和指定我的价值被忽略。但是, 如果页面又为其他一些页面元素使用, 它是如果它看作是一个透明度组了 XObject; 我值解释正常的方法, 以确定是否是孤立的页面组。

K	布尔	(可选的) 标志位指定是否是推倒的透明度组 (见小节 7.3.5, “推倒组”)。如果此标志是真实的, 集团内的对象对一个完全透明的复合最初的背景下, 如果假的, 他们是假定对集团的背景。默认值: 假的。
---	----	--

组了 XObject 的透明度的内容流定义的图形对象属于本集团。调用了 XObject 经营者是否对执行其内容流组和复合材料所产生的颜色, 形状, 不透明度到集团的母公司或网页, 好像他们来自一个基本图形对象。当应用到一个透明度组了 XObject, 请执行以下行动中除了为窗体了 XObject (第 4.9 节中所述, “表格 Objects” 正常的)

- 如果透明度组非隔离 (即在其第一组项的值字典属性为 false), 其最初的背景下, 在包围盒由了 XObject 的 BBox 项规定, 被定义为积累的颜色和组 alpha, 也就是说, 将得到的东西有结果被画在父组这一点。 (但是, 如果家长是淘汰赛组, 最初的背景是, 作为父母的相同。) 如果本集团隔离 (I 是真的), 其最初的背景下被定义为透明。
- 之前的透明度组了 XObject 的内容流的执行, 在目前的混合状态被初始化为普通图形模式, 目前的抚摸和非抚摸阿尔法常数为 1.0, 而目前的软掩码没有。

请注意: 在初始化开始时这些图形状态参数的用途执行是确保他们没有申请两次: 一次是当成员对象画到时再组和组到母公司或网页画。

• 文档在描绘对象的透明度组的 XObject 的内容流的时候复合到组中, 根据 7.2.2 节中所述, “基本合成公式的规则。” 标志的淘汰位 (k) 在组属性字典和图形的透明度相关的参数

造成这种状态的计算。

- 如果一组色彩空间 (CS) 的是在小组指定属性字典中，所有画板转换源颜色的彩色空间之前合成入组对象，并在每个点所产生的颜色是解释该颜色空间。如果没有指定组色彩空间，当时的色彩空间是动态继承父组或网页。(如果没有其他规格，指明分数，网页小组的色彩空间是从继承的本土色彩空间输出设备。)
- 经过了 XObject 的透明度组的内容流的执行，图表，IC 的状态恢复前的经营者是否调用到以前的状态 (因为它不为任何形式的了 XObject)。该组织的形状，所有物体工会油漆编入小组，由小组了 XObject 的边框，然后剪辑油漆教育署将母公司或网页，使用该组织的颜色和积累在每一点混浊。

注意：是否被调用一次以上组为给定的透明度了 XObject，每次调用被视为一个单独的透明度组。也就是说，结果是，如果该组的每个独立的背景下复合调用。应用程序执行呈现形式 XObjects 必须采取缓存这将户口的规定上述的行动会出现一个透明组只了 XObject - 1 形成了 XObject 有一组项指定的一组属性 subdictionary 亚型组 (S) 是透明度。一个普通的表格了 XObject - 一组有没有入门不受任何透明度的目的分组行为。也就是说，它包含的复合图形单独的对象。

7.5.6 模型和透明度

在透明的成像模型，图形对象组成的图案细胞一个平铺模式(见第4.6.2，“平铺模式”)可以包括透明对象和透明度组。透明合成内可能发生该模式细胞和它与背景图案是哪里画。同样，底纹图案 (第4.6.3，“底纹图案”) 复合材料它的背景下，犹如被遮荫字典与 SH 申请。

在这两种情况下，该模式的定义是，如果它被视为隐含在一个封闭非隔离透明度组：一为平铺模式，非淘汰赛 A 组基因敲除组底纹图案。该定义不继承当前它是在评估时间值的图形状态参数，这些生效只有在使用后产生的图案是画的对象。相反，

图形状态参数初始化如下：

- 一如往常的透明度群体，这些参数与透明度 (混合模式，软膜，和 a 常数) 初始化为它们的标准默认值。
- 所有其他参数初始化在内容流 (开始了自己的价值，例如一个页面或窗体了

XObject 在该模式是定义) 作为一种资源。这是所有模式的正常行为，不论是不透明或者透明成像模式。

- 在底纹图案的情况下，参数值可能会增加，这是在模式中的字典 ExtGState 项内容（见第4.6.3，“底纹模式”）。只有那些参数是影响的，如当前变换矩阵和渲染意图，被使用。参数影响路径画经营者不使用以来，立刻执行，并不涉及绘画的路径。
- 如果有一个背景底纹字典项，该模式的隐含透明度组与之前的 SH 操作指定的背景颜色填充被调用。

当该模式是用于油漆后一个图形对象，颜色，形状和不透明度值从模式的定义的评价结果被用作该对象的源颜色 (Cs)，物体的形状 (Fl) 的，不透明度和对象 (I) 在透明合成公式。这幅画在操作的时候受到了有效的图形状态参数的值，就像在绘画与稳定的色彩一对象。

不同的是不透明的成像模型，其中一个单元格可以平铺花纹图案被评估一次，然后复制无限期地填补画领域，在一般情况下的效果是透明的，如果模式被定义为每个独立瓦型，考虑到在该背景颜色每个点。然而，在通常情况下在该模式完全由画的物体与正常混合模式，此行为可以优化细胞治疗的模式，好象这是一个孤立的群体。因为在这种情况下的结果仅仅依赖于颜色，形状和图案细胞混浊，而不是那些的背景下，该模式可以计算一次细胞，然后复制，就像在不透明画。

注意：在一个平铺光栅为基础的执行情况，重要的是所有瓷砖一起作为一个单一的透明度治疗组。这样可以避免因多种工件沿标记的像素相邻瓷砖的边界。

前面的讨论适用于有色 (PaintType 1) 和本色 (PaintType 2) 图案。在后一种情况下，一个本色的限制模式的定义可能不指定颜色的延伸，以及任何透明度组的定义可能包括。有没有相应的限制，然而，就在指定的图形状态透明度相关的参数。

7.6 颜色空间和渲染的问题

本节描述透明度与颜色指定和渲染的其他方面的交互。

7.6.1 透明组的颜色空间

如在第7.5.5节，“透明度集团 **XObjects**，”一个透明组可以有它自己的明确声明或继承的色彩空间它的母公司。在这两种情况下，集团内的源对象的颜色转换为集团的色彩空间，如有需要，以及所有与合成混合计算完成在该空间（见7.2.3节，“混合色空间”）。由此产生的颜色，然后解释该颜色空间时复合组随后与背景。

根据这项安排，我们设想，全部或大部分的一块艺术品将在一个单一的色彩空间，最有可能创造，工作色彩空间它生成的应用程序。将多种色彩空间通常会使用只有当出现组装到一个独立制作网页作品。后所有的作品已被放置在页面上，来自该集团的色彩转换空间页面的设备色彩空间工作将作为最后一步，没有任何提高透明度的合成。透明成像模型并不要求本公约应遵循，但是，它的通过是原因避免颜色信息的损失和失误造成的引进不必要的色彩空间的转换。

只有一个孤立的群体可能有其自身的显式声明的色彩空间。非孤立的群体，必须继承其从父组（题目为页组特殊待遇的色彩空间，如7.3.6节所述，“页面集团”）。这是因为在非隔离组使用一个明确的色彩空间将需要转换的背景的颜色空间的颜色，该小组，以便执行合成计算。这种转换可能是不可能的（因为有些颜色可以进行转换只在一向），甚至如果可能的话，这将需要转换的色彩过多。

组颜色空间的选择有重要效果，它将产生：

- 正如7.2.3节中所指出，“混合颜色空间”，在结果的合成一个设备的色彩空间是设备依赖。对于合成计算，从而在一个设备无关的方式工作，该小组的色彩空间必须智库为基础。
- 假如选择一个智库组成的社会团体色彩空间的后果是，只有智库，基于空间可以用来指定集团内各对象的颜色。这是因为从设备转化为智库的颜色是不可能在基于CIE; 定义的转换工作，只是方向相反。见下文进一步讨论。
- 计算的合成和混合功能的一般计算线性色彩元件值的组合，前提是该组件值本

身，是线性的。基于这个原因，它通常是最好的选择组色彩空间，有一个线性伽玛函数。如果一个非线性色彩空间选定后，结果仍然是定义良好的，但外观可能不匹配用户的期望。注意，特别是该智库的 sRGB 色彩空间（见页226）是非线性的，因此可以作为一个不适合使用组色彩空间。

注意：透明成像模式的实现，应使用在尽可能多的精度在合成计算的颜色代表

在积累了论坛的结果。为了尽量减少舍入误差积累和避免了线性群色彩空间使用中产生的其他错误，更精确的需要比通常用来表示中间结果无论是原来的源数据或最终栅格化的结果。

如果一个群体的色彩空间，无论明示或指定从父继承集团是智库为基础，对绘画对象的任何设备的色彩空间的使用受以特殊待遇。设备的颜色不能直接画成这样一个小组，因为没有通常被定义为它们转化成 CIE 为基础的方法色彩空间。此问题出现在以下情况：

• **DeviceGray, DeviceRGB, 和 DeviceCMYK** 颜色空间，除非重新映射到解除故障智库为基础的色彩空间

- 算子（如 RG）的装置，指定一个色彩空间暗示，除非该空间被重新映射
- 特别颜色空间的基础或底层空间是一个设备的色彩空间，取消不到空间重新映射

建议的默认颜色空间重映射机制始终受聘当定义一个透明组，其色彩空间智库为基础的。如果指定颜色的设备，而不是重新映射，它会转换成 CIE 为基础色彩空间中的实现依赖的方式，产生不可预知的的结果。

注意：上述限制并不适用，如果该组织的色彩空间是含蓄转换为 *DeviceCMYK*，如“隐智库基于转换颜色讨论空间”页228。

7.6.2 点颜色和透明度

色彩的空间上面的讨论过程中一直关注同颜色由一个输出设备的过程中生产的着色剂的组合。过程中指定的颜色，可在设备的本土色彩空间（如直接DeviceCMYK），或者他们可能会产生一些其他颜色的转换空间，如智库为基础的(CaIRGB或ICCBased)空间。使用任何手段指定他们，混色受到转换和从该集团的色彩空间。专色是一种额外的颜色组成，用于生产过程中颜色的独立。它可能是一种额外的隔离，生产或额外的着色剂被应用到复合页（见“分离颜色空间”和234页上的“DeviceN色彩空间”238页）。颜色组成部分的价值，或色彩的专色，指定的浓度相应的点着色剂。色彩是传统派减法，而不是添加剂，价值观。

颜色是天生的设备依赖，不总是可用的。在不透明的成像模型，在每一个专色分离或元件使用DeviceN色彩空间是伴随着一个备用色彩空间和色彩空间映射到该值的色彩转换功能。这使得颜色近似与工艺着色剂着色剂时，相应的点是在设备上无法使用。

现场的颜色可容纳直截了当的透明成像模型(除有关的问题套印，在第7.6.3讨论，“叠印和透明度”)。当一个对象是透明的画专色组成部分，可在输出设备，该颜色是与相应的现场的背景颜色分量复合，是独立于流程的颜色进行合成。专色保留自己的身份，它是不受转换或从色彩空间对封闭的透明度组或网页。如果对象是一个透明度组，两件事情可能发生的元素之一。

- 该集团保持了每个单独的专色组成颜色值，独立于本集团的色彩空间。实际上，通过直接的专色通过集团的层次结构到设备，没有颜色转换完成。但是，它仍然是受混合，与使用相同专色的其他对象合成。
- 专色转换为它的替代颜色空间。由此产生的颜色然后按正常合成流程色彩规则。特别是，专色是永远不会提供一个透明度组了XObject用来定义一个软口罩；的替代颜色空间始终在这种情况下取代。

只有一个形状值和不透明度值维持在每一个点计算组的结果，它们同时适用于过程和专色组成部分。实际上，每个对象都被认为是现有涂料颜色组成部分，既过程和现场。如果没有值已明确规定在一个给定的对象，一个给定组件1.0附加值指定（或减法色彩值0.0）假定。例如，当画在指定颜色的对象DeviceCMYK或ICCBased色彩空间，颜色分量的过程是画作为指定的颜色分量，并当场与价值画添加剂1.0。同样，当绘画在一个指定的颜色色分离的对象空间，命名为专色，画指定和所有其他部件（进程的颜色和其他专色）与价值画添加剂1.0。本节讨论了7.6.3，“叠印和后果透明度。”

不透明的成像过程模型还允许颜色分量加以解决个人，好像他们是专色。例如，它可以指定一分离颜色空间命名为青色，而涂料只是一个青色组件的CMYK输出设备。但是，这种能力是非常困难的群体扩大到透明度。一般来说，在一个组中的颜色分量不是过程着色剂本身，而是被转换为只处理后，该集团所有合成（或许还有一些计算色完成着色剂以及其母公司组）。例如，如果组的色彩空间是ICCBased时，论坛没有被涂青色组件。因此，治疗过程

颜色组成部分，因为如果它是一个专色，只是一组允许的范围内继承了输出设备（天然色彩空间或隐式转换为DeviceCMYK所讨论的“隐智库进行颜色空间转换”228页）。试图在一组，指定其自己的颜色在所要求的专色转换为它的替代颜色空间的结果。

7.6.3 重印和透明度

在不透明的成像模型，叠印是受两个参数图形状态：在套印参数和叠印模式（见第4.5.6，“套印控制”）。绘画对象导致某些具体的着色剂的装置设置为标志，按目前的色彩空间和当前的颜色确定图形状态。其余的着色剂不是删除或保持不变，这取决于是否套印参数的真伪。当电流色彩空间是 DeviceCMYK，套印模式的参数可以加这种着色剂选择性标记，以适用于个别颜色分量根据组件是否为零或非零值。

由于这种套印与绘画的设备直接交易模式着色剂，独立的色彩空间，色彩的来源已指定的，它是高度依赖于设备的地址和主要生产需要，而不是设计意图。叠印通常是不透明的着色剂保留或非常暗的颜色，如黑色。这也是在调用后期生产操作，如诱捕（见第10.10.5 “，俘获支持”），着色剂的设备时，实际设置已经确定。

因此，最好是透明看成参加面貌发生空间，但着色剂叠印的设备在设备空间。这意味着，套印的决定应在输出时间，基于任何实际运作的透明度，因而合成着色剂。另一方面，类似的叠印效果，可在在一个设备独立的方式实现

通过利用混合模式的优势，如下一节所述。

混合模型和重印

如第7.6.2表示，“现货颜色和透明度”，每个图形对象这是影响所有现有的颜色画的组成部分：着色剂的全过程透明度小组的色彩空间，以及任何可用的位置着色剂。对于颜色

部件的价值还没有被指定，为1.0光源颜色值假设，当对象是完全不透明和混合模式是正常使用，这有这些组件的擦除效果。这种待遇是一致的行为与套印设置为`false`参数不透明成像模型。

透明成像模型定义了一些混合，如变黑，模式，即可以用来实现类似的效果叠加。`Darken`混合函数：

$$B(cb, cs) = \min(cb, cs)$$

在这种混合模式下，合成的结果始终是相同的背景色当源颜色为1.0，因为它是所有未指定的颜色分量。当背景是完全不透明的，剩下的结果颜色保持不变是的背景。这与成像的不透明行为是一致的模型设置为`true`的套印参数。

如果该对象或不完全不透明的背景下，上述相应的修改所述的行动。也就是说，擦除效果降低，叠印为1.0色值的对象可能会影响结果的颜色。虽然这些结果可能或未必是有用的，他们的谎言以外的叠印不透明和擦除的成像模型定义的行为境界。

当过程中颜色叠加或删除（因为专色正绘），完成以上所述的独立的混合运算每个小组的色彩空间的组成部分。如果从不同的空间输出设备的原始色彩空间，它的组件没有该设备的实际过程着色剂；影响的混合计算过程只着色剂之后，小组的成果转化到设备的色彩空间。因此，效果从不同的叠印或删除该设备的过程中着色剂直接。另一方面，这是一个完全通用的运作，使工程均匀，不论对象或计算，产生源类型颜色。

到目前为止的讨论都集中在那些颜色组件的值没有指明，而不是要删除或保留不变。然而，正常或变暗混合用于这些目的可能不适合的模式这些组件使用其指定的颜色值。特别是，使用变暗混合的模式将这些部件排除叠印黑暗用打火机的颜色之一。此外，其他一些具体的混合模式可能希望对这些组件。

PDF格式的图形状态仅指定一个当前的混合模式参数，总是适用于处理着色剂和着色剂，有时也到现场。具体来说，只有可分离，白保混合模式可以用来为现货颜色。混合模式是一个白色的保护，如果它的混合函数`B`的财产该乙（1.0, 1.0）= 1.0。（标准可分在表7.2中列出的混合模式491页，除差异和排除都是白保存。）如指定的混合模式是不分开的，白保，它仅适用于四色组成部分；正常

混合模式是专色替代。

这样可以确保当对象堆积在一个孤立的透明度组对不明成分的累积值保持，因为只有白保长的混合模式1.0使用。该集团的业绩然后可以使用套印变暗（或其他有用的模式），同时避免不必要的相互作用其成分从未集团内指定的值。

不透明重印的兼容性

由于使用混合模式来达到类似的效果不叠加利用图形中的套印控制参数直接使用状态，例如方法是有用的，只有透明的应用程序。出于兼容性与套印控制在不透明的成像模型，一个特殊的混合模式，兼容套印使用的方法，只要是有关咨询的套印图形状态参数来计算其结果。这种模式只适用于小学时画的图形对象（填充，笔触，文字，图像和阴影）。这是从来没有明确地调用，并没有发现任何PDF对象的名称；相反，它是隐式调用每当一个基本图形绘制对象而叠印启用（即在图形时的状态是真实套印参数）。

注意：透明成像模型的设计，包括增设早些时候混合模式命名兼容，其中明确援引兼容这里描述的套印混合模式。由于兼容性重印引用现在含蓄在适当时，它永远不会要指定兼容的混合模式在合成使用。它仍然是公认的有效结合为模式的缘故指数，而只是视为等同于普通。

该混合模式在**CompatibleOverprint**函数B（CB，CS）的值是要么会CB或CS，这取决于对套印模式参数，设置当前和集团的颜色空间，颜色值cs的源

- 如果套印模式是1（非零叠印模式）和当前颜色空间和群体色彩空间都是**DeviceCMYK**，那么只有混色非零值与元件更换相应的元件值的背景。所有其他元件值保留现有的背景值

不变。也就是说，在混合函数B（会CB值，CS）的是任何进程（**DeviceCMYK**）颜色组成，其源组件cs的（减法）色值是非零的，否则它会CB组件的背景。对于专色组成部分，其价值总是会CB。

- 在所有其他情况下，B值（会CB，CS）是cs的指定颜色的所有部件在目前的色彩空间，否则会CB。例如，如果当前的色彩空间是**DeviceCMYK**或**CalRGB**，该混合函数值是cs的颜色的过程部件和组件会CB现场。另一方面，如果当前的色彩空间是分离的空间代表一个专色组成部分，该值cs的组成部分，该点为全过程的所有

其他组件和现货会CB
组件。

注意：在上面的描述，这个词是指当前颜色空间的颜色空间用于画作业。这可能是目前所指定的色彩空间在图形状态参数（见4.5.1节，“颜色值”），含蓄的颜色，如经营者的RG（第4.5.7，“颜色商”），或由一个形象了XObject（第4.8项的色彩。4，“图像词典”）。在一个案件索引的空间，它指的是基地色彩空间（参见“索引颜色空间”的页面232），同样的分离和DeviceN空间的色彩还原为其候补空间，所述“分离颜色空间”和234页上的“DeviceN彩色空间”页238。

如果当前的混合模式时调用任何模式其他较正常，正在绘制的对象是含蓄，如果它被视为定义在非隔离，非基因敲除组和油漆使用的透明度在混合模式。该工作组的结果，然后用画目前在图形混合态模式。

注意：这是没有必要建立这样一个隐含的透明度，如果当前组混合模式是正常的，只是代替混合模式而画的对象产生相同的结果。还有一些额外的其中隐含的透明度组可以优化出的案件。

板20显示所有4个可能的组合融合和叠印的效果，使用屏幕融合在DeviceCMYK色彩空间模式。该标签“套印启用”是指在图形套印参数状态真实的套印模式是1。在图中，浅绿色椭圆形的上半部画是不透明（不透光率=1.0）较纯黄色底纹为背景纯洋红。在下半部分，同样的对象，画上的透明度（不透光率=0.5）。

特别的路径描绘考虑

以上讨论的叠印考虑也影响这些路径画操作相结合的灌装，抚摸在单一经营的路子。这些包括乙，乙*，乙，和b *画板（见第4.4.2节，“路径，画板”）与字形的渲染模式文字画2或6（第5.2.5，“文本渲染模式”）。对于透明合成的目的，结合填充和中风被视为一个单一的图形对象，如果他们是在一个封闭透明度组。这隐含小组的建立和使用情况如下：

- 如果叠印启用（在图形状态是真实的套印参数）和目前的抚摸和nonstroking阿尔法常数是平等的，非隔离，非淘汰赛透明度组成立。集团内，填充和中风都以一个1.0 alpha值，但与CompatibleOverprint混合模式。该小组的结果，然后与复合背景下，利用原指定的a 和混合模式。
- 在所有其他情况下，非隔离敲除组成立。在组，填充和中风都以各自目前的阿

尔法常量和当时的混合模式。结果该组，然后用复合背景，使用的是1.0 alpha值和正常的混合模式。

请注意，在显示与合并灌装和抚摸文本案渲染模式，这种行为是独立的文本参数淘汰赛

图形状态（见第5.2.7，“敲文字”）。

这些规则的目的是为了避免非不透明中风复合材料对在重叠，这将产生双重影响边境地区，通常是不可取的填充结果。特殊情况下，适用于当套印参数为true是为了向后兼容的叠印

行为不透明成像模型。如果想要的效果就无法实现与灌装和抚摸联合经营或文字渲染模式，它可以达到指定的填充及另外一个明确的目标和路径的透明度组划线。

注意：在填补了不起作用在上述第二种情况下（虽然不是中风叠印填充或中风的背景下仍然可以套印）。此外，如果套印图形状态参数是真实的，是不连续的结果是两者之间的行程和非抚摸阿尔法常数平等和不平等的价值观过渡。基于这个原因，最好不要使用套印为联合行动的灌装和抚摸抚摸和*nonstroking*如果阿尔法常数正在不同独立。

重印行为的总结

表7.14和7.15总结叠印和擦除的行为不透明和透明的成像模型，分别。表7.14显示了叠印不透明模型中使用，如4.5.6节所述，“套印规则控制。”表7.15显示了由实施混合模式的透明模型等价规则。OP和OPM的名称在统计表的套印和叠印模式图形状态参数。

表7.14 不透明成像模型中的重印行为

源颜色空间	受影响颜色分子	颜色组件的影响		
		OP FALSE	OP TRUE, OPM0	OP TRUE, OPM1
DeviceCMYK 直接指定， 不是在简化图像中	C, M, Y, 或者 K	描绘源	描绘源	如≠0.0 描绘源
	处理着色 而不是CMYK	描绘源	描绘源	描绘源

	点着色	描绘0.0	不描绘	不描绘
任何颜色空间 (包括 DeviceCMYK 的其中情况)	进行着色	描绘源	描绘源	描绘源
	点着色	描绘0.0	不描绘	不描绘
分离或者DeviceN	进行着色	描绘0.0	不描绘	不描绘
	颜色空间点着色	描绘源	描绘源	描绘源
	不是源空间点着色	描绘0.0	不描绘	不描绘

表7.15 透明成像模型中的重印行为

源颜色空间	受影响的组颜色空间的组件	混合函数B (cb, cs) 表达成TINT的值		
		OP FALSE	OP TRUE, OPM0	OPTRUE, OPM1
DeviceCMYK 直接指定, 不是在简化图像	C, M, Y或者K	cs	cs	cs, cs ≠ 0.0 cb, cs = 0.0
	处理颜色组件而不是CMYK	cs	cs	cs
	点着色	cs (=0.0)	cb	cb
任何其他的颜色空间 (包括 DeviceCMYK 的情况)	处理颜色组件	cs	cs	cs
	点着色	cs (=0.0)	cb	cb
分离或者DeviceN	处理颜色组件	cs (=0.0)	cb	cb
	源空间点着色不再源空间的	cs	cs	cs
	点着色	cs (=0.0)	cb	cb

一个组(不是 一个初级对象)	所有颜色组件	cs	cs	cs
-------------------	--------	----	----	----

色彩元件值的代表色值作为消减这些表因为叠印通常用于消减着色剂，如油墨而不是添加剂的如显示屏荧光粉。该混合模式，因此如果它描述为参数，返回了减法减法的结果。但实际上，（像所有的混合模式）视为添加剂价值观颜色分量；消减组件必须补充之前和之后的混合应用功能。

请注意两个表之间的重要区别。在表7.14，过程颜色组件正在讨论的是实际设备着色剂，颜色元件的输出设备的原生色彩空间（DeviceGray, DeviceRGB, 或DeviceCMYK）。在表7.15，这个过程色组成部分是那些小组的色彩空间，这并不一定是作为输出设备相同（甚至可以像CaIRGB或ICCBased的东西）。基于这个原因，该组色彩空间混色组件不能被视作他们

在DeviceN色彩空间分离或专色（见第7.6.2，“专色和透明度”）。之间的这种不透明和透明的规则差异套印和擦除时（包括页面仅在一个透明组组，如果它的色彩空间是从本机色彩空间不同的输出设备）。有没有专色元件的差别待遇。

表7.15已在底部一个额外的行。它适用于当画一个对象，它是透明而不是一组基本对象（填充，中风，文本，图像，或着色）。如第7.6.2表示，“现货颜色和透明度”，是考虑到油漆的颜色部分，不论是过程和现场一组。颜色，没有任何明确的组对象画组件有添加剂的颜色值1.0 (0.0减法色彩)。是因为没有信息对保留的组件，实际上集团内画，兼容叠印是不可能在这种情况下，混合的模式恢复到正常，没有考虑的套印和叠印模式参数。（甲透明度识别应用程序可以选择一个更适合混合模式，如变黑，产生类似的效果叠加。）

7.6.4 渲染参数和透明度

不透明的成像模型有几个图形状态参数与处理色彩渲染：当前色调（见6.4.4节，“半色调词典”），传输功能（第6.3节，“传递函数”），使意图（“渲染有所图”页230），和黑色代和底色搬迁函数（6.2.3节，“从DeviceRGB转换为DeviceCMYK”）。所有这些渲染参数可以指定对每个对象的基础上，他们控制如何呈现一个特定的

对象。当所有的对象是不透明的，它很容易界定这意味着什么。但是，当他们是透明的，不止一个对象可以有助于在一个给定的点的颜色，它目前还不清楚其中的渲染参数适用于透明物体的一个领域重叠。同时，透明

成像模型应与不透明模型不透明物体时，只画一致。

此外，一些渲染参数，半色调和传输功能，尤其是一可应用于只有在最后一个给定的点的颜色众所周知。在透明的存在，这些参数必须从那些受到不同的待遇(渲染意向，黑色一代，底色搬迁)适用每当颜色必须从一个色彩空间转换为另一个。当对象是透明的，对象的呈现不会发生当指定的对象，但在以后的时间。因此，对于前一类渲染参数，实现必须随时跟踪渲染每个参数从他们的时间，直到实际发生的指定的时间点。这意味着这些渲染参数必须符合相关的页面区域，而不是单个对象。

中间色调和转换函数

半色调和传递函数，用于在任何网页上的特定点上实际上在这些绘画的最后一个(最顶层)基本图形对象内附这一点时间，但只有当对象是完全不透明。(只有小学对象是相关的；呈现一个组对象相关参数被忽略。)在任何一点的最上面的对象被定义为基础的最顶层

在整个页面对象栈，有一个非零的值在该物体的形状(巧的)点(即，对此点内的对象)。一个对象被认为是完全不透明如果下列所有条件，在当时的对象是持有画：

- 目前阿尔法在图形状态(抚摸或nonstroking，根据涂装作业常量)为1.0。
- 目前在图形模式下的混合状态正常(或兼容，这被视为等同于正常)。
- 目前在软掩模图形状态是无。如果对象是图片了XObject，没有在自己的形象字典SMask项。
- 上述三项条件是当时也确实做经营者的为包含引用的对象，以及对任何直接祖先组。
- 如果当前的颜色是平铺模式，在其模式定义的所有对象细胞也满足上述条件。

总之，这些条件确保只有对象本身的颜色有助于在给定的点，完全模糊了背景。如需部分页面其最上面的对象是不透明或不完全从来没有在所有油漆，默默认半色调和页面传递函数的使用。

注意：如果一个图形对象，画上叠印启用，也就是说，如果适用（抚摸或 nonstroking）在图形套印参数的状态是真实的半色调和传递函数使用在给定的点必须确定每种颜色分量独立。叠印隐式调用的兼容，叠印混合模式（见“不透明叠印兼容性”页536）。一个对象被认为是一个给定组件不透明只有兼容，彩色套印产生源（不是背景颜色）的该组件。

渲染意图和颜色转换

呈现意图，黑色代和底色搬迁参数需要处理有所不同。呈现意图影响从智库转换为基础的色彩空间到目标色彩空间，同时考虑到目标空间的色域（即色彩范围可以复制）。而在不透明的空间成像模型的目标始终是天然色彩输出设备的空间在透明模式，这有可能成为本集团色彩空间的透明度在其中一组对象被绘制。

渲染的目的是在需要的时候这种转换必须执行，也就是说，当画一组对象或在一智库的色彩空间指定到父有不同的颜色空间群。这不同于从目前的半色调和传递函数，其值时，只用所有颜色合成已经完成，目前正在执行的光栅。

在所有情况下，呈现意图用于转换为对象的颜色（不论一个基本的对象或一个透明度组）取决于意图参数与对象相关联。特别是：

- 当绘画发展成为一个具有透明度，以不同的颜色空间群，呈现意图使用智库进行颜色的基本对象是在目前实施的渲染图形意图状态在绘画时运作。
- 当画一个透明组，其色彩空间智库基成母公司有不同的色彩空间，呈现的意图是采用

在当前的渲染效果在当时的操作员的意图是否是应用于小组。

- 当页面组色彩空间智库为基础，呈现意图使用颜色转换到输出设备的色彩空间是本机的默认呈现为页面的意图。

注意：由于可能存在一个或多个嵌套的透明度有不同的群体智库为基础的色彩空间，一个基本源对象的颜色可能会被转换在不同阶段对设备的色彩空间中，在渲染效果在每一个阶段的意图控制。在每个阶段呈现的意图正确的选择取决于相对的源和目标色彩空间。它指定明确的文件制作，而不是由PDF技术规范规定的，因为没有单一的政策管理呈现意图是为所有的情况。

类似的方法工程黑白和底色迁移功能，它们可被用于从DeviceRGB期间只转

DeviceCMYK

色彩空间：

- 当画一个带有DeviceRGB颜色基本对象直接进入透明组，其色彩空间DeviceCMYK，所用的功能目前黑代，实际上底色迁移中的作用图形状态在绘图手术时间。
- 当画一个透明组，其色彩空间DeviceRGB成家长组，其色彩空间DeviceCMYK，功能用的都是那些实际上当时的经营者是否适用于本集团。
- 当页面组色彩空间DeviceRGB与本土色彩输出设备的空间DeviceCMYK，用于转换颜色的职能该设备的颜色空间是页面的默认功能

7.6.5 PostScript 兼容性

由于PostScript语言不支持透明成像模型，PDF 1.4的消费者应用程序必须有一定的转化途径外观的文档，它使用的透明度，以一个纯粹的不透明描述为在PostScript输出设备打印。类似的技术也可以用来这些文件转换为一个可以用PDF 1.3和正确地看待形式早前消费者。从透明的转换页的内容，形式需要一些不透明结合形状分解和prerendering平缓栈页面上的透明物体，执行所有必要的透明度计算，并说明最终的外观采用不透明的对象只。

无论网页包含的内容需要透明，可被夷为平地确定本页面的资源简单的分析，这是没有必要分析内容流本身。转换为不透明的形式是不可逆转的，因为所有的透明度如何产生的影响，丢失的信息。

要执行的透明度计算正确，应用程序需要知道输出设备的原生色彩空间。这是没有问题时，应用控制设备直接输出。但是，当生成PostScript输出，应用程序已无从得知的本土色彩空间的方法PostScript输出设备。一个不正确的假设会破坏任何校准智库的颜色在页面上出现。这个问题可以解决，要么两种方法：

- 如果整个页面由智库的颜色，平坦的颜色单一智库的色彩空间而不是一个设备的色彩空间。首选的色彩空间为此可以很容易地确定是否有一个组的页面属性字典（在页面对象组条目）指定一个智库的色彩空间（见第7.5.5节，“透明度集团XObjects”）。
- 否则，扁平化的颜色与预定的一些假设设备校准色彩空间。在生成的PostScript输出，油漆的平顶颜色一智库进行颜色校正后的空间由于之间使用着色剂，并当

场将它们转换为替代颜色空间的选择会影响颜色的扁平化过程的结果，决定还必须在作跋可用来设置转换点着色剂承担。（这不同于严格不透明画，其中的决定可以推迟到生成的PostScript代码时被执行。）

第 8 章 交互功能

本章描述 pdf 的特性，允许用户利用鼠标和键盘在屏幕上与文档交互（关于多媒体功能应用，这是在第 9 章介绍，“多媒体功能”）：

- 设置参数来控制文档在屏幕上的显示方式（8.1 节，“浏览器参数”）
- 通过导航工具移动文件有多种方式（第 8.2 节，“文档级导航”，8.3 节，“页级导航”）
- 为文档添加注释，声音，视频和其他辅助信息（第 8.4 节，“注释”）
- 指定事件触发操作（第 8.5 节，“操作”）
- 采用互动的形式从用户收集信息（第 8.6 节，“互动形式”）
- 数字签名，验证一个用户的身份和这个文档内容的有效性（第 8.7 节，“数字签名”）
- 测量属性，使现实世界的事物显示成在页面上的一个对象（第 8.8 节，“测量属性”）

8.1、用户参数

在文档目录了浏览器参数选项中（见第 3.6.1 节，“文档目录”）指定了一个选项（PDF1.2）控制文档是要在屏幕上或在打印。如果没有指定这样的选项，查看和打印的申请应当与前用户的参数设置相一致。表 8.1 显示了用户参数的内容。（见在附录 H 73 页的执行情况说明）

表 8.1 用户参数

KEY	TYPE	VALUE
HideToolbar	boolean	(可选) 一个标志指定是否隐藏当前工作文档的工具栏。默认值：假
HideMenubar	boolean	(可选) 一个标志指定是否隐藏当前工作文档的菜单项。默认值：假
HideWindowUI	boolean	(可选) 一个标志指定是否隐藏文档窗口中接口选项(如滚动条和导航控制)。只留下该文件的内容显示。默认值：假
FitWindow	boolean	(可选) 一个标志指定是否调整文档的窗口以适应显示的第一页的大小。默认值：假
CenterWindow	boolean	(可选) 一个标志指定文档窗口是否是否在屏幕中心。默认值：假
DisplayDocTitle	boolean	(可选；PDF 1.4) 一个标志指定是否在窗口标题栏显示文档标题，它是从文档的信息

		中获取的（见第 10.2.1 节，“文档信息”）。如果为 <code>false</code> , 在标题栏显示 PDF 文档的包含的内容。默认值：假。
<code>NonFullScreenPageMode</code>	<code>name</code>	(可选) 文档的页面模式，指定如何显示在退出全屏模式下： <code>UseNone</code> 无论文档概要或略图，图像轮廓可见 <code>UseOutlines</code> 文档概要可见 <code>UseThumbs</code> 文档缩略图可见 <code>UseOC</code> 可选内容组面板可见 这个项目有意义只有在 <code>PageMode</code> 项的值（见第 3.6.1 节，“文件目录”）是 <code>FullScreen</code> 时；否则它会被忽略。默认值： <code>UseNone</code> 。
<code>Direction</code>	<code>name</code>	(可选; PDF 1.3) 主要用于文字的阅读顺序： <code>L2R</code> 从左到右 <code>R2L</code> 从右到左(包括垂直书写系统, 如中文, 日语和韩语) 这个项目没有在该文件的内容或页码直接影响但可用于确定网页的相对定位当并排显示或打印的n的一面。默认值： <code>L2R</code>
<code>ViewArea</code>	<code>name</code>	(可选; PDF 1.4) 一个页面区域范围的名称将会显示, 当文档显示在屏幕上的时候。这个值是指定页的有关边界的关键（见“页对象” 119页和第10.10.1节，“页范围”）。如果页对象中的页面边界没被定义, 使用它的默认值, 如见119页表3.27。默认值： <code>CropBox</code> 。 注：此项主要是使用来解释或操纵页面边界，见第10.10.1节“页边界。” 大多数的PDF应用可以忽略它。
<code>ViewClip</code>	<code>name</code>	(可选; PDF 1.4) 页面边界名称, 它其中的一个页面的内容被裁剪, 当浏览屏幕上的文档时。该值是指定页面对象相关的页边界的关键（见 119 页“页对象” 和第 10.10.1 “节，页边界”）。如果指定页面边界没有在页面对象中被定义, 使用它的默认值, 如页 119 表 3.27。默认值： <code>CropBox</code> 。 注：此项主要是使用来解释或操纵页面边界，见第10.10.1节“页面边界”， 大多数的PDF应用可以忽略它。
<code>PrintArea</code>	<code>name</code>	(可选; PDF 1.4) 页边界名代表的一个页面区域将会被提供用于打印, 当打印文档时。该值是指定页面对象相关的页边界的关

		键（见119页“页对象”和第10.10.1节“页边界”）。如果指定页面边界没有在页面对象中被定义，使用它的默认值，如页119表3.27。默认值：CropBox。 注：此项主要是使用来解释或操纵页面边界，见第10.10.1节“页边界。”大多数的PDF应用可以忽略它。
PrintClip	name	页边界名代表的一个页面的内容被裁剪，当打印文档时。该值是指定页面对象相关的页边界的关键（见119页“页对象”和第10.10.1节“页边界”）。如果指定页面边界没有在页面对象中被定义，使用它的默认值，如页119表3.27。默认值：CropBox 注：此项主要是使用来解释或操纵页面边界，见第10.10.1节“页边界。”大多数的PDF应用可以忽略它。
PrintScaling	name	(可选；PDF 1.6) 当该文档打印对话框显示时，页缩放比例选项被选中。有效值是None，这表明该打印对话框没有页面缩放比例；AppDefault，这表明应用程序应使用当前的打印缩放比例。如果此项中有无法识别的值，应用程序使用当前的打印缩放比例。 默认值：AppDefault。 注：如果打印对话框被抑制，它的参数是直接由应用程序提供，该项的值依然也会被使用。

8.2、文档级导航

本节中所描述的功能允许一个PDF查看器与用户交互，两种形式全局浏览文档：

- 作为同一个等级的大纲，显示文档的内部结构
- 由一种微型形式，让缩略图代表该文档

大纲中的每个项目或每个小图片可以与文档中的一个页面相对应，用户可以点击鼠标，直接跳转到目的页面。

8.2.1、目标

一个目标定义了一个特定的视图文件，由下列元素组成：

- 文档页面显示

- 文档窗口位于页的位置
- 当显示页面时，放大率（放大）因子的使用

目标可以与大纲项目关联（见 8.2.2 节，“文档纲要”），注释（在 587 页“链接注释”），或动作（“616 页的”跳转动作“和 617 页”远程访问“）。在每一种情况下，当大纲项目或注释打开或跳转时，特别指定的文档会被显示。此外，可选在文档对话框的 `OpenAction` 项（第 3.6.1 节，“文档对话框”），当文档打开时，可以指定一个目标显示。一个目标可以被明确的指定，通过一个参数数组定义它的属性或间接的名字。

明确的目标

表 8.2 显示了在 PDF 文件中，指定目标的句法形式。在每一种情况下，页是间接引用的页面对象。所有的坐标值（左，右，顶部和底部）是用默认的用户空间坐标系表示。该页的边界框是最小的矩形包围的所有内容。（如果任何一方的边界框超出了页面的裁剪框，作用的方框来代替相应的方框，见第 10.10.1 节“页边界”，中进一步讨论）。

注：没有页面对象可以指定与远程跳转相关联（见第 617 页“远程跳转行动”），因为目标页是在一个不同的 PDF 文件。在这种情况下，页面参数指定的范围内，不是在当前文档页面对象的远程文档页码。

TABLE 8.2 Destination syntax 表 8.2 目标语法

[page /XYZ left top zoom]	逐页显示指定的页面，使用坐标（左，上）定位，在窗口的左上角位置和利用缩放放大页上的内容。 <code>null</code> 值时，左，上，或缩放指定参数的当前值保持不变。 <code>0</code> 和 <code>null</code> 值有相同的含义
[page /Fit]	逐页显示指定的页面，其放大的内容刚够以适应无论横向和纵向窗口的整个页面。如果所需的水平和垂直放大系数是不同的，使用较小的两个，从其他页面的窗口进入页面。
[page /FitH top]	逐页显示指定的页，窗口上边垂直坐标的顶部和页上放大的内容足够适应窗口的页的整个宽度。
[page /FitV left]	显示的页面指定的页面上，左边的水平坐标定位在窗口的左边缘和网页上的内容放大刚好以适应窗口的整个页面的高度。
[page /FitR left bottom right top]	逐页显示指定的页面，其放大的内容足够填充窗口横向和纵向，左坐标，下坐标，右坐标，顶部及指定的矩形。如果所需的横向和纵向的放大系数不同，

	使用两个中最小的，在其他方面，窗口的中心是一个方框。
[page /FitB]	(PDF 1.1) 逐页显示指定的页面，其放大的内容正好适合窗口的横向宽和纵向高。如果所需的水平和垂直放大系数不同，使用两个中较小的，其他部分的窗口中心弹出对话框。
[page /FitBH top]	(PDF 1.1) 逐页显示指定的网页，与窗口顶部上边的垂直坐标和页面上放大的内容刚够以适应窗口边界框的整个宽度。
[page /FitBV left]	(PDF 1.1) 逐页显示指定的页面，左边的水平坐标定位在窗口的左边缘，页面上放大的内容刚够以适应窗口边框的整个高度。

命名目的

目标位置可以间接地通过一个具有名称的目标或一个字符串表示，而不直接用明确的语法定义。如表格 8.2 所示。当目标位置在另一个 PDF 文件中，这种能力尤其有用。一个到第六章的链接，在另一个文档中可能是由目标文件的名称表示，而不是一个明确的文档页码，比如第六章的开始。那么文档中章节的位置，在无效链接的情况下是不能改变的。一个注释或概述（大纲条目），如果它们有指示的目标位置，那通常都有一个相关的操作，例如一个跳转操作或者一个线程操作，如果有 F 条目，目标位置就在 F 条目指定的操作的文件中，如果没有，目标文件就在当前文件中。

在 PDF1,1 目标与目标位置对应的定义是由 Dests 项决定（见上）这个条目的值是一本字典，字典的关键是一个目标位置，对应值就是定义目标位置的一个数组或有 D 条目的一本字典，它的值就是这样一组。后者允许额外属性与目标位置关联，也使跳转操作被用作指定目标位置的对象。

在 PDF1.2 中，字符串和目标位置之间的通讯是由文档名字字典的 Dests 项定义的。这个条目的值为一命名树，其将名字字符串映射到目标位置。命名树的 key 可以看作是为了显示目的的文本字符串。与命名树值有关的目标位置可以是一个数组或字典，之前篇章已描述了。

注意：字符串作为目的名使用是 PDF1,2 的特点。如果在其版本的 pfd 兼容性方面有要求，那么只有名称对象能被用来指定命名的目标位置。PDF1,2 的文件也能兼容两种类型。然而，如果向后兼容性不予考虑，应用程序应该在目标命名树中用字符串形式表示。

8.2.2、文件大纲

一个 PDF 文件可以随意地在屏幕上显示文档大纲。允许用户能在一个文件与另一个文件间交替转换。大纲包括大纲目录的树形结构的层次树，这个层次树作为一种目录表格来向用户显示文档结构。用户用鼠标能交替开启和关闭个人条目。当一个条目打开时，层次结构的孩子项就出现在屏幕上，每个孩子项轮流打开关闭，选择性地暴露或隐藏进一步的层次结构。当一个条目关闭时，在层次结构后面的项全部被隐藏。点击文本可见的条目激活条目，会导致浏览器跳转到目标位置，或者触发一个与条目有关的操作。

一个文件大纲层次结构的根源是文件目录中大纲条目既定的层次，表格 8.3 显示了字典的内容。在层次结构中，每一个个体大纲条目都被定义为一个大纲条目字典。这个条目在每一级层次结构上形成一个链表，通过它们的函数和下一个条目绑在一起，从第一个到最后一个父项目访问。当在屏幕上显示时，该条目在给定的层次内，以它们在链表中出现的顺序出现。

表 8.3 大纲纲要规则项

KEY	TYPE	VALUE
Type	name	字典对这个类型的 PDF 描述；如果存在，必须是大纲纲要规则。
First	dictionary	(如果有任何打开或关闭大纲条目操作，必须是一种间接参考) 一个大纲条目字典代表大纲中的最高项目
Last	dictionary	(如果有任何打开或关闭大纲条目操作，必须是一种间接参考) 一个大纲条目字典代表大纲中的 the last top-level item
Count	integer	(如果文件中有任何打开的大纲条目) 在大纲水平范围内，打开条目的总数。如果没有打开的大纲条目，这个条目应该忽略
Title	text	(必须) 文本为该项显示在屏幕上。
Parent	dictionary	(必须；必须是一个间接的参考) 在大纲结构中该项的父项目。一个顶层项的父项是大纲规则本身。
Prev	dictionary	(除了每个级别的第一个项，其他都是必须的；必须是一个间接的参考) 在大纲层次结构中以前的项。
Next	dictionary	(除了每个级别的最后一个项，其他都是必须的；必须是一个间接的参考) 在大纲层次结构中的文本项。
First	dictionary	(必须，如果该项目有任何的后裔；必须是一个间接的参考) 该项在大纲中层次结构中，第一个直接孩子。
Last	dictionary	(如果该项有任何的派生，则该项必须；它必须是一个间

		接的参考)在大纲中的层次结构中, 最后这个项目的直接子女。
Count	integer	(必须, 如果该项目有任何的后裔。如果该项被打开, 它打开在较低层次大纲的文档的所有数目。如果项目是关闭的, 一个负整数的绝对值指定了多少后续将出现, 如果该项目已重新开放。)
Dest	name, string, or array	(可选; 如果 A 项存在, 则不允许)当该项有效, 目标被显示(见第 8.2.1 节, “目的地”; 也可见附录 H 第 75 页)。
A	dictionary	(可选; PDF 1.1; 如果 Dest 项存在, 则不允许) 当该项有效的时候, 该动作将发生(见第 8.5 节, “行动”)。
SE	dictionary	(可选; PDF 1.3; 必须是一个间接的参考) 该项牵涉的结构元素(见第 10.6.1 节, “结构层次”)。 注意: 用一个结构元素联系一个项目大纲的能力(例如, 一章的开始), 这是 pdf 规范 1.3 的特点。向后兼容早期的 pdf 版本, 例如一个项指定了一个对用一个页面区域的目的地(Dest), 那里指定结构元素被显示的内容。
C	array	(可选; PDF 1.4) 一个在 0.0 到 1.0 范围的三元素数组, 代表了在 deviceRGB 颜色区域的组件将被使用来绘制大纲项的文本。默认值: [0.0 0.0 0.0]。
F	integer	(可选; PDF 1.4) 一组特定的标志, 用于显示纲要项目内容(见表 8.5)。默认值: 0。

项目概要的 F 项的值 (pdf1.4) 是一个 32 位无符号整型, 它包含了指定的标志类型参数以便显示。在标志字内的位图被从 1 (低位) 到 32 (高位) 标记。

表 8.5 显示标志的含义;所有未定义的标记位被保留且必须为 0。

表 8.5 纲要项目标志		
BIT POSITION	NAME	MEANING
1	Italic	如果设置, 显示纲要为斜体
2	Bold	如果设置, 显示纲要为粗体

例 8.1 显示了一个典型的大纲和大纲项目。见附录 G 为一个完整的轮廓层次结构的例子

例 8.1

```
21 0 obj
<< /Count 6
/First 22 0 R
```

```
/Last 29 0 R
>>
endobj
22 0 obj
<< /Title (Chapter 1)
/Parent 21 0 R
/Next 26 0 R
/First 23 0 R
/Last 25 0 R
/Count 3
/Dest [3 0 R /XYZ 0 792 0]
>>
Endobj
```

8.2.3、缩略图

一个 PDF 文档可以定义缩略图，成为其页面内容的缩影。一个查看应用程序可以显示在这些图像屏幕上，让用户按一下它的缩略图就可浏览它相应的页面：

注：缩略图图像不是必需的，并可能为一些页面，或者其它的。

在页面对象中，页面缩略图是一个图像对象（见 119 页“页对象”）。它具有通用的结构（第 4.8.4 节，“图像词典”），但只有宽度，高度，色彩空间，`BitsPerComponent`，和解码项有意义；如果存在，表 4.39 在 310 页上列出的其它项都可以被忽略。（如果一个子项被指定，其值必须是图像）图像的色彩空间必须是 `DeviceGray` 或 `DeviceRGB`，或基础这些之一的一个索引空间的。例 8.2 显示了一个典型的缩略图图像定义。

例 8.2

```
12 0 obj
<< /Width 76
/Height 99
/ColorSpace /DeviceRGB
/BitsPerComponent 8
```

```
/Length 13 0 R
/Filter [/ASCII85Decode /DCTDecode]
>>
stream
s4IA>! "M;*Ddm8XA, lT0!!3, S!/(=R!<E3%!<N<(!WrK*!WrN,
...Omitted data...
endstream
endobj
13 0 obj % Length of stream
...
Endobj
```

8.3、页面级导航

本节描述 PDF 格式的特点，使用户可以浏览从文档一页面到另一页：

- 通过编号或以其他方式识别单个页（见章节 8.3.1）
- Article threads，它们是文档内容项目在逻辑上的连接，而没有物理的连接（见第8.3.2节）
- 演示是一个放映幻灯片的过程，它从一个页面到下一个自动或根据用户控制的页面（参见 8.3.3 章节）。

对于另一个页面级导航的重要形式，见 587 页 “链接注释”。

8.3.1、页面标志

PDF文件中的每个页面是由一个确定的整数标示，用于索引该页在文档中的相对位置。此外，一个文档可以选择（PDF 1.3），以确定在屏幕上是直观浏览每一页标签或者打印页面。页面标签和页面索引不必一致：该索引是固定的，从文档开始的第一页，通过连续标记，但标签可以以任何的方式指定，以适合特定文件。例如，如果文档从12页开始正面的内容罗马数字编号，剩余的文档用阿拉伯数字标记，第一页将有一个页索引0 和一个页标记i，第十二页将有索引11和页标记xii，同时，第十三页将有索引12和页标记1

使用页面标签的目的是，一个文档可分割为不同的范围标签，这是一系列使用相同编号系统的连续页面。一个范围内页面的编号顺序按升序排列。一个页面

标签的数字部分包括一个表示其范围的位置标签，一个可选的标签前缀表明它的范围。例如，在附录中的页面可能被标记，它是用十进制数字及以字符为前缀的 A -；结果页面的标签将是一个 A - 1, A - 2, 等等。

一个文档的标签定义的范围是在文档 **PageLabels** 项目录(见第 3.6.1 节,“文档对话框”)。此项的值是一个编号树 (第 3.8.6, “编号树”), 其关键是每个页面的索引标签范围。相应的值是一个标签页定义, 是在该范围内页面标签的特点。该编号树必须包括一个索引值为 0 的页。表 8.6 显示了一个页标签内容。 (见实施笔记在附录 H 76)

例 8.3 展示了一个用标签页的文档

i, ii, iii, iv, 1, 2, 3, A- 8, A- 9, ...

例 8.3

1 0 obj

```
<< /Type /Catalog
/PageLabels << /Nums [ 0 << /S /r >> % A number tree containing
4 << /S /D >> % three page label dictionaries
7 << /S /D
/P (A- )
/St 8
>>
]
>>
...
>>
Endobj
```

表 8.6 页面标志规则项

KEY	TYPE	VALUE
Type	name	(可选) PDF 对象的类型描述, 如果它存在, 对于一个页标签必须为 PageLabel
S	name	(可选) 编号样式用于每一页的数字标签: D 十进制阿拉伯数字 R 大写罗马数字

		r 小写罗马数字 A 大写字母 (A到Z用于前26页, AA到ZZ用于下一26页, 依此类推) a 小写字母 (a到z用于前26页, aa到zz用于下一26页, 依此类推) 没有默认的编号样式; 如果没有S项存在, 页面标签仅仅由一个没有数字编号的前缀组成。例如, 如果是P项 (下) 指定标签的前缀内容, 每一页被没有页码的进行简单标记。 (如果P项也缺少或为空, 页面标签是一个空字符串)
P	text string	(可选) 在此范围内的标签页的标签前缀
St	integer	(可选) 第一页标签范围中的编号部分的值。随后的页面是从这个值按顺序编号, 必须大于或等于1。默认值: 1。

8.3.2、协议

有些类型的文档可能含有内容项目的序列, 它逻辑上相连接而物理顺序上是不连接。例如, 一个信息可能开始在第一页上, 其他相关的分散在一个或多个不相邻的页面上。为了表示对物理不连续, 但逻辑上相关的项, 一个 PDF 文件可以定义一个或更多的序列 (PDF1.1)。对一个文档的顺序流的定义是一文档线; 个别项目的内容组成本文档被称为 *beads*。PDF 查看器应用程序可以提供导航, 以使用户可以从一个线索到下一个线索。

可选的线程在文件目录项 (见第 3.6.1 节, “公文目录”) 持有的线程字典 (表 8.7) 定义文档的一系列文章。每一个线程内个别珠是由一个珠词典 (表 8.8)。线程字典的 F 切入点, 在第一线珠, 琉璃珠是在通过链接在一起的 N (下) 和 V (前) 项双向链表顺序。此外, 每一页上的文章珠出现, 页面对象 (见页对象“页 119”) 的 B 项应该包含在制定秩序, 其值是对网页上的珠子间接引用的数组。

表 8.7 线程规则项

KEY	TYPE	VALUE
Type	name	(可选) 的PDF对象的类型描述, 如果存在, 必须为线程的线程
F	dictionary	(必须, 必须是间接引用) 在这个线程中的第一个头。
I	dictionary	(可选) 线程信息包含一个线程的有关信息, 如标题, 作者, 和创建日期。这个内容和文件的信息相似 (见第 10.2.1 节, “文档信息”)。

表 8.8 节点规则项

KEY	TYPE	VALUE
Type	name	(可选) 规则描述的 pdf 对象类型; 如果操作, 必须是一个节点规则的节点。
T	dictionary	(必须是所需的线程的第一节点; 对所有其他可选的; 必须是间接引用) 节点所属的线程。 注意: 在 pdf 规范 1.1 中, 该项仅仅被线程的第一个节点准许。在 pdf 规范 1.2 和更高版本中, 它被一些节点准许, 但是仅是第一个是必须的。
N	dictionary	(必填, 必须是间接引用) 线程中的文本节点。在最后的节点中, 该项指向开始。
V	dictionary	(必填, 必须是间接引用) 在这节点之前的节点。在第一节点, 此节点指向最后。
P	dictionary	(必填, 必须是间接引用) 页对象描述在本页面 bead 出现的页。
R	rectangle	(必需) 在页上的 bead 的指定矩形的位置。

例 8.4 shows a thread with three beads.

例 8.4

```

22 0 obj
<< /F 23 0 R
/I << /Title (Man Bites Dog) >>
>>
endobj
23 0 obj
<< /T 22 0 R
/N 24 0 R
/V 25 0 R
/P 8 0 R
/R [158 247 318 905 ]
>>
endobj
24 0 obj
<< /T 22 0 R
/N 25 0 R
/V 23 0 R

```

```
/P 8 0 R
/R [322 246 486 904 ]
>>
endobj
25 0 obj
<< /T 22 0 R
/N 23 0 R
/V 24 0 R
/P 10 0 R
/R [157 254 319 903 ]
>>
Endobj
```

8.3.3、显示

有些 PDF 查看器应用程序可能会允许在一个文件显示一个演示文稿或幻灯片放映，自动或根据用户的控制从一个页面前进到下一个页面。另外，PDF 1.5 引入了同一个页面不同状态切换的功能（见 566 页“子页面导航”）。

注：PDF1.4 引入了一个不同的机制，作为替代显示，为幻灯片显示，在 9.4 节中所述，“代替显示”。

一个页面对象（见 119 页“页对象”）可能包含两个可选的项，Dur 和 Trans (PDF1.1)，在页显示模式，以指定如何显示。Trans 项包含一个临时的描述类型和过渡期间的视觉使用当通过一个演示从另一页移动到指定页面。表 8.9 显示了过渡时期的内容。（有些项目过渡风格的需要，正如在表中）。

页面中的 Dur 指示页面的持续显示（也称为其提前的时间）：显示该页面自动显示到下一页之前以秒计算的最长时间。（在规定的时间被中断之前用户可以手动设置时间）如果页对象中 Dur 项没有被指定，页面不会自动前进。

表 8.9 过渡项

KEY	TYPE	VALUE
Type	name	(可选) PDF 对象的类型介绍，如果存在，必须为一个过渡。
S	name	(可选) 当通过演示从其他地方移动页面时，使用这过渡类型。

		<p>默认值: R。屏幕显示分裂开, 揭示新的一页。该线可以是水平或垂直移动, 并可以从中心向外边缘移动, 由 Dm 和 M 项分别指定。</p> <p>Split 两行席卷屏幕上, 显示新的一页。该线可以是水平或垂直移动, 并可以从该页面一边或从中心向外移动, 分别由 Dm 和 M 项指定。</p> <p>Blinds 多行, 均匀分布在屏幕上, 并在同一方向同步扫描以显示新的一页。该线可以是水平或垂直的, 按项中指定的 DM。水平线向下移动; 垂直线右移。</p> <p>Box 一个矩形框从页面的一边或从中心向外在页面内掠过, 由 M 项指定, 以显示新的一页。</p> <p>Wipe 一条在屏幕上简单的线条, 通过 Di 项设置, 在指定的方向上从一边到另一边, 以此来显示新页。</p> <p>Dissolve 旧的一页逐步溶解以展示新的页面。</p> <p>Glitter 类似于溶解, 只不过是要整个页面从屏幕的一侧由 Di 项中指定的方向解散。</p> <p>R 新的页面简单替换, 没有特殊的过渡效果; D 项将被忽略</p> <p>Fly (PDF 1.5) 这是在由 Di 指定的方向上, 飞出或者飞入的变化 (如由 M 指定), 或者当 Di 是 None, 这个位置就是屏幕以外的位置。</p> <p>Push (PDF 1.5) 在屏幕上关闭旧的一页幻灯片, 而新的一页幻灯片由 Di 指明的方向由旧页内推动进入。</p> <p>Cover (PDF 1.5) 屏幕上新幻灯片由地指明的方向覆盖旧页面。</p> <p>Uncover (PDF 1.5) 在由 Di 指定的方向上, 关闭旧页幻灯片, 露出由 Di 指定方向新页幻灯片。</p> <p>Fade (PDF 1.5) 新页面透过旧页面逐渐变为可见。</p>
D	number	(可选) 以秒计算的过渡效果的持续时间。默认值为 1
Dm	name	<p>(可选; Split 和 Blinds 过渡方式) 在这个里, 指定的过渡效果发生:</p> <p>H 横向 V 纵向 默认值: H。</p>
M	name	(可选; Split, Box 和 Fly 过渡方式) 为指定的过渡效果方向: I 从页面的边缘外进入 O 由页面中心向外
Di	name or number	<p>(可选; Wipe, Glitter, Fly, Cover, Uncover 和 Push 过渡方式) 指定的过渡动作效果, 逆时针方向从左到右的变换效果。(区别于页对象的旋转项, 那是从顶部顺时针的)</p> <p>下面的数值有效:</p> <p>0 从左到右 90 自下而上(只 Wipe 方式) 180 从右到左(只 Wipe 方式) 270 从上到下</p>

		315 从左上角到右下角(只Glitter方式) 唯一有效的名称值是None; 当SS值不是1.0时, 它只有Fly过渡方式有意义。 默认值:0
SS	number	(可选; PDF 1.5; Fly过渡方式)开始或结束的变化模式。如果M指定一个向内的过渡, 这个变化模式是从SS到1.0以上的过渡。如果M指定一个向外的过渡, 这个变化模式是从1.0到SS的过渡。 默认值:1.0
B	boolean	(可选; PDF 1.5; Fly过渡方式)如果为真, 它将以不透明的矩形框飞出。 默认值:false.

图8.1说明了连续的过渡期 (d在过渡项) 与连续的显示之间的联系 (Dur在页对象中)。请注意, 过渡期为1页 (页图2) 到指定管辖的, 从另外一个网页的网页过渡, 在过渡到下一个页面是由页面的过渡期管理。页面指定的连续过渡(在图中的页2)控制这页到其他页的过渡; 页的过渡通过下一页的连续过渡来控制。



图 8.1 Presentation timing 显示时间

例 8.5 显示了一个页面中的参数, 它用来显示 5 秒。页显示前, 有 3.5 秒的过渡期, 过渡期时两条垂直扫描线从中心向外页面的边缘展开。

例 8.5

```
10 0 obj
<< /Type /Page
/Parent 4 0 R
/Contents 16 0 R
/Dur 5
/Trans << /Type /Trans
/D 3.5
```

```
/S /Split  
/Dm /V  
/M /0  
>>  
>>  
Endobj
```

子页面导航

子页面导航 (PDF 1.5)，不仅可以浏览页面，而且可以查看同一页之间的不同状态。例如，在单页的 PDF 文稿中，有一个子项可以单独打开和系列打开和关闭。在这样的一个例子中，子项会代表可选内容（见第 4.10 节，“可选内容”），每一页的状态会代表作为导航节点。

注：当用户进入演示模式并且演示模式结束时恢复它，查看器应用程序会保存可选项的状态。这将确保了临时的改变，例如，不会影响文档的打印。

导航节点（见表 8.10）按照用户进行导航的要求执行指定动作，例如，按箭头键。该页面上所形成的下一步和导航动作双向链表节点上一个项目。页上的主节点是由页面可选的 PresSteps 项来确定（见表 3.27）。

注意：在演示模式时，建议浏览器仅仅关联导航节点（见第 8.3.3 节，“显示”）。

表 8.10 导航节点规则项

KEY	TYPE	VALUE
Type	name	(可选) 该类型的 PDF 对象描述；它必须是一个导航节点。
NA	dictionary	(可选) 当用户导航前进，行动执行的顺序。
PA	dictionary	(可选) 当用户导航反向时，行动执行的顺序。
Next	dictionary	(可选) 接下来的导航节点，如果有的话
Prev	dictionary	(可选) 以前的导航节点，如果有的话。
Dur	number	(可选) 查看器应用程序自动前进到下一个导航节点的以秒计的最长时间。如果此项不指定，不会发生自动前进。

一个查看器应用程序应该支持当前的导航节点。当用户浏览到一个页面，如果页面有 PresSteps 项，该条目指定的节点成为当前节点。（否则，没有当前节点）如果有向前导航要求提出（诸如导航箭头键下）而且有当前导航节点时，发生以下情况：

1. 由NA指定的操作顺序（如果存在）被执行。

注意：如果 NA 指定了导航到另一页的动作，导航到另一页动作发生了，下一步就不会发生。

2. 由下一个指定的节点（如果存在的话）成为新的当前导航节点。

同样地，如果有一个请求导航反向，且有一个当前导航节点时，发生以下情况：

1. 执行由PA（如果存在）指定的操作顺序。

注意：如果 PA 指定导航到另一个页的操作，这导航到另一个页面的动作执行了，那么之前的不会显示。

2. 由上一个指定的节点（如果存在的话）成为新的当前导航节点。

当节点之间导航时，可以指定过渡效果。这些效果类似于上一节中指定的页面过渡。然而，他们使用一个不同的机制，见 632 页上的“转移行动”。

注：“前进”和“后退”是由用户操作决定，如按向右或向左箭头键，而不是由一个动作目标页决定。

如果有一个请求导航到另一个页面（不论是否有当前节点）且该页包含一个

PresSteps 项，发生下面情况：

1. 由PresSteps代表的导航节点成为当前节点。

2. 如果导航的要求前进或导航请求是随机访问（如通过点击链接），由NA指定的动作会执行以及下一步指定的节点将成为新的当前节点，如上所述。

如果导航的要求是后退，由 PA 指定的动作和 Prev 指定的节点将成为新的当前节点，如上所述。

3. 该浏览器将使新的一页成为当前页并显示它。许多由 Trans 指定的过渡页面会被执行。

8.4、注释

注释联系对象，如说明，声音或视频在一个 PDF 文档的页面的一个位置，或者提供了一种利用鼠标和键盘与用户的交互的方法。PDF 格式包括了多种标准注解类型，在 8.4.5 节中详细介绍，“注释类型。”

很多标准注释类型可以在任何公开或关闭的状态下显示。关闭时，他们出现在一些独特的形式，页面等作为一个图标，一个框，或图章，这取决于具体的注释类型。当用户通过点击激活，如打开一个弹出式窗口显示文字说明（图 8.2）

通过其播放声音或电影，或其它，它也体现了其关联的对象、注释。

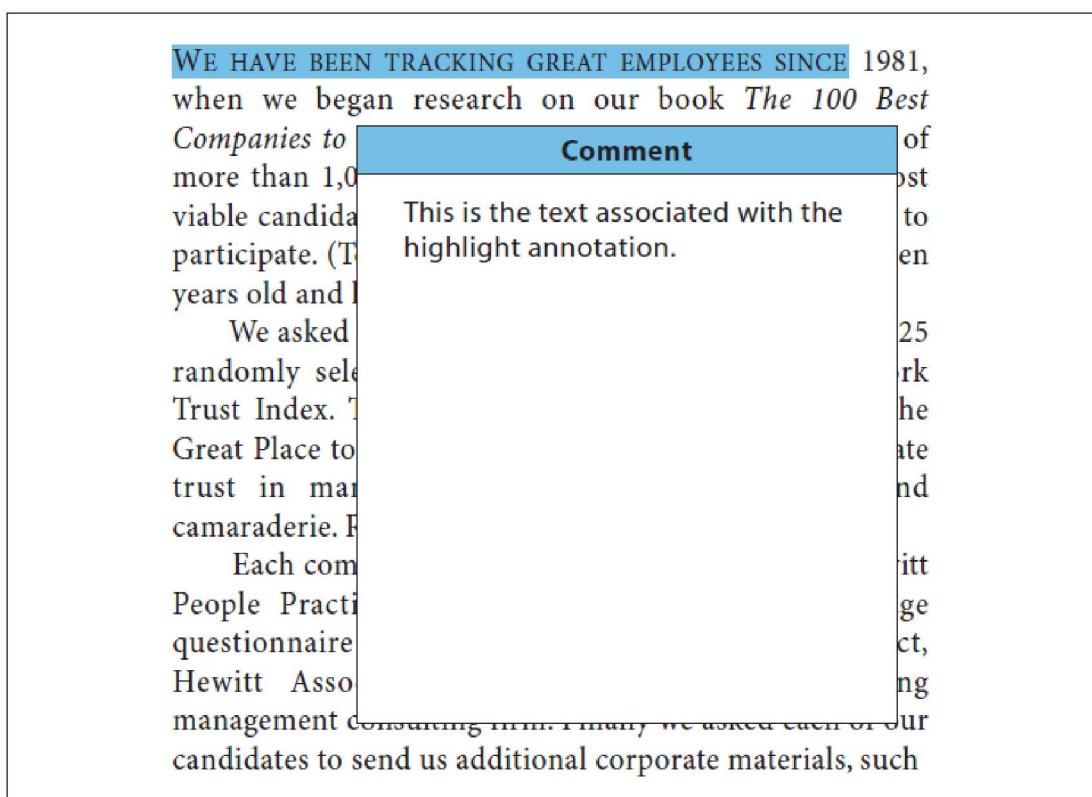


图 8.2 打开注释

浏览器可以允许用户通过使用键盘（特别是 Tab 键）浏览一个注解页面，执行情况的说明见 PDF 1.5 的 77 页附录 H，在一个页对象中，PDF 可能会用可选的标签进入有序的导航（见表 3.27）。以下是此项可能的值：

- R (行顺序)：注解查看在整个页面水平行方向。行内一个方向被确定，在浏览器参数中的方向项（见第8.1节，“浏览器首选项”）。在最上面行的第一个注解被访问。当遇到一行结束，下一行中的第一个注解被访问。
- C (列顺序)：注释查看在垂直向上和向下运行的页面列。列被设置在浏览器参数中的方向项（见第8.1节，“浏览器首选项”）。在第一列的顶部第一个注释被查看。当遇到一列结束，在下一列的第一个注释被访问。
- S (结构顺序)：注解以他们在结构树（见10.6节，“逻辑结构”）中出现的顺序被浏览。对于那些不包括在结构树中的注解顺序依赖于应用程序。

注：以上说明，假设页面在 *Rotate* 项指定方向上被浏览。

每个注释类型的反应，是由称为注释处理程序的软件模块来执行。标准注释类型被直接建立在 PDF 浏览器应用程序中，为处理程序是的；其他类型的处理程序可以插件扩展方式被供应。

8.4.1、注释规则

在页面对象项中可选的 `Annots` 项（见 119 页“页对象”）持有的注释数组，每个代表与所给出的页面相关联的注释。表 8.11 显示了所有注释必需和可选的项。这项可能包含其他针对特定的注释类型的项，参见 8.4.5 节中，“类型的描述“对细节的描述。

注：一个给定的注释可从只有一页 `Annots` 数组引用。试图共享一个多页的注释可能产生不可预知的行为。这项规定只适用于注释本身，而不是附属的对象，它可以被多个注释共享而不会造成任何问题。

表 8.11 所有注释规则通用项

KEY	TYPE	VALUE
Type	name	(可选) PDF 对象的类型描述，如果存在，必须为一个注解
Subtype	name	(可选) 该注释的类型描述，见第 580 页上的特定值表 8.16。
Rect	rectangle	(可选) 矩形注释，确定了用户在页上默认区域注释的位置。
Contents	text	(可选) 注释以文本的形式显示，或者如果这种类型的注释不显示文字，这种注释内容的以可读的形式替代表述。在这两种情况下，当取该文档的内容支持部分用户或作其他用途时，这个文本是非常有用的（见第 10.8.2 节，“替代说明”）。参阅第 8.4.5 章节，“注释类型”中对这个项目的每个注释类型含义更多的细节。
P	dictionary	(可选；PDF 1.3；在 FDF 文件中未使用) 间接引用对象的页面与本注释关联。 注：此项是必需的移交逃犯行动（PDF 1.5 相关的屏幕说明，见 602 页“屏幕注释”以及 630 页“演示动作”）。
NM	text	(可选；PDF 1.4) 注释的名称，唯一标识页面上所有注释的一个文本字符串。
M	date	(可选；PDF 1.1) 批注最近修改的日期和时间。首选的格式是一个在 3.8.3 节所描述的日期字符串，但浏览器被希望能够接受和显示的任何格式的字符串。（具体情况见附录 H 第 78 页）
F	integer	(可选；PDF 1.1) 指定各种特性的注释标志集（见第 8.4.2 节，“注释标记”）。默认值：0.
BS	dictionary	(可选；PDF 1.2) 边框样式的注释指定了它边界的特征（见 8.4.3 节，“边框样式”，也可见附录 H 第 79

		和86页情况说明)。 注：此项还指定了宽度和由线、方、圆的线条绘制的虚线模式，和墨迹注释。见对边界说明（如下）的其他信息。
AP	dictionary	(可选；PDF 1.2) 指定的外观，在页面上注释如何形象的被展现（见第8.4.4节，“外观流”，也注意到在附录H79和80页）。个人注释处理程序可能会忽略此项目，并提供自己的外观。
AS	name	(如果AP包含一个或多个子项，则该项必填；PDF 1.2) 该批注的外观状态，它从外观子项选择适用的外观流（见第8.4.4节，“外观流”，并具体情况见附录H第79页）。
Border	array	(可选) 指定的注释边界的参数数组。边界被指定为一个圆角矩形。 在PDF 1.0中，该数组包含三个数字，用以确定横向圆角半径，垂直角半径，边框宽度，默认的用户空间中的所有单位。如果角半径是0，边界已方形（不四舍五入）角度；如果边框宽度为0，没有边界设置。（情况说明见附录H 81页）。例如，一个[0 0 1 [3 2]]的边框值指定了一个边框是1单位宽、方角，3个单位和2个单元的交替的虚线绘制。在PDF 1.1中，可能有四分之一的数组元素，一个可选的数组定义了一个破折号的样式并且在绘制边界时会被使用。数组和图形状态参数以相同形式被指定（见187页“虚线模式”）。例如，[0 0 1 [3 2]]值指定一个边界项：宽，方角，3个单元，2单元的差距交替破折号绘制。请注意，如果没有活跃状态被指定，该状态被假定为0。（具体情况的说明见附录H 82页） 注：在PDF 1.2或更高版本，此项目可能会被忽略以支持BS项（见上文），见附录H86页的具体情况的说明。 默认值：[0 0 1]。
C	array	(可选；PDF 1.1) 范围在0.0到1.0的三个元素的数组，代表在DeviceRGB色彩空间色彩的组成。这种颜色是用于下列用途： •该批注的图标关闭时的背景 •注释的弹出窗口的标题栏 •链接注释的边界
A	dictionary	(可选；PDF 1.1) 当注释被激活，一个操作将要被执行，（参见8.5节，“动作”）。 注：在链接注释中时如果Dest项存在，本项不准许（见587页“链接注释”）。另外请注意，在视频注释中每项都有不同的含义（见601页“视频注释”）。

AA	dictionary	(可选; PDF 1.2) 一个额外的动作, 定义了注释在响应各种触发事件的行为(见第8.5.2节, “触发事件”)。在规范发布时, 此项仅被窗口注释使用。
StructParent	integer	(如果注释是一个结构内容元素, 该项必须; PDF 1.3) 在父结构树中, 注释项的整型值(见797页“从内容项中查找结构元素”)。
OC	dictionary	(可选; PDF 1.5) 可选的内容组或内容成员(见4.10章节, “可选的内容”), 它们定义了可选内容注释的特性。在注释被显示之前, 它的清晰度取决于一项值, 该项与在F项中定义的注释标志一样。如果它被隐藏, 注释会被跳过, 就像在文档中它根本不存在似的。

8.4.2、注释标记

注释 F 项的值是一个 32 无符号整型, 它包含指定指定各种注释参数的标志。标志位从低位 1 到高位 32。表 8.12 显示了标志的具体含义; 所有没定义的标志位被保留且必须置 0

表 8.12 注释标志		
BIT	POSITION	NAME
1	Invisible	如果设置, 如果它不属于标准注释类型, 将不显示注释, 同时, 没有注释句柄是可用的。如果清零该项, 通过它的显示设置项使用一个指定的显示流, 显示一个未知的注释(将第 8.4 节, “显示流”)。
2	Hidden	(PDF 1.2) 如果设置, 无论它的注释类型是什么或者注释句柄是否有效不会显示或打印注释或者允许它与用交互。在这种情况下, 屏幕空间受限, 隐藏和显示注释被用于关联显示流(见 8.4.4 节, “显示流”), 以显示类似于在线帮助系统备用的信息。(见附录 H 的第 83 页)
3	Print	(PDF 1.2) 如何设置, 当页面被打印时, 打印注释。如果清零, 无论它是否在屏幕上显示, 都将打印注释。这非常有用, 例如, 在打印页面, 注释展示出来的交互按钮, 没有明确的服务目的。(具体细节见附录 H83 页)。
4	NoZoom	(PDF 1.3) 如果设置, 不会为了页面匹配放大而缩放注释的显示。无论页面如何放大, 页面上注释的位置(被固定在矩形注释的左上角)保持固定。见下文作进一步讨论。
5	NoRotate	(PDF 1.3) 如果设置, 不会为匹配页面的旋转而旋转注释的显示。无论页如何旋转, 矩形保持在页面注释左上角的固定位置, 见下文作进一步讨论。
6	NoView	(PDF 1.3) 如果设置, 在屏幕上不显示注释或允许它与用户交互。这个注释可能被打印, 但是会刻意隐藏屏幕显示和

		用户交互的目的。
7	ReadOnly	(PDF 1.3) 如果被设置，不允许用户和注释交互。这只是可能显示或者打印（取决于NoView的设置和打印标志）但是，不会回应鼠标点击或回应鼠标动作而改变它的显示。注意：这个标志做为注释控件被忽略；它的功能是被和信息栏有关联的ReadOnly标志包含。
8	Locked	(PDF 1.4) 如果设置，不允许注释被删除或它的属性（包括位置和大小）被用户修改。无论如何，这个标志不是限制改变注释的内容，例如 a form field 的值(见具体细节附录 H 第 84 页)。
9	ToggleNoView	(PDF 1.5) 如果被设置，对于某些事件，NoView 标志项取反。一个典型的用法是仅仅当鼠标划过它的时候，出现一个注释。

如果 NoZoom 标志被设置，注释保持在屏幕上相同的固定大小且不会受被显示页的放大级数的影响。类似的，如果 NoRotate 标志被设置，当页面旋转时，注释保持它原来在屏幕上的方向（通过改变页对象的 Rotate 项；见 119 页“页对象”）。

无论是哪种情况，通过确定注释方框左上角坐标，因而注释的位置是固定的，它通过注释元素中的 Rect 项来定义，同时在用户默认区域被注释。当用户默认区域被缩放或旋转，在原来的区域和改变了的区域，注释方框其他三个坐标位置是不同的。浏览器将会自动的变化。尽管如此，它不会改变注释的 Rect 项，该项描述了未缩放和未旋转的注释的联系。

例如，图 8.3 显示了一个注释，当页面顺时针 90 度旋转，它的 NoRotate 标志被标记。注释保持在左上角同时指向默认的用户区域；该注释围绕一个中心。

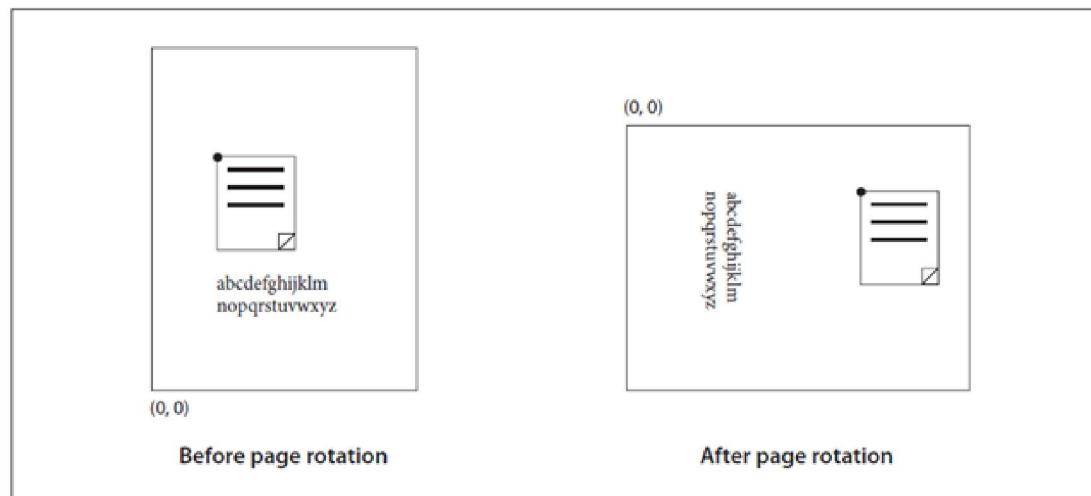


图 8.3 坐标调整与 NoRotate 项

8.4.3、边框类型

当显示或者打印时，一个注释可能被一个可选的边框围绕。如果存在，边缘被完整的绘制在矩形注释的内侧。在pdf规范1.1中，边缘的参数通过Border项来定义（见570页表8.11）。从pdf规范1.2开始，在边缘类型注释的BS项中，许多注释的类型被特定的边框参数替代

注意在附录H第86页。每个元素也被指定宽度使用，同时用直线、方形、圆形和ink注释绘制线条做为边缘样式。表8.13总结了边缘样式的内容。如果Border或BS项存在，边缘用宽度为1个像素的实线来绘制。

表 8.13 边框类型规则项		
KEY	TYPE	VALUE
Type	name	(可选) pdf对象的类型描述；如果存在，必须是一个边界类型设置选项的边缘。
W	number	(可选) 以像素为单位的边缘宽度。如果这个值是0，没有边缘。默认值：1.
S	name	(可选) 边缘样式： S (实心)一个围绕注释的实心矩形框。 D (虚线)一个围绕注释的虚线矩形框。这个虚线被D项定义（见下）。 B (倾斜)在页面表面上，一个模仿凹凸的矩形框。 I (插入)一个模拟刻纹的矩形隐藏在页面的表面。 U (强调)沿着长方形底部的一行注释 其他边缘样式可以以后在定义（详细细节见附录H第86页）。 默认值：S.
D	array	(可选)一个虚线数组定义一个虚线的样式用于绘制一个虚线边缘（上文D边框样式）。这个数组被定义与图像状态的虚线条有相同的格式（见187页“虚线条样式”） 虚线没有被指定且假设为0。例如，D项指定了一个用3个像素和2个像素交替绘制的边缘。 默认值：[3]。

从pdf规范1.5开始，许多注释（方形，圆形和多边形）可能有一个BE项，它是一个有效的设置选项定义的边缘，有效应用到注释的边框。它的选项在表8.14中列出。

表 8.14 边框效果规则项

KEY	TYPE	VALUE
S	name	(可选)一个象征边界效应的名字。可能值是： S 无效果：边界通过注释设置选项的BS项来描述 C 边界表现阴影。宽度和由BS指定的虚线数组有效。 默认值：S.
I	number	(可选；仅仅 S 值是 C 时，该值有效)一个描述效果强度的数字。支持值的范围从 0 到 2。默认值：0。

8.4.4、显示流

从 pdf 规范 1.2 开始，一个注释可以指定一个或多个显示流供，在早期版本中，简单边缘和可用的颜色参数列表可供选择。显示流使注释直观的显示，并用不同的方式去影响与用户的交互。显示流是由 **Xobject**（见 4.9 节，“构成 XObject”）构成：在注释内部一个独立的内容流被提供。下面的方法用来演示，从 **Xobject** 显示的（被 **Matrix** 项定义；见表 4.45）坐标系统到在默认用户区域的注释的矩形框。

算法8.1

1. 显示范围（被**Bbox**项定义）使用矩阵来改变，以在随机方向上产生一个四边形。这改变的显示是包含了这个四边形的最小的垂直矩形。
2. 一个矩阵A计算尺度，改变转换盒的外观以配合该批注的矩形的边缘（由矩形入口指定）。转换外观盒的图的左下角（最小的X和Y坐标的角）和右上角（最大X和Y坐标的角）对应注释的矩形的相应角落。
3. 矩阵和 A 相连组成矩阵 AA 然后从表面坐标系统映射到默认的用户空间：

$$AA = A \quad Matrix$$

如果**NoZoom**或**NoRotate**标志被设置（见8.4.2章节，“注释标志”），注释可能会进一步缩小、旋转。许多注释应用的变换也被应用于关于它的显示。

在pdf规范1.4版本中，一个注释显示可以包含透明性。如果显示的流设置选项不包含一个**Group**项，它会被做为**non-isolated**、**non-knockout**的透明集合处理。否则，在集合设置选项中，定义的**isolated** and **knockout**的值被使用。

变换的集合被综合一个背景，它由一些以前描绘的注释页面内容构成，它使用了融合 **Normal** 模式。（见具体细节附录 H 第 88 页）

注意：如果一个透明注释显示，它透过没有使用显示流显示的注释，那这效果是现实相关的。这是因为每个注释有时不遵从 Adobe 图像模型绘制。加强一个透明注释显示效果是现实相关的。

显示能被定义成三种不同的方式：

- 当注释没有与用户交互，正常显示。打印时也使用这种显示方式。
- 当用户移动光标进注释的有效区域且没有按下鼠标键时，翻转显示。
- 当鼠标在注释的有效区域被按下或长按，将向下显示。

注意：这个鼠标术语象征了一个通用的指向设备，它控制屏幕上光标的位置，同时至少有一个按钮能被按下、长按。见第 8.5.2 节，“事件引发”进一步讨论。

正常、翻转、向下显示都被定义在一个显示设置中，它依次是在注释设置项中 AP 项的值。（见 570 页表 8.11）。表 8.15 列出了显示项了的内容。

表 8.15 显示规则项

KEY	TYPE	VALUE
N	stream or dictionary	(必须) 注释的普通显示
R	stream or dictionary	(可选) 注释的转动显示。默认值：N 项的值。
D	stream or dictionary	(可选) 注释的向下显示。默认值：N 项的值。

显示设置中的每个项目可能包含一个单一的外观流或外观流的子项。在后一种情况下，子项定义了多重显示流对应于不同的注释的显示状态。例如，一个注释较代表了一种查看交互，它有名为开和关的两个显示状态。它的出现可能会被定如

```
/AP << /N << /On formXObject1
/Off formXObject2
>>
/D << /On formXObject3
/Off formXObject4
>>
>>
```

在检查和未经检查的状态中，formXObject1 和 formXObject2 定义复选框的正常外观，并提供 formXObject3 和 formXObject4 视觉反馈，如当用户点击它时，加粗其轮廓。（当用户移动光标在复选框上时在没有按下鼠标按钮，没有出现 R 项的定义是因为没有特殊需要。）之间的外观检查，未选中状态，是由在注释字

典坚定的选择项。(见570页表8.11)。检查显示状态与否是由注释设置中AS项觉得的。

注：其中一些标准的PDF注释类型通过插件扩展，如送注释作为所有自定义注解以及由第三方定义的类型。如果插件在一个特定的注释类型上不可用，PDF浏览器应用程序应显示其正常的注释(N)的外观。浏览器应用程序也应尝试提供平日的合理，如没有显示(如果注释的作为入指定的一个外观状态而没有出现在字典中定义的外观)。如果常规注释指定的一个外观状态而没有出现在设置外观的定义中，浏览器应用程序也应尝试提供一个合理的解决方法。

为方便管理出现的重复的显示流，在PDF文档的名称条目(见3.6.3节，“名字选项”)中，AP可以包含一个名字的映射名称的字符串显示流。这个名字字符串有没有标准的含义；没有PDF对象涉及显示流的名字。

8.4.5、注释类型

PDF格式支持标准注释类型，在表8.16中列出。下面部分详细描述了每种类型。插件可以添加新的扩展注释类型和标准的类型，以后它们可能会进一步增加。(见附录H第89页。)在表8.16的第一列中的值代表了注释值的子项值。第三列指示是否注解一种标记注释，如下面，“标记注解说明”。该节还提供了更多关于不同内容的信息项值的注释类型。

表 8.16 注释类型

ANNOTATION	TYPE	DESCRIPTION	MARKUP?
Text	文本注释	Yes	第 586 页“文本注释”
Link	连接注释	No	第 587 页“连接注释”
FreeText	(PDF 1.3) 开放文本注释	Yes	第 588 页“开放文本注释”
Line	(PDF 1.3) 行注释	Yes	第 590 页“行注释”
Square	(PDF 1.3) 正方形注释	Yes	第 593 第“正方形和圆形注释”
Circle	(PDF 1.3) 圆注释	Yes	第 593 页“正方形和圆形注释”
Polygon	(PDF 1.5) 多边形注释	Yes	第 595 页“多边形和折线注解”
PolyLine	(PDF 1.5) 折线注释	Yes	第 595 页“多边形和折线注解”
Highlight	(PDF 1.3) 高亮注释	Yes	第 596 页“文本标记注解”
Underline	(PDF 1.3) 强调注释	Yes	第 596 页“文本标记注解”
Squiggly	(PDF 1.4) 弯曲强调注释	Yes	第 596 页“文本标记注解”

StrikeOut	(PDF 1.3) 删除线注释	Yes	第 596 页 “文本标记注解”
Stamp	(PDF 1.3) 橡皮图章注释	Yes	第 598 页 “橡皮图章注解”
Caret	(PDF 1.5) 插入符号注释	Yes	第 597 页 “插入符号注释”
Ink	(PDF 1.3) 墨迹注释	Yes	第 598 页 “墨迹注释”
Popup	(PDF 1.3) 弹出注释	No	第 599 页 “弹出注释”
FileAttachment	(PDF 1.3) 文件附件注释	Yes	第 600 页 “文件附件注释”
Sound	(PDF 1.2) 声音注释	Yes	第 600 页 “声音注释”
Movie	(PDF 1.2) 视频注释	No	第 601 页 “视频注释”
Widget	(PDF 1.2) 插件注释	No	第 603 页 “插件注释”
Screen	(PDF 1.5) 屏幕注释	No	第 602 页 “屏幕注释”
PrinterMark	(PDF 1.4) 打印机的标志注释	No	第 605 页 “打印机的标志注释”
TrapNet	(PDF 1.3) 网络限制注释	No	第 605 页 “网络限制注释”
Watermark	(PDF 1.6) 水印注释	No	第 606 页 “水印注释”
3D	(PDF 1.6) 3D 注释	No	第 747 页 “3D 注释”

标记注释

在 8.4.1 章节中, “Annotation Dictionaries” 提及, 注释的意义不同于注释的类型。通常, 它作为注释的文本显示, 或如果这个注释不显示文本, 以通俗易懂的方式替代注释的描述。无论如何, 当提前文档内容支持有问题的用户或做为其他用途, 内容项是非常有用的(见 10.8.2 节, “Alternate Descriptions”)。

许多注释类型被定义为标记注释, 因为它们主要是为了标记 PDF 文件(见表 8.16)。这些说明文字作为注释的一部分出现, 可通过其他方式显示在浏览器中, 如, 在评论窗格。

标记注释可分为以下几组:

- 自由文本注释直接在页面上显示文本。注释的内容项指定了要显示的文本。
- 大部分其他标记注解有一个关联的可能包含文字的弹出式窗口。当弹出窗口被打开时, 注释的内容项指定要显示的文本。这些内容包括文本、线条、方形、圆形、多边形、折线、突出、强调、弯曲、下划线、删除线、橡皮擦、标记、插入符号、签名和文件附件说明。
- 声音批注没有弹出式窗口, 但也可能有关联由目录项中指定的文本。

注意: 当内容被分成段落, 应使用回车(没有的话, 例如, 使用一个换行符)。

注意: 一个子集被称为标记注释的文本标记注释(见 596 页, “文本标记注释”)。

其余的注释类型不被视为标记注释：

- 这典型的弹出注释类型不是它自己显示的；它和种注释标记连续在一起，种注释标记利用它显示文本内容。

注意：如果它没有更上一层注释，弹出注释的内容项有意义；这种情况下，它显示了注释的文本内容。

- 对于所有其他的注释类型（链接，视频，插件，打印和TrapNet），内容规定了注释内容的替代描述，当提取文档内容以支持部分用户或其他目的，它十分有用（见10.8.2章节，“替代描述”）。

表8.17列出条目适用于所有标记注释。

表 8.17 标记注释附加指定项		
KEY	TYPE	VALUE
T	text string	(可选；PDF 1.1) 当开放和活跃，标签的文本将显示在注释的弹出窗口的标题栏。按照惯例，此项标识添加注解的用户。
Popup	dictionary	(可选；PDF 1.3) 间接引用到一个弹出注释或编辑与文字关联这个注释。
CA	number	<p>(可选；PDF 1.4) 不断模糊的值，它用于绘制注释（见7.1节，“透明度概述”以及7.2.6节，“形状和模糊计算”）。此值适用于在其封闭状态注释所有的可见元素（包含它的背景和边框），但是不包括注释打开时出现的弹出窗口。</p> <p>指定的值不使用（见8.4.4节，“显示流”）；在这种情况下，显示流必须指定任何透明度。（但是，如果浏览器重新生产注释的显示流，它可能将CA值纳入显示流的内容之中。）</p> <p>隐式的混合模式（见第7.2.4节，“混合模式”）是正常的。</p> <p>默认值：1.0.</p> <p>注意：如果没有明确的显示流被定义，它根据执行情况绘制，这意味着不必遵循Adobe图像模型。在这种情况下，该项的效果也依赖于执行情况。</p>
RC	text string or text stream	(可选；PDF 1.5) 当注释被打开时，文本字符串（见642页“丰富的文本字符串”）被显示在弹出式窗口中。
CreationDate	date	(可选；PDF 1.5) 注释被创建的时间和日期（见3.8.8节，“日期”）
IRT	dictionary	(如果RT项存在，该项必须，否则可选；PDF 1.5) 这注释是“应答”一个引用注释，注释必须在文

		档相同页。这两者之间的关系是由注释的 RT 项指定。如果这项在 pdf 文件中（见第 8.6.6 节，“形成数据格式”）存在，它的类型不是一个 dictionary，但是，一个文本字符串包含了这注释被替代的 NM 项的内容，以便注释不是在相同的 pdf 文件中回应。
Subj	text string	(可选; PDF 1.5) 文本代表一个主题的简短描述，该描述正在由注释解释。
RT	name	(可选; 只有 IRT 存在，该项才有意义; PDF 1.6) 一个指定的名字将（这“回应类型”）注释和一个由 IRT 指定的项关联到一起。 有效值： R 注释被看做是回应由 IRT 中指定的注解。浏览器应用程序不应该显示单独的答复注释，而应以评论的形式显示。 Group 由 IRT 指定的注释被分组；见下面的讨论。 默认值: R
IT	name	(可选; PDF 1.6) 一个描述标记注释意图的名字。允许浏览器请区分不同用途和单一标记注释类型的行为。如果此项不存在或它的值与注释类型相同，注释没有明确的意图，并在浏览器应用程序中表现出通用方式。 在 PDF 规范 1.6，自由文本（表 8.21），线条（表 8.21）和多边形（表 8.21）注释已被定义了意向，他们的值枚举在相应的表中。

在 pdf 规范 1.6 中，一个注释设置被分组，以至于当用户与它们交互的时候，它们一个功能单一项。构成一个主要注释的组它必须没有 IRT 项，且一个或更多的附属注释必须有涉及主注释的 IRT 项，同时 RT 项的值在一个组。

主要注释中的许多项目是由“属性组”创建的；在相应的条目下属注释应该被忽略。这些条目的内容（或 RC 和 DS）。控制集合中的注释的选项，例如移动、剪切和复制，被浏览器以的在选项组动作创建

注：主注释回复而不是附属注释；也就是说，没有一组 RT 值。

注释状态

从 pdf 规范 1.5 开始，注释可能有一个特定作者的状态与它们相关。这个状态不是被注释本身指定的，而是在一个独立的文本注释中。这个文本注释依靠它的 IRT（“答复”）项（见表 8.20）引用原本的注释。

表 8.18 注释状态

STATE	MODEL	STATE
Marked	Marked	注释并被用户标记
	Unmarked	注释并没有被用户标记（默认）。
Review	Accepted	用户同意更改。
	Rejected	用户不同意更改。
	Cancelled	这种更改已被取消。
	Completed	这种更改已经完成
	None	用户对更改没有任何表示（默认）。

注解可以被看作是最初默认状态。由一个用户所做状态的更改表现在与文本注释相关的以下项目：

- 由T项（见表8.17）指定用户
- IRT项（见表8.20）引用原来的注释。

状态和状态模式（见表8.19）更新了原有的注释状态为指定的用户。

在应答前加入文字注释应答指定的用户，额外的状态会改变。

文本注释

文本注释是一个附加到PDF文档中一个点的“便条”。当关闭时，注释显示为一个图标；开启时，它显示一个弹出窗口，它包含在查看器应用程序中所选择的字体和大小的说明文字。文字注释不能是缩放和旋转的页面；他们表现得好像NoZoom和NoRotate注释标志（见573页表8.12）的总集合。表8.19显示了针对这种类型的注释条目。

表 8.19 文本注释指定的附加项

KEY	TYPE	VALUE									
Subtype	name	(必须) 该注释的类型描述；必须为文本注释。									
Open	boolean	(可选) 一个指定是否应先公开显示注释的标志。默认值：false（关闭）。									
Name	name	(可选) 该图标的名称，用于显示注释。查看器应用程序应提供预定义的图标，至少遵循下面标准名称： <table style="margin-left: 20px;"> <tr> <td>Comment</td> <td>Key</td> <td>Note</td> </tr> <tr> <td>Help</td> <td>NewParagraph</td> <td>Paragraph</td> </tr> <tr> <td>Insert</td> <td></td> <td></td> </tr> </table> 也可以支持其他名称。默认值：Note。 注：如果存在注释的AP项，优先接管名称项，见570页表8.11和第8.4.4章节，“显示流。”	Comment	Key	Note	Help	NewParagraph	Paragraph	Insert		
Comment	Key	Note									
Help	NewParagraph	Paragraph									
Insert											
State	text	(可选；PDF 1.5) 原注释设置的状态；见上面“注释状态”。 默认：“Unmarked”，如果 StateModel 是“Marked”；									

		如果 StateModel 是 “Review”, “None”。
StateModel	text	(如果 State 存在, 否则可选; PDF 1.5) state model 对应于 State; 见上面“注释状态”

示例 8.6 显示了一个文本注释定义。

例 8.6

```
22 0 obj
<< /Type /Annot
/Subtype /Text
/Rect [266 116 430 204]
/Contents (The quick brown fox ate the lazy mouse.)
>>
endobj
```

Link Annotations 连接注释

一个链接注释表示一个链接到其他地方的超文本链接（见 8.2.1 节，“目的地”）或将要执行的操作（第 8.5 节，“动作”）。表 8.20 显示了这种类型的注释具体的注解条目。

表 8.20 连接注释指定的附加项

Subtype	name	(必须) 字典描述的注释类型; 必须是一个行注释的连接。
Dest	array, name, or string	(可选; 如果 A 项存在, 不允许) 当注释被激活, 一个注释被显示 (见第 8.2.1 节, “目标”; 也可见附录 H 第 90 页)。
H	name	(可选; PDF 1.2) 当在它的有效区域内鼠标按下或者保持按下状态, 注释的强调描述会产生有效反应。 N (无) 没有强调 I (倒置) 倒置注释边框内容。 O (轮廓) 倒置注释边框。 P (下推) 显示注释, 就如同它是被推低的页面。详细情况见附录 H 第 91 页。 默认值: I. 注意: 在 pdf 规范 1.1 中, 总是通过反相注释矩形内的颜色来突出。
PA	dictionary	(可选; PDF 1.3) 一个 (见 624 页 “URI 动作”) 与这个注释相关联的 URI 动作。当网络捕获 (第 10.9 节, “网络捕获”) 改变了一个从 URI 到跳转动作的注释 (616 页, “跳转动作”), 它用该项来保持来

		自 URI 操作的数据，以至于，它可以改变目标页使跳转操作随后被删除。
QuadPoints	array	(可选; PDF 1.6)一个 $8 \times n$ 个元素的数组，它指定了默认用户区域中 n 个四边形坐标，且构成该范围的连接时有效的。每个四边形坐标按 $x_1\ y_1\ x_2\ y_2\ x_3\ y_3\ x_4\ y_4$ 顺序被给出，它指定了四边形按逆时针方向的四个顶点。方向指向，如当提供一个突出的边框样式，这个四边形的底部是由 (x_1, y_1) 和 (x_2, y_2) 形成的线条。 如果该行不存在，或者浏览器不能识别它，这个被 Rect 项指定的区域将被使用。如果数组中的任何坐标指定的是矩形外的区域，该项应该被忽略。

示例 8.7 显示了在目的地文件中，一个链接注释跳转到其他地方。

例 8.7

```
93 0 obj
<< /Type /Annot
/Subtype /Link
/Rect [71 717 190 734]
/Border [16 16 1]
/Dest [3 0 R /FitR -4 399 199 533]
>>
endobj
```

开放的文本注释

一个自由文本注释 (PDF 格式 1.3) 显示文本直接在页面上。不同于一般的文本 (见 586 页“文本注释”), 一个开放的文本注释没有开放或关闭状态；取而代之的是被弹出窗口显示，且文本始终可见表 8.21 显示了指定注释类型的项。

639 页的“变化文本”描述了使用这些项在注释中去生成文本显示的过程

表 8.21 开发文本注释指定的附加项		
KEY	TYPE	VALUE
Subtype	name	(必须)注释类型的描述；它必须是一个开放的文字注释开发文本。
DA	string	(必须)默认的显示字符串被用于格式化文本(见639页“变化文本”)。 注：如果注释字典的 AP 项存在，优先替代 DA 项；见 570 页

表 8.11 和第 8.4.4 节, “显示流。”		
Q	integer	(可选; PDF 1.4)一种指明的编码形式(理由)被用于显示注释的文字: 0 左对齐 1 中间对齐 2 右对齐 默认值: 0 (左对齐).
RC	text string or text stream	(可选; PDF 1.5)丰富的文本字符串(见 642 页“丰富的文本字符串”)生成的注释外观。
DS	Text string	(可选; PDF 1.5)page 642. 一个默认字符串类型, 见 642 页“丰富的文本字符串”的描述。
CL	array	(可选; PDF 1.6)一个四元素或六元素的数组指定一个对自由文本注释的附加标注线。六元素 [$x_1 \ y_1 \ x_2 \ y_2 \ x_3 \ y_3$] 代表默认用户空间中的开始点、拐点和结束点坐标, 见下面图 8.4。四元素 [$x_1 \ y_1 \ x_2 \ y_2$] 代表开始点线条结束点坐标。
IT	name	(可选; PDF 1.6)一个描述自由文本注释意图的名字(见表 8.17)。PDF 规范 1.6 中的有效值是 FreeTextCallout, 它意味着注释的目的是要作为一个标注同时 FreeTextTypeWriter 意味着注释的功能是作为一个点击方式或打字机对象。

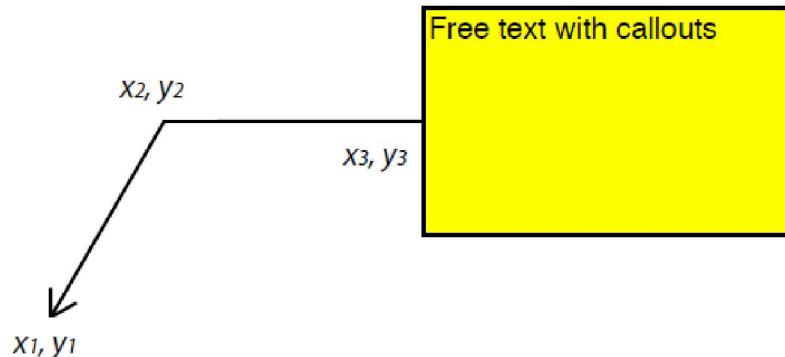


图 8.4 开发的标注文字注释

行注释

一个行注释(PDF 1.3)显示了一个简单的在页面上直线的行。当它别打开时,

它显示出一个弹出窗口，该弹出窗口包含了相关注释的文本。表 8.22 显示了指定的注释类型字典项。

TABLE 8.22 Additional entries specific to a line annotation 行注释指定的附加项		
KEY	TYPE	VALUE
Subtype	name	(必须) 字典描述的注释类型；它必须是一个行注释的行。
L	array	(必须) 一个四元素的数组 [x1 y1 x2 y2]，它指定了在用户区域的线条的开始和结束坐标。 注意：如果 LL 项存在，这个值代表线条指向的终点，而不是自身的终点。见图 8.5。
BS	dictionary	(可选) 一个边界样式指定了宽和线条样式，以便在绘制线条时使用。 注意：如果存在，注释的 AP 项，它将优先于 L 和 BS 项；见 570 页表 8.11 和 第 8.4.4 节，“显示流”。
LE	array	(可选；PDF 1.4) 两个名称的数组指定了用于绘制路线的行结束样式。这第一和第二个数组项，通过在行数组中的第一和第二对坐标，(x1, y1) 和 (x2, y2)，依次指定了线条终点结束样式。表 8.23 显示了可能的值。默认值：[/None /None]。
IC	array	(可选项；PDF 1.4 版后) 该数组有 3 个从 0.0 到 1.0 范围的数，它们指定了注释（在 Device RGB 范围内）的 interior color（内部颜色）：该颜色用于填充注释线段的结尾（见表 8.23）。如果没有这个参数，注释线的结尾将设为透明。
LL	number	(当有 LLE 时是必需的，否则是可选项；PDF 1.4 版后) 带箭头注释线的长度是在默认的用户空间范围内，从每行的终点垂直扩展 行本身，如图 8.5 所示。好的方面，当箭头线从起始点到结束点是顺时针方向出现（用 L 表示）；坏的方面指出了该位置的方向。 默认值：0（没有箭头线）。
LLE	number	(可选项；PDF 1.6 版后) 一个非负数描述了从箭头线固定的 180 度方向扩展的长度，如图 8.5。
Cap	boolean	(可选项；PDF 1.6 版后) 若为真，YONG 参数 Contens 或 RC 说明的文本将复制作为注释线标题的外观，如图 8.6 所示。该文本用某种方式渲染来分配到显示空间，这需要考虑多种因素，例如书写方向。
IT	name	(可选项；PDF 1.6 版后) 一个名字用来描述该行注释的意图（另见表 8.17）。PDF 1.6 版中的有效值是 LineArrow，这意味着注释目的是要作为一个箭头，LineDimension 意味着注释目的是要作为一个尺寸

		线。
--	--	----

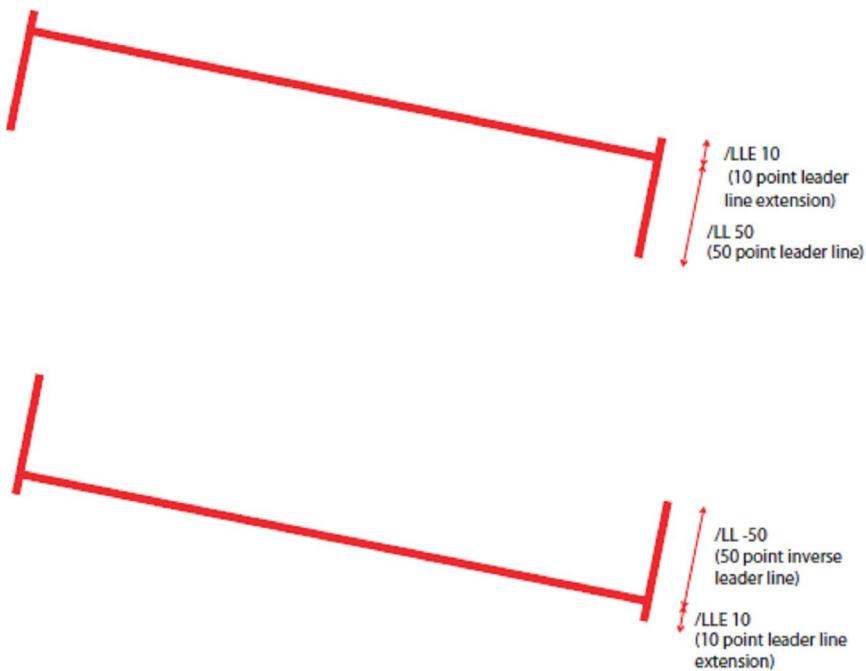


FIGURE 8.5 Leader lines

图 8.5 箭头线

This is a caption

This is a caption that is longer than the line

图 8.6 带标题的注释线

表 8.23 注释线结束款式		
名字	外观	描述
方形		一个方形用注释的内饰颜色填充, 如果有的话
圆形		一个圆形用注释的内饰颜色填充, 如果有的话
菱形		一个菱形用注释的内饰颜色填充, 如果有的话

开放式箭头		两条短线以锐角相交来形成开放式箭头
闭合式箭头		两条短线以锐角相交来形成开放式箭头（如上）在和第三线连接在一起，形成一个封闭的三角箭头用注释的内饰颜色填充，如果有的话
无		线结束没有其他东西
柄型		(PDF1.5 版) 在端点垂直线本身短线
右端开放式箭头		(PDF1.5 版) 从两个相反方向短线形成的开放式箭头
右端闭合式箭头		(PDF1.5 版) 一个封闭的三角箭头方向相反的形成的闭合式箭头
切口型		(PDF1.6 版) 在端点从线本身垂直方向约 30 度顺时针方向的短线

圆形和方形注释线：

圆形和方形注释线 (PDF1.3 版) 分别在页面上显示为矩形或椭圆。当打开时，它们显示一个弹出式窗口包含文本的相关说明。该矩形或椭圆形内嵌到注释矩形框，会被注释词典的 `Rect` 参数重新定义（见 570 页的表 8.11）。图 8.7 中画了两个注释，每个线宽 18 个像素。尽管名叫正方形和圆形，注释矩形宽度和高度的不必相等。表 8.24 显示了注解词典中针对这些类型注释的条目。

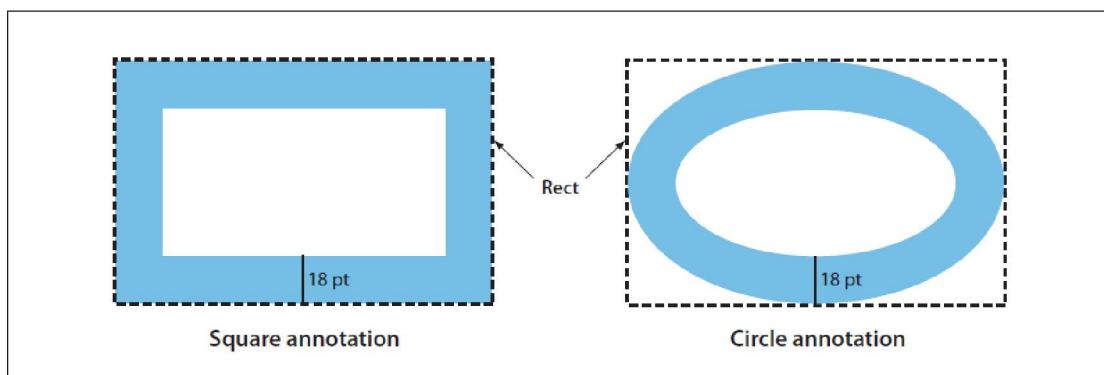


图 8.7 方形和圆形注释

表 8.24 为方形或圆形注释指定的附加项

KEY	TYPE	VALUE
Subtype	name	(必须) 字典描述的注释类型; 对于一个方形或圆形注释必须分别是方形或圆形。
BS	dictionary	(可选) 一个边框类型字典 (见 576 页表 8.13) 指定了在绘制矩形或椭圆形时线宽和线条类型。 注意: 如果注释字典的 AP 项存在, 优先于 Rect 和 BS 项; 见 570 页表 8.11 以及第 8.4.4 章节, “显示流”。
IC	array	(可选; PDF 1.4) 颜色区域, 在由 0.0 到 1.0 范围构成的三个元素的数组, 它 填充注释的矩形或椭圆形框内部颜色 (见表 8.23) 如果该项不存在, 该注释内部是透明的。
BE	dictionary	(可选; PDF 1.5) 边界效应字典描述了应用到 BS 项描述的边界的效果。 (见表 8.14).
RD	rectangle	(可选; PDF 1.5) 4 个数字描述了两个长方形数值差异: 注释的 Rect 项和方形或圆形的底层实际边界。这种差异可能发生的情况下是, 边界效应 (由 BE 来描述) 导致该矩形的大小增加超出了方形或圆形框。 在默认用户区域, 这四个数字依次对应的左, 上, 右, 和底部 Rect 坐标和方形或圆形之间的区别。每个值必须大于或等于 0。在顶部和底部的总和必须大于 Rect 的高度, 同时, 左, 右的总和必须大于 Rect 的宽。

多边形和折线注解

页面上多边形注释 (PDF 1.5) 显示关闭多边形。多边形可能有一些顶点被直线连接。多边形注释 (PDF 1.5) 类似与多边形, 唯一例外的是第一个和最后一个顶点是不直接连接的。

表 8.25 多边形或折线注释额外指定的项目

KEY	TYPE	VALUE
Subtype	name	(必须) 字典描述的注释类型; 对于多边形或折线标注,

		必须分别是多边形或折线。
Vertices	array	(必须)一个数字数组，代表在用户默认区域中横向和纵向的分别对每一个顶点的交叉坐标。
LE	array	(可选；唯一有意义的折线注释)一个两元素的数组指定了线条终点样式。这第一个和第二个数组元素通过在在顶点数组的第一和最后的坐标，指定了线条终点样式的端点定义。 默认值：[/None /None]。
BS	dictionary	(可选)一个边框类型字典(见576页表8.13)指定了在绘制线条时的宽度和线条样式。 注意：注释字典的AP项如果存在，将优先于BS项；见570页表8.11和第8.4.4节“显示流”。
IC	array	(可选；PDF 1.4)三元素的数组在0.0到1.0的范围内，在DeviceRGB范围内指定了内部颜色的构成，以便填充注释的结束行(见表8.23)。如果该项不存在，该结束行的内部是透明的。
BE	dictionary	(可选；唯一有意义的多边形注释)一个有效的边框描述通过BS项应用到边界描述的效果(见表8.14)。
IT	name	(可选；对于多边形注释才有意义；PDF 1.6)一个名字描述了多边形注释的意图(见表8.17)。在pdf规范1.6中，仅仅有效的值是PolygonCloud，它意味着注释是为作为云对象的功能。

文本标记注解

文本标记注释，在文档文本中以高亮、空心（pdf规范1.3）或锯齿形（“弯曲”）、下划线（pdf规范1.4）方式显示。当打开时，它们显示了一个弹出窗口，该窗口包含了注释的文本内容。表8.26显示了注释字典指定的注释类型的项目。

表 8.26 指定文本标记注释的额外项		
KEY	TYPE	VALUE
Subtype	name	(必须)字典描述的注释类型；它必须依次是高亮，下划线，波浪线，或弯曲下划线，下划线，删除线或标注。
QuadPoints	array	(必须)一个8元素的数组指定了用户默认区域的n个四方形的坐标。每一个四边形包含字或者在文本中的相邻字集的相关注释。这四边形坐标依次由x1 y1 x2 y2 x3 y3 x4 y4给定，它们指定了四边形在逆时针方向上的四个顶点（见图8.8）。该文本涉及连接点边缘(x1, y1)和(x2, y2)。(x1, y1) and (x2, y2). (详情见附录H第92页) 注意：如果存在，注意注释字典的AP优先于QuadPoints项；见表8.11和第8.4.4节“显示流”。

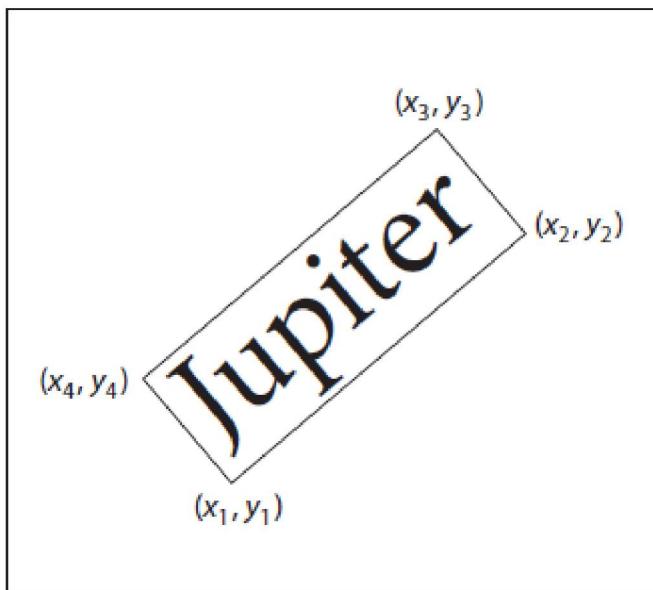


图 8.8 QuadPoints specification

插入符合注释

创建一个注释 (PDF 1.5) 是一种可见的符号，它指示编辑文本存在。条 8.27 列出了插入符号注释指定的项目。

表 8.27 创建注释指定的附加项

KEY	TYPE	VALUE
Subtype	name	(必须) 字典描述的注释类型；它必须是插入符号注释的符号
RD	rectangle	(可选；PDF 1.5) 4组数字描述了两个长方形数值差异：注释的Rect项和底层插入符号的实际边界。例如，当一起显示插入符号的一段指定符号，这种差异可能会出现。 四个数字依次对应于默认用户区域的左，上，右，和底部的矩形坐标。每个值必须大于或等于0。在顶部和底部的总和必须比Rect的高度较低，且左和右的总和必须比Rect的宽度减少。
Sy	name	(可选) 一个指定标记的名字，它被用于和插入符号相关联： P 关联插入符号的一种新的段落标记()。 None 没有标记关联插入符号。 默认值：None.

橡皮图章注解

橡皮图章注释 (PDF 1.3) 显示文本或图像就如同他们是在页面上橡皮图章的盖章。当打开时，它显示了一个弹出窗口，该弹出窗口包含了相关说明的文本。表 8.28 显示了这个注释类型指定的规则注释项。

表 8.28 橡皮图章注释附录的指定项		
KEY	TYPE	VALUE
Subtype	name	(必须) 规则描述的注释类型；必须是一个橡皮图章注释。
Name	name	(可选) 图标的名称在显示注释时被使用。浏览器将提供指定的图标注释，它至少是以下标准名称： Approved Experimental NotApproved AsIs Expired NotForPublicRelease Confidential Final Sold Departmental ForComment TopSecret Draft ForPublicRelease 其他名称也可以支持。默认值：Draft 注意：注释规则的 AP 项如果存在，将优先于 Name 项；见 570 页表 8.11 和第 8.4.4 节“显示流”。

墨迹注释

一个墨迹注释 (pdf1.3) 代表一个由一个或更不相交的路径组成的手绘“书写”。当打开时，它显示了一个弹出窗口包含了相关说明的文本。表 8.29 显示了注释类型指定的规则项。

表 8.29 墨迹注释的附加指定项		
KEY	TYPE	VALUE
Subtype	name	(必须) 规则描述的注释类型；它必须是墨迹注释的连接。
InkList	array	(必须) 一个你阵列的数组，每个元素代表一个划过路径。每个数组是一个在默认用户区域的系列水平和垂直交叉坐标，它指定了沿着路径的点。当绘制时，这些点被连接成直线、曲线。（见附录 H 第 93 页）
BS	dictionary	(可选) 一个边框类型规则（见 576 页表 8.13）指定了线宽和绘制路径的线条样式。 注意：这注释规则的 AP 项如果存在，将优先于 InkList 项；见 570 页表 8.11 和第 8.4.4 章节“显示流”。

弹出注解

一个弹出注释 (pdf1.3) 显示文本在一个弹出窗口进入和编辑。它通常不会

单独出现，而是与一种标记注释，它的父注释和被使用来编辑父类的文本出现。它没有显示流或它自己的关联行动，且被父类注释规则的**Popup**项确定（见583页表8.17）。表

8.30 显示了指定这种注释类型的注释规则项。

表 8.30 弹出注释附加指定项		
KEY	TYPE	VALUE
Subtype	name	(必须) 规则描述的注释类型；必须弹出一个弹出注释。
Parent	dictionary	(可选；必须是间接引用) 父注释与弹出窗口关联。 注意：如果这项存在，父注释内容、M, C, 和 T 项（见570页表8.11）凌驾在弹出的注释本身之上。
Open	boolean	(可选) 一个指定标志无论弹出窗口最初是否打开显示。 默认值：false (关闭)。

文件附件注解：

文件注解（PDF 1.3）包含参考文件，这种参考文件通常嵌入在 PDF 文件中（详见 3.10.3 节，嵌入式文件流），其执行情况说明 95 参见附录 H。举一个例子，数据表就可以使用文件附件注解链接到基于表中数据的电子表格文件，通过激活注解来提取嵌入式文件，并使用户有可能查看此文件或者保存将其到文件系统中。表 8.31 显示了这种类型注解的具体注解词典项。

注解词典的内容条目会详列与附加文件有关的说明文本。应用查看器应该使用此条目而不是由注解的文件分隔符条目定义的文件规范字典（详见表 3.40）中的可选降序条目（PDF 1.6）；其执行情况说明 95 参见附录 H。

表 8.31 文件附加注释的额外项		
关键字	格式	值
		(必选) 本规范描述的注释格式；必须与文件附件注释 FileAttachment 属性值相等
		(必选) 与注释相关的文件
		(可选) 被用在显示注释的图标名称。浏览软件应该提供事先定义的图标，且至少是以下标准中的一种： Graph PushPin

		<p>Paperclip Tag</p> <p>对额外的名称的支持良好。默认值：PushPin</p>
注意：注释字符集的 AP 项，如果存在，其比 Name 项更重要；见 570 页的表 8.11，和 8.4.4 节的“流界面”		

声音注解：

声音注解（PDF 1.2）类似于文本注释，只是它不是文字说明，而是来自计算机的麦克风的录音或由文件导入的声音。当注解被激活时，声音就会播放出来。

这种注解在很多方式下都和文本注释表现相同，通过一个不同的图标（默认为扬声器）来区别表示它代表的是声音。表 8.32 显示了这种类型注解的具体注解词典项。声音对象将在 9.2 节声音讨论。

表 8.32 音频注释的额外项

关键字	格式	值
Subtype		(必选) 本规范描述的注释的额外项；必须等于音频注释的 Sound。
Sound		(必选) 当注释是有效的时候（见 9.2 节“音频”），其定义在要播放的音频文件中的音频对象。
Name		(可选) 被用在显示注释的图标名称。浏览软件应该提供事先定义的图标，且至少标准名称 Speaker 和 Mic. Additional 名称需要得到良好的支持。默认值：Speaker 注意：注释字符集的 AP 项，如果存在，其比 Name 项更重要；见 570 页的表 8.11，和 8.4.4 节的“流界面”

影片注解：

影片注解（PDF 1.2）包含通过计算机屏幕和扬声器呈现出的动态图像和声音。当注解被激活时，影片就会播放出来。表 8.33 显示了这种类型注解的具体注解词典项。影片对象将在 9.3 节影片讨论。

表 8.33 其他条目具体到影片注解

关键词	类型	重要性
Subtype	名称	(必填) 这说明字典类型的注释, 电影必须标注 Movie。
Movie	字典	(必需) 描述一个电影字典影片的静态特性(见第 9.3 节, “Movies”)。
A	布尔或字典	(可选) 标志或字典指定是否注释和如何激活影片播放时。如果这个值是一个字典, 它是一个电影激活词典(见第 9.3 节, “Movies”)指明了如何播放影片。如果该值是布尔 true, 影片应发挥激活使用默认参数。如果该值为 false, 电影不应该播放。默认值: true。

屏幕注解:

屏幕注解(PDF 1.5)具体说明了可以播放媒体片段的页面区域。它也可以作为可以触发动作的对象。第 630 页的“移交行为”讨论了屏幕注解和移交行为的关系。表 8.34 显示了这种类型注解的具体注解词典项。

表 8.34 其他条目具体到屏幕上批注

关键词	类型	重要性
Subtype	名称	(必填) 这说明字典类型的注释, 屏幕必须标注 Screen。
MK	字典	(可选) 外观特征词典(见表 8.36)。这本词典 I 进入的规定产生的图标的外观采用所指的屏幕批注的 AP 项。

除了上表列出的条目, 屏幕注解还通过以下方式使用注解字典中的共同条目:

- P 条目需要由移交行为引用的屏幕注解。它必须引用一个有效的页对象, 并且注解也必须在该页的 Annots 使得行为有效。
- AP 条目指的是一种其正常外观为视觉外观提供屏幕注解的外观词典(见表 8.15); 当没有播放媒体片段时, 这种外观词典就用于打印和默认显示。
- AA 条目指的一种包含 PDF1.5 介绍的四个条目的附加行为字典, 它支持多媒体。

构件注解:

交互表格(见 8.6 节, 互动表格)使用构件注解(PDF 1.2)来呈现域的外观和管理用户的交互行为。

为了方便, 当一个域只拥有单一的相关联构件注解时, 域字典(见 8.6.2 节, 域字典)和注解字典的目录可以被合并成一个包含有关域和注解的条目的字

典（因为这两种字典没有冲突，所以这并没有任何含糊之处）。表 8.35 显示了这种类型注解的具体注解词典项；交互表格和域将在 8.6 节详细讨论。

表 8.35 其他条目具体到一个小部件注释		
关键词	类型	重要性
Subtype	名称	(必填) 这说明字典类型的注释，一个小部件必须标注 Widget。
H	名称	(可选) 注释的高亮模式，视觉效果时要使用的鼠标按钮按下或作用在局部： N (无) 无突出。 I (倒) 反转矩形的注释内容。 O (提纲) 倒置注释的边界。 P (推) 显示批注的外表下，如果有的话（见第 8.4.4，“外观流”）。如果没有出现下降的定义，抵消矩形标注的内容显示为网页，如果它被推向下方的表面。 T (切换) 同为 P (这是首选)。 一个突出模 P 除了外表覆盖下的任何定义的注释。默认值：1。
MK	字典	(可选) 外观特征词典（见表 8.36）将页面用于建造一个充满活力的外观流的视觉呈现在指定注释。 在本项名称 MK 的意义只是历史，并没有直接的意义。

MK 条目可以用来为构件注解的外观流提供包含附加信息的外观特征字典，表 8.36 列出了此字典的目录。

表 8.36 外貌特征字典中的选项		
关键域	类型	取值
R	整形	(可选的) 描述该部件逆时针相对于该页旋转的度数。取值必须是 90 读内的。默认值：0
BC	数组	(可选的) 一个指定部件标注颜色的数组，取值范围 0.0 到 1.0。该数组中的元素觉得了定义颜色的颜色空间： 0: 无颜色，透明的 1: DeviceGray 3: DeviceRGB 4: DeviceCMYK
BG	数组	(可选的) 一个指定控件注释背景颜色的数组，取值范围是 0.0 到 1.0，数组中的颜色决定着颜色空间，如上面的 BC 一样。
CA	文本字符串	(可选，仅针对按钮域) 控件注释的正常标题，当他不和用户进行交互的时候显示。

		注：不同于下面列出的其余项目，只适用于小部件说明相关领域的按钮，CA 可以被用于任何一种类型的按钮域。包括查找箱和单选按钮。
RC	文本字符串	(可选，仅针对按钮域) 控件标注的过渡标题，当用户的光标指到它的灵敏区而不用点击鼠标就会显示。
AC	文本字符串	(可选，仅针对按钮域) 控件标注备用标题，当在灵敏区按下鼠标的时候会显示。
I	流	(可选的，仅针对按钮域，必须是一个间接的引用) 定义控件标注正常图标的表格 XObject，当没有与用户交互的时候显示。
RI	流	(可选的，仅针对按钮域，必须是一个间接的引用) 定义控件标注正常图标的表格 XObject，当用户的光标指到它的灵敏区而不用点击鼠标就会显示。
IX	流	(可选的，仅针对按钮域，必须是一个间接的引用) 一个定义控件备用图标的表格 XObject，当在灵敏区按下鼠标的时候会显示。
IF	字典	(可选，仅针对按钮) 一个图标调整字典，用来指定如何在一个注释方框中显示控件注释图标。如果有，图标调整字典适用于所有的注释图标。
TP	整型	(可选，仅针对按钮) 一个表面再哪里放置和他的图标相关的控件注释标题的文本。 0 无图标，仅针对标题 1 无标题，仅有图标 2 标题在图标下 3 标题在图标上 4 标题在图标的右边 5 标题在图标的左边 6 标题直接叠加在图标上 默认值：0

打印机标识注解

打印机标识注解 (PDF1.4) 代表添加到页面用来协助生产人员定义多片工作使用的组件和保持生产的一致输出的图像符号，比如一个注册对象、颜色条、或者剪切标志。进一步讨论见 10.10.2 节打印机标识。

阀门网注解

阀门网注解 (PDF1.3) 定义 PDF 文件的页面陷阱特征。(捕捉器是为页面彩色边界添加标志的过程，由此来避免当打印页面时由于不当注册染色剂产生的意料之外的视觉品)。一个页面至多有一个捕捉器，其子类型条目有一个有用的阀门网并且通常是页对象的 Annots 阵列中的最后一个元素。进一步讨论见 10.10.5

节捕捉支持。

水印注解

水印注解 (PDF 1.6) 用来表示打印在一个指定大小和位置页面上的图形，而不管打印页的尺寸。水印注解字典 (见表 8.37) 的固定打印条目是一个包含指定的大小和注解位置值的字典 (见表 8.38)。

水印注解没有弹出窗口或其他互动元素。当要在将水印注解显示在屏幕上时，查看器应用程序使用媒体的注释框的尺寸相同的页面大小，以便在滚动和缩放行为和其他的注解相同。

注意：由于许多打印设备没有不可打印页边，当定位水印注解到页面边缘时，应考虑到这样的页边。

表 8.37 水印注释指定的附加项

KEY	TYPE	VALUE
Subtype	name	(必须) 规则描述的注释类型；必须是一个水印注释的水印。
FixedPrint	dictionary	(可选) 一个固定的定义规则指定了这个注释如何被相对的绘制目标媒体的尺寸。如果该项不存在，注释任何特殊考虑的绘制目标媒体的尺寸。 注意：如果目标媒体的尺寸在绘制时不知，相对于被页面 MediaBox 项指定的尺寸来进行绘制。 (见表 3.27).

表 8.38 固定打印规则项

KEY	TYPE	VALUE
Type	name	(必须) 必须是 FixedPrint.
Matrix	array	(可选) 这个矩阵用来改变渲染前的注释矩形框。 默认值：单位矩阵 [1 0 0 1 0 0]. 它建议为该项提供一个合理的补偿，以便不用打印边缘。
H	number	(可选) 翻译水平相关内容的数量，如目标媒体宽的百分比 (或如果不清晰，本页面的宽度)。1.0 代表 100% 和 0.0 代表 0%。负值不建议使用，因为它们可能导致内容被提请关闭该页面。 默认值：0.
V	number	(可选) 翻译水平相关内容的数量，如目标媒体宽的百分比 (或如果不清晰，本页面的宽度)。1.0 代表 100% 和 0.0 代表 0%。负值不建议使用，因为它们可能导致内容被提请关闭该页面。 默认值：0.

当用**FixedPrint**项绘制水印注释时，下面情况可能出现：

注释的矩形框（由**Rect**项指定）转换为原来的样子且 被**FixedPrint** 规则的**Matrix**项产生一个任意方向的四边形。

转换的注释矩形框被定义为最小的直立矩形，它包括四边形；它被使用在注释矩形的地方，这涉及到 577 页算法 8.1 步骤 2 和 3。

此外，给定一个矩阵 B ，它映射到一个缩放和旋转的页面进默认用户区域，一个新的矩阵被计算，它取消了 B 且翻译了原本的打印页面进默认的用户区域。这种转变是用于确保正确的缩放和对齐。

例 8.8 显示了一个水印注释，它从距打印页左边缘 1 英寸和距打印页顶 1 英寸的范围，打印一个文本字符串。

例 8.8

```
8 0 obj % 水印标记示例
<<
/Length...
/Resources...
/BBox ...
>>
stream
...
BT
/F1 1 Tf
36 0 0 36 0 -36 Tm
(未新建) Tx
ET
...
endstream
Endobj
```

```
9 0 obj % 水印标记示例
<<
/矩形...
/类型/注释
/图标类型/水印标记
/FixedPrint 10 0 R
/AP <</N 8 0 R>>
>>
% 在字典页里
/注释[9 0 R]
10 0 obj % 紧缩打印
<<
/类型/缩放打印
/矩阵[1 0 0 1 72 -72] % 右间距1英寸，下间距1英寸
/H 0
/V 1.0 % 垂直铺满整个页面
>>
endobj
```

在某些特殊情况下，PDF 页面大小和打印纸一样，水印标记自定义打印应按照以下方式：

当工作区标志设为可见时（也就是，一页 PDF 分多页打印），每页注释设定特定大小、指定位置，确保任何附加内容都清晰显示。

当选择了 n-up 打印方式（也就是，多页 PDF 打印成一页），水印设定特定大小，指定位置打印，印纸的尺寸限定在页面的一部分。这样可以使任何附加内容不与其它页面重叠，清晰显示。（示例 96 见附录 H.）

8.5、操作

并不是简单地跳到文档的一个位置，注释和要点是一个浏览器程序操作（PDF1.1），就像运行一个程序，播放一首音乐，或者改变注释的格式。当注释选项可用时，可使用注释中的 A 选项（见 P570 表 8.11 和 P555 表 8.4）。

在 PDF 1.2 中，不同的环境可能实现一个操作（见 8.5.2 节，触发事件）。

另外，当文档时打开状态时，文档目录中的可选入口也是一个特定的操作（见 3.6.1 节“文档目录”）。PDF 包括了不同的标准操作类型，详见 8.5.3 节“操作类型”。

8.5.1、操作字典

操作字典描述了操作的特点和具体实现。表 8.39 列出了操作字典中可选的公共操作入口。该字典可能附加了特定的操作类型；详见 8.5.3 节“操作类型”。

表 8.39 操作字典中可选的公共操作入口		
键值	类型	值
Type	命名	(选项) 描述 PDF 目标类型，如果存在必须是一个操作字典
S	命名	描述操作类型，见 P615 表 8.44
Next	字典或数组	(选项；PDF1.2) 下一个操作或是操作序列的下一个。该值并不是一个按顺序的单独的操作字典或操作字典数组。更多讨论见后面文档。

操作字典入口 (PDF 1.2) 允许一连串的操作组合在一起。例如，鼠标点击注释链接，播放音乐，跳到另一个页面，然后开始一段视频播放。也就是下一个入口并没有限制为一个操作而是包含了很多操作，每一个又是它自身的一个入口。因此这些操作会组成一个树而不是一个简单得链表。每一个数组中的操作执行都按特定顺序，一个接着一个如此继续。程序浏览器提供了模拟情形下的可能行为。例如，自我指示行为执行不只一次，关闭文档的操作不会结束整个运行。程序也应该提供给用户结束全部操作的机制。PDF 1.5 介绍了事务操作，可以允许图像控制一序列操作；见 P632 ”事务操作“

提示：没有任何一个附属操作可以更改他自己或操作树中其他操作的操作字典。这种操作更改的影响在操作树中未被定义。

8.5.2、触发事件

一个批注，页对象，或 (PDF 1.3) 表单交互可能包含一个 AA 入口，指明了一个附加操作字典 (PDF 1.2) 延伸了一些可以操作可以触发操作的执行

在 PDF 1.4 中，文档目录字典（见 3.6.1 节，“文档目录”）可能也包括了触发事件的 AA 入口可以影响整个文档。表 8.40 到 8.43 显示了这种字典类型的全

部内容。(见附录中注释 97 98)。PDF 1.5 介绍了四种触发事件来支持多媒体：

P0 和 **PC** 入口和页对象的 **O** 和 **C** 入口有类似的功能(见表 8.41)。但是，批注连结这些触发器允许批注对象包含自身，很大程度上简化了权限的设置。例如，扑住包含了这些可以在页间移动和复制的操作，并不需要更改打开或关闭操作。

PV 和 **PI** 入口也允许了打开的页与页的不同。在任何时候，应用程序只打开了一个单独的页面，然而其他页面都可见，这就靠页面布局实现。

注意：这些触发事件，标志的值是由批注的 **F** 入口定义(见 8.4.2 节“批注标志”)与触发事件是否发生无关

表 8.4.0 批注附加操作字典入口

键值	类型	值
E	字典	(选项；PDF1.2) 当光标移到批注活动区域该操作被激活
X	字典	(选项；PDF1.2) 当光标移出批注活动区域该操作被激活
D	字典	(选项；PDF1.2) 当鼠标在批注活动区域被按下该操作被激活(D 表示“按下”)
U	字典	(选项；PDF1.2) 当鼠标在批注活动区域释放该操作被激活(U 表示“弹起”)
Fo	字典	(选项；PDF1.2) 当批注活动区域得到焦点该操作被激活(D 表示“按下”)
Bl	字典	(选项；PDF1.2)(B 大写, L 小写) 当批注活动区域失去焦点该操作被激活(BI 表示“模糊”)
P0	字典	(选项；PDF1.5) 当包含批注的页面被打开(例如，当用户从下一页跳到前一页)。该操作在 O 操作(见表 8.41) 后执行，并且执行文档目录中打开操作(见表 3.25)
PC	字典	(选项；PDF1.5) 当包含批注的页面被关闭(例如，当用户从下一页跳到前一页)。该操作在 C 操作(见表 8.41) 后执行，并且执行文档目录中打开操作(见表 3.25)
PV	字典	(选项；PDF1.5) 当包含批注的页面被打开在浏览器应用程序中用户接口中可见时该操作被执行
PI	字典	(选项；PDF1.5) 当包含批注的页面被打开在浏览器应用程序中用户接口中永远不可见时该操作被执行

表 8.4.1 页面对象附加操作字典入口

键值	类型	值
O	字典	(选项；PDF1.2) 当包含批注的页面被打开(例如，当用户从下一页跳到前一页)。该操作独立于任何定

		义在文档目录中的操作(见 3.6.1 节“文档目录”), 在该操作后被执行 (详见附录 H 标注 99)
C	字典	(选项; PDF1.2) 当包含批注的页面被关闭 (例如, 当用户从下一页跳到前一页)。该操作应用于已经关闭的页面, 在任何页面被关闭时被执行 (详见附录 H 标注 99)

表 8.4.2 页面对象附加操作字典入口

键值	类型	值
K	字典	(选项; PDF1.3) 当用户当鼠标移动到一个文本区域或组合框或改变列表框中选项时, 一个 java 脚本被执行。该操作可以检查是否有键按下并改变它。
F	字典	(选项; PDF1.3) 当一个区域格式化之前, 一个 java 脚本被执行。该操作可以在格式化之前改变区域的值
V	字典	(选项; PDF1.3) 当一个区域值被改变时, 一个 java 脚本被执行。该操作可以检查新的值是否有效 (V 代表“确认”)
C	字典	(选项; PDF1.3) 当另一个区域的值改变时, 该区域值被重新计算时, 一个 java 脚本被执行 (C 代表“计算”)。重新计算的顺序在交互表单字典中的 CO 入口定义 (见 8.6.1 节, “交互表单字典”)

表 8.4.3 目录字典附加操作字典入口

键值	类型	值
DC	字典	(选项; PDF1.4) 关闭一个文档时执行一个 java 脚本 (DC 表示“关闭文档”)
WS	字典	(选项; PDF1.4) 保存一个文档时执行一个 java 脚本 (WS 表示“将保存”)
DS	字典	(选项; PDF1.4) 保存文档后执行一个 java 脚本 (DS 表示“已保存”)
WSP	字典	(选项; PDF1.4) 打印文档时执行一个 java 脚本 (WP 表示“将打印”)
DP	字典	(选项; PDF1.4) 打印文档后执行一个 java 脚本 (DP 表示“已打印”)

E (进入), X (退出), D (向下), U (向上) 触发事件, 这些术语若要鼠标指示正确需要按以下特点:

按下一个选项按钮, 然后释放。如果不只一个鼠标按钮, 这个选项通常用鼠标左键。

位置——也就是, 屏幕上哪个位置正确。通常由屏幕光标确定。

焦点——也就是，哪个文档正在与用户交互。在很多系统中，这个是由插入点，矩形区域，或颜色的改变确定。

PDF 应用程序必须确保相应的操作正确执行。与鼠标相关的触发事件应遵循以下限制：

当鼠标键被弹起时，E(进入)事件被触发

在 E 事件执行前不会触发 X(退出)事件

在 E 事件和 D 事件执行前不会触发 U(向上)事件

在重叠批注区域，进入第二个批注区域可以使 X 事件被触发

注意：与触发事件 K(按键)，V(解析)，C(计算)有关的在按钮区域不被定义（详见 P647”按钮区域“）。这些事件中某个触发事件操作只影响它自己，并可以在事件外发生。例如，即使 F 事件用来触发格式化某个区域的操作，一个触发事件用来计算或者改变文档是可能的。这些区域相关的触发事件通过与用户交互或者创建，例如对用户交互表单字典中需要显示入口的响应（见 8.6.6 节”表单数据格式“），FDF 数据的重要性（见 8.6.6 节”表单数据格式“），JavaScript 操作（P668 ”JavaScript 操作“）。例如，用户改变一个区域的值会触发一个级联的计算，会更新文档中其他区域的参数。

8.5.3、操作类型

PDF 支持表 8.44 中的标准操作类型。下面详细描述了每种操作类型。插件扩展可以添加新的操作类型。

表 8.44 操作类型

操作类型	描述	具体章节
跳转	跳转到本文档的某一个位置	“跳转操作”在 616 页
远程跳转	(“Go-to remote”) 跳转到另一个文件的某个地方	“远程跳转操作”在 617 页
嵌入跳转	(“Go-to embedded”; PDF1.6) 跳转到嵌入文件的某个地方	“嵌入跳转操作”在 617 页
启动	启动一个应用程序，通常是打开一个文件	“启动操作”在 621 页
线程	开始读一个文章线程	“线程操作”在 623 页
统一资源标	指向一个统一资源标识符	“统一资源标识符操

识符		作”在 624 页
声音	(PDF1.2) 播放声音	“声音操作”在 625 页
影片	(PDF1.2) 播放影片	“影片操作”在 626 页
隐藏	(PDF1.2) 设置一个注解隐藏标记	“隐藏操作”在 626 页
预定义	(PDF1.2) 在视图应用中执行一个预定义的操作	“预定义操作”在 628 页
提交表单	(PDF1.2) 将数据提交到一个统一的资源位置	“提交表单操作”在 662 页
重置表单	(PDF1.2) 将区域设置为它们的默认值	“重置表单操作”在 666 页
导入数据	(PDF1.2) 从文件向区域中导入数据	“导入数据操作”在 667 页
Java 脚本	(PDF1.3) 执行一段 JavaScript 语言	“Java 脚本操作”在 626 页
设置 OCG 状态	(PDF1.5) 设置非强制控件组的状态	“设置 OCG 状态操作”在 626 页
播放	(PDF1.5) 控制多媒体控件的播放	“播放操作”在 630 页
翻译	(PDF1.5) 使用一个翻译词典更新一个文件的显示	“翻译操作”在 632 页
转换到 3D 视图	(PDF1.6) 将当前的视图转换到 3D 视图	“转换到 3D 视图操作”在 632 页
	(PDF1.3)	

注：PDF 先前版本的说明书描述了一个设置状态的操作类型；现在这种操作类型已经过时，并不建议使用这种操作。在 PDF1.2 中定义的一个附加的操作类型---no-op 操作从来没被使用过，它将不再定义而被忽略。

跳转操作

跳转操作将视图转换到一个特定的位置（页，位置，和放大要素）。表 8.45 列出了在词典条目中这个操作的具体操作类型。

表 8.45 跳转操作具体的附加条目

关键词	类型	值
S	命名	(要求) 词典描述的这个操作类型, 当执行跳转操作时必须跳转
D	命名, 字符串, 数组	(要求) 跳转的目的地 (8.2.1 部分的“目的地”)

具体描述了跳转操作在条目 A 中的联系注解, 当直接在 Dest 条目中具体描述目的地时, 简要条目 (见 587 页的表 8.20 和 555 页的表 8.4) 的效果相同。例如, 在使用了一个跳转操作例 8.9 中显示的这个联系的注解, 和在 588 页的例 8.7 有相同的效果, 它直接具体描述了目的地。但跳转操作和 PDF1.0 不兼容, 然而使用一个直接的目的地是可取的。

例 8.9

```
93 0 obj
<< /类型 /注释
/替代类型 /链接
/矩形 [71 717 190 734]
/边框 [16 16 1]
/A << /类型 /操作
/S /跳转
/D [3 0 R /FitR -4 399 199 533]
>>
>>
Endobj
```

远跳转操作

远跳转操作和普通的跳转操作类似, 但跳转的目的地在另外的一个 PDF 文件中, 而不是当前的文件。表 8.46 列出了在词典条目中这个操作的具体操作类型。

注: 远跳转操作不能在嵌入的文件中使用, 见 617 页的嵌入跳转操作。

表 8.46 远跳转操作具体的附加条目

关键词	类型	值
S	命名	(要求) 词典描述的这个操作类型, 当执行远跳转操作时必须远跳转
F	指定的文件	(要求) 文件中的目的地是明确的
D	命名, 字符串, 数组	(要求) 跳转的目的地(见 8.2.1 部分的“目的地”)。如果这个值是一个数组, 定义了一个明确的目的地(如下面 551 页“详细目的地”的描述), 它的第一个元素必须是一个远程文件中的页码而不是一个间接的在当前文件中提到一个页。第一页的页码是 0.
新窗口	布尔型	(非强制; PDF1.2) 是否在一个新窗口中打开文件的标志。如果这个标志值为假, 目的文件在同一个窗口将当前文件覆盖。如果这个值是缺省的, 应用视图将根据当前使用者的先前选择操作。

嵌入跳转操作

嵌入跳转操作 (PDF1.6) 和远跳转操作类似, 但允许跳转到一个嵌入到另外的 PDF 文件的 PDF 文件或者从嵌入到另外的 PDF 文件的 PDF 文件跳转出(见 157 页的“嵌入文件流”)。嵌入的文件可能和附加的注解(见 600 页的“文件附加注解”)或者嵌入文件名树(见 3.6.3 部分;“名字词典”)相联系。嵌入文件可能反过来包含嵌入文件。表 8.47 列出了在词典条目中嵌入跳转操作的具体操作类型。

嵌入跳转操作提供了一个非常方便的连接, 在一个嵌入文件的不同层次和在相同或不同目录下的另一个文件之间。使用了以下术语:

源是包含了嵌入跳转操作的文件。

目标是要跳转的目的文件。

在操作词典中的 T 条目是一个指定了目标和源的关系的目标词典, 在一个文件系统中, 用一个相关的路径描述了两个文件之间的物理关系这样十分相同的方法。在达到最后目标之前, 目标词典可能重复的指定一个或更多的中间目标。每个中间目标被指定作为当前的文件。最初的, 这个源文件是当前的文件。

注：目标词典无限循环是错误的（如，指向自身的目标词典）。视图应用应该试图发现这样的情况，如果发现了这样的操作，应该拒绝执行。

子文件是嵌入在另一个 PDF 文件中的文件。

父文件是包含了嵌入文件的文件。

根文件是没有嵌入在另外 PDF 文件中的文件。目标和源可能包含在根文件或嵌入文件中。

表 8.47 嵌入跳转操作具体的附加条目

关键词	类型	值
S	命名	(要求) 词典描述的这个操作类型，当执行嵌入跳转操作时必须嵌入跳转
F	指定的文件	(非强制要求) 目标文件的根文件和源文件的根文件相联系。如果是缺省，源和目标共享根文件。
D	命名，字符串，数组	(要求) 目标跳转的目的地（见 8.2.1 部分的“目的地”）。
新窗口	布尔型	(非强制；PDF1.2) 如果为真，目的文件在一个新窗口中打开；如果为假，目的文件在同一个窗口将当前文件覆盖。如果缺省，应用视图将根据当前使用者的先前选择操作。
T	词典	(非强制性的如果 F 存在，否则要求) 一个目标词典（见表 8.48）详细指定了目标文件的路径信息。每一个目标词典用到目标的全路径指定了一个元素，可能目标词典具体描述了附加的元素。

表 8.48 具体描述目标词典的条目

关键词	类型	值
R	命名	(要求) 指定了当前文件和目标（可能是中间目标）的关系。有效值是 P（目标文件是当前文件的父文件）和 C（目标文件是当前

		文件的子文件)
N	字符串	(如果 R 的值是 C 并且目标在嵌入文件名树中则要求; 否则必须是缺省的) 在嵌入文件名树中的文件名
P	整数, 字符串	(如果 R 的值是 C 并且目标和文件的附加注释相联系则要求; 否则必须是缺省的) 如果这个值是一个整数, 它具体指定了当前文件的页码 (以 0 开始) 包括附加注释的文件。如果这个值是字符串, 它在当前的文件中具体指定了一个命名的目的地, 它提供了附加注释的页码。
A	整数或文本字符串	(如果 R 的值是 C 并且目标和文件的附加注释相联系则要求; 否则必须是缺省的) 如果这个值是一个整数, 它具体指定了注释在注释数组 (见标 3.27) 中的下标 (从 0 开始) 在 P 指定的页。如果这个值是文本字符串, 它具体描述了 NM 在注释词典中的值 (见表 8.11)
T	词典	(非强制性) 一个目标词典详细指定了目标文件的路径信息。如果这个条目缺省, 当前的文件就是包含目的地的目标文件。

例 8.10 说明了几种源和目的的可能关系。对一个嵌入跳转操作来说, 每一个显示对象是一个操作词典。

例 8.10

```
1 0 obj % 链接到子文件
<< /类型 /操作
/S /嵌入跳转
/D (第 1 章)
/T << /R /C
/N (嵌入文件) >>
```

>>

Endobj

2 0 obj % 链接到父文件

<< /类型 /操作

/S /嵌入跳转

/D (第一章)

/T << /R /P >>

>>

endobj

3 0 obj % 链接到兄弟或姊妹

<< /类型 /操作

/S /嵌入跳转

/D (第一章)

/T << /R /P

/T << /R /C

/N (另外一个嵌入文件) >>

>>

>>

endobj

4 0 obj % 链接到额外文件中的嵌入文件中

<< /类型 /操作

/S /嵌入跳转

/D (第一章)

/F (某些 PDF 文件)

/T << /R /C

/N (嵌入文件) >>

>>

endobj

5 0 obj % 从一个嵌入文件到一个普通文件的链接

<< /类型 /操作

/S /嵌入跳转

```
/D (第一章)
/F (某些 PDF 文件 )
>>
Endobj
6 0 obj % 链接到一个孙子文件
<< /类型 /操作
/S /嵌入跳转
/D (第一章)
/T << /R /C
/N (嵌入的文件)
/T << /R /C
/P (一个目的地名)
/A (注释名)
>>
>>
>>
endobj
7 0 obj % 从源的父文件链接到一个表亲文件
<< /类型 /操作
/S /嵌入跳转
/D (目的地)
/T << /R /P
/T << /R /C
/N (嵌入的文件)
/T << /R /C
/P 3
/A (注释名)
>>
>>
>>
>>
```

Endobj**启动操作**

启动操作启动了一个应用程序或打开或打印一个文件。表 8.49 列出了在词典条目中嵌入跳转操作的具体操作类型。

为了启动指定的应用程序，可以任选的 Win, Mac, 和 Unix 条目允许操作词典包括指定的平台参数。如果对于指定的平台没有这样的条目，就使用 F 条目。表 8.50 列出了对于 Windows 平台指定的平台启动参数。对于 Mac 操作系统和 UNIX 平台，在出版本文档时，参数还没有定义。

表 8.49 启动操作附加的指定项

KEY	TYPE	VALUE
S	name	(必须) 规则描述的动作类型；它必须产生一个启动动作。
F	file	(如果 Win, Mac, or Unix 中没有一个项存在，那么该项必须使用) 这个应用于启动或文档被打开或被打印。如果该项不存在且浏览器不认可备选条目，那么它没有任何动作。
Win	dictionary	(可选) 一个规则包含了窗口指定启动参数(见表 8.50；详情也可见附录 H 第 100 页)。
Mac	(undefined)	可选) Mac 的具体启动参数；尚未定义。
Unix	(undefined)	(可选) UNIX 的具体启动参数；尚未定义。
NewWindow	boolean	(可选；PDF 1.2) 一个指定的标志，无论是否在新窗口中打开目标文件。如果这个标志是假，那么在相同的窗口中，目标文件将替代当前的文件。如果该项不存在，浏览器将根据当前用户的倾向而采取操作。如果由 F 项指定的文件不是一个 PDF 文件，该项被忽视。

表 8.50 窗口启动参数项

KEY	TYPE	VALUE
F	string	(必须) 应用的文件名以在标准的 Windows 路径格式，被启动或者文档被打开或被打印。如果名称字符串包含了一个反斜杠字符()，反斜杠自己前面必须先有一个反斜杠。 注：此值必须是一个简单的字符串，它不是一个文件规范。
D	string	(可选) 一个在标准 DOS 线条的字符串指定默认目录。
O	string	(可选) 执行一个字符串指定的操作： open 打开一个文档。

		Print 打印一个文档。 如果 F 项指定一个替代文档的应用，该项被忽略，且该应用被启动。默认值：打开。
P	string	(可选)一个字符串参数被传递给由 F 项指定的申请。如果 F 指定一个文件，那么该项将省略。

线程操作

一个线程操作跳转到一个指定上一篇文章线程的节点(见第 8.3.2,“协议”),无论当前是否在同一文档。表 8.51 显示了这个动作类型指定的规则项的动作。

表 8.51 线程操作指定的附加项		
KEY	TYPE	VALUE
S	name	(可选)规则描述的操作类型；它必须是一个线程操作的线程。
F	file	(可选)文件包含线程。如果该项被忽略，这个线程在当前文档中。
D	dictionary, integer, or text string	(必须)目标线程，它指定下面的形式之一： 间接引用一个线程规则(见第8.3.2节，“协议”)。在这种情况下，线程必须在当前文件。 线程的索引在它的文件对话框的线程数组中(见章节 3.6.1, “文档对话框”)。这第一个线程在数组的索引0。(见 561 页表 8.7)。如果两个或者更多线程有相同的标题，文档对话框线程数组的第一个将被显示执行。
B	dictionary or integer	(可选)目标线程的节点，它指定下面的形式之一： 一种间接涉及规则节点(见第8.3.2节，“协议”)。 在这种情况下，线程必须在当前文档中。 线程中的节点索引。这在线程中第一个节点的索引是0

URI 操作

统一资源标识符 (URI) 是标识 (解析) 一个互联网上的资源的一个字符串，最典型的是一个超文本链接目标的文件，虽然它也可以标识查询或其他实体。(URI 在 Internet RFC 2396 的统一资源标识符 (URI): 通用语法中有描述；见参考文献)

URI 行为就是去解决 URI。表 8.52 列出关于这类型行为的行为字典的详细说明。(见附录 H 的执行注释 101 和 102)

表 8.52 URI 操作的其它条目

关键字	类型	值
S	name	(必需的) 描述的行为类型; URI 操作必须是 URI。
I	string	(必需的) 用于解决的统一资源标识符, 7-bit ASCII 编码。
IsMap	boolean	(可选的) 标志, 指定当当 URI 已解决时, 是否跟踪鼠位置 (见下文)。默认值: false。 这个条目只在用户发生点击注解行为时使用; 忽略此项, 当行为与大纲项目或文档 OpenAction 条目有关系的行为。

如果 IsMap 标志为 true, 并且用户已经通过点击注释触发了 URI 行为, 在行为发生时, 相对注释方框右上角, 此时鼠标位置坐标应该从设备空间转换到用户空间(即与 URI 行为相关注释的 Rect 值)。例如, 如果在用户空间的鼠标坐标是 (xm, ym), 注释矩形从左下角 (llx, lly) 到右上角的 (urx, ury), 则最后坐标 (xf, yf) 值如下:

$$\begin{aligned} xf &= xm - llx \\ yf &= ury - ym \end{aligned}$$

如果结果坐标 (xf, yf) 是分数, 应该圆整到最接近的整数值。然后将它们附到 URI 上解决, 用逗号隔开, 在前面加一个问号, 如下所示:

<http://www.adobe.com/intro?100,200>

为了支持 URI 的行为, PDF 文档目录(见 3.6.1 “文档目录”)可能包含一个值在 URI dictionary 的 URI 条目。在发布时, 只给这样的 dictionary 定义一个条目。(见表 8.53)。

表 8.53 URLdictionary 条目

关键字	类型	值
Base	string	(可选的) 这个 base URI 用于解决相关的 URI 参考。这个文档中的 URI 行为可能指定 URI 局部格式, 相对基址说明。如果没有指定 base URI, 这些局部 URI 将被相对文档本身的位置被说明。这个条目的用法类似于 body 元素<BASE>, 如 HTML4.01 规范所描述(见参考书目)。

此 base 项允许文档的 URI 记录文档内容可能被访问出界的位置。例如, 如一个文档已经移到新的位置, 但是对其它文档的相关链接并没有移动, 这时 base 项能使这些链接到其它文档的实际位置, 而不是已移动的文档。

发声操作

发声操作 (PDF1.2) 是通过计算机的扬声器播放一个声音, 表 8.54 显示了指定此类操作的操作的全部条目, 发声在 9.2 节 “Sounds” 中讨论。

表 8.54 其它发生操作条目

Key	Type	Value
S	name	(必需的) 操作类型描述, 发声操作必须是 sound
Sound	Stream	(必需的) 声音对象定义播放的声音(见 9.2 节“sounds”)和附录 H 执行注释 103
Volume	number	(可选) 声音播放幅度, 范围-1.0~1.0, 见附录 H 执行注释 105, 默认值: 1.0。
Synchronous	boolean	(可选) 指定声音同步还是异步播放的标志, 见附录 H 执行注释 105, 如果此标志是 true, 浏览器应用程序保留控制, 只是允许用户取消声音, 或等声音完全放完。默认值: false。
Repeat	boolean	(可选) 指定是否循环播放声音的标志。如果有这个条目, Synchronous 条目被忽略。默认值: false。
Mix	boolean	(可选) 标志指定是否混合声音到正播放的声音, 见附录 H 执行注释 106, 如果 Mix 为 false, 播放此声音前先会停止已在播放的任何声音, 可用来停止一个循环播放的声音(见上 repeat 条目)。默认值: false。

视频操作

视频操作(PDF1.2)可在悬浮和窗口或在视频注释矩形内播放一个视频(见 601 页“Movie Annotations”和 9.3 节, “Movies”), 视频注释必须与相关一个页相联系, 此页是连接注释目标, 或是包含视频操作的大纲项目, 或是与视频操作相关的页对象。(见附录 H 执行注释 107)

注: 视频操作本身并不保证有视频的页会在试图播放它前播放, 这些页操作必须明确地进行。

视频操作内容类似于激活视频(见 742 页表 9.31), 见表 8.55 其它的条目, 视频注释相关的激活内容有默认值, 任何指定视频操作的信息都会覆盖这些值。

表 8.55 其它视频操作条目

Key	Type	Value
S	name	(必需的) 操作类型描述, 视频操作必须是 Movie。
Annotation	dictionary	(可选) 对指定有播放视频注释的一个间接引用。
T	ext string	(可选) 有播放视频的视频注释题目。 注: 必须包含一个 Annotation 或 T, 但不能两个都包含
Operation	name	(可选) 视频播放操作; Play; 播放视频, 使用 Mode (见 742 页表 9.31) 指定的模式, 如果此视频在暂停状态, 将被先重定位到开始处再播放 (如果有 Start 指定的点, 就从此指定点处开始播放)。 Stop: 停止播放 Pause: 暂停播放 Resume: 恢复播放 默认值: Play

隐藏操作

隐藏操作 (PDF 1.2) 就是通过设置或清除它们的 Hidden (见 8.4.2 节, “Annotation Flags”), 隐藏或显示一个或多个在屏幕上的注释, 这种操作可用结合外观流或触发事件 (8.4.4 节, “Appearance Streams” 和 8.5.2 节 “Trigger Events”) 在屏幕弹出帮助信息, 比如, 当用户在页内激活区里内外滚动光标时, 注释其它操作字典中 E (enter) 和 X (exit) 触发事件可被用来展示或隐藏注释。这能用于弹出帮助标签或工具提示, 来描述点击页内位置的行为。表 8.56 显示了指定此型操作的条目 (见附录 H 执行注释 108 和 109)。

表 8.56 隐藏操作的其它条目

Key	Type	Value
S	name	(必需的) 操作类型描述, 隐藏操作必须为 Hide
T	dictionary, string, or array	(必需的) 当一条或多条注释要隐藏或显示, 用下边任一方 式指定: <ul style="list-style-type: none">● 间接引用注释● 字符串给出完全合格的交互式表格域名, 与交换表格相关的部分注释或批注会受到影响 (见 638 页 “域名”)

● 序列或字符串

H boolean (可选) 指示是否隐藏 (true) 或显示 (false) 注释标志。

默认值: false。

命名操作

表 8.57 列出了几个 PDF 浏览器应用程序均需支持的命名操作 (PDF1.2); 新的操作将来可能还会添加 (见附录 H 执行注释 110 和 111);

表 8.57 命名操作

NAME	ACTION
NextPage	到文档的下一页
PrevPage	到文档的上一页
FirstPage	到文档的第一页
LastPage	到文档的最后一页

注: 阅读器应用程序可能支持其它的, 非标准的命名操作, 但是任何使用它们的文档都不是便携的。如果阅读器面对一个与阅读平台不相适应命名操作, 或如果浏览器不认识这个命名, 就不会有操作。

表 8.58 显示了在命名操作中的操作条目。

表 8.58 隐藏操作的其它条目

Key	Type	Value
S	name	(必需的) 操作类型描述, 命名操作必须为 Named
N	name	(必需的) 执行操作名称 (见表 8.57)

Set-OCG-State (ocg: official card game) 操作

`set-ocg-state(pdf1.5)` 设置一个或多个可选内容组状态(见 4.10 节“Optional Content”。

表 8.59 显示了指定此类操作的其它条目。

表 8.59 Set-OCG-State 其它条目

Key	Type	Value
S	name	(必需的) 操作类型描述, 对 <code>set-OCG-state</code> 操作必须为 <code>SetOCGState</code>
State	array	(必需的) 序列包含任意数目的命名对象 (ON, OFF, 或 Toggle), 命名对象跟随一个或多个可选内容组, 这序列元素 排列从左到右; 每一个名字被应用到子组, 直到遇见下一个 名字: ON: 设置子组状态为 ON OFF: 设置子组状态为 OFF Toggle: 转换子组状态
PreserveRB	boolean	(可选) 如果是 (true), 代表单选按钮在可选内容组的状态 关系 (就像当前配置被条目 RBGroups 指定; 见 346 页表 4.51), 当 State 序列的状态被应用时, PreserveRB 应该被 保存。也就是, 如果一个组在状态序列中被设置为 ON (ON 或 Toggle), 其它任何属于单选按钮组的变为 OFF, 如果一个 组被设置为 OFF, 对其它组没有任何作用。 如果 PreserveRB 为 false, 不论怎样, 单选按钮状态关系都 被忽略。 默认值: true

当 `set-OCG-state` 操作被执行, State 序列从左到右处理。每个命名被应用到序列子组,
直到遇到下一个命名, 如下例:

例 8.11

```
<< /S /SetOCGState  
/State [/OFF 2 0 R 3 0 R /Toggle 16 0 R 19 0 R /ON 5 0 R]
```

>>

一组可出现多个 state 序列；根据最新的命名，每次碰到它，它状态就会被设置一次，比如：如果序列包含 [/OFF 1 0 R /Toggle 1 0 R]，则组状态在操作执行后会变为 ON。ON、OFF 和 Toggle 序列没有要求的顺序，序列中的多个次序可能包含有同样的名字。

注：尽管规范允许一个组在状态序列多次出现，并不是试图执行动画或其它连续绘制操作。PDF 执行程序允许积累所有状态改变，并在重画前，只是同步应用网络改变到所有作用组。

表演操作

- 表演操作（PDF1.5）控制多媒体内容的播放（见 9.1 节“多媒体”）。操作使用如下：
- 开始进行对象转换（见 9.1.2 节“Renditions”），就如与屏幕注释相关（见 602 页“屏幕注释”）。除非另行指定，否则屏幕注释指出那里进行表演。
- 停止，暂停，恢复播放表演
- 触发的 JavaScript 脚本后可能执行自定义操作。

表 8.60 列出了表演操作的条目。

表 8.60 表演操作其它条目

Key	Type	Value
S	name	（必需的）操作类型描述，对表演操作必须为 Rendition
R	dictionary	（当 OP 存在且值为 0 或 4 时要求，否则为可选）表演对象（见 9.1.2 节“Renditions”）
AN	dictionary	（当 OP 存在且值为 0 或 4 时要求，否则为可选）屏幕注释间接引用（见 602 页“屏幕注释”）
OP	integer	（当 JS 不存在则要求，否则可选）行为触发时，操作进行，有效值是： 0：如果没有表演与 AN 指定的注释相关，执行由 R 指定的表演及相关注释，如果表演和注释已经相关，则停止，把注释与新的转换相关。 1：停止所有进行的和 AN 指定相关注释的表演，并除去联系。如没有表演进行，不产生作用。 2：暂停所有进行的和 AN 指定相关注释的表演，如没有表演

进行，不产生作用。

3: 恢复所有与 AN 指定注释的表演操作。如没有 表演在进行或转换被暂停，不产生任何效果。

4: 执行由 R 指定它和 AN 指定注释相关的表演，如果有表演已何注释相关，恢复处在暂停状态的表演，否则不发生作用。

JS string or (如果 OP 不存在则要求，否则可选) 当操作被触发时，包含
 stream JavaScript 脚本的字符串或流会被执行。

JS 条目或 OP 条目必须存在。如两个都存在，如果浏览器应用程序不能执行 JavaScripts 脚本，OP 相关反馈执行，如果 OP 有未确定的值且无 JS 条目，则操作无效。

一些情况，暂停 (OP 值是 2) 或恢复 (OP 值是 3) 操作可能无意义 (比如，对一个 JPEG 图片)，或播放器可能不支持它。这种情况下，用户会应该被通知执行操作失败。

在表演操作执行前，浏览器应用程序必须确保两条，一是屏幕注释 P 条目引用一个有效页对象，二是注释存在页对象 Annots 序列 (见表 3.27)。表演可能在屏幕注释矩形内进行，即使注释本身不可见；例如，如果它的 Hidden 或 NoView 标志 (见表 8.12) 被设置。因为其位置不能被浏览器显示，所有屏幕注释而不可见。然而，如果视图改变它又变得可见，如通过滚动。

过渡操作

过渡操作 (PDF1.5) 能在连续操作中控制绘制，如在 8.5.1 节 “Action Dictionaries” 中讨论。在操作中 Next 条目能指定连续的行为，当这一系列开始，浏览器应用程序通常会暂停绘制，并在它们结束时再重新绘制，如果过渡操作在序列中存在，浏览器应该确保页面可视区域的状态，就像它完成先前的操作，并且使用指定的过渡播放它 (见表 8.61) 一旦这过渡完成，绘制应该被再次暂停。

表 8.61 过渡操作其它条目

Key	Type	Value
S	name	(必需的) 操作类型描述，对过渡操作必须为 Trans
Trans	dictionary	(必需的) 更新显示使用的过渡 (见表 8.9)。

Go-To-3D-View 操作

Go-To-3D-View (PDF1.6) 定义了一个 3D 注释并且指定了注释使用的视图 (见 9.5 节, “三位艺术”)。表 8.62 显示了 Go-To-3D-View 的各个操作条目。

表 8.62 Go-To-3D-View 操作其它条目

Key	Type	Value
S	name	(必需的) 操作类型描述, 对过渡操作必须为 GoTo3Dview。
TA	dictionary	(必需的) 设置视图的目标注释
V	(various)	(必需的) 视图使用, 可为下列类型之一 <ul style="list-style-type: none">● 3D 视图 (见 9.5.3 节 “3D Views”)● 整型值, 指定 3D 流中 VA 序列下标 (见表 9.35)● 文本字符串, 相匹配于 VA 序列一个视图的 IN 条目 (见表 9.37)● 名字, 表示 VA 序列的第一个 (F), 最后一个 (L), 下一个 (N), 前一个 (P), 或默认 (D) 条目, 见下面讨论。

V 条目选择视图去应用于 TA 指定的注释, 此视图可能是一个已被 3D 流 (见表 9.35) VA 条目预先定义好视图, 或是这里指定的唯一的视图。

如果已定义的视图被名字 N (next) 或 P (previous) 指定, 它应该通过下面的方法解释。

- 当末视图通过 VA 序列指定应用, N 和 P 表示下一个和前一个条目, 分别在 VA 序列 (如必要就循环)。
- 当末视图不是通过 VA 手段指定, 使用 N 和 P 应该导致恢复到默认视图。

8.6、交互表格

交互表格 (PDF1.2) 有时就像 AcroForm 引用, AcroForm 是一个为聚集用户交互信息于的汇集体, 一个 PDF 文档可能包含任意数目的域, 域是任意页组合, 所有的这些组成一个单独的、全局的交互表格贯穿到整个文档, 这些域的任意子集能从文档里导入或导出; 见 8.6.4 节 “Form Actions”。

注：交互表格不应该与 Xobjects 混淆（见 4.9 节“Form XObjects”）尽管名字类似，但这两个是不同的，是两种无关的对象类型。

每个在文档交互表格上的域通过一个域字典定义（见 8.6.2 节“Field Dictionaries”）。为了定义和命名目的，域可以被层次组织，并能够从它们域层次上的祖先继承属性。在域家谱中一个子域可能也包含部分父域页上外观注释（见 603 页“Widget Annotations”），子域是局部注释的父域叫做终端域。

为了方便，当域只有一个相关的局部注释，则域字典内容和注释字典（8.4.1 节“Annotation Dictionaries”）可能融进一个的包含条目的字典中，条目从属域和注释。（这并没有含糊之处，由于来了两种字典的内容并不冲突。）如果这个对象定义了一个外观流，则域外观必须与对象当前值相一致。

注：如果域包含文本，文本内容先前并知道，域可能需要动态构建外观流而不是静态的定义一个外观字典：见 639 页“Variable Text”。

8.6.1、交互表格字典

文档交互表格的内容和特性被一个交互表格字典定义，交互表格字典参考文档目录的 AcroForm 条目（见 3.6.1 节“文档目录”）。表 8.6.3 显示了此字典的内容。

表 8.63 交互表格字典条目

Key	Type	Value
Fields	array	(必需的) 文档根域的参考序列 (在域家谱里没有父辈域)。
NeedAppearances	boolean	(可选) 一个标志，其指定是否对文档中的局部注释构建外观流和外观字典（见 639 页“文本变量”），默认值：false
SigFlags	integer	(可选: PDF1.3) 一组标志，指定各种文档水平的相关签定域统计（见下表 8.64 和 658 页“Signature Fields”），默认值：0
CO	array	(如果任何在文档的域有包含 CO 条目的操作字典，CO 就是必需的；PDF1.3) 一个对有计算操作域字典有间接引用的序列，需要定义计算次序，因为当任何域值改变，这些值将会被重新计算（见 8.5.2 节，“事件触发”）。
DR	dictionary	(可选) 资源字典(见 3.7.2 节，“Resource Dictionaries”) 包含默认的资源（如字体，模式，或颜色空间），资源将会被

窗体域外观流使用，至少，此字典必须包含一个 `Font` 条目指定默认的资源名和字体，为文本显示使用。(见附录 H 执行注释 112 和 113)

`DA` `string` (可选) 一个文档范围内可变文本域 `DA` 属性默认值 (见 639 页“文本变量”)。

`Q` `integer` (可选) 一个文档范围内可变文本域 `Q` 属性默认值 (见 639 页“文本变量”)。

`XFA` `stream or array` (可选: PDF1.5) 一个流或包含 XFA 资源的序列，其格式被 XML 数据包规范描述 (见参考书目)。此条目值必须是一代表的 XML 数据包的全部内容的流，或代表组成的 XML 单个数包的字符串序列或流。

见 8.6.7 节，“XFA Forms” 获取更多信息。

注: 在 PDF1.5 规范的早期版本，此条目值仅被定义为一个流；

见附录 H 执行注释 114。

交互表格 `SigFlags` 条目值是一个无符号 32 位的整形，包含标志指定各种相关签定域文档层的统计 (见 658 页“签定域”)。在这个字内的 bit 位置被标记从 1 (`low-order`) 到 32 (`high-order`)，表 8.64 显示了标志意思；所有未定义标志位必须设为 0

表 8.64 签定标志

BIT POSITION	NAME	MEANING
1	<code>SignaturesExist</code>	如设置，文档至少包含一个签定域，此标志允许浏览器应用程序使能用户交互条项 (如菜单项或按钮)，与签字过程相关的交互条项不用浏览全部文档扫描签定域的存在。
2	<code>AppendOnly</code>	如设置，如果文件被保存 (写入) 改变它先前内容，相对于增长更新，文档包含签定可能无效。文件更新只是追加新信息到前版本末尾是安全地 (见 G.6 节“更新示例”) 浏览器应用程序能使用此标志去满足用户完全保存要求，完全保存就是会报警框，警告签定会无效，并要求用户在继续操作前明确认。

8.6.2、域字典

每个文档交互表格域用域字典定义，它必须是一个间接对象，域字典可能被层次化组织成一个或多个树形结构。许多域属性是可继承的，也就是如果它们不对给定域明确指定，它们的值就是从父域继承来的，可继承属性被指定在下面表中，指定（必需的：可继承）意味着属性必须为每个域定义，它自己的域字典是否明确，或是通过父域继承。表 8.65 显示条目是通用于所有类型的域字典，条目仅属于下面的有关章节描述的一个特定域类型。

表 8.65 域字典通用条目

KEY	TYPE	VALUE
FT	name	(终端域要求：可继承) 字典域类型描述： Btn Button (见 647 页“按钮域”) Tx Text (见 653 页“文本域”) Ch Choice (见 656 页“选择域”) Sig (PDF 1.3) 签定(见 658 页“签定域”) 注：此条目可能出现在非终端域（下一级也是域），提供可继承 FT 值，然而，终端域逻辑上没有自己的类型，其仅仅是一个可继承属性的容器，任类型终端域可继承这些属性。
Parent	dictionary	(如果此域是域家谱中另一个域的子域，必需有；否则不要求) 域是这个域的直接父域（如果任何域的 Kids 序列包括这个域）。域最多有一个父域，也就是它能被包括在最多一个其它域的 Kids 序列。
Kids	array	(有时需要，描述如下) 对其第一级子域间接引用的序列。 在非终端域，Kids 序列要求参考第一级子域的域字典，在一个终端域，Kids 序列一般必须参考一个或多个分散的与此域相关的局部注释，然而，如果只有一个相关的局部注释，且它的内容已经被融入域字典，则 kids 必须忽略。
T	text string	(可选) 部分域名（见下边“域名”，和附录 H 执行注

释 115 和 116)

TU	text string	(可选; PDF1.3) 在域必须定义用户接口时 (比如指向域的错误或状态信息), 备用域名是用来取代实际域名的。当在为了有禁止权限的用户访问或其它目的抽取文档内容时, 文本在也是有用的。
TM	text string	(可选; PDF1.3) 映射名, 当从文档抽取接口表格域数据时使用。
Ff	integer	(可选; 可继承) 一系列标志指定各种域特征 (见表 8.66)。默认值: 0
V	(various)	(可选; 可继承) 域值, 格式变化依赖域类型, 参考无效的域类型描述, 获取进一步信息。
DV	(various)	(可选; 可继承) 默认值就是域复位的恢复值 (见 666 页 “复位操作”) 其值格式类同 V。
AA	dictionary	(可选; PDF1.2) 为了各种的触发事件 (见 8.5.2 节“触发事件”), 附加操作字典定义域行为, 此条目在注释字典 (见 8.4.1 节 “注释字典”) 刚好与 AA 条目有相同意思。

域字典 Ff 条目的值是一个无符号 32 位整形, 其包含指定各种域特征的标志位, 位位置在标志 word 被从 1 (low-order) 到 32 (high-order) 计数。在表 8.66 的各个标志类同于所有域类型。仅应用指定各种域类型的标志在描述那些类型的章节讨论, 所有的未定义标志位必须设为 0

表 8.66 类同所有域类型的域标志

BIT	POSITION	NAME
1	ReadOnly	如果设置, 用户可能不能改变域值, 任何相关的局部注释将不会与用户交互, 也就是它们不会响应鼠标点击或改变外表去响应鼠标操作, 此标志有益于那些域值被计算或从导入数据库的域。
2	Required	如设置, 域必须在它被提交操作输出时有值 (见 662 页

“提交操作”）。

3	NoExport	如设置，域一定不能被提交操作导出（见 662 页“提交操作”）。
---	----------	----------------------------------

域名

在域字典里的 T 条目（见 637 页表 8.65）有文本字符串定义域的部分域名。完全合格的域名称不被清晰的定义，而是用域的局部域名和父域名构建出来的。对一个没有父域的域，其局部域名和全域名是一样的。对一个其它域的子域，其全域名的形成是这样的，附加子域局部域名到父域的全域名后，并通过一个点（.）隔离：

parent' s_full_name.child' s_partial_name

比如，局部域名为 PersonalData 的域有一个子域，子域局部名是 Address，Address 也有一个子域，名为 ZipCode，那么这最底层域的域全名是：

PersonalData.Address.ZipCode

所以，所有后裔域继承父域的全域名作为构成其自己全域名的前缀。

可能的在不同域字典会有同样全域名，如果它们是同一个有名父域的后代，同时没有它们自己局部域名（T 条目）的域。如此域字典是同样底层域的不同体现；它们应该仅会在指定它们可见外观属性时不同。特别是有同样去域名的域字典必须有同样的域类型（FT），值（V），和默认值（DV）。

文本变量

当一个域的内容和属性事先已知，它的可视外观能被一个定义在 PDF 文件的外观流指定（见 603 页 8.4.4 节“外观流”和“局部注释”）。然而在有些情况下，域包含的文本其值可能直到看时才能知道，在文档显示时填充的文本，用户通过键盘和滚动列表等交互方式确定其内容文本类型。

在这种情况下，PDF 文档不能为显示域提供静态定义的外观流，相反，浏览器应用程序必须在观看时动态构建一个外观流，展示在表 8.67 的字典条目提供了关于域外观的一般信息，能与指定包含构建外观流的文本组合。

表 8.67 使用与所有包含变量文本域的条目

KEY	TYPE	VALUE
DA	string	（必需的，可继承）默认外观字符串，它包含连续有效

内容页图标或文本状态操作，文本状态操作定义如文本大小和颜色的属性。

Q	integer	(可选, 可继承) 代码指定输出文本的输出方式: 0: 左对齐 1: 居中 2: 右对齐 默认值: 0 (左对齐)。
DS	text string	(可选; PDF1.5) 默认字符串类型, 描述在 642 页“文本字符串详解”
RV	text string or	(可选; PDF1.5) 文本字符串详解, 642 页“文本字符串详解”有描述。

新外观流变成了外观字典里的平常外观 (N), 与域局部注释 (见 579 页表 8.15) 相关。
(如果局部注释没有外观字典, 浏览器应用程序必须创建一个, 并保存它到注释字典的 AP 条目中。)

在 PDF1.5, 表格域有文本标志集详解 (见表 8.73) 指定相关描述信息, 如 642 页“多文本字符串”中描述。对那些域, 下边描述通常不会使用, 并且当值变化时, 全部的注释外观值在重新生成。

就像所有外观流, 对单文本域外观流类是标段 Xobject, Xobject 的表格字典使其内容初始化如下:

- 资源字典 (Resources) 用自交互表格字典 DR 条目 (见表 8.63) 资源表格创建;
可同时参考附录 H 执行注释 117
- 表格坐标系中边界框 (BBox) 的左下角坐标设置为 (0,0), 其右上角坐标根据注释框矩形尺寸获取 (局部注释字典中的 Rect 条目)。
- 其它外观流表格字典的条目被设置为其默认值 (见 4.9 节“Xobjects 表格”)。
包括下边标记内容章节的外观流, 代表了部分输出文本流。

例 8.12

```
/Tx BMC % Begin marked content with tag Tx
q % Save graphics state
...Any required graphics state changes, such as clipping...
```

```
BT % Begin text object
...Default appearance string (DA)...
...Text-positioning and text-showing operators to show the variable text...
ET % End text object
Q % Restore graphics state
EMC % End marked content
```

BMC (标记内容开始) 和 EMC (标记内容结束) 操作符均在 10.5 节讨论, “标记内容”。
q (保存图形状态) 和 Q (储存图片状态) 都在 4.3.3 节讨论, “图形状态操作符”。BT (文本对象开始) 和 ET (文本对象结束) 在 5.3 节讨论, “文本对象”。见例 8.16。

在显示域变量文本时, 包含任何图形或文本状态的操作符默认外观字符串 (DA) 需要建立图形状态参数, 比如文本字体大小和颜色。只有操作符被允许在文本对象中时, 文本对象才出现在这个字符串 (见 167 页图 4.1), 但这个字符串至少必须包括一个 Tf (文本字体) 操作符连同字体和大小, 指定 font 的值必须与一个在默认资源字典的 Font 条目资源名匹配 (应用交互表格的 DR 条目; 见表 8.63)。size 为 0 意味着字体大小自定义 ; 它的大小被作为一个注释方框高度函数计算。

默认外观字符串应包含最多一个 Tm (文本矩阵) 操作。如这个操作符存在, 浏览器应用程序应该取代有水平和垂直位置值的转换部件, 是否恰当, 基于域值、(Q) 属性, 以及使用的任何布置规则, 如果默认外观字符串不包含 Tm 操作符, 浏览器应该插入一个到外观流 (有适当的水平和垂直转换模块) 在默认外观字符串之后, 和文本定位和文本显示操作变量文本之前。

要新一个现存外观流去反映新的域值, 浏览器应用程序应首先从文档的 DR 字典 (见表 8.63) 复制所有需要的资源到流的 Resources 字典, (如果 DR 和 Resources 字典包含有同名资源, 则已在资源字典应不受影响, 不能用 DR 字典的对应值取代), 接着浏览器应用程序应取代现存的外观流内容, 从/Tx BMC 到 EMC 更换为适当的新内容, 如例 8.12 显示的。(如果现存的外观流没有包含标记为 Tx 的内容, 则新内容应该被附在原流的后边。) 可参考附录 H 执行注释 118。

多文本字符串

PDF1.5 开始, 就像标记注释, 可变文本表格域的文本内容能包括格式类型信息。这些多文本字符串是完全格式的 XML 文档, 它部分符合 XFA 文本规范, 这本身是一个 XHEML1.0

字典子集，加上一些 CSS2 类型约束属性（见所有这些标准引用目录），这部分描述了此规范的基本成分。

表 8.68 列出了 XHTML 元素，它们支持多文本字符串，`<body>`元素是根元素；它要求的属性在表 8.69，其它元素（`<p>`和``）包含了封闭文本，封闭文本可能采用列在表 8.70 类型属性。这些类型属性是 CSS 内联类型属性，声明格式 `name:value`，每个声明用一个分号隔开，就像 646 页例 8.13 图示。

表 8.68 使用在多文本字符串的 XHTML 元素

ELEMENT	DESCRIPTION
<code><body></code>	XML 文档根元素，表 8.69 列出了此元素的必要属性
<code><p></code>	包含的文本解释为一个图表，可采用表 8.70 列出的类型属性。
<code><i></code>	包含文本以斜体显示
<code></code>	包含文本以粗体显示
<code></code>	组文本专为应用类型的目的（使用属性在表 8.70）、

表 8.69`<body>`元素属性

ATTRIBUTE	DESCRIPTION
<code>xmlns</code>	多文本字符串元素默认的命名空间。 必须为 <code>xmlns="http://www.w3.org/1999/xhtml"</code> <code>xmlns:xfa="http://www.xfa.org/schema/xfa-data/1.0".</code>
<code>xfa:contentType</code>	必须为 "text/html"。
<code>xfa:APIVersion</code>	标释生成多文本字符串软件的字符串，格式必须为 <code>software_name:software_version</code> , 这里 <code>software_name</code> 通过名字定义软件，不可以包含空间 <code>software_version</code> 定义软件版本，它包含一系列的由小数点分隔的整数，每个整数是一个版本号码，最左边的值是最大版本号，随着值向右逐渐变小，当比较几个字符串，版本是按照一定顺序组成的。比如“5.2”就新于“5.13”，因为 2 小于 13，并不能像十进制数那样对待字符串，当组成的字符串有不同的数字部分，拥有更少段的字符串隐含的加到右边，用 “0” 填补，使组成段数字的数目相同。
<code>xfa:spec</code>	对 PDF1.5, 多文本字符串符合 XFA 规范版本 2.02 和更新版本都支持，

对 PDF1.6, 版本 2.2 和更新版本支持。

表 8.70 多文本字符串的 CSS2 类型属性

ATTRIBUTE	VALUE	DESCRIPTION
Text-align	keyword	水平对齐, 可取值: 左, 右, 中
Vertical-align	decimal	据此调整包含文本的基线的数额, 正值表示是上标, 负值是下标。格式值<十进制>pt, 通常符号在前, “pt”跟随, 如: -3pt, 4pt.
Font-size	decimal	包含文本的字体大小, 格式值<十进制>pt.
Font-style	keyword	指定包含文本应该使用正常还是斜体显示, 可取值: 正常、斜体。
Font-weight	keyword	包含文本字体的重量, 可取值: 正常, 粗体, 100, 200, 300, 400, 500, 600, 7000, 800, 900 注: 通常正常为 400, 粗体为 700
font-family	list	用于显示包含文本的字体名或字体名列表(如果列表提供, 第一个包含的图形被使用去指定文本。)
font	list	快捷 CSS 字体属性格式: 字体: <font-style> <font-weight> <font-size> <font-family>
color	RGB value	包含字体的颜色, RGB 值, 在 sRGB 颜色空间指定(<http://www.srgb.com>)。可为下列两种格式之一: #rrggbb, 每个部分有 2 个 16 进制位 rgb(rrr, ggg, bbb) 是每个部分一个十进制值 注: 尽管被指定得颜色值用 sRGB 解释, 但是它们会转化为非 ICC 颜色空间的值, 当用于产生注释外观时。
text-decoration	keyword	下边关键字之一 <u>underline</u> : 包含文本应该有下划线 <u>line-through</u> : 包含文本应该有画线穿过

font-stretch	keyword	字体家族指定一个正常、压缩或放大，支持值从最窄的到最宽的是超压缩，极压缩，压缩，半压缩，正常，半扩展，扩展，极扩展和超扩展
--------------	---------	---

多文本字符串的指定由两个，一是可变文本格式字典的 RV 条目（见表 8.67），一是标记注解字典 RC 条目（见表 8.17），多文本字符串可能被分装成文本流（见 3.8.2 节“文本流”）。格式域使用多文本流应该也有多文本标志设置（见表 8.73）。

默认的字符串形式需要用自由文本注释的 DS 条目（见表 8.21）或变量格式域（见表 8.67）指定。字符串指定类型属性的默认值，用于任何类型属性，这些属性不能被明确指定为注释或域，所有表 8.70 的属性在默认类型字符串都是合法的，除了 RV 和 RC 条目，都用于产生外观，下边条目被兼容 PDF1.5 的浏览器忽略：注释用的 Contents 条目、自由文本注释 DA 条目、V、DA 和格式域 Q 条目。

注：标记注释和自由文本注释（见 581 页“标记注释”）不会使用默认的类型字符串，因为它们外形使用平台控件实施，所有要求浏览器应用程序选择一个适当的系统显示字体。

当一个格式域或注释包含多文本字符串，字符串平面文本（字符数据）应该也被保存（对格式域的 V 条目和注释的 Contents 条目）。使早期浏览器应用程序可读和编辑数据（尽管有格式信息损失）。当文件被保存时 DA 条目应该也被写出。

如果用不支持 PDF1.5 浏览器编辑包含多文本字符串的文档，即使相应的平面文本可能已经改变，但是多文本字符串不会被保存（因为浏览器不认识它们）。当支持 PDF1.5 浏览器从文档里读多文本字符串，它必须检查是否相应的平面文本已经通过使用下面的操作改变：

1、创建一个新的包含多文本字符串字符的平面文本字符串。字符参考（如）应该被转换为它们等量字符。

注：不要试图去保存有指定标记元素的格式。比如：尽管<p>元素意味新一行，但是在相关平面文本中，生成返回不会被保证。

2、如果任一个值使用 UTF-16 编码，如果必要，就尽量使其它的值用 UTF-16 编码。

3、组合结果字符串。

如果字符串不等，就假定域已经被早期浏览器修改，同时新的多文本字符串应从平面文本中创建。

当多文本字符串指定字体属性，浏览器应用程序应使用 15.3 节 CSS2 规范描述（见目录）的字体。强力推崇在默认资源字典中给定过程字体，就如在表 8.63 的 DR 条目指定；见附录 H 执行注释 119。

下面的例子图示了多文本局部注释字典的条目，DS 条目指定默认字体，RV 条目包含两个多文本绘图；第一个指定默认字体中的粗体和斜体绘图；第二个绘图改变字体尺寸。

例 8.13

```
/DS (font: 18pt Arial)          % Default style string using an abbreviated font
% descriptor to specify 18pt text using an Arial font
/RV (<?xml version="1.0"?><body xmlns="http://www.w3.org/1999/xhtml"
  xmlns:xfa="http://www.xfa.org/schema/xfa-data/1.0/"
  xfa:contentType="text/html" xfa:APIVersion="Acrobat:6.0.0" xfa:spec="2.0.2">
<p style="text-align:left">
<b>
<i>
  Here is some bold italic text
</i>
</b>
</p>
<p style="font-size:16pt">
  This text uses default text state parameters but changes the font size to 16.
</p>
</body> )
```

8.6.3、域类型

交互格式支持下列的域类型：

按钮域代表屏幕交互控件，用户能用鼠标操作。它们包括按钮，复选框和单选按钮。

文本域是空间框，在其中用户可以从键盘输入文本。

选择域包含几个文本项目，最多一个可能被选为域值，有滚动列表框和复合框。

签字域代表用户身份认证和文档内容有效性电子签名。

下面的章节详细地描述了每个域类型，将来可能还增加其它类型。

按钮域

按钮（域类型 Btn）是屏幕上的交互控件，用户能用鼠标来操纵它，有三种类型的按钮：

下压按钮是一个纯交互控件，可立即响应用户输入，触发恢复不保留状态（见 648 页“Pushbuttons”）。

复选框在开和关两种状态切换（见 648 页“Check Boxes”）。

单选框控件包含一组相关按钮，每个都可开或关。通常在一组这样的控件中，任何时间最多有一个单选按钮被选择，选择任一个单选按钮，一组内其它按钮自动去选。（这个规定也有例外，见 650 页“Radio Buttons”记录）

各种类型的按钮控件通过 Ff 标志区分，如表 8.71 所示。

表 8.71 按钮控件使用的标志项

BIT POSITION	NAME	MEANING
15	NoToggleToOff	(只对单选按钮) 如设置 1：有一个单选按钮必须一直被选择；单击正被选择的单选按钮没有效果，如为 0 点击已选择的去去选它，就没有任何按钮被选。
16	Radio	设为 1：控件是一组单选按钮，设为 0：控件是一个复合框。这些标志只有在 Pushbutton 标志位 0 时才有意义。
17	PushButton	为 1：域是一个下压按钮，不会存留值。
26	RadiosInUnison	(PDF1.5)，如为 1：在单选按钮控件里的一组单选按钮同时表示 ON 状态，将会一起打开或关闭，也就是如果一个被选，所有都被选；为 0：按钮们互斥（类同 HTML 单选按钮操作）

下压按钮

下压按钮是域类中最简单的控件，它有一个 Btn 控件类型和按钮标志（见表 8.71）设置。因为这型控件不保存值，它不能用域字典中 V 和 DV 标志项（见 637 页表 8.65）；

复选框

复选框控件代表一个或多个复选框，当用户使用鼠标和键盘操纵，在 ON 和 OFF 两

种状态切换。控件类型是 `Btn` 并且其 `PushButton` 和 `Radio` 标志（见表 8.71）都清除。每个状态有一个单独的外观，外观通过外观字典中局部注释（见 8.4.4 节“外观流”）的外观流控制。关状态外观是可选的，但如果存在，必选储存在名字为 `Off` 的外观字典中。推崇（但是并不要求）`on` 状态名字是 `Yes`。

域字典（见 637 页表 8.65）中 `V` 条目有名字对象，它代表复选框的外观状态，它用于从外观库中选择适当的外形状态。

例 8.14 显示了典型的复选框定义。

例 8.14

Example 8.14

```
1 0 obj
<< /FT /Btn
/T (Urgent )
/V /Yes
/AS /Yes
/AP << /N << /Yes 2 0 R /Off 3 0 R>>
>>
endobj
2 0 obj
<< /Resources 20 0 R
/Length 104
>>
stream
q
0 0 1 rg
BT
/ZaDb 12 Tf
0 0 Td
(8) Tj
ET
```

```
Q
endstream
endobj
3 0 obj
<< /Resources 20 0 R
/Length 104
>>
stream
q
0 0 1rg
BT
/ZaDb 12 Tf
0 0 Td
(8) Tj
ET
Q
endstream
endobj
```

从 PDF1.4 开始，对复选框和单选框按钮域字典就包含一个可选的 `Opt` 项（见表 8.72），其有一个文本字符串序列代表域中每个控件注释的输出值。它用于下列目的：

- 表示复选框和单选框控件在非拉丁书写系统的输出值。由于名字对象在外观库被限制到 `PDFDocEncoding`，所有它们不能用拉丁文表示。
- 允许单选按钮或复选框被单独确定，即使它们有同样的输出值。

例如，在一个或几个页重复一组复选框，并且需要的动作是当用户确定一个复选框，则相应在每个其它页面的复选框也被选择，在这种情况下，每个相应的复选框是 `Kids` 复选框控件序列的一个子集。

注：对单选按钮，如果 `RadiosInUnison` 标志位设为 1，则同样的行也会发生。如果为 0，在控件中最多有一个单选按钮被置 1，见附录 H 执行注释 120.

表 8.72 复选框和单选按钮控件的其余指定条目

KEY	TYPE	VALUE
Opt	array of text strings	(可选; 可继承; PDF1.4) 序列对每个在单选按钮或复选框控件的 Kids 序列的局部注释包含一个条目，条目都是一个文本字符串，代表相应局部注释的 ON 状态。 当这个条目出现，名称用来在代表在 AP 库每个注释的 ON 状态，名字用来计算生成类同注释 Kids 序列编号位置（从 0 开始）。即使在 OPT 序列有两个或多个同样的值，也能够区分注释。比如，两个单选按钮可能同样为 ON 状态，但是如果 RadiosInUnison 标志为 0，它们中只有一个可以被用户确定。

单选按钮

单选按钮控件是一组相关按钮。就像复合框，单个的单选按钮有两个状态，on 和 off。一个单选按钮可能不会被直接关掉，除非打开另一个按钮，通常，一组单选按钮在任何时间（定义为一个单选按钮控件的子控件）至多有一个按钮在 on 状态，选择任一个会自动去选所有其它单选按钮。

注：当在一个域的多个单选按钮是 on 状态，且 RadiosInUnison 标志位 1，会出现一个特例，这种情况下，一个 on，其它全 on。

域类型是 Btn, Pushbutton 标志（见表 8.71）为 0，且 Radio 标志位 1。这型的按钮控件还有一个 NoToggleToOff 标志，如为 1，则必须有一个单选按钮被选，此时，点击已被选的按钮不产生任何作用，如果 NoToggleToOff 设为 0，点击已选择按钮就会去选它，就再没有按钮被选。

在中单选按钮字典里（见 637 页表 8.65），Kids 条目有一个代表组中各个按钮局部注释的序列，父域 V 条目有一个名字对象对应任何一个当前处在 on 状态的子域；这个条目的默认值是 Off。例 8.15 显示了对一组单选按钮对象的定义。

例 8.15

```
10 0 obj %单选按钮控件
<< /FT /Btn
    /Ff ...
    /T (Credit card)
    /V /MasterCard
    /Kids [ 11 0 R
              12 0 R
            ]
>>
endobj

11 0 obj %第一个单选按钮
<< /Parent 10 0 R
    /AS /MasterCard
    /AP << /N << /MasterCard 8 0 R
          /Off 9 0 R
>>
>>
>>
endobj

12 0 obj % 第二个单选按钮
<< /Parent 10 0 R
    /AS /Off
    /AP << /N << /Visa 8 0 R
          /Off 9 0 R
>>
>>
>>
endobj

8 0 obj %外观流在 " on "状态
<< /Resources 20 0 R
```

```
/Length 104

>>

stream

q
0 0 1 rg

BT
/ZaDb 12 Tf
0 0 Td
(8) Tj

ET
Q

endstream

endobj
9 0 obj %外观流在 "off"状态
<< /Resources 20 0 R

/Length 104

>>

stream

q
0 0 1 rg

BT
/ZaDb 12 Tf
0 0 Td
(4) Tj

ET
Q

endstream

endobj
```

像复选框域，单选按钮域能使用在控件库 (PDF1.4) 可选的 Opt 条目，对非拉丁字符 (见表 8.72) 使用 Unicode 编码。Opt 有一个文本字符串系列与代表域 Kids 序列里各按钮局部

注释对应。

文本控件

文本控件（控件类型 Tx）是一个箱子或空间，在里边用户能从键盘输入文本。文本可能被限制为单行或页可能被允许扩展多行，这依赖于控件库 Ff 条目多行标志的设置。表 8.73 显示了适应于此类型控件的标志。

表 8.73 文本域指定标志

BIT POSITION	NAME	MEANING
13	Multiline	如设置，控件能包含多行文本；0：控件文本限制在一行。
14	Password	如设置，域用于输入安全密码，所以输入值不应该屏幕反馈可见。从键盘输入字符类型的应该取代为一些不可读的符号，如星号或原子符。 为保护密码秘密，浏览器应用程序应绝不保存在 PDF 文件的文本框值，如果这个标志为设置。
21	FileSelect	(PDF1.4) 如果设置，文本输入到框表示文件路径名，其文件内容将被设为框值。
23	DoNotSpellCheck	(PDF1.4) 如果设置，框内输入的文本不能被拼写检查。
24	DoNotScroll	(PDF1.4) 如果设置，框不能滚动（单行文本不能水平滚动，多行文本不能垂直滚动）去容纳超过它定义文本矩形的更多文本，一旦框满，多余的文本就不会被接受。
25	Comb	(PDF1.5) 只有在文本框字典（见表 8.74）的 MaxLen 条目存在，且 Multiline, Password, 和 FileSelect 标志清除才有意义。如设置，域被自动拆分成相等的空间位置或 combs，如 MaxLen 的值，并且文本被放置在那些 combs 中。

26

RichText

(PDF1.5) 如果设置, 这个域应该表示多文本字符串 (见 642 页“多文本字符串”)。如果这个域有值, 则框字典 (表 8.67) 的 RV 条目指定多文本字符串。

域文本有一个在域字典 V (value) 条目的文本字符串 (或流, 从 PDF1.5 开始)。文本字符串或流的内容用于构建外观流去显示此域, 如在 639 页“Variable Text”描述。此文本代表一种类型 (字体, 大小, 颜色等), 就如 DA 指定的 (default appearance) 字符串。

如果 FileSelect 标志 (PDF1.4) 被设置, 就如 file-select 控件的域功能。这种情况, 域文本代表文件路径名, 路径名内容将被做为域值提交:

- 对域传到 HTML 的表格格式, 提交使用 MIME 内容形式 multipart/form-data, 如在网络 RFC2045 多功能 Internet 邮件扩充服务 (MIME) 描述, 在第一部分互联网信息格式中 (见参考书)。
- 对表格数据格式 (FDF) 提交, 在 PDF 域字典 V 条目 (见 676 页“FDF 域”) 值是一个定义选择文件的文件规范 (3.10 节“文件规范”)。
- XML 格式不支持文件选择控件; 所有, 这种情况没有提交值。

除了通常条目适用于所有域 (见 637 页表 8.65) 和域包含可变文本 (见表 8.67), 域, 对文本域的域字典能包含显示在表 8.74 的其它条目。

表 8.74 文本域其它标记条目

KEY	TYPE	VALUE
MaxLen	integer	(可选; 可继承) 在文本域的最大字符长度。

例 8.16 是一个对典型文本域的对象定义。

例 8.16

```
6 0 obj
<< /FT /Tx
    /Ff ... %设置 Multiline 标志
    /T (Silly prose)
```

```
/DA (0 0 1 rg /Ti 12 Tf )  
  
/V (The quick brown fox ate the lazy mouse)  
  
/AP << /N 5 0 R >>  
  
>>  
  
endobj  
  
5 0 obj  
  
<< /Resources 21 0 R  
  
/Length 172  
  
>>  
  
stream  
  
/Tx BMC  
  
q  
  
BT  
  
0 0 1 rg  
  
/Ti 12 Tf  
  
1 0 0 1 100 100 Tm  
  
0 0 Td  
  
(The quick brown fox ) Tj  
  
0 -13 Td  
  
(ate the lazy mouse. ) Tj  
  
ET  
  
Q  
  
EMC  
  
endstream  
  
endobj
```

选择域

选择域（域类型 Ch）包含几个文本条目，其中一个或多个可能被选择为域值。此项目可能以下面两种形式之一交给用户：

- 滚动列表框

- 包含下拉列表的复合框并能进行文本编辑的方框，在框内用户能输入没有预先定义的值。

表 8.75 指定选择域的域标志项

BIT POSITION	NAME	MEANING
18	Combo	1: 控件是复合框；0: 控件是列表框
19	Edit	1: 复合框包含编辑文本框像下拉列表；0: 它只包含下拉列表。只有 Combo 为 1 才有意义。
20	Sort	1: 此域的选择项目应被安阿拉伯顺序排列，此标志是表格使用工具授权，不能被浏览器应用程序使用。浏览器应该仅仅按顺序显示选项，如出现在 Opt 序列（见表 8.76）的序列。
22	MultiSelect	(PDF1.4) 1: 多个域选项可能被同步选择；0: 此时最多一个被选。
23	DoNotSpellCheck	(PDF1.4) 1: 输入域的文本不能被拼写确定。此标志只 Combo 和 Edit 标志都为 1 时常有效。
27	CommitOnSelChange	(PDF1.5) 1: 定点设备一选择，新值就被提交。此选项使应用程序在选择一发生就执行动作，不用要求用户退出域，如果清除，直到用户退出域，新值才提交。

如表 8.75 各种选择域类型通过 Ff 条项标志区分，表 8.76 指定域字典条目指定选择域。

表 8.76 指定选择控件其余标志项

KEY	TYPE	VALUE
Op	array	(可选) 提供给用户的一个选项序列。每个序列元素要么是文本字符串表示一个可用选项，要么表示一个包含两个文本字符串的序列：选项的输出值和作为选项名字的显示文本（见附录 H 操作注释 121）。 如此条目不存在，没有选择提供给用户

Tl	integer	(可选) 对滚动列表框, 顶部索引 (下标指在 Opt 序列第一个列表可见选项), 默认值: 0
	array	(有时要求, 有时可选; PDF1.4) 对选择域其允许多选择(多项标志设置), 升序排列的整形序列, 基于 0 排列指示当前选项的 Opt 序列。当在 Opt 序列两个或多个有不同的名字的元素有相同的输出值或序列选择控件的值时, 此条目要求有。另外, 此条目允许但是不要求, 如这些被这条目定义的项目不同那些域定义 V 条目时 (见下), 则 V 条目先执行。

Opt 序列指定选择域选项列表, 其中每个都被一个文本字符串显示在屏幕上显示。每个 Opt 序列包含的元素, 或是自身文本字符串, 或两元素序列, 它的第二个元素是文本字符串, 它的第一元素是代表输出值文本字符串, 当从文档输出值到交互表格域时, 这个值要被使用的。

这控件字典的 V (value) 条目 (见 637 页表 8.65) 定义一个或多个在选择域的当前选择。如果域不允许多重选择, 既 MultiSelect 标志 (PDF1.4) 没设置, 或支持多选但当前仅一个项目被选, V 是一个代表选择项目名的文本字符串, 就如域字典的 Opt 序列中给定。如果多个条目被选, V 就是这种字符串序列。(代表在双元素 Opt 序列项目, 命名字符串是两元素序列的第二个。) V 默认值是 null, 表示当前没有项目被选。

例 8.17 显示了一个典型的选择区定义。

例 8.17

```
<< /FT /Ch
     /Ff ...
/T (Body Color )
/V (Blue)
/Opt[ (Red)
      (My favorite color )
      (Blue)
]
```

>>

签名域

签名域 (PDF1.3) 是一表格域，其中包含数字签名 (见 8.7 节，“数字签名”)。域字典代表一个签名域，它可能包含列在表 8.77 的其它条目，就像在表 8.65 描述的标准条目。域类型 (Ff) 是 Sig，且区值 (V) 是签名域，它包含签名并指定各种签名域属性 (见表 8.98)。

在签名域签名需要更新至少 V 条目和 AP 条目，AP 条目通常和局部注释相关。导出一个签名域典型是输出 T, V 和 AP 条目。

就像其它区，签名域实际可能通过一个局部注释字典描述，局部注释字典包含条目属于注释和域 (见 603 页“局部注释”)，在此注释的方框 (Rect) 给出其页上域位置。不能可见的签名域应该有一个默认高和宽的注释方框。

签名域的局部注释的外观字典 (AP) 定义页上域的可见外观 (见 8.4.4 节，“外观流”)。关于怎样 Adober 处理数字签名外观的信息技术说明 Digital Signature Appearances 中(见目录)。

表 8.77 指定签名域的其它条目项

KEY	TYPE	VALUE
Lock	dictionary	(可选；必须是间接引用；PDF1.5) 在这个签名域被签署时，指定将被上锁的一组格式域。表 8.78 列出在这字典中的条目。
SV	dictionary	(可选；必须是间接引用；PDF1.5) 字典种子值 (见表 8.79)，它包含的信息限制应用到此签名区的属性。

域字典 SV 条目值是一个种子值字典，它的条目 (见表 8.79) 提供签名的时候使用限制信息。它的 Ff 条目指定是否其它字典中的条目项是必须使用或仅仅推荐使用。

注：种子值字典可能包含属于多处理器私有条目的种子值。一个给定处理器应该仅使用那些永久属于本身的条目，而忽略其它。

表 8.78 签名域锁定字典标志项

KEY	TYPE	VALUE
Type	name	(可选) 字典描述的 PDF 对象类型; 1: 签名域锁存字典必须为 <code>SigFieldLock</code>
Action	name	(必需的) 一个名字, 它与 <code>Fields</code> 结合指示一组应被锁区。有效值是: <code>All</code> : 文档所有区 <code>Include</code> : 所有 <code>Fields</code> 指定的区 <code>Exclude</code> : 所有 <code>Fields</code> 没有指定的区
Fields	array	(必需的, 如果 <code>Action</code> 值包括 <code>Include</code> 或 <code>Exclude</code>) 包含区名的字符串序列。

表 8.79 签名域种子值字典条目

KEY	TYPE	VALUE
Type	name	(可选) 此字典描述的 PDF 对象类型; 如存在, 种子值字典必须为 <code>SV</code> 。
Filter	name	(可选) 指示将用于签字区的签字处理程序。
SubFilter	array	(可选) 指示可接受签名编码的名称序列。序列中第一个支持签名处理程序编码的名字实际上应在它签署时编码。
V	integer	(可选) 要求签名域签字处理程序的最小版本号。
Cert	dictionary	(可选) 认证种子值字典 (见表 8.80) 包含关于在签字时使用的认证方面的信息。
Reasons	array	(可选) 在签署文档时, 指定可能原因的字符串序列。
MDP	dictionary	(可选; PDF1.6) 字典中有一条目, 它的 <code>key</code> 是 <code>P</code> , 并且它的值是在 0 到 3 的整数, 值 0 定义签署是一般 (<code>non-author</code>) 签署 (见 8.7 节, “数字签署”)。值 1 到 3 用于自定义签名, 并且要与在 <code>DocMDP</code> 转换参数字典中的 <code>P</code> 值对应 (见表 8.100)。

		如果 MDP 不存在或不包含 P 条目，没有规则根据 签字类型或许可定义。
TimeStamp	dictionary	(可选; PDF1.6) 包含两个条目的时间戳字典: URL 一个指定时间戳服务器的字符串 URL 提 供一个时间戳兼容于 RFC3161, 网络 X.509 公钥时间戳的协议 (见目录)。 Ff 一个整形值, 为 1 (签名要求有时间戳) 或 0 (签名不要求有时间戳), 默认值: 0
LegalAttestation	array	(可选; PDF1.6) 指定可能合法认证 (见 8.7.4 节, “合法内容认证”) 的字符串序列。
Ff	integer	(可选) 指定字典中指定条目项的解释的一组标 志位。Ff 为 1 意味着签署者仅需要为此条目使 用指定值。Ff 为 0 意味着其它值也被允许。位 置 1 (Filter); 2 (SubFilter); 3 (V); 4 (Reasons); 5 (LegalAttestation)。默认值: 0。

表 8.80 认证种子值字典条目

KEY	TYPE	VALUE
Type	name	(可选) 字典描述的 PDF 对象类型; 如果存在, 认证种子值字典必须 为 SVCert。
Subject	array	(可选) 一个允许被签署的包含 DER-encoded X.509v3 认证的字符串 序列, X.509v3 认证描述在 RFC 3280, 网络 X.509 公匙协议, 认证和 证书吊销列 (见参考书)。
Issuer	array	(可选) 包含 DER-encoded 的 X.509v3 可接受认证发行人的字符串序 列, 如果签署者的认证链到任何指定的发行人 (直接或间接), 该认证 被认为是可接受签署。
OID	array	(可选) 字符串系列, 包含符号认证策略的对象身份 (OIDs), 必须出 现在签署过程认证。此字符串示例如: (2.16.840.1.113733.1.7.1.1)。 此域只有在 Issuer 值非空才有效。认证策略扩展描述在 RFC 3280 (见 目录)。

URL	string	(可选) 如果匹配认证没被发现, URL 能用于招收一个新凭据。
Ff	integer	(可选) 一组位标志, 指示此字典中指定条目项的解释, 此标志为 1 意思是签署者要求仅使用此条目的指定值, 标志为 0 意味着其它值也被允许。Bit 1 (Subject); Bit 2 (Issuer); Bit 3 (OID)。
默认值: 0		

8.6.4、表格操作

除了那些在 8.5.3 节 “操作类型” 描述, 交互表格支持四种特定的操作类型:

- **Submit-form** 操作就是传递名字和交互表格选择域值到一个指定统一的资源位置 (URL), 一般的网络服务器地址会处理它们并返回一个响应。
- **Reset-form** 操作复位交互表格域为它们设置的默认值。
- **Import-data** 操作从指定文本导出 FDF (表格数据格式) 数据到文档交互表格。
- **JavaScript** 操作 (PDF1.3) 引起一个脚本被编译和用 JavaScript 解释器执行。

Submit-form 操作

Submit-form 操作提交名字和已选交互表格域到 URL。一般网络服务器地址会处理它们并返回一个响应。表 8.81 展示了指定此类型操作的操作字典条目项。

操作字典的 Flags 条目标志值是一个无符号 32 位整形, 其包含指定操作各种特征的标志。在标志字中的位位置编号从 1 (顺序低位) 到 32 (顺序高位)。表 8.82 显示了标志位的意义; 所有未定义标志位保留并必须设为 0

表 8.81 指定目项

KEY	TYPE	VALUE
S	name	(必需的) 字典操作类型描述; 对 Submit-form 必须为 SubmitForm。
F	file specification	(必需的) URL 文件规范 (见 3.10.4 节, “URL 规范”) 在网络服务器中给出统一的资源位置 (URL) 脚本, 服务器会处理这些提交。
Fields	array	(可选) 序列定义提交中包含或排除的域, 这些依赖 Flags 条目的 Include/Exclude 标志设置 (见表 8.82)。每个序列元素或是一个对域字典的间接引用, 或是一个代表完全合格域名的字符串。两种元素可能混合在同一个序列。 如果这个条目被忽略, Include/Exclude 标志被忽略, 并且所有

的在文档交互标单的域被提交,除了哪些 `NoExport` 标志(见 638 页表 8.66) 被置 1。(没有域值的也可能被排除,依赖 `IncludeNoValueFields` 标志设置;见表 8.82)。见下文本表 8.82 的进一步讨论

Flags integer (可选; 可继承) 指定操作各种特征的一组标志 (见表 8.82)。
默认值: 0.

表 8.82 submit-form 操作标志

BIT POSITION	NAME	MEANING
1	Include/Exclude	0: <code>Fields</code> 序列 (见表 8.81) 指定哪个域被提交。(所有指定域的子集域也都被提交) 1: <code>Fields</code> 序列说明哪些域被排除。所有在文档交互界面的域都被提交除了那些列表在 <code>Fields</code> 序列的和哪些其 <code>NoExport</code> 标志被设 1 的。(见 638 页表 8.66)。
2	IncludeNoValueFields	1: 所有域区分通过被 <code>Fields</code> 序列和 <code>Include/Exclude</code> 标志被标出, 不管它们是否有值 (在域字典的 V 条目)。对没有值的域, 只有其域名被传送 0: 没有值的域不被提交。
3	ExportFormat	只有 <code>SubmitPDF</code> 和 <code>XFDF</code> 标志为 0 时才有意义。如为 1, 域名和值以 <code>HTML</code> 表格格式提交。如为 0, 它们以 <code>FDF</code> (表格数据格式) 传送; 见 8.6.6 节, “表格数据格式。”
4	GetMethod	1: 域名和值通过 <code>HTTP GET</code> 请求提交, 0: 它们使用 <code>POST</code> 请求提交, 这个标志只有在 <code>ExportFormat</code> 标志为 1 才有意义; 如果 <code>ExportFormat</code> 标志为 0, 它也必须为 0.
5	SubmitCoordinates	如为 1, 鼠标点击坐标可导致 <code>Submit-form</code> 操作当做表格数据的一部分传送, 其坐标值相对于域局部注

释方框左上角的。它们表示的数据格式是：

`name. x=xval&name. y=yval`

如果出现，`name` 是域映射名（`TM` 在域字典）；否则，`name` 是域名。如果 `TM` 条目值只是一个空格字符，之后的名字和点被压缩，结果格式是：

`x=xval&y=yval`

这个标志只有在 `ExportFormat` 标志被置 1 时才有意义，如果 `ExportFormat` 标志为 0，这个标志必须也为 0.

6	<code>XFDF</code>	(PDF1.4) 只有在 <code>SubmitPDF</code> 标志为 0 才有意义，如为 1，域名和值被像 XFDF 提交。
7	<code>IncludeAppendSaves</code>	(PDF1.4) 只有在表格以表格数据格式（即 XFDF 和 <code>ExportFormat</code> 标志都被置 0）提交才有意义。如果为 1，提交的 FDF 文件包括的内容有所有对底层 PDF 文档的增加更新，就像在 FDF 字典（见 673 页表 8.89）的 <code>Differences</code> 条目项。如果为 0，增加更新不会被包括。
8	<code>IncludeAnnotations</code>	(PDF1.4) 只有在表格以表格数据格式提交（即 XFDF 和 <code>ExportFormat</code> 标志都被置 0）才有意义。如果为 1，提交的 FDF 文件包括所有隐含在 PDF 文档的局部注释（见 581 页“局部注释”）。如果为 0，局部注释不会被包括。
9	<code>SubmitPDF</code>	(PDF1.4) 如设为 1，文档被像 PDF 提交，使用 MIME 内容类型 <code>application/pdf</code> （网络 RFC2045 描述，多用途网络邮件扩展（MIME），部分一：网络信息体格式；见目录）。如为 1，除了 <code>GetMethod</code> ，所有其它标志被忽略。
10	<code>CanonicalFormat</code>	(PDF1.4) 如设为 1，任何提交域值表示数据被转换为标准格式，标准格式描述在 3.8.3 节，“数据”（格式域作为日期解释本身不能在域中清楚指定，但是 JavaScript 代码可处理它。）

11	Exc NonUserAnnots	(PDF1.4) 只有在表格以表格数据格式提交(即 XFDF 和 ExportFormat 标志都被置 0 且 IncludeAnnotations 标志被设置为 1) 才有意义。如设置, 它仅包含一些局部注释, 它们的 T 条目(见表 8.17) 匹配当前用户名, 就像远程服务器决定哪个表格被提交。(对局部注释 T 条目指定的文本标签, 将在标题条上弹出窗口显示注释, 被假定代表用户名签定注释) 这允许多用户合作定义一个远程 PDF 文档且不会影响他人的注释。
12	Exc FKey	(PDF1.4) 只有在表格被一表格数据格式(即 XFDF 和 ExportFormat 标志都被置 0) 提交时才有意义。如为设置, 已转化的 FDF 排除 F 条目。
14	EmbedForm	(PDF1.4) 只有在表格以表格数据格式(即 XFDF 和 ExportFormat 标志都被置 0) 提交时才有意义。如为设置, 则已提交 FDF 的 F 条目是一个包含嵌入文件流的文件格式, 其代表来自 PDF 文件的 FDF 正被提交。

这组域的名字和值要被提交, 这组域通过操作字典 (表 8.81) 的 Fields 序列定义, 还有 Flags 条目 (表 8.82) 的 Include/Exclude 和 IncludeNoValueFields. 每个 Fields 序列元素定义一个交互表格域, 要么通过间接参考它的域字典 (PDF1.3), 要么通过它完全规格的域名(见 638 页“域名”)。如果 Include/Exclude 标志为 0, 则提交包含所有列表在 Fields 序列的域, 还有任一个在域家谱的子域。如果 Include/Exclude 标志设置为 1, 则除了列在 Fields 的序列, 提交包含文档交互表格的所有内容。

注: 域字典 Ff 条目的 NoExport 标志 (见 637 页表 8.65) 优先于 Fields 序列和 Include/Exclude 标志的操作, NoExport 标志被设置的域绝不能在 submit-form 操作。

域名和值可能通过下面的任一种格式被提交, 其依赖于操作的 ExportFormat, SubmitPDF 和 XFDF 标志 (见参考目录) 的设置。

- HTML 表格格式 (描述在 HTML 4.01 规范)
- 表格数据格式 (FDF) 描述在 8.6.6 节, “表格数据格式”; 也可见附录 H 执行注释 122。
- XFDF 是一本基于 XML. XFDF 的 FDF 版本, 它被描述在 Adobe 技术文档 XML 表格数据格式规范, 版本 2.0XML 描述在 W3C 文档可扩展标记语言 (XML) 1.1

- PDF (在这种情况, 提交全部文档而不是单个域和值)。

每个域提交的名字是它的全域名 (见 638 页“域名”), 提交的值由它域字典 V 条目指定。

注: 在 FDF 提交的下压按钮域, 提交值是在域局部注释字典的 AP 条目, 如果 submit-form 操作字典没包含 Fields 条目, 此按钮域不会被全部提交。

没有值的域 (也就是, 其域字典不包含 V 条目) 通常不被包括提交。submit-form 操作的 IncludeNoValueFields 标志能覆盖这行为, 如这个标志被设置, 这些无值域仅通过名字包括入提交, 且没有相关值。

Reset-Form 操作

Reset-Form 操作重设已选择的交互格式域为它们的默认值; 也就是, 它设置在域字典中的 V 条目值到 DV 条目 (见 637 页表 8.65)。如果没有默认值为域定义, 它的 V 条目被除去, 对能没值的域 (如下压按钮), 其操作没效果。表 8.83 显示了这种类型操作的操作字典条目。

操作字典 Flags 条目的值是个无符号 32 位整型, 其包含指定操作各种特征的标志。在这个标志字内的 bit 位置被排号从 1 (low-order) 到 32 (high-order)。在发布时, 这种类型只有一个标志; 表 8.84 显示它的意思。所有未定义的标志 bits 被保留并必须设为 0

表 8.83 指定 reset-form 操作的其它条目

KEY	TYPE	VALUE
S	name	(必需的) 字典描述的操作类型; 对 reset-form 操作必须是 ResetForm。
Fields	array	(可选) 一序列定义那个域被重设或那个被从重设中排除, 依赖于在 Flags 条目 (见表 8.84) 中 Include/Exclude 标志的设置。每个序列元素要么是与字典的间接应用, 要么 (PDF1.3) 是字符串代表域完全合格名。两种元素可能被混合在同一个序列。 如果这条目被忽略, Include/Exclude 标志被忽略; 所有在文档交互界面的域被重设。
Flags	integer	(可选; 可继承) 一组标志指定各种操作的特征 (见表 8.84)。默认值: 0

表 8.84 reset-form 的操作条目

BIT POSITION	NAME	MEANING
1	Include/Exclude	如 0: Fields 序列 (见表 8.83) 指定那些域重设。(所有在域级系中指定域的子域也都被重设。); 如果 1: Fields 序列指示那些域被从重设中排除; 也就是, 所有在文档交互格式的域除了那些列在 Fields 序列的域都被重设。

Import-Data 操作

Import-Data 从一个指定文本(见 8.6.6 节,“格式数据格式”)中导入格式数据格式(FDF)数据到文档交互格式。表 8.85 显示指定这种类型操作的操作字典条目。

表 8.85 指定 Import-Data 操作的其它条目项

KEY	TYPE	VALUE
S	name	(必需的) 操作类型的字典描述; Import-Data 操作必须设为 ImportData。
F	file specification	(必需的) 导入数据的 FDF 文档。(见附录 H 执行注释 123 和 124)

JavaScript 操作

JavaScript 操作 (PDF1.3) 导致一个脚本通过 script 解释器被编译执行。依赖脚本的属性, 各种在文档的交互格式可能更新它们的值或改变它们的可见外观, 网景通讯公司的客户端 JavaScript 参考和 Adobe Acrobat JavaScript 的脚本参考 (见目录) 给出关于 JavaScript 脚本内容和作用的详细的描述。表 8.6.6 显示了此类型操作的操作字典条目。

表 8.86 指定 JavaScript 操作的其它条目项

KEY	TYPE	VALUE
S	name	(必需的) 操作类型的字典描述; JavaScript 操作必须设为 JavaScript。
JS	file	(必需的) 要被执行的包含 JavaScript 脚本的字符串或

specification 流。

注: `PDFDocEncoding` 或 `Unicode encoding` (后者定义通过 `Unicode` 前缀 `U+FEFF`) 用于编码字符串和流的内容。(见附录 H 执行注释 125)

为了支持 `JavaScript` 脚本使用有参函数, 在 PDF 文档名字字典 (见 3.6.3 节, “名字字典”) 的 `JavaScript` 条目能包含名字树, 其映射名字字符串到文档水平的 `JavaScript` 操作。当文档被打开, 所有在这个名字树的操作被执行, 定义 `JavaScript` 函数为了其他文本脚本的使用。

注: 在名字字典中, 关联独立 `JavaScript` 操作的名字字符串仅像一个组织和封装脚本的方便工具。名字随意且不能承受任何有关 `JavaScript` 的名字空间。

8.6.5、命名页

文档名字字典 (见 3.6.3 节, “名字字典”) 的可选 `Pages` 条目 (PDF1.3) 包含一个名字树, 它映射名字字符串到文档内独立页。命名一页允许它被在两种方式引用:

数据导入操作能加命名页到文档进入 `FDF` 正被导入, 或为一页或为一个按钮外观。

`JavaScript` 操作执行的脚本能到有名页到当前文档作为一个正常页。

要对用户可见的有名页应该被留在页树 (见 3.6.2 节, “页树”), 并且这应该是一对引用在合适名字字典 `Pages` 树的页节点。如果页不能被浏览器应用程序播放, 它就应该被从名字字典的 `Templates` 引用。不可见页应该有一个 `Template` 而不是 `Page` 对象类型, 并且应该没有 `Parent` 或 `B` 条目 (见 119 页表 3.27)。不管是否页被命名在 `Pages` 或 `Templates` 树, 或被加到一个文档通过 `import-data` 或 `JavaScript` 操作, 新复制不被本身命名。

8.6.6、格式数据格式

这节描述格式数据格式 (`FDF`), 文件数据格式用于交互格式数据 (PDF1.2)。`FDF` 被使用当提交格式数据给服务器, 接受相应, 并合并到交互格式。它也能用于导出格式数据到 `stand-alone` 文件, 能被保存、电传播, 和导回到相应的 PDF 交互格式。另外, PDF1.3 开始, `FDF` 能用于定义一个注释容器, 其独立于它们应用的 PDF 文档。

`FDF` 是基于 PDF 的; 它使用同样的句法 (见 3.1 节, “词汇公约”) 和基础对象类型 (3.2 节, “对象”), 本质上相同的文件结构 (3.4 节 “文件结构”)。然而, 它在下列方式不同

于 PDF:

- 交叉参考表（3.4.3 节，“交叉参考表”）是可选的；
- FDF 文件不能被升级（见 3.4.5 节，“更新升级”）。对象可能仅第 0 代，没有两个对象有同样对象号；
- 文档结构比 PDF 更简单，由于 FDF 文档体仅包含一个要求对象。
- 流长度可能不会通过一个间接引用对象指定；

FDF 使用 MIME 内容类型应用程序/vnd. Fdf。在 Windows 和 UNIX 平台，FDF 文件有扩展名. fdf；在 Mac OS，它们有文件类型 ‘FDF’。

FDF 文件结构

FDF 文件构建本质上是与 PDF 文件同样的方式，但是仅包含哪些要求为交互格式和注释数据导出和导入元素，它包含三个要求元素和一个可选元素（见图 8.9）：

单行头定义文件复合的 PDF 字典版本号

体包含对性组成文件内容

可选 cross-reference table（交叉引用表）包含关于文件对象信息

跟踪器给出在文件体内的各种对象位置

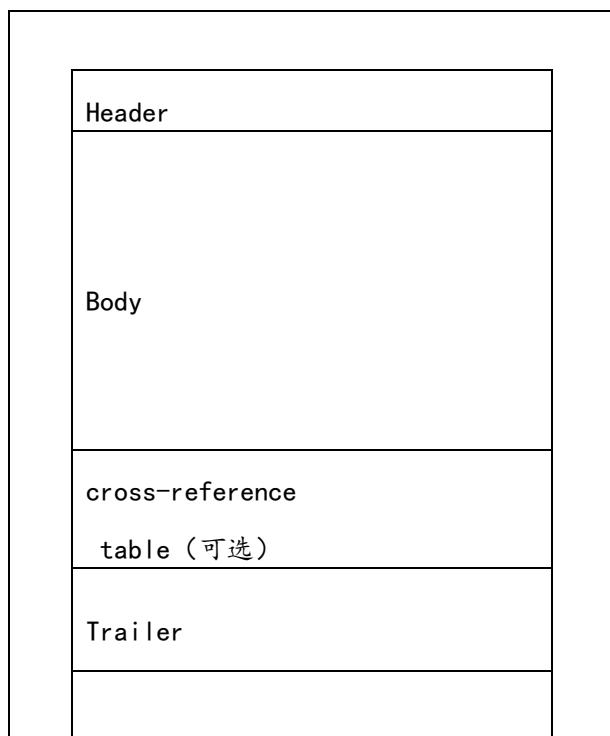


图 8.9 FDF 文件结构

FDF 头

FDF 文件的第一行是 **header**, 通常是为定义符合文件 PDF 字典版本。然而, 由于历史原因。这个版本号现在被固定且必须读

%FDF-1.2

实际版本号现在通过在 FDF 目录字典的 **Version** 条目给定 (见“FDF 目录,”下; 也见附录 H 执行注释 126)。

FDF 体

FDF 文件体包含一序列间接对象, 其代表文件目录 (见 672 页“FDF 目录”) 和任一其它目录可能引用的对象。对象是同样的基础类型, 其描述在 3.2 节, “**Objects**”。就像在 PDF, FDF 可为直接或间接对象。

FDF 追踪器

FDF 文件追踪器使能读文件应用程序在文件体内去寻找重要对象。最后一行文件仅包含文件结束标志 **%%EOF**。此标志前是 FDF 追踪器字典, 含有关键字 **trailer**, 被一列一个或多个关键值对包进双角括号内。唯一要求的关键是 **Root**, 它的值是一个文件目录间接引用到的字典 (见表 8.87)。追踪器可能选择性地包含其它对象条目, 其被应用在目录内。

表 8.87 FDF 追踪字典条目

KEY	TYPE	VALUE
Root	dictionary	(必需的; 必须是间接引用) FDF 文件目录对象 (见下边“FDF 目录”)。

所以, 追踪器有总体结构

trailer

<< /Root c 0 R

key2 value2

...

```
keyn valuen
>>
%%EOF
```

这里 C 是文件目录字典的对象编号。

FDF 目录

FDF 文件对象层级的根节点是目录字典，通过在文件按追踪器字典中的 Root 条目定位（见上“FDF 追踪器，”）。如显示在表 8.88。唯一要求在目录的条目是 FDF；它的值是 FDF 字典（表 8.89），相反它包含对其它对象应用描述文件内容。目录可能也包含一个可选 Version 条目定义 FDF 规范遵循的版本

表 8.88 FDF 目录规范条目项

KEY	TYPE	VALUE
Version	name	(可选； PDF1.4) PDF 规范版本是这个 FDF 文件遵循的（如例 1.4），前版本在文件头（见 671 页“FDF 头”）的中指定。如果头指定早期版本，或如果没有此条目，文档遵循头部指定的版本。 注：此条目值是一个名字对象，不是一个数字，在 FDF 文件被写时，所有必须在前边有一个反斜杠字符（例如，/1.4）。
FDF	dictionary	(必需的) 这个文件的 FDF 字典（见表 8.89）。
Sig	dictionary	(可选； PDF1.5) 签字字典，说明文档已被签署

表 8.89 FDF 目录规范条目项

KEY	TYPE	VALUE
Status	string	(可选) 状态字符串要被显示指定操作结果，一般是提交格式操作（见 662 页“提交格式操作”）。字符串使用 PDFDocEncoding 编码。（见附录 H 自行记录 127）这个条目和 Pages 条目可能不会都出现。
Pages	array	(可选； PDF1.3) 用于追加 FDF 页序列字典（见 679 页“FDF 页”）描述到一个目标 PDF 文档的新页。Fields 和 Status

条目可能不会和这个条目一起出现。

Encoding	name	(可选; PDF1.3) 编码要用于任一 FDF 域值、选项 (V 或 Opt 在域字典; 见 677 页表 8.92) 或域名, 是以一个字符串而不是统一码前缀 U+FEFF 开始。(见附录 H 自行记录 128) 默认值: PDFDocEncoding.
Annots	array	(可选; PDF1.3) 一 FDF 注释序列字典 (见 681 页 “FDF 注释字典”)。这序列能包括除去 Link, Movie, Widget, PrinterMark, Screen, 和 TrapNet 以外的任何标志类型注释。标准类型列在 580 页表 8.16。
Differences	stream	(可选; PDF1.4) 当它被打开时, 流包含所有增加到底层 PDF 文档中更新的字节 (见 3.4.5 节, “增长更新”)。提交格式操作, 是提交文档到一个远程服务器, 就像 FDF 使它的 <code>IncludeAppendSaves</code> 标准被设置 (见 662 页 “” 提交格式操作)。流内容在提交时, 允许任何数字签名 (见 8.7 节 “数字签名”) 被提交给服务器。增加更新在提交发生前一刻被自动执行, 为了记录所有对文档的改变, 注意文档被打一次打开时, 提交总包括增加的全部更新, 即使它们中的一些可能已经被包括进提交过程了。 注: 尽管 Fields 或 Annots 条目 (或两个) 可能随着 Differences 一起出现, 但并不保证它们的内容将被连在一起, 特别是, 如果 Differences 包含一个数字签名, 只有在 Differences 给定格式域值, 流才能被认为是可信的签名。
Target	string	(可选; PDF1.4) 浏览框架名, 在其中底层 PDF 文档将被打开。这类似于 HTML <code><href></code> 标签的目标属性行为。
EmbeddedFDFs	array	(可选; PDF1.4) 文件说明序列 (见 3.10 节“文件规格”), 代表其它嵌入的 FDF 文件 (3.10.3 节, “嵌入文件流”)。
JavaScript	dictionary	(可选; PDF1.4) JavaScript 字典 (见表 8.91), 定义文档层次 JavaScript 脚本

在 FDF 字典指定 EmbeddedFDFs 条目的嵌入 FDF 文件可能被加密。除了一般的嵌入式文

件流，流字典要代表这样一个加密 FDF 文件，就必须包含其它在表 8.90 显示的条目，去确定用于加密文件 FDF 加密算法的调整号。尽管 FDF 加密机制被从 PDF 文件加密中独立出来，PDF 文件加密在 3.5 节“加密”调整 1（发布时定义的唯一一个）描述，用于一个类似 RC4 基于 40bit 的加密钥匙加密算法，加密钥匙通过 MD5 hash 方法计算，填充用户使用一个密码作为输入，计算被指定为 100 页算法 3.2 阶段 1 和 2；前 5 个结果字节是嵌入式 FDF 文件的加密钥匙。

表 8.90 加密 FDF 文件用嵌入式文件流字典的其它条目

KEY	TYPE	VALUE
EncryptionRevision	integer	(必需的，如果 FDF 文件被加密；PDF1.4) FDF 用于加密文件的加密算法修订号，发布时定义的唯一有效值是 1

FDF 字典 JavaScript 条目的 JavaScript 字典，其 JavaScript 脚本是文档层次的全局量，而不是独立域。字典能包含脚本定义 JavaScript 函数为其它文档脚本使用，同时脚本将被在 FDF 文件导入前后立即执行，表 8.91 显示了此字典内容。

表 8.91 JavaScript 字典中的条目项

KEY	TYPE	VALUE
Before	string	(可选) 字符串或流包含 JavaScript 脚本的会被宰 FDF 文件在将要导入前执行。
After	string	(可选) 字符串或流包含 JavaScript 脚本的会被宰 FDF 文件在被刚导入后执行。
AfterPermsReady	string	(可选；PDF1.6) 包含 JavaScript 脚本的字符串和流将被执行在 FDF 文件被导出后，且在 PDF 文档的使用权已经被决定（见 692 页“UR”）。 注：验证使用权要求整个文件出现，在这种情况，此脚本推迟执行直到请求出现。
Doc	array	(可选) 定义其它 JavaScript 脚本序列会被加到那些被定义在文档名字典（见 3.6.3 节“字典名”）的 JavaScript 条目。序列包含一个元素平均数，成对组

织。每对的第一个元素是一个名字，第二个是一个字符串或流定义相应于那名的脚本，每个定义脚本被加到那些已经在名字典，并会被执行在定义 **Before** 条目的脚步之前。像 668 页“**JavaScript 操作**”描述。这些脚本用于定义 **JavaScript** 函数为其它文档脚本使用。

FDF 域

每一个在 FDF 文件的通过 FDF 域定义描述的域，表 8.92 显示了此类型定义的内容。多数条目有相同格式和意思就像在域定义（637 页表 8.65, 640 页表 8.67, 654 页表 8.74, 657 页表 8.76）相应条目项，或者一个局部定义字典（570 页表 8.11 和 603 页表 8.35）。除非在表中指出，导入一个域引起在 FDF 域字典条目项值取代那些域中相应的条目，其有同样完全合格名在目标文档中。（见附录 H 执行注释 129–134）

表 8.92FDF 文件字典中的条目项

KEY	TYPE	VALUE
Kids	array	(可选) 一序列包含此域低级孩子域。 注：不像在 PDF 文件的域子集，它必须被指定为间接对象引用，FDF 域可能是直接或间接对象。
T	Text string	(必需的) 局部域名（见 638 页“ 域名 ”）
V	(various)	(可选) 域值，它依赖域类型格式化变量；见 8.63 节私有域类型描述获取更多信息。
Ff	integer	(可选) 一组标志量指定各种特征（638 页表 8.66, 648 页表 8.71, 653 页表 8.73 和 656 页表 8.75）。 当导入到交互格式，此条目值取代在格式相应域字典的 Ff 条目，如这个域出现，任一 SetFf 和 ClrFf 条目被忽略。
SetFf	integer	(可选) 一组标志要被设置（turned on）在格式相应域字典的 Ff 条目，在 SetFf 中 Bits 为 1 引起 Ff 相应位置 1。这条目被忽略，如果一个 Ff 条目出现在 FDF 域字典。

ClrFf	integer	(可选) 在格式响应域字典的 Ff 条目的一组标志将被清除 (turned off)。在 ClrFf 中位置 1 的导致在 Ff 对应位被设置为 0。如 SetFf 条目页出在在 FDF 域字典, 它被应用在这条目前, 这条目被忽略如果 Ff 条目出现在 FDF 域字典。
F	integer	(可选) 一组指定域局部注释的各种特性的标志(见 8.4.2, “注释标志”)。当导入一个交互格式, 此条目的值取代对应在格式注释字典的 F 条目, 如此域出现, SetF 和 ClrF 条目, 都会被忽略。
SetF	integer	(可选) 一组标志将被设置 (turned on) 在格式相应局部注释字典的 F 条目。在 SetF 中 Bits 为 1 的为 0 导致对应在 F 的 bits 位被设置为 1。这个条目被忽略, 此条目被忽略如果 F 条目出现在 FDF 域字典中。
ClrF	integer	(可选) 一组在格式对应局部注释字典的 F 条目的标志要被清除 (turned off)。在 ClrF 中 Bits 为 1 的导致对应在 F 条目中位被设置为 0。如果 SetF 条目也出现在 FDF 域字典, 它在这条目前应用。这条目被忽略如 F 条目出现在 FDF 域字典。
AP	dictionary	(可选) 一个外观字典指定一个下压按钮域外形(见 648 页“下压按钮”)。外形字典内容显示如 579 页表 8.15, 除非 N, R 和 D 条目的值必须全是流。
APRef	dictionary	(可选; PDF1.3) 字典有对外部 PDF 文件应用的包含页去用于按钮域外观。这字典类似于字典外观(见 579 页表 8.15), 除了 N, R, 和 D 条目值必须全是有名页应用字典 (681 页表 8.96)。这条目被忽略如 AP 条目存在。
IF	dictionary	(可选; PDF1.3; 只是按钮域) 一个图标适应字典(见表 8.93)指定这样显示按钮域图标在它局部注释的注释方框内。
Opt	array	(必需的; 只是选择域) 提交给用户的选项序列。

		每个序列元素能采用下边两种格式之一。
		文本字符串代表一个可用选项
		两元素序列包含一个文本字符串表示一个可用选项之一，且一个默认外观字符串去构建项目外观动态地在浏览时（见 639 页“变量文本”）
A	dictionary	（可选）当这个域局部注释被激活时，要执行的操作（见 8.5 节，“操作”）。
AA	dictionary	（可选）一个额外操作字典定义域行为去响应各种触发事件（见 8.5.2 节“触发事件”）。
RV	text	（可选；PDF1.5）一个多文本字符串，如描述在 642 页“多文本字符串”。

在 FDF 域字典按钮表示域的可选 IF 条目中，有图表适应字典（PDF1.3）指定怎样在它的局部注释矩形显示按钮图表。表 8.93 显示此类型字典的内容。

表 8.93 图表适应字典条目项

KEY	TYPE	VALUE
SW	name	（可选）图表应该被缩放到注释矩形内的情况： A 全尺寸 B 只有当图表大于注释矩形时缩放 C 只有当图表小于注释矩形时缩放 D 不缩放 默认值：A
S	name	（可选）缩放类型使用： A 变形缩放：缩放图表去适合注释矩形确切的，没有考虑它的原有宽高比（宽高比） P 比例缩放：缩放图表区适应注释矩形的宽和高，答案保持图表原始宽高比。如要求水平和垂直缩放因素不同，使用两者更小，中心图表在注释矩形在其它尺寸。 默认值：P.

A	array	(可选)两树序列在 0.0 和 1.0 表示剩余空间部分去安排在图表左下角。一个 [0.5 0.5] 值集中它在矩形。这条目被使用只有如果图表被适当缩放。默认值: [0.5 0.5]
FB	boolean	(可选; PDF1.5) 如为 true。表示按钮外观应该被缩放去完全适应注释边界, 而不用考虑边界线宽; 见附录 H 执行注释 135。默认值: false。

FDF Pages

可选 Pages 域在 FDF 字典 (见 673 页表 8.89) 包含一序列的 FDF 页字典 (PDF1.3) 描述新页要被加到目标文档。表 8.94 显示这种类型字典内容。

表 8.94FDF 页字典条目项

KEY	TYPE	VALUE
Templates	array	(必需的) 一个 FDF 模板字典序列 (见表 8.95) 描述有名页其作为一个页模板。
Info	dictionary	(可选) 一个 FDF 页信息字典包含其它信息关于页, 在发布时, 没有已定义的对此字典的条目。

FDF 模板字典包含描述一个有名页信息, 有名页就像一个模板。表 8.95 显示这类型字典内容。

表 8.95FDF 模板字典条目项

KEY	TYPE	VALUE
TRef	dictionary	(必需的) 一个有名页参考字典 (见表 8.96) 指定模板位置。
Fields	array	(可选) 一个对 FDF 域字典参考序列 (见 677 页表 8.92) 描述根域要被导出 (那些在层次域没有父辈域的)。
Rename	boolean	(可选) 标志指定是否导入从模板域可能被重命名当名字与现存域名冲突; 见下进一步讨论。默认值:

true。

从模板导入的域名有时可能与目标文档现存的域相冲突。如果同样模板页超过一次被导入或两个不同模板有同名域，这就可能发生。如果模板字典 **Rename** 标志的值是 **false**，域不会被重命名；这导致目标文档的多个同名域，在每次 FDF 文件给一个给地域名提供属性时，所有同名的域都被更新。（见附录 H 执行注释 136 和 137）

FDF 模板字典的 **Tref** 条目，有一个描述外部模板位置或页的有名页参考字典的元素。

表 8.96 显示这类型字典内容。

表 8.96 FDF 有名页参考字典条目项

KEY	TYPE	VALUE
Name	string	(必需的) 参考页名。
F	File	(可选) 文件包含有名页，如果这条目不再，它假
specification		定页驻留在相关 PDF 文件。

FDF 注释字典

每个在 FDF 文件的注释必须有一个 **Page** 条目（见表 8.97）指示资源文档到附后注释。

表 8.97 FDF 文件注释字典条目项

KEY	TYPE	VALUE
Page	integer	(必需的，对在 FDF 文件的注释) 注释应出现的序页码，这儿页 0 是第一页。

8.67 XFA 格式

PDF1.5 介绍了对基于 Adobe XML 格式构架的交互格式 (XFA) 的支持。交互格式字典（见表 8.63）的 XFA 条目，指定了一个 XFA 资源，它是一个包含格式信息的 XML 流。XFA 资源格式在 XML 数据包装规范中被描述，XFA 格式在 XFA 规范，版本 2.2（见参考）中被详细描述。

XFA 条目可能是一个流，包含整个 XFA 资源，或一个序列指定的组成 XFA 资源的独立包。资源包括下面信息，但不会被限制：格式模板（指定模板包），用于描述格式特性，包括它的域、计算、验证和格式化；数据（在数据集包指定），代表格式状态；配置信息（在配置包指定），要求去适当处理格式模板和相关数据。

每个包代表一个完整地 XML 元素。除了第一个包和最后一个包，它指示开始和结束的图标是 xdp:xdp 元素。例 8.18 显示了 XFA 条目包括了一个包序列；

例 8.18

```
1 0 obj           XFA entry in interactive form dictionary
<< /XFA [(xdp:xdp) 10 0 R XFA resource specified as individual packets
(template) 11 0 R
(datasets) 12 0 R
(config) 13 0 R
(/xdp:xdp) 14 0 R ]
>>
endobj
10 0 obj
stream
<xdp:xdp xmlns:xdp="http://ns.adobe.com/xdp/">
Endstream

11 0 obj
stream
<template xmlns="http://www.xfa.org/schema/xfa-template/2.1/">
... remaining contents of template packet...
</template>
Endstream

12 0 obj
stream
<xfa:datasets xmlns:xfa="http://www.xfa.org/schema/xfa-data/1.0/">
... contents of datasets packet...
</xfa:datasets>
Endstream
```

```
13 0 obj
stream
<config xmlns="http://www.xfa.org/schema/xci/1.0/">
... contents of config node of XFA Data Package...
<config>
endstream
14 0 obj
stream
</xdp:xdp>
endstream
```

例 8.19

```
1 0 obj XFA entry in interactive form dictionary
<< /XFA 10 0 R >>
Endobj

10 0 obj
stream
<xdp:xdp xmlns:xdp="http://ns.adobe.com/xdp/">
<template xmlns="http://www.xfa.org/schema/xfa-template/2.1/">
... remaining contents of template packet...
</template>
<xfa:datasets xmlns:xfa="http://www.xfa.org/schema/xfa-data/1.0/">
... contents of datasets packet...
</xfa:datasets>
<config xmlns="http://www.xfa.org/schema/xci/1.0/">
... contents of config node of XFA Data Package...
<config>
</xdp:xdp>
endstream
Endobj
```

当一个 XFA 条目出现在一个交互格式字典中，这个 XFA 资源能提供大多数关于格式的信息；特别是，全部有关如统计和有效的格式信息。其它在交互格式的条目必须与 XFA 资源信息相符合。当用 XFA 资源创建或修改 PDF 文件时，应用程序应该按照如下指导：

- PDF 交互格式域对象必须出现所有在 XFA 资源指定的域。XFA 域值必须与 PDF 域对象对应的 V 条目相符合。
- XFA 脚本对象模型 (SOM) 指出一个名字转换，其必须用于连接交互格式名字和在 XFA 资源的域名。关于这个模块个信息在 XFA 规范，版本 2.2 (见参考) 是有效的。
- A 条目或 AA 条目 (见表 8.11) 不应该出现在域注释字典中，在域也有 XFA 资源指定操作时。两种未定义方法指定域的操作行为。

8.7、数字签名

数字签名 (PDF1.3) 能用于去认证用户身份和文档内容，当它被签署时，保存关于签名者和文档状态信息。签名可能是纯运算，如一个公/私加密文档文摘，也可能是一个生物识别格式，如手写签名、指纹或视网膜扫描。具体的签名格式通过相关签字处理程序生效。第三部分处理程序是作者鼓励使用 Adobe 签署它们处理名称：见附录 E。

签名信息被包括在签署字典，它的条目列在表 8.98。签字处理程序能使用或忽略那些条目，其被标记可在表中选择，但是如果它们被使用，鼓励用标准方法使用。另外，签字处理程序可能增加它们自己私有条目。为了避免名字重复，建议关键字对所有这样的私有变量用周期前缀 (.) 加上注册程序名。

签名是通过计算文档的数据摘要 (或部分数字) 创建，并且在文档中存储摘要，为了确认签名，摘要被重复计算，并和存储在文档的摘要作比较，不同于摘要中的值表示自从文档被签署后又已发生过修改。

有两个技术计算全部或部分 PDF 文件内容重复出现的摘要：

- 一个字节摘要通过一系列的文件字节计算，通过签字字典中的 ByteRange 条目来指示的。范围一般是全文件，包括除了签字值本身 (内容条目) 的签字字典。当一个字节范围摘要出现，所有在签字字典中的值要求都是直接对象。见附录 H 执行注释 138。
- 在内存中对象摘要 (PDF1.5) 被选择性地通过内存中对象子树计算。从参考对象开始，参考对象典型是根对象。结果摘要随同怎么被计算它的信息放置在签字参考字典中，它的条目项列在表 8.99。Transform Method 条目明确说明了用于计算摘要的一般手段，并且指定了计算的可见部分。转换方法在 8.71 节“转换方法”中详细描述。

PDF 文档可能包含下列标准签署类型：

- 一个或多个（或常见）签字。这些签字出现在签字格式域（见 658 页“签字域”）。对应每个签字格式的签字字典是格式域值（如其 V 条目指定）。签署字典必须包含一个 **ByteRange** 条目代表一个字节范围摘要，如上边描述。签字通过计算和比较它和存储在签署里的摘要验证。

注：如果一个签字文档被修改并通过增长更新保存（见 3.4.5 节，“增长更新”），对应于原始签名字节数据被保存。所以如果签字有效，可能会像它在签署退出一样，重新创建文档状态。

- 最多一个 MDP（修改检测和预防）签字（PDF1.5），也参考作者或认证签名，MDP 签字的签字字典必须是一个签字域值，并且必须包含一个 **ByteRange** 条目。它可能也被许可字典（见 8.7.3 节，“许可”）**DocMDP** 条目引用。签署字典必须包含一个参考字典（见表 8.99）的签字，参考字典有 **DocMDP** 转换方法。见 690 页“**DocMDP**”怎样创建和验证这些签字。
- MDP 或日常签字的签字字典可能也有一个签字参考字典，其有 **FieldMDP** 转换方法；见 695 页“**FieldMDP**”。
- 最多两个签字使用权（PDF1.5）。它的签字字典是参考许可字典（不是来自签字域）中的 UR 或 UR3（PDF 1.6）条目；见表 8.103。字典必须包含签字参考字典，其有 UR 转换方法。见 692 页“**UR**”，获取有关这些签字怎样被创建和验证的信息。
- 在 PDF 文件目录中的 **Sig** 条目（见 672 页 FDF 目录）指定签字字典。

表 8.98 签字字典条目项

KEY	TYPE	VALUE
Type	name	(可选) 此字典描述的 PDF 对象类型；如果出现，签字字典必须是 Sig 。
Filter	name	(必需的；可继承) 签字名处理程序首选使用名字，当验证签字时。如果 Prop_Build 条目不出现，它也是签字处理程序的名字，签字处理程序用于创建签字，如果 Prop_Build 存在，它能用于决定创建签名

(其通常类同于 Filter, 但是不被要求) 的名字处理程序。确定签名时, 应用程序可能替代不同处理程序。只要它支持指定的 SubFilter 的格式。例如签字处理程序是 Adobe.PPKLite, Entrust.PPKEF, CICI.SignIt, 和 VeriSign.PPKVS。

SubFilter	name	(可选) 一个名字描述签字字典中签字编码值和关键信息。一个应用程序可能使用任何支持这种格式验证签名的处理程序。 对公钥加密签名的定义值是 adbe.x509.rsa_sha1, adbe.pkcs7.detached, and adbe.pkcs7.sha1 (见 8.7.2 节, “签字互通性”)。
Contents	string	(必需的) 签字值, 当 ByteRange 出项, 值是一个 16 进制字符串 (见 32 也“16 进制字符串”), 它代表字节范围摘要值。如果 ByteRange 不出现。除了 Contents 条目, ByteRange 值是一个签字字典对象摘要。 对公钥签字, Contents 通常要么是一 DER-encoded PKCS#1 二进制数据对象, 要么是 iGeDER-encoded PKCS#7 二进制数据对象。
Cert	Array or string	(当 SubFilter 是 adbe.x509.rsa_sha1 时必需; 否则不使用) 字符串代表使用 X.509 认证连, 如果连只有一个条目, 签名和认证签名使用公钥密码或字符串。签字认证必须首先出现在序列; 它用于确认在 Contents 的签字值, 并且其他认证不用于确认签字认证的真实性。 如果 ubFilter 是 adbe.pkcs7.detached 或 adbe.pkcs7.sha1, 这个条目不使用, 并且认证连必须放入在 Contents 的 PKCS#7 包里。
ByteRange	array	(对所有参考许可字典 UR3 条目签字使用权和签字域部分的所有签字都是必需的) 一个整数对序列 (开始字节偏移, 字节长度) 精确描述计算的字节范围。多

个不连续字节用于描述一个数字，这个数字不包括签字值（Contents 条目）本身。

Reference array (可选；PDF1.5) 签字参考字典序列（见表 8.99）。

Changes array (可选) 指示文档变化的三个整数序列，此文档已经在前次签名和这次签名完成：按此顺序，页号码改变，域号码改变，并且域数字填入。（见附录 H 执行注释 138）

注：签字顺序是取决于 ByteRange 值。由于每个签字导致一个增加保存，后面签字有更长的值。

Name text (可选) 签署文档的人或权利名。当它不可能从签字中抽取出名字时，此值应被使用；比如，从签署者中认证。

M date (可选) 签字时间，依靠签字处理程序，这可能是一个正常的未经证实的计算机时间，也可能是一个使用可确定方法从一个安全时间服务器产生时间。
这个值应该仅在签字时间无效时使用；比如。时间戳能被嵌入 KCS#7 二进制数据对象（见 697 页“PKCS#7 签字”）。

Location text (可选) CPU 主机名或签字物理位置。

Reason text (可选) 签字原因，如（我同意…）。

ContactInfo text (可选) 签署者提供一个信息，使接受者能去联系签署者确定签署；比如，电话号码。

R integer (可选) 用于创建签字过程中的签字处理程序版本。
注：PDF1.5 开始，这个条目不被推崇，且信息应该存储在 Prop_Build 字典。

V integer (可选；PDF1.5) 签字字典格式版本。它对应于 SubFilter 值内容的签字字典用法。值是 1，如果 Reference 字典被认为是关键的签字验证。

默认值：0

Prop_Build dictionary (可选；PDF1.5) 能被签字处理程序使用记录信息的字典，信息是捕获的签字计算机环境状态，软件组建

日期, 版本和操作系统。

Adobe 发布一个独立规范, PDF 签字为 Acrobat 6.0 构建字典规范, 对字典使用提供执行引导

Prop_AuthTime integer (可选; PDF1.5) 自从签字者被最后认证的秒数。它旨在签字拒绝声明中使用。如果这个值未知, 应该被忽略。

Prop_AuthType name (可选; PDF1.5) 签字者使用认证方法。它旨在签字拒绝声明中使用。有效值包括 PIN, Password 和 Fingerprint。

注: 签字字典中的条目能被概念为不同的字典; 由于历史和加密原因它们在一个字典中。类别是签字属性 (R, M, Name, Reason, Location, Prop_Build, Prop_AuthTime 和 Prop_AuthType); 关键信息 (Cert 和 Contents 部分, 当签字值是 PKCS#7 对象); 参考 (Reference 和 ByteRange); 和签字值 (Contents 当签字值是一个 PKCS#1 对象)。

表 8.99 签字参考字典条目项

KEY	TYPE	VALUE
Type	name	(可选) 字典描述的 PDF 对象类型; 如果出现, 对签字参考字典必须是 SigRef。
TransformMethod	name	(必需的) 转换方法名字 (见 8.71 节, “转换方法”), 当签字有效, 转换方法指导对象摘要计算或修改分析。有效值是: DocMDP 用于检测文档相关签字域修改, 域被文档原有者签署; 见 690 页 “DocMDP”。 UR 用于检测文档修改, 其使有文档使用权的的签名无效; 见 692 页 “UR” FieldMDP 用于检测指定在 TransformParams 格式域表的修改; 见 695 页 “FieldMDP”。 Identity 使用签定单个对象时, 通过在签署参考字典 (见表 8.99) 中 Data 值指定。此转换方法支持 FDF 文件签署。详细见 696 页 “身份”。

TransformParams	dictionary	(可选) 字典指定由 TransformMethod 指定转换方法的参数 (可变数据)。除了 Identity 每个方法都使用它们自己的参数集。详细见上边指定节关于个别转换参数字典。
Data	(various)	(必需的, 当 TransformMethod 是 FieldMDP 或 Identity) 通过对文档中对象的间接参考, 摘要被计算, 或对象修改分析应该被执行。对转换方法而不是 FieldMDP 和 Identity, 此对象被隐式定义。
DigestMethod	name	(可选) 当计算摘要时, 定义将使用的算法名, 有效值是 MD5 和 SHA1。默认值: MD5.
DigestValue	string	(必需的, 在一些情况) 当存在, 计算摘要值。见 8.7.1 节, “转换方法”有关此条目需要时的细节。
DigestLocation	array	(必需的, 当 DigestValue 被要求且 TransformMethod 是 FieldMDP 或 DocMDP 时) 两个整数的序列指定 PDF 文件 DigestValue 字符串位置。整数分别代表偏移值和长度, 以字节单位的

8.7.1、转换方法

转换方法, 随同转换参数, 决定在对象摘要计算或调整比对中, 那些对象被包括, 那些排除。下面章节讨论它们使用的转换方法类型、转换参数。附录 I, “对象摘要计算,” 详细描述计算对象摘要算法。

注: 所以转换方法从签字字典排除对象摘要。

DocMDP

DocMDP 转换方法用于检测对文档作者 (应用第一个签字的人) 签字域的修改。文档仅能有一个包含 DocMDP 转换方法的签字域; 它必须是文档中的第一个签署域。它让作者指定文档允许应用什么改变, 且怎样使作者签字无效。

如前面的讨论, “MDP” 代表修改检测和预防。这签字使能作者指定的不允许改变

的检测，另外，不允许更改也能被阻止，当签字字典被许可字典中 DocMDP 条目引用（见 8.7.3 节。“许可”）。

注：当创建一个作者签名，应用程序鼓励去创建一个合法认证字典（见 8.7.4 节，“合法内容认证”），字典能够指定所有可能导致意想不到文档翻译的内容，还有作者对这样内容的认证。这字典能用于建立作者意图，如果文档完整性被质疑。

在 DocMDP 转换参数字典（见表 8.100）的 P 条目指定作者规范，哪些对文档的改变将是无效签名，（这些对文档的改变将阻止，如果签署字典是引用在许可字典中 DocMDP 条目）。P 的值为 1 指示文档将会是最终的；也就是任何改变使签名无效，值为 2 和 3，许可修改，修改与格式域或评论工作流程相适应。

DocMDP 对象摘要经过一个在文档中的 PDF 对象子集被计算。特别是这个子集包含对象是不允许被直接或间接修改，由于被转换参数字典指定。附录 I 描述对象摘要计算。

验证 MDP 签名

为了验证一个 MDP 签名，一个应用程序首先确定字节范围摘要。接着，确定任何已经做修改的文档是转换参数允许的，确认使用下面一种技术：

- PDF1.5 要求在签署要被存储在签字参考字典（见表 8.99）DigestValue 条目时，计算对象摘要值，所以要当验证时，应用程序能比较这条目和它的计算对象摘要。如果值是不同，签字是无效。
- 在 PDF1.6，DigestValue 不被要求。一旦字节范围摘要有效，文档的部分被签字字典（见表 8.98）ByteRange 条目指定，在签字时，相应的文档状态是已知的。所以应用程序能比对当前和已签字的文档，查看是否已有任一对象修改，修改是不被转换参数允许的。

见附录 H 执行注释 140

表 8.100 DocMDP 转换参数字典条目项

KEY	TYPE	VALUE
Type	name	（可选）字典描述的 PDF 对象类型；如果出现，对转换参数字典必须为 TransformParams。
P	number	（可选）访问许可授予这个文档。有效值是： 1 没有改变对文档是允许的，任何对文档的改变废止签字。

2 许可改变被填到格式，实例化页模板，签字；其它改变废止签字。

3 许可改变相同 2，同时注释产生，删除和修改；其它改变废止签字

默认值：2

V	name	(可选) DocMDP 转换参数字典版本。PDF 值是 1.5 和早期是 1.2。(注意到这个值是名字对象，不是数值。) 默认值：
		1.2

UR

UR 转换方法用于检测对文档的改变，其将会废止使用权签字（见 8.7.3 节，“许可”）。使用权签字用于使能其它交互特征，其在具体浏览器应用程序默认是无效的（如 Adobe Reader）。签字用于验证许可已经被真正授权，转换参数字典（见表 8.101）指定其它权利，如果签字是有效的其被使能，如果签字由于文档已经被修改被废止，也就是不允许或签署者身份不被授权执行许可，其它权利也不被授予。

比如，Adobe 系统授权可能使 Adobe 阅读器其它特征给，使用公钥密码，它使用认证权去确定公共钥匙认证给文档创建者，有了这就能进入商业关系，Adobe 阅读器确认使能签字权利，使用来自 Adobe 授权权威的认证，其它 PDF 浏览器应用程序自由为它们自己目的使用同样机制。使用权签字认证依赖签字字典是否是引用许可条目的 UR 或 UR3 条目（见附录 H 执行注释 141）：

- UR：签字时，通过一组在文档的 PDF 对象子集，应用程序计算对象摘要；也就是，对象不被直接或间接修改，通过许可操作，就像被转换参数字典指定。附录 I 描述对象摘要计算。这个摘要的计算值被存储在签字参考字典（见表 8.99）的 DigestValue 条目。应用程序在验证时能比较这个条目和它的计算对象摘要。如果值不同，签字无效。
- UR3 (PDF1.6)：在签字字典（见表 8.98）中的 ByteRange 条目要求出现，首先，应用程序确定字节范围摘要，在签署时，决定是否 ByteRange 指定的文档部分对应于文档状态，接下来，应用程序检测当前文档版本，去看是否任何对象的修改是不被转换参数允许的。

表 8.101 UR 转换参数字典字典条目项

KEY	TYPE	VALUE
-----	------	-------

Type	name	(可选) 字典描述的 PDF 对象类型; 如存在, 对转换参数字典必须是 <code>TransformParams</code>
Document	array	(可选) 文档指定其它文档范围使用权利的序列名字, <code>FullSave</code> 是唯一定义值, 它允许用户连同修改格式和/或注释数据保存文档。(见附录 H 执行注释 142)
Msg	string	(可选) 能用于指定任何信息的字符串, 就像是加使用权到文档的理由。
V	name	(可选) UR 转换参数字典版本。对 PDF1.5 的值是 2.2, (主要这个值是一个名字对象, 不是一个数字) 如一个未知版本出现, 无权使能。默认值: 2.2
Annots	array	(可选) 指定对文档其它注释相关使用权利序列名字。在 PDF1.5 和后续版本的有效名字是 <code>Create</code> , <code>Delete</code> , <code>Modify</code> , <code>Copy</code> , <code>Import</code> , 和 <code>Export</code> , 这允许用户区执行一个有名操作到注释。 下列增加在 PDF1.6 名字。仅当签字字典(见表 8.103)参考签署字典 UR3 条目时, 它们被允许; 见表 8.103 (见下 <code>FormEx</code>) <code>BarcodePlaintext</code> 许可文本格式域数据像一个平面文本二维条码被编码 <code>Online</code> 允许在线评论; 也就是从服务器上传和下载标记注释。 <code>SummaryView</code> 允许用户交互显示总结文档中的评论 (标记注释)
Form	array	(可选) 一个序列名指定其它格式域相关文档使用权, 有效名字在 PDF1.5 是: <code>FillIn</code> 允许用户保存一个文档, 文档上格式填充已被完成。 <code>Import</code> 允许用户导入格式数据文件以 FDF, XFDF 和文本 (CSV/TSV) 格式。 <code>Export</code> 允许用户导出格式数据文件像 FDF 或

XFDF.

SubmitStandalone 允许用户提交格式数据，当文档没有在网路浏览器打开。

SpawnTemplate 允许新页被实例化从有名页模板。下列名字被增加在 PDF1.6。它们被允许只有当签字字典被引用从许可字典中的 UR3 条目；见表 8.103（然而，见下 FormEx）

BarcodePlaintext 允许文本格式域数据被编码像一个明文两维条码。

Online (PDF1.6) 允许使用格式指定在线机制就像 SOAP 或 Active Data Object

FormEx array (可选；仅当签字字典参考许可字典 UR 条目时允许；PDF1.5) 一个指定其它格式域相关使用权的序列名字。有效名仅是 BarcodePlaintext，它允许文本格式域数据被编码像一个明文两维条码。

Signature array (可选) 一个指定其它签字相关对文档使用权的序列名。仅有的定义值是 Modify，这允许用户应用一个数字签名到一个现存签名格式域或清除有符号签名格式域。

EF array (可选；PDF 1.6) 对文档的有名嵌入式文件指定其它相关签名使用权的序列名。有效名是 Create, Delete, Modify, 和 Import，它们允许用户对有名嵌入文件执行有名操作。

P boolean (可选；PDF 1.6) 如为 true，应该限制所有消费应用程序的被 Adobe Reader 授予文档许可，当允许字典中其它条目使能权利许可。默认值：false。

FieldMDP

FieldMDP 转换方法通过一列格式域对象计算一个对象摘要，并且检测那些格式域值改变，在表 8.102 列出了参数字典其它转换参数条目

表 8.102 FieldMDP 转换参数字典条目

KEY	TYPE	VALUE
Type	name	(可选) 此字典描述 PDF 对象类型; 如存在, 对转换参数字典, 必须是 TransformParams。
Action	name	(必需的) 伴随 Fields 序列, 名字描述对象摘要包括 (签署应用后, 不在允许改变) 的格式域。有效值是: All 所有格式域 Include 只是那些指定域的格式域 Exclude 只是那些没有被指定在 Field 的 s 格式域。
Fields	array	(必需的, 如果操作是 Include 或 Exclude) 包含域名的一个字符串序列。
V	name	(可选) 转换参数字典版本。对 PDF1.5 和 1.2。(注意这个值是名字对象, 不是数字。) 默认值: 1.2

当文档用于格式域工作流程, 下列发生:

- 作者指定格式域能被填充不需验证作者签字, DocMDP 转换参数字典 P 条目被设置为 2 或 3 (见表 8.100)。
- 作者也能指定在一个特定接受者已经签字这文档后, 任何对指定格式域的修改应该使接受者的签名无效。每个指定接受者都有一个独立签字域, 每个有相关签字域锁字典 (见表 8.78) 指定的格式域应该对用户是上锁的。
- 当接受者签了域, 签字参考、签字和转换参数字典被创建, 在转换参数字典的 Action 和 Fields 条目从对应域在签字域锁字典被复制。
- 注: 做复制因为所有在签字字典的对象必须是直接对象, 如果字典包含一个字节范围签字。(即使 FieldMDP 签署时对象签署, 任何引用签字域的签字字典也必须有一字节签字范围)。所有, 转换参数字典不能间接地参考签字域锁字典。

对象摘要通过所有转换参数字典指定的格式域计算, 以阿拉伯顺序排列 (详细见附录 I)。指定格式域通过标记它们为只读上锁阻止改变。当授权者签字确定, 所有对格式域的改变都能被发现。

FieldMDP 签字域通过类 DocMDP 签字的方法生效。详细见 691 页 “有效 MDP 签字”

Identity

当计算的对象摘要非常广泛时，`Identity` 转换方法被使用；也就是，没有对象被排除。整个对象树在进行，从在签字参考字典（见表 8.99）`Date` 指定对象开始。任何对象内容的改变废止签字，这方法用于支持 FDF 文件签收。通过 FDF 目录对象计算摘要。

8.7.2、签署互用性

目的是使 PDF 消费应用程序允许签字处理程序的互用性；就是一个带有从供应商处理程序的 PDF 文件签字必须能够用一个来自不同供应商的处理程序验证。

签字字典的 `SubFilter` 项指定签字值和关键信息编码，`Filter` 条目指定首选检验签字使用的处理程序，处理程序指定它们支持 `SubFilter` 编码；所以不仅首选处理程序其它处理程序也能用于验证签字，如果必须或要求。

有几个 `SubFilter` 项已定义值，所以基于公钥加密标准发布被 RSA 安全和部分标准引用的是互联网工程任务组（IETE）和公钥基础（PKIX）工作组；见应用参考。

PKCS#1 签字

PKCS#1 标准支持几个公钥加密算法和摘要方法，包括 RSA 加密，DSA 签名，和 SHA-1 and MD5 摘要（见应用参考）。对签署 PDF 文件使用 PKCS#1，唯一 `SubFilter` 的推崇值是 `adbe.x509.rsa_sha1`，其使用 RSA 加密算法和 SHA-1 再要方法。签字者认证连存储在 `Cert` 条目。

PKCS#7 签字

当 PKCS#7 签名使用，`Contents` 值是 DER-encoded PKCS#7 二进制数字对象包含签字。
`SubFilter` 能采用下面一种值：

`adbe.pkcs7.detached`: 没有数据包括在 PKCS#7 符合数据域。

`adbe.pkcs7.sha1`: 字节范围 SHA1 摘要被包含在 PKCS#7 符合数据域和类型数据信息。

PKCS#7 对象必须符合网络 RFC2315 PKCS#7 规范；加密信息句法，版本 1.5（见参考）。至少它必须包括签名者的 X.509 签字认证。这个认证被用于确定 `Contents` 的签署值。PKCS#7 对象可能选择性地从签字者的信任链包含一个或多个引用认证；见附录 H 执行注释 143。

PKCS#7 对象可能选择地包含下列属性：

- 时间戳信息作为一个无签署属性：时间戳使用必须遵从 RFC3161 且必须像在附录 A 的

RFC3161 描述去计算和嵌入到 PKCS#7 对象。

- 废止信息是一个签署属性：这属性能包含所有废止信息，其是必须区执行废止检查签署者的认证和它的发行认证。

下列对象标识符标识 Adobe 的废止信息属性：

```
adbe-revocationInfoArchival OBJECT IDENTIFIER ::=  
{ adbe(1.2.840.113583) acrobat(1) security(1) 8 }
```

废止信息属性值能包括任何下列数据类型：

- 认证废止列表 (CRLs)，描述在 RFC3280 (见参考)：CRLs 通常大，所以不推崇到嵌入式 PKCS#7 对象。
- 在线认证状态协议 (OCSP) 响应，在 RFC2560，X.509 网络公钥基础设施在线证书状态协议---OCSP(见参考) 描述：通常小并且尺寸固定的被括进 PKCS#7 对象的建议数据类型。
- 消费废止信息：格式不是使用规范规定，而是像一个 OCTET STRING 被编码。应用程序应该决定数据类型，其包通过查看相关 OBJECT IDENTIFIER 包含在 OCTET STRING 。

Adobe 的废止信息属性值有 ASN.1 类型 RevocationInfoArchival：

```
RevocationInfoArchival ::= SEQUENCE {  
    crl          [0] EXPLICIT SEQUENCE of CRLs, OPTIONAL  
    ocsp         [1] EXPLICIT SEQUENCE of OCSP Responses, OPTIONAL  
    otherRevInfo  [2] EXPLICIT SEQUENCE of OtherRevInfo, OPTIONAL  
}  
  
OtherRevInfo ::= SEQUENCE {  
    Type   OBJECT IDENTIFIER  
    Value  OCTET STRING  
}
```

对字节范围签字，Contents 是一个 16 进制字符串由 “<” 和 “>” 分隔。它必须在空间上精确在 ByteRange 指定范围。由于 PKCS#7 对象长度是不被完全预测的，在写 PKCS#7 到文件分配空间时，它通常需要在字符串尾部（在 “>” 分隔符前）用 0 给 Contents 垫值。

PKCS#7 对象是一个签字信息包装者，也就是当使用 PKCS#1 能被发现在签字字典中。

PKCS#7 签字时尺寸可预测且紧凑，131 字节对 1024 位 RSA 签字，在十

六进制编码前。

迅速转化 PKCS#1 签名值为 PKCS#7 签名值是有可能的，但是反向转化是不可能的，原因是 PKCS#1 签字能被处理程序验证通过，并不直接支持 PKCS#1，但是 PKCS#7 签字不能被仅支持 PKCS#1 的处理程序验证使用。当附加 PKCS#7 能力不需要时，PKCS#1 签名推广到所用情况。

8.7.3、权限

在文档目录（见表 3.25）Perms 条目指定权限字典（PDF1.5）。每个在字典（见表 8.103 当前定义条目）的条目指定权限处理程序的名字，扩展处理程序访问文档权限。这些权限类似于那些安全处理程序定义权限（见 99 页表 3.20）但不要求文档被加密。一个权限（比如，填充格式域能力）要被一个文档实际授权，它必须被每个权限处理程序允许。它出现在权限字典如安全处理程序。

表 8.103 权限字典条目项

KEY	TYPE	VALUE
DocMDP	dictionary	<p>(可选) 直接引用到一个签署字典（见表 8.98）。这个字典必须包含 Reference 项，其是一个签署引用字典（见表 8.99）其有一个 DocMDP 转换方法（见 690 页“DocMDP”）和对应转换参数。如果此条目存在，消费应用程序应该执行在 DocMDP 转换参数字典 P 属性指定的权限并且应该验证对应签字基于是否这些权限的任意已经被违反。</p>
UR	dictionary	<p>(可选) 一个签署字典是被用于指定和验证授予此文档的其它能力（使用权限）；也就是，浏览器应用程序交互特征使能默认是不可用的。比如，Adobe Reade 不允许使用默认保存文档，但是 Adobe 系统可能授权许可其使能保存特定文档在 Adobe Reade。签署被用于验证权限已经被 Adobe 系统授予。签署字典必须包含一个 Reference 条目其是一个签署引用字典有 UR 转换方法（见 692 页“UR”）。</p>

对这种方法的转换格式参数指示那个附加权限应该授权于文档。如果签字有效，Adobe Reader 允许为文档指定许可，除了应用程序默认许可。
签字字典必不能包含 ByteRange 条目。

UR3	dictionary	(可选；PDF1.6) 签字字典指定和验证使用权利。在 UR 条目上描述应用于 UR3，除了签字字典必须包含 ByteRange 条目。详细见 692 页“UR”。
-----	------------	--

8.7.4、合法内容认证

PDF 语言提供了许多能呈现 PDF 文档变量外形的功能。这些功能能被用来构建一个文档，当考虑一个签署 PDF 文档合法影响时，有意或无意误导文档接受者，这些情况是相关的。

所以，必须有一个机制，通过它文档接受者能决定文档是否能被信赖。主要方法是去接受唯一其包含作者签名的文档（一个有 DocMDP 签名定义什么被允许区改变在文档：见 690 页“DocMDP”）。

当创建作者签名，应用程序应该也创建一个合法认证字典。它的条目项显示在表 8.104。这个字典是文档目录（见表 3.25）Legal 条目值。它的条目指定所有内容可能提供进一步的此内容的澄清通过 Attestation 条目。重审应该为自身建立它们相信的作者和文档内容。在对文档合法挑战情况，任何疑问内容能审查在这字典的信息内容

表 8.104 合法认证字典条目

KEY	TYPE	VALUE
JavaScriptActions	integer	(可选) 文档中 JavaScript 操作号（见 668 页“JavaScript 操作”）
LaunchActions	integer	(可选) 文档中 launch 操作号（见 621 页“launch 操作”）
URIActions	integer	(可选) 文档中 URI 操作号（见 624 页“URI 操作”）
MovieActions	integer	(可选) 文档中 movie 操作号（见 626 页“movie 操作”）
SoundActions	integer	(可选) 文档中 sound 操作号（见 625 页“sound 操作”）
HideAnnotationActions	integer	(可选) 文档中 hide 操作号（见 627 页“hide 操作”）

GoToRemote	integer	(可选) 文档中 remote go-to 操作号 (见 617 页“remote go-to 操作”)
AlternateImages	integer	(可选) 文档中交互印象号 (见 317 页“Alternate Images”)
ExternalStreams	integer	(可选) 文档中外部流号。
TrueTypeFonts	integer	(可选) 文档中实体类型字体号 (见 384 页“TrueType Fonts”)
ExternalRefXobjects	integer	(可选) 文档中参考 Xobjects 号 (见 331 页“Reference XObjects”)
ExternalOPdicts	integer	(可选) 文档中 OPI 字典号 (见 907 页“Open Prepress Interface (OPI) ”)
NonEmbeddedFonts	integer	(可选) 文档中非嵌入式字体号 (见 5.8 节“Embedded Font Programs”)
DevDepGS_OP	integer	(可选) 文档中对图像状态参数 OP 的参考号 (见表 4.8)
DevDepGS_HT	integer	(可选) 文档中对图像状态参数 HT 的参考号 (见表 4.8)
DevDepGS_TR	integer	(可选) 文档中对图像状态参数 TR 的参考号 (见表 4.8)
DevDepGS_UCR	integer	(可选) 文档中对图像状态参数 UCR 的参考号 (见表 4.8)
DevDepGS_BG	integer	(可选) 文档中对图像状态参数 BG 的参考号 (见表 4.8)
DevDepGS_FL	integer	(可选) 文档中对图像状态参数 FL 的参考号 (见表 4.8)
Annotations	integer	(可选) true 如果选择内容在文档中 (见 8.4 节“Annotations”)
OptionalContent	boolean	(可选) 认证, 文档作者创建, 解释任何其它在字典的条目存在或任何内容影响文档合法完整性存在的字典条目。(见 4.10 节“Optional Content”)
Attestation	string	(可选)

8.8、测量性能

如那些被 CAD 软件创建、可能包含图表的 PDF 文档，打算代表现实对象。用户经常需要关于此文档中关于相应真实世界对象的大小、测量单位和在 PDF 用户空间它们关系单位的信息。

这信息使能浏览器应用程序施行测量，测量结果使用文档创建者的预期单位表示，这个内容中的测量需要输入一组 n 维坐标对，典型函数的结果

$$\{(x_0, y_0), \dots, (x_{n-1}, y_{n-1})\}$$

并产生单个数作为输出依赖测量类型。比如，距离测量等于

$$\sum_{i=0}^{n-2} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}$$

对 $n >= 2$ 。

PDF1.6 开始，这些信息可能被保存在测量字典（见表 8.106）。测量字典提供关于测量单位的信息，单位关联一个已知文档矩形区域作为视口。

视口（PDF1.6）是一个页矩形区域。可选 VP 项在页字典（见表 3.27）指定视口字典指定一个视口字典序列，它的条目显示在表 8.105。如果需要，视口允许不同测量比例（由 Measure 条目指定）或使用在不同页区域。

VP 序列的字典指定顺序。由于视口可能重叠，决定视口在页上任何点使用，从最后一个开始迭代反向检查序列中字典，且它的第一个 Bbox 条目包含选择点。

注：任何测量都可能包含有多个视口，就像在两个点指定距离应该使用指定在视口第一点的信息。

表 8.105 视口字典条目项

KEY	TYPE	VALUE
Type	name	(可选) 这字典描述的 PDF 对象类型；必须是 Viewport 对视口字典。
BBox	rectangle	(必需的) 在用户空间的矩形坐标指定页上视口位置。 两个矩形的坐标对必须被指定以统一格式；也就

是，左下角跟随右下，相对坐标系统测量。这顺序决定测量坐标系统在这个视口的方向（也就是，x 轴和 y 轴正向延伸方向），这可能有不同转向对这页。

注：这个矩形坐标是独立于测量坐标系统原定，指定由 **Measure** 指定的测量字典 0 条目（见表 8.107）

Name	text	(可选) 一个描述文本字符串或视口标题，在用户交互使用。
Measure	dictionary	(可选) 测量字典（见表 8.106）指定尺寸和单位，其应应用于在视口内容测量。

测量字典指定一个备用坐标系统为页区域，随同视口字典，它提供转化页内坐标系统坐标到测量坐标系统坐标所需要的信息。测量字典提供信息对格式结果值到文本格式为显示在图形用户接口。

表 8.106 显示了测量字典条目。PDF1.6 只是定义一个类型坐标系统，它被定义为一个 x 轴和 y 轴是垂直的并有线性增加的单位（分别对右和上）。其它子类型被许可，提供弹性测量使用其它坐标系统。

表 8.106 测量字典字典条目项

KEY	TYPE	VALUE
Type	name	(可选) 这字典描述 PDF 对象类型；对测量字典必须是 Measure 。
Subtype	name	(可选) 指定的一个用于测量的坐标系统类型名。 默认值：RL，它指定直线坐标系统

表 8.107 显示了条目中的其它直线测量字典。许多在这个字典的条目是数字格式序列（见表 8.108）。每个数字格式字典代表一个特殊的测量单位（如英里或尺）。它包含关于每个在文本表示单位和计算单位数字信息的因素。

数字格式序列指定所有一个专门测量要使用的单位，每个序列包含一个或多个数字单位字典，以粒度减小为顺序（比如，一个格式字典数字指定英尺应该先于一个指定英寸。）所有在序列的元素包含文本字符串串联在一起，指定单位应该怎样被显示。比如，一个 1.4505

英里可能被表示为“1.45005mi”，其要求对 miles 有一个数字格式字典，或像“1 mi 2,378 ft 7 5/8 in”，这将要求三个字典（为 miles, feet 和 inches）。

表 8.107 直线测量字典条目项

KEY	TYPE	VALUE
R	text	(必需的)一个文本字符串表示绘制比例在对应词字典区域，一般认为单位缩写应该被使用，要么匹配那些在这个域的那些数字格式序列要么那些通用使用比例。比如，在构建绘画的一个通用尺寸是“1/4 in = 1 ft”，默认表示 1/4 inches 用户空间等于 1 英尺在现实世界测量。 如果比例尺寸在 x 和 y 向不同，两个尺寸都应该被指定；比如，“in X 1 cm = 1 m, in Y 1 cm = 30 m”。
X	array	(必需的)一个测量沿 x 轴改变数字格式序列，如果 Y 不存在，y 轴同理。这个序列第一个元素包含比例尺寸，比例尺寸转换从默认用户空间单位到沿着轴测量坐标系统大单位。 X 和 Y 轴方向在测量坐标系统，且独立于页自转，这些方向通过包含视口（见表 8.105）Bbox 条目决定。
Y	array	(必需的，当 x 和 y 尺寸有不同单位或转换因素)一个数字格式序列测量沿 y 轴的改变，这个序列第一个元素包含比例尺寸，比例尺寸是转换从默认用户空间单位到沿着 y 轴测量坐标系统大单位。
D	array	(必需的)一个数字格式序列为任方向的距离测量，序列第一个元素指定到大距离单位转换从通过 x 轴第一个元素代格式位，尺寸因素从 X, Y (如果出现) 和 CYX (如果 Y 存在) 被用于转换从默认用户空间到适当单位在应用距离功能前。
A	array	(必需的)一个数字格式序列测量区域，序列第一个元素指定转换到大区单位从单位表示用 X 轴第一个元素。尺寸因素从 X, Y (如果出现) 和 CYX (如果 Y 存在) 被用于转换从默认用户空间到适当单位在应用区域功能前。

T	array	(可选) 一个数字格式序列测量角度。第一个在序列的元素指定转换到大角度单位度。尺寸因素从 CYX (如果存在) 被用于转换从默认用户空间到适当单位在应用角功能前。
S	array	(可选) 数字格式化直线斜率测量序列。序列的第一个元素指定最大坡度单位转换，通过第一个元素在 Y 除以第一个元素 X 来表示。在应用斜率函数之前，这些尺寸因素 X, Y (如果出现) 和 CYX (如果 Y 存在) 用于从默认用户空间到适当单位的转换。
O	array	(可选) 两个数字序列指定默认用户空间测量坐标系统原点。x 和 y 方向增长值从这个原点由视口 BBox 条目决定 (见表 8.105)。 默认值：视口 BBox 条目指定矩形的第一坐标对 (左下角)
CYX	number	(可选；只有当 Y 出现才有意义) 一个因素转化最大单位沿着 y 轴到最大单位沿着 x 轴，要求在一些计算下 (距离，面积和角度) 这里单位必须相等；如果被指定，这些计算不能执行 (其在如 x 代表时间，y 代表温度的情况下)。其它计算 (改变 x, y 和坡度) 不要求这些值。

X 和 Y 条目在测量字典中被数字格式化，分别地指定单位使用于测量 X 和 Y 方向，比率在用户空间单位和具体单位。Y 出现只有当 x 和 y 测量是在不同的单位或有不同比例；这种情况，适当时，CYX 条目被用于转换 y 值为 x 值。

表 8.108 数字格式字典条目项

KEY	TYPE	VALUE
Type	name	(可选) 此字典描述的 PDF 对象类型；对数字格式字典必须为 NumberFormat。
U	text	(必需的) 文本字符串指定一个标签显示单位表示这字典在用户接口；它被推崇在标签使用一个统一组织的缩写。
C	number	(必需的) 转换因素用于倍值用一个值在局部单位，其先前数字格式元素包含一个以这字典单位。当条目在序列中第一个数字格式字

典，它的意义（也就是，它要倍乘的）依赖在直线测量字典中的那个条目参考数字格式序列。

F	name	(可选；有意义只有对数字格式序列的最后字典) 名字指示是否和用什么方式显示部分值从转换结果到通过 C 条目的字典单位，有效真是：
D		D D 条目指定显示十进制精度
F		F D 条目指定显示有分母的小数
R		R 没有小数部分；元整到最近值。
T		T 没有小数部分；截断整个单位的小数
		默认值：D
D	integer	(可选；只有对数字格式序列的最后字典才有意义) 正整数指定精度或分母的分数：
		当 F 的值是 D，这个条目代表一个十进制显示精度；它必须为 10 的倍数。低位 0 可能被截断除非 FD 是 true。
		默认值：100 (百分号，对应的两个十进制小数位)。
		当 F 值是 f。这条目代表小数的分母显示。小数可能被缩减除非 FD 的值是 true。默认值：16
FD	boolean	(可选：只有对数字格式序列最后字典时才有意义) 如果 true，分数值根据 D 条目格式化可能没有它的分母减少或低阶零截断。
		默认值：false
RT	text	(可选) 文本被用在数字值显示的几千个顺序之间。空字符串指示没有文本被增加。
		默认值：逗号 (", ")，美国的公约。
RD	text	(可选) 文本被用于十进制点显示数值，空字符串指示默认值应被使用。
		默认值：逗号 (", ")，美国的公约。
PS	text	(可选) 文本被串联到用 U 指定的标签左边，空字符串指示没文本应被加。
		默认值：单个空字符串 ("")。
SS	text	(可选) 文本要被串联到 U 指定标签后边。空字符串指定没有文本应被增加。

默认值：单个空字符串（“ ”）。

0 name (可选) 一个名字指定被 U 指定的标签顺序去计算单位值。有效值是：

S 标签是值的后缀

P 标签是值的前缀

注：被 PS 和 SS 指定的字符被串联在考虑这个条目前。

默认值：S

使用一个数字格式序列创建一个文本字符串包含适当格式单位去显示在用户接口，应用算法 8.2；

算法 8.2

在直线测量条目中（见表 8.107）其参考数字格式序列决定初始测量值，比如 X 条目指定用户空间单位，且 T 条目指定度。

倍乘值指定第一个在序列的数字格式字典的 C 条目之上，其转化测量到在序列指定的最大粒度单位，适当时应用 RT 值。

如果结果没有包含非零小数部分，串联在 U 条目指定的标签按照 O 指定顺序的。在从 PS 和 SS 加上空间，格式然后被完成。

如果这儿是非零小数部分并且没有多元素在序列，格式化小数部分像被后字典中 RD, F, FD 条目指定。串联被 U 条目指定标签按照 O 指定的顺序。在加空间从 PS 和 SS 之后，格式被完成。

如果有非零小数部分和许多元素在序列，进行到序列的下一数字格式字典，倍乘它的 C 条目通过部分产生在前一阶段，适当地应用 RT 值，然后进行到阶段 3。

注：在这个过程的元素的串接假定是从左到右的顺序。文档使用从右到左的语言能修改过程和条目意义适当的生产正确的结果。

例 8.20

显示了一个措施字典其指定在 x 或 y 的改变以英里表达：距离被以英里，英尺，英寸表示，面积以亩表示，给出一个例子距离标度是 1.4505miles。格式文本产生通过应用数字格式序列为：

“1 mi 2,378 ft 7 5/8 in” .

例 8.20

```
<</Type /Measure  
/Subtype /RL  
/R (1in = 0.1 mi)  
/X [ <</U (mi)           % x offset represented in miles  
/C .00139             % Conversion from user space units to miles  
/D 100000  
]  
/D [<< /U (mi) /C 1 >> % Distance: initial unit is miles; no conversion needed  
<< /U (ft) /C 5280 >> % Conversion from miles to feet  
<< /U (in) /C 12       % Conversion from feet to inches  
/F /F /D 8 >>         % Fractions of inches rounded to nearest 1/8  
]  
/A [<</U (acres)        % Area: measured in acres  
/C 640 >>              % Conversion from square miles to acres  
]  
>>
```

第 9 章 多媒体特性

本章主要是描述了支持嵌入式并显示多媒体内容的 PDF 特性。它包括以下几节：

- 9.1 节，“多媒体”描述了 PDF 1.5 版本中介绍的完整的多媒体参数。
- 9.2 节，“音频”，和 9.3 节，“视频”，描述了自从 PDF 1.2 版本以来就支持的特性。
- 9.4 节，“Alternate Presentations”，描述了在 PDF 1.4 版本中的幻灯片功能。
- 9.5 节，“3D 技术”，描述了在文档中嵌入 3D 图形技术，在 PDF 1.6 版本中有相关的介绍。

9.1、多媒体

PDF 1.5 介绍了综合语言概念来对以下参数进行使能：

- 任意可以嵌入到 PDF 文件媒体类型。（见附录 H 注意 144 列出的推荐使用 Acrobat 6.0 视图的媒体类型）。

- 嵌入式媒体，以及 PDF 文件中引用的外部媒体，可被多种播放软件播放。
(在一些情况下，播放软件可以对自身进行查看)。

注：播放这个术语可以使用在很多种类的媒体类型上，并且对音频或是视频也没有限制。例如，它可以适用于像 JPEG 这类的静态图像。

- 媒体对象可能拥有多个的播放方案，这可在播放时基于目前的条件如支持的带宽进行选择。
- 文件作者可以控制播放时间的条件，如哪些软件可被用来播放特定的媒体类型。
- 媒体对象可以使用多种途径进行播放；例如，在一个浮动的窗口上也能和其在静态页面上播放时具备一样的效果。
- 媒体未来的扩展结构能够通过当前视图软件合适的方法进行控制播放媒体。作者可以控制扩展结构未来的发展。
- 文件作者要能够适用多媒体的无障碍使用要求。
- 在线媒体类型可以有效的，甚至是非常大的作用。

下面列出的是多媒体的功能，并指出了每个专题在哪里有讨论：

- 9.11 节，“可行性”，描述了当媒体对象在一个特殊系统中可播放时的决定条件。
- 播放行为（见 630 页的“播放行为”一节）可以在开始播放多媒体内容时使用。
- 播放时的屏幕注释（见 602 页的“屏幕注释”）。
- 播放模式：媒体播放（见 718 页的“媒体播放”）定义了可播放媒体的特性，而选择播放（见 719 页的“选择播放”）初始选择了可以播放的媒体。
- 播放媒体包括了什么可以播放（见 9.1.3 节，“媒体模式对象”），怎么播放（见 9.1.4 节，“媒体播放因素”），以及在哪里播放（见 9.1.5 节，“媒体屏幕因素”）。
- 9.1.6 节，“其他媒体对象”，描述了多种在上述列出的用来参考的 PDF 对象。

注意：在下面章节中一些描述的特性在多媒体集成语言（SMIL 2.0）标准（见参考文献）中是有相关元素的。

9.1.1、可行性

当播放多媒体内容时，查看器软件必须经常对播放器以及使用（例如，播放大小和播放时间）建议作出决定。在做出这些决定时，查看器必须决定对象的可行性的使用。如果一个对象是不可行的，这个媒体就无法播放。如果对象是可行的，即使在缺少理想条件，这个媒体也可以播放。

这里有几项在多媒体对象单中对可行性有影响的因素。特别的，一些对象定义了两项可将其拆分两项的因素：

- **MH** (“必须实现”）：本项中指定的选项都是必须实现的；其他选项包含的对象被视为不可行的。
- **BE** (“最好能实现”）：这是最好能够实现的选项；但是，如果无法实现，包含的对象仍然是可行、有效的。

MH 和 **BE** 都是选项的集合，定义了同样的选项。在他们中所有的选项都是可行的，他们都可以对这些进行设置，或是只有一个对其设置，或是都不设置。例如，媒体播放参数选项（见表 9.14）允许播放选项通过 **MH** 和 **BE** 选项（见表 9.15）中的 **V** 选项方法对其进行设置。如果在 **MH** 选项中指定的播放选项不生效，则对象将被视为不可行，并且播放不能正常进行。如果在 **BE** 选项（而不是在 **MH** 选项）中的 **V** 选项，则播放仍然进行，不受影响：播放软件能够自动获取能使其更好播放的选项。

使用这个机制，作者可以指定最低要求（**MH**）和最优方案（**BE**）。他们还可通过旧软件界面指定要添加到多媒体选项的选项。如果在 **MH** 选项中存在无法被显示器显示的选项，则该对象将被视为不可行的。如果是 **BE** 中的选项不能被正常显示，则其活动性和可行性不受影响。除非有另外的申明，如果它的 **MH** 选项中包含有不能识别的关键字或是不能被关键字识别的部分，则应该考虑到不可行的情况。

以下的规则适合于 **MH** 和 **BE** 中的选项，其行为与其他的 PDF 选项不同：

- 如果一个选项是必须的，如果选项不管是在 **MH** 还是在 **BE** 中都要符合要求。
- 如果一个不存在于上述两个集合中的任何一个的可选选项，它将默认是保存在 **BE** 集合中（如果只保存到一个集合中）。
- 如果在 **MH** 和 **BE** 中存在同一个选项，则在 **BE** 集合中的这个选项将被视为无效。

- 如果 MH 或是 BE 集合中的一个选项值是一个集合或是阵列，它将作为决定可行性的一个最小单位。也就是说，所有在集合或阵列中的选项对于包含对象必须是可行的。

注意：当决定选项是否是可行时，由于彼此之间可能存在一些依赖关系，因此不要求对每一个选项都做出检查。也就是说，一个浏览软件或是播放器可能一次性（即使是不同的集合）的检查所有的选项来决定它们的数值是否有效、可行。

以下媒体对象都拥有 MH 和 BE 集合。除非在个别章节中指出，否则都符合以上的描述：

- 简述（表 9.2）
- 媒体剪辑数据（表 9.11）
- 媒体剪辑章节（表 9.13）
- 媒体播放参数（表 9.15）
- 媒体屏幕参数（表 9.18）

9.1.2、简述

存在两种播放对象：

- 移交媒体（见 718 页的“移交媒体”）是一个基本的媒体对象，可以通过它来指定要播放的内容，以及如何、在哪播放。
- 媒体选择（见 719 页的“媒体选择”）包含了一个有序的选择列单。这个清单可能包含有其他的媒体选择类型，构成了一个媒体选择树。浏览软件应该按顺序播放树（见 9.1.1 节，“可行性”）中的第一个媒体选择类型。

表 9.1 显示了常见的播放词汇。在播放词汇中的 N 词条指定了一个查找命名树（见 125 页的表 3.28）。JavaScript 行为（见 668 页的“JavaScript 节”），例如，就是用了这个技术。由于引用树的值必须是间接引用的，因此建议所有的播放词汇都作为间接的对象使用。

注意：移交目录不要求有词条树名。当对其使用时，浏览软件必须保证该名称报此与 N 词条（例如，如果用户界面允许更改名称）的值一致。建议（不是要求）文档不对不同多媒体使用同一名称。

MH 和 BE 词条可能会存在于一个或是其他的词汇选项中，正如在 9.11 节中的，“可行性”描述的。对于译码，这些词汇中都有单一的 C 词条（见表 9.2），C 词条的值是一个媒体标准词汇，其中明确规定了必须符合移交的标准以使之可行

(见表 9.3)。

正如在 9.11 节中描述的，媒体行为准则有点不同于其他的 MH/BE。通过所有的词条指定的这些规范标准不管是在 MH 还是 BE 中，都必须满足。唯一例外的是，如果 BE 中的一个词条是无法被浏览软件识别的，它将不对对象的可行性造成影响。如果一个在 MH 和 BE 中都存在的词条，则这个词条在这两者中都会有各自的评价，其中优先考虑 MH (符合 BE 的将被忽略)。

表 9.1 所有共同的选项

关键字	格式	值
Type	name	(可选) PDF 对象词汇格式描述；如果存在，必须提交 移交的对象
S	name	(必选) 移交本词典的类型描述。可能成为移交媒体的 MR 或是选择移交的 SR 项。如果浏览软件没有识别到此 项的值，则操作视为非法的。
N	Text string	(可选) 一个 Unicode 编码的文本字符串指定在用户界 面中的移交名称，并且利用 JavaScript 进行对该名称的 查找。
MH	dictionary	(可选) 项的集合 (见表 9.2) 必须对移交的可行性进 行实现
BE	dictionary	(可选) 项的集合 (见表 9.2) 需要，而且只能在“最 好能实现”中被实现

表 9.2 MH/BE 中的移交选项

关键字	格式	值
C	dictionary	(可选) 一个媒体标准 (见表 9.3) 注：在 9.1.1 节的“可行性”中描述了媒体类型标准不 同于 MH/BE。其所有方案都必须满足 MH 或是 BE 中指定 的标准。唯一例外的是如果是在 BE 中，则浏览软件无法 识别，但这不影响对象的可行性。

表 9.3 标准媒体中的选项

关键字	格式	值
Type	name	(可选) 本标准描述的 PDF 对象的格式; 如果存在, 必须成为 MediaCriteria 中的一项
A	boolean	(可选) 如果有指定, 则此项的数值必须符合作者的要求, 如为了使对象可行而对音频元素进行描述。相当于 SMIL 中的 systemAudioDesc 属性。
C	boolean	(可选) 如果有指定, 则此项的数值必须符合作者为了显示说明性内容, 从而使对象可行。相当于 SMIL 中的 systemCaptions 属性。
O	boolean	(可选) 如果有指定, 此项的数值必须符合作者为了使对象可行而对音频进行叠加。
S	boolean	(可选) 如果有指定, 此项的数值必须符合作者的要求, 从而使对象的字幕可行。
R	integer	(可选) 如果指定, 系统的带宽 (bits/秒) 必须大于或等于这个字符的数值, 并使对象有效。与 SMIL 的 systemBitrate 属性相等。
D	dictionary	(可选) 一个字符集合 (见表 9.4) 指定了为了保证对象可行的最低深度要求。等于 SMIL 的 systemScreenDepth 属性值。
Z	dictionary	(可选) 一个字符集合 (见表 9.5) 指定了为了保证对象可行的最小的屏幕大小。于 SMIL 的 systemScreenSize 属性值相等。
V	array	(可选) 一个软件识别对象的阵列 (见 736 页的“软件识别字符集”。如果这个阵列存在并且非空, 则浏览软件为了实现对象的可行必须保证一个或是更多的对象存在于这个阵列中。)
P	array	(可选) 一个包含有一个或是两个对象名称的阵列指定了最小的和可选择的 PDF 语言版本的最大值, 与文档目录 (见表 3.25) 中的 Version 词条具备相同格式。如果

这个阵列是存在且非空的，则最大版本号必须完全的支持于浏览软件，并且为了使对象可行必须保证其在指定的范围。

L array (可选) 一个语言标识（见 865 页的“语言标识”）的阵列。如果这个阵列是存在且非空的，则浏览软件上使用的语言必须符合语言标识，或者只存在基本的代码来匹配基本标识的代码，从而使得对象可行。这等于 SMIL 的 `systemLanguage` 属性值。

9.4 在最低深度集合中的字符项

关键字	格式	值
Type	name	(可选) 本规范描述的 PDF 格式对象；如果存在必须等于最低深度字符集中的 <code>MinBitDepth</code> 值。
V	integer	(必选) 为了使其有效、可行，而使用一个明确的整数（0 或是大于 0 的整数）描述了监视器最小的屏幕高度（bits 为单位）。负数不能使用。
M	integer	(可选) 监视器说明符（见表 9.28）指定了监视器 V 值，并对其测试。如果这值是无法被识别的，则对象不可行。 默认值：0.

表 9.5 最小屏幕的字符项

关键字	格式	值
Type	name	(可选) 本规范描述的 PDF 对象格式；如果存在，必须为对象的 <code>MinScreenSize</code> 属性值。
V	name	(必选) 一个含有两个非负整数的阵列。由 M 指定的监视器宽度和高度（像素）必须分别大于或等于阵列中的第一和第二个整数值，从而使对象可行。
M	integer	(可选) 一个监视器说明符（见表 9.28）指出了监视器的 V 值，且对其测试。如果这个值无法被识别，则对象不可行。默认值：0.

播放媒体

表 9.6 列出了在媒体播放字符集中的字符。它指出了媒体被播放 (C), 怎么播放 (P), 以及在哪里播放 (SP)。一个播放媒体对象当且仅当对象通过引用有效的 C, P 和 SP 字符时是有效、可行的。

在播放引用没有任何意义的输入时, C 可被省略。这要求 P 是存在且引用包含 PL 阵列的媒体播放参数字符 (见表 9.14), 引用媒体播放字符集 (见 733 页的“媒体播放字符集”) 有一个非空的 MU 数组或是一个非空的 A 数组。

表 9.6 媒体播放字符集中的额外项

关键字	格式	值
C	dictionary	(可选) 一个媒体剪辑字符集 (见 9.1.3 节, “媒体剪辑对象” 指定了当媒体播放对象运行时什么应该被播放。
P	dictionary	(如果 C 不存在则必选, 其他情况可选) 一个媒体播放参数字符集 (见 9.1.4 节, “媒体播放参数”) 指出了应该被播放的播放媒体对象。
SP	dictionary	(可选) 一个媒体屏幕参数字符集 (见 9.1.5 节, “媒体屏幕参数”) 指定了播放媒体对象应该被播放。默认值: 一个媒体屏幕参数字符集的字符项 (表 9.17) 包含了所有的默认值。

播放选择器

一个播放选择器指定了存在于 R 字符 (见表 9.7) 的对象阵列。在这个阵列中的对象存在优先权, 且最优的放在前面。在播放时间, 阵列中的对象会被评价并且第一个可行的播放媒体需要被播放。如果对象中的一个是选择器, 则选择器轮流受到评价, 相当于有效的深度搜索树。注意的是, 一个播放选择器可能是不可行的; 在这种情况下, 不对其关联的播放媒体进行评价 (实际上, 会跳过这个树的分支)。

可以使用这类机制, 例如, 指定一个大的视频剪接用在高带宽的机器上, 以及指定一个小的视频剪辑用在低带宽的机器上。

表 9.7 播放选择器的附加字符

关键字	格式	值
R	array	(必选) 一个播放对象的阵列。在阵列中的第一个可行播放媒体对象，或嵌套一个阵列中的选择器，应该被使用。允许这个阵列为空。

9.1.3、媒体剪辑对象

这里存在两种类型的媒体剪辑对象，由 S 自类型决定，它会是 MCD 媒体剪辑数据（见 720 页的“媒体剪辑数据”）或是媒体剪辑章节（见 723 页的“媒体剪辑章节”）的 MCS。这个所有媒体剪辑词汇的共同项都列在表 9.8 中。

表 9.8 所有媒体剪辑词汇中的共同项

关键字	格式	值
Type	name	(可选) 本规范描述的 PDF 格式；如果存在，必须为媒体剪辑词汇中的 MediaClip。
S	name	(必选) 本规范描述的媒体剪辑子类型。可能是媒体剪辑数据（见 720 页的“媒体剪辑数据”）的 MCD 或是媒体剪辑章节（见 723 页的“媒体剪辑章节”）的 MCS。如果浏览软件不能识别这个词条，则媒体剪辑将被视为不可行。
N	text string	(可选) 媒体剪辑的名称，用于用户界面。

媒体剪辑数据

一个媒体剪辑数据字符集定义了可播放的媒体剪辑数据。例如，它可能引用一个 URL 网址或是演示视频，从而在 PDF 文件中嵌入一个电影。表 9.9 列出了它包含的项。

表 9.9 在媒体剪辑数据词汇表中的额外项

关键字	格式	值
D	file specification	(必选) 一个完整的文档标准或是指定了实际的媒体数据的 XObject。

ion or stream		
CT	string	(可选, 不允许从 XObjects 中取值) 在 D 中的数据字符标识。这个字符必须符合 Internet RFC 2045, Multipurpose internet mail extensions (MIME) 第一部分的规范标准的内容格式: Internet Message Bodies (见 Bibliography) 中的格式。
P	dictionary	(可选) 一个字符集许可标识 (见表 9.10) 包含了控制媒体数据的标识。默认值: 一个媒体字符集允许标志包含的默认值。
Alt	array	(可选) 一个阵列用来提供替代文本描述媒体剪辑数据无法播放时的方法; 见 869 页的“多种文本阵列描述语言”。
PL	dictionary	(可选) 一个媒体播放字符集 (见 733 页的“媒体播放字符集”) 标识符除此以外的信息, 标准和非标准用于播放媒体的播放器。 注: 如果媒体播放字符集是非可行的, 则媒体剪辑数据也是非可行的。
MH	dictionary	(可选) 这个字符集的字符 (见表 9.11) 必须为媒体剪辑数据提供使之有效的方法。
BE	dictionary	(可选) 这个字符集的字符 (见表 9.11) 只能在“最好实现”中实现。

媒体剪辑数据对象如果被不包括 Type 词条的 D 词条引用, 则必须是不可行的, 其中 Type 词条是不被承认或是引用对象不是一个字符集合或流。注意这个不包括简单的文件规范 (见 3.10 节的“文件规范”)。

如果 D 参考一个嵌入文件流 (见 3.10.3 节“嵌入文件流”) 的文件规范, 这个嵌入的文件流如果存在 Subtype 字符的话将被忽略, 并且这个媒体剪辑数据字符集的 CT 字符将使用这个数据的格式。

如果 D 参考了 XObject, 相关的播放器是的隐式的浏览软件, 若为其他格式, 则 XObject 结构会被提交。例如, 媒体播放因素字符集 (见表 9.14) 中的 F 和 D

项适用于 XObject 格式，与它们适用 QuickTime 一致。

若媒体类型不是 XObjects，则媒体剪辑对象必须提供足够的信息允许浏览软件去找出合适的播放软件。这可以根据以下的一个或是两个信息来实现：

- CT 字符指定了媒体（首选方法）的内容格式。如果这个字符存在，则选择的任何一个播放器必须支持这个内容格式。
- PL 字符指定了一个或是更多的可以用来播放这个引用媒体的播放器。如果 CT 存在，则建议使用这个字符。可查看附录 H 的注 145 的详细介绍。

P 词条指定一个允许的媒体字符集（见表 9.10），它指定了通过浏览软件使媒体参考数据的方法。这些字符集允许作者控制他们使用数据从而允许对其进行复制。如果字符集包含不能被识别的字符或是字符含有不能被识别的数值，则它被认为是非可行的，并且浏览软件无法播放媒体。

表 9.10 媒体允许的外部字符集

关键字	格式	值
Type	name	(可选) 本规范描述的 PDF 对象格式；如果存在，必须等于媒体允许字符集中的 MediaPermissions。
TF	string	(可选) 一个指定的字符串可以接受写入临时文件用来播放一个媒体剪辑。有效的数据有： (TEMPNEVER) 不允许。 (TEMPEXTRACT) 当文档允许外部内容时允许；例如，当使用者使用 bit 5 时允许（见表 3.20）设置。 (TEMPACCESS) 当文档允许外部内容是允许，除了可达到的目的外；例如，当使用 bit 5 或 10 允许对其设置（见表 3.20）。 (TEMPALWAYS) 总是允许。 默认值：(TEMPNEVER) 一个不承认的值是 (TEMPNEVER)

在媒体剪辑数据中 MH 和 BE 字符集（见表 9.11）中的的 BU 词条为媒体数据

规定了一个基本 URL。相对于在媒体（例如指向辅助文件或是使用超级链接）中的 URLs 应该考虑 BU 的值来对其进行解决。以下是 BU 词条中应该要注意的：

- MH 字符集中的 BU 和基本 URL 不能实现（例如，播放器不接受 URLs），媒体剪辑数据不可行。
- 决定对象可行时不要求进行校验，不管基本的 URL 是否有效（例如，目标机存在的情况）。
- 媒体中的全部的 URLs 不产生影响。
- 如果媒体本身包含一个基本的 URL（例如，在 HTML 中定义的<BASE>元素），这个值将优先赋给 BU。
- BU 是完全独立的与文档目录（见 3.6.1 节，“文档目录”）中的 URI 字符数值无关。
- 如果 BU 不存在，媒体将被嵌入到文档中，PDF 中的 URL 在当其数值为 BU 字符在 BE 字符集中的数值时使用；也就是说，实现一个比较有价值的基 URL。

表 9.11 在媒体剪辑数据 MH/BE 字符集中的外部字符

关键字	格式	值
BU	string	(可选) 一个绝对的 URL 使用作为在解决媒体数据中的相对的 URLs 的绝对 URL。

媒体剪辑组

一个媒体剪辑组字符集（见表 9.12）定义了一个连续分节的另外媒体剪辑对象（即 next-level 媒体剪辑对象）。例如，一个媒体剪辑组可以定义一个 15 分钟片段的媒体剪辑数据对象表示一个两小时的电影。这个 next-level 媒体剪辑对象，由 D 字符指定，既是一个媒体剪辑数据对象或是其他的媒体剪辑对象。然而，通过媒体剪辑组中的 D 字符的联机必须终止一个媒体剪辑数据对象。如果 next-level 媒体对象是不可行的，则媒体剪辑组也是不可行的。

表 9.12 在媒体剪辑组字符集中的额外的字符

关键字	格式	值
D	dictionary	(必选) 媒体剪辑组或是媒体剪辑组对象中的媒体剪辑

数据对象 (`next-level` 媒体对象) 定义了一个连续的可分节的剪辑对象。

Alt	<code>array</code>	(可选) 一个数组, 提供了在媒体剪辑组不能被播放时的一种文字表述方法; 见 869 页的“多种语言数组”。
MH	<code>dictionary</code>	(可选) 字符集的字符 (见表 9.13) 在可行的媒体剪辑组中必须是可行的。
BE	<code>dictionary</code>	(可选) 字符集的字符 (见表 9.13) 只能在“最好能实现”节中为可行的。

在媒体剪辑组中的 `MH` 和 `BE` 字符集 (见表 9.13) 中的 `B` 和 `E` 字符通过指定开始和结束的偏移定义了参考 `D` 的部分 `next-level` 媒体对象。依靠媒体类型, 这个偏移可能是由时间, 框架, 或是标记 (见 732 页中的“媒体偏移字符集”) 对其进行定义。`B` 和 `E` 不要求指定同样的偏移格式。

以下是适用于这些偏移的规则:

- 对于没有意义的媒体类型 (例如 JPEG 图像), `B` 和 `E` 是可忽略的, 且对可行性不产生影响。
- 当 `B` 或 `E` 被时间或是框架指定, 它们的值被认为为 `next-level` 媒体剪辑的开始。然而, 当 `E` 指定一个偏移, 且超出了 `next-level` 媒体剪辑的长度时, 最终的值将会被取代, 但不对可行性造成影响。
- 当 `B` 或 `E` 被标记指定时, 与基本媒体剪辑数据对象的绝对偏移是一致的。如果这个偏移不在 `next-level` 媒体剪辑 (如有) 的范围内, 或者如果这个标记不存在这个基本的媒体剪辑中, 将忽略其的存在, 并不对可行性造成影响。
- 如果绝对偏移值是从所有的媒体剪辑组链中的 `B` 词条的值衍生出来大于或等于从所有 `E` 词条中衍生出来的绝对偏移值, 则将定义一个空的范围。一个空范围是合法的。
- 在媒体剪辑组中 `MH` 字符集的 `B` 或 `E` 词条必须在播放时间实现, 会导致媒体剪辑组不可行。(如果词条的值不可行或是如果播放器不支持该值, 则词条将无法实现; 例如, 播放器不支持标记)。
- 如果 `B` 或 `E` 是在媒体剪辑组中 `MH` 字符集中, 则所有的 `B` 或 `E` 词条, 分别在更深的层次 (接近媒体剪辑数据) 将被认为是在 `MH` 字符集中 (即使他

们实际上是在 BE 字符集中)。

- 如果在 BE 字符集中的 B 或 E 词条不被支持，将在播放时间被忽略。

表 9.13 在媒体剪辑组 MH/BE 字符集的额外字符

关键字	格式	值
B	dictionary	(可选) 一个媒体偏移字符集 (见 732 页的“媒体偏移字符集”) 在媒体剪辑组的开始部分指定了 next-level 媒体对象中的偏移。默认值: next-level 媒体对象的开始位置。
E	dictionary	(可选) 一个媒体偏移字符集 (见 732 页的“媒体偏移字符集”) 在媒体剪辑组的开始部分指定了 next-level 媒体对象中的偏移。默认值: next-level 媒体对象的结束位置。

9.1.4、媒体播放参数

媒体播放参数字符集指定了一个媒体对象怎么被播放。它从媒体播放中进行引用 (见 718 页的“媒体播放”)。

表 9.14 媒体播放参数字符集的额外字符

关键字	格式	值
Type	name	(可选) 本规范描述的 PDF 对象格式; 如果存在, 则为媒体播放参数字符集的 MediaPlayParams。
PL	dictionary	(可选) 一个媒体播放字符集 (见 733 页的“媒体播放字符集”) 标识符除此以外的信息, 标准和非标准用于播放媒体的播放器。 注: 如果媒体播放字符集是非可行的, 则媒体剪辑数据也是非可行的。
MH	dictionary	(可选) 一个字符集的字符 (见表 9.13) 必须使媒体播放参数实现为可行的。
BE	dictionary	(可选) 一个字符集的字符 (见表 9.13) 只能被“最好能实现”实现。

V	integer	(可选) 一个整数指定一个音量等级比作为理想的音量等级。零值相当于静音；负值是非法的。默认值：100。
C	boolean	(可选) 一个标志指定是否显示特定显示控制器在播放时的使用界面（例如，播放/暂停/停止控制）。默认值：假
F	integer	(可选) 该以何种方式在符合播放区域播放一个媒体类型。 0 缩放媒体的宽度和高度缩放，同时保持纵横比。这样，媒体和播放区域有最大的接触面显示所有的媒体内容。等同于 SMIL 中的“meet”值。 1 缩放媒体的宽度和高度缩放，同时保持纵横比。这样，播放区域可以被完全填充，以及媒体内容的数额与播放区域的不适合将降到最小。等同于 SMIL 的“slice”值。 2 媒体的宽度和高度独立缩放，使媒体和播放区域一致，外表不一定保留原先的比例。这与 SMIL 中的“fill”值相同。 3 媒体不进行缩放。如果媒体区域是加宽或是加高的，则用户界面设置为滚动的。这与 SMIL 中的“scroll”值相同。 4 媒体不进行缩放。只有部分媒体区域与播放区域相交。与 SMIL 的“hidden”值相同。 5 使用播放器的默认设置（作者不偏好）。 默认值：5。 如果词条是 BE 字符集中的字符，则一个无法识别的值应视为默认值。如果该条目存在于 MH 字符集中，它有一个无法识别的值，对象应被视为非可行的。
D	dictionary	(可选) 一个媒体字符集（见表 9.16）。默认值：描述固定时间的词汇（见下文）。
A	boolean	(可选) 如果 true，当媒体被激化时将自动播放。如果为 false，激活时，媒体应先暂停（例如，第一帧将不被播放）。有关媒体会被暂停。默认值：真。

RC	number	(可选) 指定的期限 D 的重复迭代次数; 类似 SMIL 中的 repeatCount 属性。Zero 标识重复下去。负的值是非法的; 非负整数值是合法的。默认值: 1.0。
----	--------	--

D 词条的值是媒体时间字符集, 在表 9.16 中显示。它指出了持续时间(对应于 SMIL 中的一个简单时间概念)。这个持续时间可能是明确的时间长度, 它可能是无穷大的, 或是可能是媒体内在的时间(例如, 在两个小时的 QuickTime 电影的内在时间就是两个小时)。这个内在时间当媒体剪辑组(见在 723 页的“媒体剪辑组”)被使用时可能会被修改; 这个内在时间是不同于开始和结束之间的时间偏移长度。对于一个没有时间概念的媒体类型(如 JPEG 图片), 持续时间被认为是有无限大的。

如果简单时间比内在时间长, 则播放器将在媒体的简单时间过后的最后状态冻结媒体。对于视觉媒体类型, 最后的画面(帧)会被显示。对于听觉性媒体, 媒体在逻辑上是被冻结的, 因此不能继续播放声音。

注意: 在这个情况下, RC 词条指出了一个重复的计数, 适用于简单时间; 因此, play-pause 词条序列是一致重复 RC 时间。

表 9.16 媒体持续字符集的额外字符

关键字	格式	值
Type	name	(可选) 本规范描述的 PDF 对象格式; 如果存在, 必须等于媒体持续字符集中的 MediaDuration。
S	name	(必选) 媒体持续字符的子集。有效值为: I 持续时间等于相关媒体的内在时间 F 持续时间是无穷大的 T 持续时间由 T 字符指定 如果浏览软件不能识别这个字符的数值, 则媒体持续时间被视为不可行。
T	dictionary	(如果 S 的值等于 T 为必选; 其他的相反) 指定的时间长度描述了详细的持续时间(见表 9.24)。取值不能为负数。

9.1.5、媒体屏幕参数

媒体屏幕参数字符集（见表 9.17）指出了一个应该被播放的媒体对象。它包含在 9.1.1 节的“可行性”中讨论的 MH 和 BE 字符集（见表 9.18）。所有的媒体剪辑与一个特殊的文档有关并且当该文档关闭时必须停止播放。

注意：我们推荐浏览软件不允许使用浮动的窗口以及全屏模式，除非由作者特别的指出需要这样的操作。原因是基于文档安全性的考虑，如在窗口中播放一些用户无法观看到的内容，则其存在一些安全隐患。如果可以传达这些内容给用户，则这项建议可能被放宽；例如，在浮动窗口中显示文本。

表 9.17 媒体屏幕参数的额外字符集

关键字	格式	值
Type	name	(可选) 本规范描述的 PDF 对象格式；如果存在，则等于媒体屏幕参数字符集的 MediaScreenParams 值。
MH	dictionary	(可选) 这个字符集的字符（见表 9.18）必须由可行的媒体屏幕参数进行设置。
BE	dictionary	(可选) 这个字符集的字符（见表 9.18）只能由“最好能实现”实现。

表 9.18 媒体屏幕参数 MH/BE 字符集项

关键字	格式	值
W	integer	(可选) 媒体对象应该播放的窗口格式： 0 浮动窗口 1 覆盖率其他窗口的全屏窗口 2 隐藏窗口 3 与播放媒体相关的屏幕注释（见 602 页的“屏幕注释”）占用了屏幕的一个长方形区域。 默认值：3。无法识别 MH 中的值：对象是不可行的；在 BE 中：按照默认值处理。
B	array	(可选) 在 0.0 到 1.0 范围内的三个数组成的一个数组指出了在该媒体开始播放时显示区域背景颜色的 DeviceRGB 颜色空间的组成。如果媒体对象不完全覆盖

整个区域或是它有透明的试图，这个颜色将被使用。忽略隐藏的窗口。

默认值：执行定义。这个浏览软件应该选择一个基于 **W** 的合理的值；例如，浮动窗口的默认窗口背景或是全屏下优先使用的背景颜色。

注意：如果一个媒体格式拥有一个固有的背景颜色，**B** 不需要对其进行覆盖。然而，**B** 颜色在媒体在透明区域或是其他不能完全覆盖整个窗口的情况下是可见的。

0 **number** (可选) 在 0.0 到 1.0 范围内的一个数指出了连续的不透明的数值用在通过 **B** 对背景颜色进行绘制。1.0 以下的数值表示窗口透明的；例如，在一个浮动窗口后面显示窗口即使这个媒体不能覆盖整个浮动窗口。0.0 的数值表示全部透明并且与 **B** 无关。全屏和隐藏窗口被忽略。

默认值：1.0 (完全不透明)

M **integer** (可选) 监视器说明符（见表 9.28）指出了监视器在一个可以显示浮动或是全屏窗口的多窗口系统下。忽略其他格式。

默认值：0 (文本终端)。在 **MH** 中的值不被承认：对象是不可行的；在 **BE**：与默认值相同处理。

F **dictionary** (如果 **W** 值为 0 则必选；其他相反) 一个浮动窗口参数字符集（表 9.19）指出了浮动窗口显示的大小，位置和使用选项。

在媒体屏幕参数 **MH/BE** 字符集中的 **F** 词条是一个浮动窗口参数字符集，并且在 9.19 中列出了具体情况。这个词条在浮动窗口参数字符集中被当作存在于 **MH** 或是 **BE** 字符集中，而实际则是参照而已。也就是说，是对所包含参数分别作评估，而不是对整个系统做一个整体的评价。（也有可能是同时存在于 **MH** 和 **BE** 中的 **F**

词条。在这种情况下，如果给定的参数都是在浮动窗口的参数字符集，则以 **MH** 字符集中的字符为先)。

D, **P** 和 **RT** 词条被用来指定浮动窗口区域。一旦创建，浮动窗口的大小和位置就不再依赖于其他的窗口，即使初始大小或是位置是相对其他窗口计算得到的。

R, **P**, **RT** 和 **O** 这类无法被识别的字符处理如下：如果它们嵌入到 **MH** 字符集中，则浮动窗口参数对象（即媒体屏幕参数对象）必须被认为是不可行的；如果它们嵌入在 **BE** 字符集中，他们应该被按照默认值处理。

表 9.19 浮动窗口参数的额外字符集

关键字	格式	值
Type	name	(可选) 本规范描述的 PDF 对象格式；如果存在，必须等于浮动窗口参数字符集的 FWParams 属性值。
D	array	(必选) 包含两个非负整数表示浮动窗口的宽度和高度，均以像素为单位。这些数值对应于媒体显示区域的尺寸，但不包括项，如标题栏和操作栏。
RT	integer	(可选) 相对于浮动窗口的窗口应该被放置的位置： 0 文档窗口 1 应用窗口 2 完整的虚拟桌面 3 由媒体屏幕参数 MH 或 BE 字符集中的 M 指定的监视器（见 9.22） 默认值：0。
P	integer	(可选) 浮动窗口（包括标题栏和操作栏等）的相对于窗口的位置由 RT 指定： 0 左上角 1 上中心 2 右上角 3 中心左侧 4 中心

-
- 5 右边
 - 6 左下角
 - 7 下中心
 - 8 右下角

默认值：4。

0	integer	(可选) 指出如果浮动窗口完全放置或是部分偏移屏幕 (这就是没有完全在物理监视器上可见) 将会出现什么情况。
0		不采取任何特别行动
1		移动和/或调整窗口大小使之适应屏幕
2		考虑的对象不可行

默认值：1。

T	boolean	(可选) 如果为 true，则浮动窗口应该有一个标题栏。 默认值：true。
UC	boolean	(可选；当 T 为 true 时是有意义的) 如果为 true，浮动窗口应该包括允许使用者使用界面元素关闭浮动窗口。

默认值：true。

R	integer	(可选) 指出浮动窗口是否可由使用者来调整大小：
0		不能调整大小
1		当宽度比保留的情况下允许调整大小
2		不保留宽度比的情况下调整大小

默认值：0。

TT	array	(可选；当 T 为 True 时是有意义的) 数组提供了在浮动窗口标题栏上显示的文本。见 869 页的“多种文本数组”。如果不存在，则浏览软件可以提供默认的文本。
----	-------	---

媒体偏移字符集

一个媒体偏移字符集（表 9.20）指出了在媒体对象中的偏移值，S（子类型）指示了如何实现指定的偏移；就时间而言（例如，“10 秒”），就帧而言（例如“20 帧”）或是就标记而言（例如“第一章”）。不同的媒体类型支持不同的偏移类型。

表 9.20 所有媒体偏移字符集的常见项

关键字	格式	值
Type	name	(可选) 本规范描述的 PDF 对象格式；如果存在，必须是媒体偏移字符集中的 MediaOffset 属性值。
S	name	(必选) 媒体偏移字符集的子类。有效值有： T 一个媒体偏移时间字符集（见表 9.21） F 一个媒体偏移帧字符集（见表 9.22） M 一个媒体偏移标记字符集（见表 9.23） 如果浏览器不能识别这项的值，则播放将被视为不可行。

表 9.21 媒体偏移时间字符集额外项

关键字	格式	值
T	dictionary	(必选) 时间长度字符集（见表 9.24）指定了媒体对象的时间偏移。无法预计的时间长度不允许包含在内容中。如果媒体的时间长度字符集是不可行的则媒体偏移时间字符集是不可行的。

表 9.23 媒体偏移帧字符集的额外项

关键字	格式	值
F	integer	(必选) 指定了媒体对象中的帧。帧数从 0 开始计数；不可预计的帧数是不允许的。

表 9.23 媒体偏移标记字符集的额外项

关键字	格式	值
M	text	(必须) 此文本字符串指出了媒体对象中的一个命名偏

string 移值。

时间长度字符集

时间长度字符集指定了时间的长度；表 9.24 显示了它包含的项。

表 9.24 时间长度字符集包含项

关键字	格式	值
Type	name	(可选) 本规范描述的 PDF 对象格式；如果存在，则等于时间长度字符集中的 <code>Timespan</code> 属性值。
S	name	(必选) 时间长度字符集的子集。当前值只允许为 S(普通时间长度)。如果浏览器没有识别到这项值，则播放将被视为不可行。
V	number	(必选) 时间长度的秒数。允许非整数的数值。允许使用负数，但是在一些特定情况下则不允许（所有不允许负值的情况在 PDF1.5 中有定义）。

注意：此词条只有在 S 词条设置为 S 时才被要求。在未来定义此子类时不需要使用这个词条了。

9.1.6、其他多媒体对象

本节定义了一些被之前章节应用的格式集合。

媒体播放字符集

一个媒体播放字符集能够被媒体剪辑数据（见 720 页的“媒体剪辑数据”）和媒体播放参数（见 9.1.4 节的“媒体播放参数”）字符集，并且允许它们指定哪些播放器可以或是不行被用来播放相关的媒体。这个媒体播放字符集引用了为每个播放器提供详细信息的媒体播放器消息字符集（见下面的“媒体播放器消息字符集”）。

表 9.25 媒体播放字符集的额外项

关键字	格式	值
Type	name	(可选) 本规范描述的 PDF 对象的格式；如果存在，则

		必须等于媒体播放字符集中的 MediaPlayers 属性值。
MU	array	(可选) 一个媒体播放器消息对象 (见表 9.26) 的数组指定了播放器的设置, 其中必须有一个被使用在媒体对象上。
<p>注意:任何在 NU 中指定的播放器都有效的从 MU 中移除。 (例如, 如果 MU 指定的播放器从版本 1 到 5 以及 NU 指定的同一个播放器版本 1 和 2, MU 有效的是 3 到 5 版本号。)</p>		
A	array	(可选) 一个媒体播放器消息对象 (见表格 9.26) 的数组指定播放器的设置。其中任何一项都可能被使用在媒体对象上。如果 MU 存在且不为空, 则 A 将被忽略。
NU	array	(可选) 一个媒体播放器消息对象 (见表 9.26) 的数组指定播放器的设置。不能被使用在媒体对象上 (即使他们在 MU 中都是有明确规定)。

MU, A 和 NU 指定了一个或多个媒体播放器消息对象。(一个空的数组被当作不存在)。媒体播放器消息对象允许指定播放重叠的范围 (例如, MU 可能包含一个媒体播放器消息字符集描述 X 播放器版本 1 到 10, 而另外一个描述从 3 到 5)。

一个不可行的媒体播放消息对象被 MU, NU 或 A 引用, 被当作不存在在原来的阵列中, 并且媒体播放消息对象包含相同的软件校验字符集 (见 736 页的“软件校验字符集”) 在逻辑上被认为是存在于 NU 中。同样的规则适用于包含一个部分软件无法识别的标识字符集的媒体播放消息对象。

媒体剪辑数据和媒体播放参数字符集能够使用播放进行操作, 并且每一个都参考媒体播放器字符集, 可能存在在两种媒体播放器字符集之间的内容出现冲突。在播放时间, 浏览软件应该使用以下演示决定是否一个在机器上出现的播放器能够使用。这个播放器能够被使用在任何条件为真的允许环境下:

演示 9.1

1. 播放器不支持一直的内容格式。
2. 在 NU 数组的每个字符集中发现播放器。
3. 这两个字符集拥有非空的 MU 数组且不存在播放器, 或者说是只有字符集

有非空的 MU 数组，且不包含有播放器。

4. 不管字符集有非空的 MU 数组，内容格式未知，且播放器在两个字符集中的 A 词条中未发现。

如果没有一个条件符合，则播放器可以使用。

注意：如果播放器匹配在 PID 中的一个项的信息，如 9.2 描述的那样，则其将在 NU, MU 或是 A 词条中“被发现”。

媒体播放消息字符集

媒体播放消息字符集提供了关于特定媒体播放器的各种信息。其项（见表 9.26）允许将信息与播放器的特殊版本或是版本范围进行相关。自 PDF 1.5 开始，只有 PID 项提供了有关播放器的信息，这将在下节的“软件识别字符集”中讲到。

表 9.26 媒体播放消息字符集的额外项

关键字	格式	值
Type	name	(可选) 本规范描述的 PDF 对象格式；如果存在则必须等于媒体播放消息字符集的 MediaPlayerInfo 属性值。
PID	dictionary	(必选) 一个软件对象标识符（见下面的“软件对象标识符”）指定了播放器名称、版本和为媒体播放器提供消息对象的操作系统。
MH	dictionary	(可选) 字符集包含必须使这个对象可行的选项。 注意：当前，没有为这个字符集定义
BE	dictionary	(可选) 字符集包含只能在“最好能实现”中的项。 注意：当前，没有为这个字符集定义

软件标识字符集

软件标识字符集允许软件被通过名称、版本号范围和操作系统识别；它指定了列在表 9.27 中的选项。一个浏览软件使用这些信息来决定是否给媒体播放器可使用的环境。如果字符集包含不能被播放器识别的关键字，它将只对部分作出验证。这个播放软件可能会或者不会决定将软件作为可行的标识符，依据其中使

用的上下文。

以下过程是判断一个软件是否被认为是匹配的软件校验对象：

演示 9.2

- 软件的名称必须与 **U** 词条（见下面的“软件 URIs”）指定的名称匹配。
- 软件版本必须在 **L**, **H**, **L1**, 和 **H1** 指定的选项中（见下面的“版本数组”）。
- 本机的操作系统名称必须准确的匹配一个存在于 **OS** 数组中的元素。如果数组不存在或是为空，则匹配默认为成立。

表 9.27 软件标识字符集额外项

关键字	格式	值
Type	name	(可选) 本规范描述的 PDF 对象格式；如果存在，必须为软件标识字符集额外项的 SoftwareIdentifier 属性值。
U	string	(必选) 一个 URI 标识的软件（见下面的“软件 URIs”）。
L	array	(可选) 在较低的范围内约束软件版本，且指定该软件对象标识符（见下面的“版本数组”）。默认值：数组[0]。
L1	boolean	(可选) 如果为 True ，则由 L 和 H 定义的时间下边界是在范围内的；即，软件版本必须大于或等于 L （见下面的“版本数组”）。如果为 False ，则不在范围内。默认为： true 。
H	array	(可选) 软件版本约束范围的上边界指定了对象标识符（见下面的“版本数组”）。默认值：空数组。
H1	boolean	(可选) 如果为 true ，则由 L 和 H 定义的时间上边界是在范围内的；即，软件版本必须小于或等于 H （见下面的“版本数组”）。如果为 false ，则不在范围内。默认值为： true 。
OS	array	(可选) 字符串数组表示了操作系统标识符指出了对象支持的操作系统。该值与 SMIL 2.0 的 systemOperatingSystem 属性值相等。可能没有对数组校验的多个副本。一个空数组被认为是表示所有的操作系统。默认值：空数组。

软件 URIs

U 词条是一个 URI（通用资源标识符）指定了一款软件。它被按照结构解释；

目前唯一定义了结构的是 vnd. adobe. swname。这个结构名称是 case-insensitive；如果没有被浏览器识别，则这个软件必须识别出不匹配。这个结构的语法是：

“vnd. adobe. swname:” software_name

其中 software_name 与定义在 Internet RFC 2396 中的 reg_name 相等，统一资源标识符(URI)：通用语法；见参考文献。Software_name 被认为是一个 UTF-8 编码字符序列，转义成网址（见 878 页的“URL 字符”）。也就是说，覆盖原先的软件名称，software_name 必须转义，并且作为一个 UTF-8 字符使用。实际的软件名称必须区分大小写的比较时尚的名称。

软件名称属于二等的名称（见附录 E）。例如，URI 是较容易改变的

Vnd. adobe. swname:ADBE_Acrobat

版本数组

L, H, L1 和 H1 都是被用在指定范围的软件版本上。L 和 H 的版本数组中包含 0 或非负整数表示权限号码。在主版本号中的第一个整数数字，以及随后的越来越小的整数数字。H 必须大于或等于 L，根据以下规则比较版本数组：

演示 9.3 比较版本数组

1. 一个空版本的数组视为无穷；也就是说，它被看作比其他除了另外一个空数组外的任何版本的数组还要大。两个空数组是等价的。
2. 当比较包含不同数量元素的数组时，较小的数组就填充 0 值时的元素的个数相等。例如，当比 i jao[5 1 2 3 4] 与 [5] 时，后面这个数组将被当作 [5 0 0 0 0]。
3. 从第一个元素开始比较数组中的对应元素。当有不同的字符出现时，包含较大元素的数组被认为是较大的版本。如果没有不一样的元素，则版本一样。
注意：如果一个版本数组包含负数，它将被认为是不可行的，就像软件校验被取消。

监视器区分符

一个监视器区分符是一个整数指出了系统上的一个物理监视器，它可以是表 9.28 中的一个数值：

表 9.28 监视器区分符值

值	描述
0	监视器包含文档窗口的最大部分
1	监视器包含文档窗口的最小部分
2	初级监视器。如果没有监视器被认为是初级监视器，则默认为 0 的情况
3	最大颜色深度的监视器
4	最大区域的监视器（以像素为单位）
5	最高的监视器（以像素为单位）
6	最宽的监视器（以像素为单位）

对于其中的一些数值，可能是播放时间的“纽带”；例如，两个监视器可能拥有同样的颜色深度。纽带是实现依赖关系的方法。

9.2、声音

一个声音对象 (PDF 1.2) 是包含定义了一个声音通过电脑话筒的声音信号值。这个声音进入一个声音设备或是声音行为字符集(见 601 页的表 8.32 和 625 页的表 8.54) 标识一个声音对象当设备被激活时表示将要发出声音。

由于声音对象是一个流，它可以包含对所有的流来说都能够正常使用的任何标准，正如在 38 页中的表 3.4 所描述的。特别是，如果它包含一个 F (文件规范) 项，声音将被定义在外部文件中。这个声音文件必须自我进行描述，包含所有播放此声音需要的信息；在 PDF 文件中不需要任何额外的信息。

注意：AIFF, AIFF-C (mac 操作系统), RIFF (.wav), 以及 snd (.au) 格式的文件都是自我描述的。

如果没有 F 词条存在，声音对象本身需要包含的样本数据和所有其他声音需要定义的信息。表 9.29 显示了对于一个声音对象来说额外的字符集。

表 9.29 特有声音对象的额外字符集

关键字	格式	值
字		

Type	name	(可选) 本规范描述的 PDF 对象格式; 如果存在, 必须等于声音对象的 Sound。
R	number	(必选) 采样频率, 以每秒采样的值
C	integer	(可选) 声音的渠道数。默认值: 1. (详细情况见附录 H 的标注 146)
B	integer	(可选) 样品每个通道的位数。默认: 8.
E	name	(可选) 样本数据的编码格式: Raw 未指定或是在 0 到 2^8 -1 无符号数 Signed 两个互为补充的数值 muLaw μ -law 编码的样本 A-law A-law 编码的样本
默认值: Raw。		
CO	name	(可选) 用于对声音样本数据的压缩格式。(这是从压缩流中分离出来通过声音对象的 Filter 词条指定的; 参加 38 页的表 3.4 和 3.3 节, “过滤器”)。如果这个词条存在, 则没有使用任何声音压缩; 这个数据包括样本以 R 样本每秒每个渠道播放的波形。
CP	(various)	(可选) 对使用的音频压缩格式可选择的参数选项。 注意: 在出版时, 没有为 CO 和 CP 提供标准值。

示例值存储在流中有重要作用的第一位(当示例位数超过 8 位时遵循大端)。示例是不是按照 8 位重复出现的将被封装成连续的数据, 以较有意义的底部开始。如果示例横跨一个字节区域, 则将最重要的信息存储在第一个字节中, 在随后的字节显示非重要的信息。对于双声道, 示例将被存储在交错的格式中, 每个样本左通道的数据(通道 1)放在相应右声道(通道 2)的前面。

为了最大限度的使 PDF 文档包含嵌入的声音, 推荐 PDF 浏览器和插件的扩展支持至少要包含有以下的格式(假设有足够的硬件和平台操作系统支持播放声音的话):

- R 8000, 11, 025, 或是 22, 050 为每秒的样本数
- C 1 或 2 通道

B 每个通道支持 8 或 16 位

E Raw, Signed 或 muLaw 编码

如果编码 (E) 采用的是 Raw 或是 Signed, R 必须是 11,025 或是 22,050 每个通道的样本数。如果编码是 muLaw, R 必须为 8000 每个通道样本数, C 必须为通道 1, B 必须是每通道 8 位。音频播放软件应该能够支持格式之间的转换, 降低频率, 并结合所需的平台通过更良好的渠道来实现播放。

9.3、电影

注意: 这部分描述的特征是即将过时的, 它们的用法是不再被推荐的。它们被一般多媒体框架取代, 这在 9.1 的“多媒体”中有描述。

PDF 有一个功能, 可以通过电影注解的方法, 把电影嵌入到一个文档中。参见“[601](#) 页电影注解”。除了名称, 电影可以全部由声音组成, 而没有在屏幕上显示可见图片。电影注解字典中的 Movie 和 A 条目, 分别地参考 movie dictionary (表格 9.30) 描述电影的状态特征和 movie activation dictionary (表格 9.31), 具体说明了它是如何出现的。

表格 9.30 电影字典的条目

关键字	类型	值
F	file specification	(必需的) 标识自我描述电影文件的说明文档 注意: 自我描述电影文件的说明文档的格式是未被指定的, 没有可移植性的保证
Aspect	array	(可选的) 电影边框的宽度和高度, 像素被指定为 [width height]。一个包含全部声音而没有可见图片的电影, 它的条目应该被省略。参见附录 H 的执行记录 147
Rotate	integer	(可选的) 旋转电影角度的度数大小与页面顺时针有关。 这个度数值必须是 90 的倍数。默认值: 0
Poster	boolean or stream	(可选的) 一个具体指定是否以及怎样去显示电影中 poster image 的帧或流。如果这个值是流, 它包含图片 Xobject (参见 4.8 “Images”), 显示为 poster。如果他的布尔值是 true, 这个 poster image 应该重新从电影中获取; 如果是 false, 应该没有 poster 显示。参见附录

		H 的执行注释 148。默认值: false
--	--	------------------------

表格 9.31 电影动作字典条目

关键字	类型	值
Start	(various)	<p>(可选的) 播放的电影片段开始时间。电影时间值在时间单元中基于 <code>time scale</code> 表示, 这些 <code>time scale</code> 定义了每秒钟单元的数目。默认时间数值范围定义在电影数据中。开始时间是名义上地非负 64 比特整数, 具体说明如下:</p> <ul style="list-style-type: none">● 如果它被描述为一个整数 (服从于整数的执行极限, 如附录 C 定义), 应该被这样阐述● 如果不是被描述成一个整数, 它应该被明确指定为一个 8 字节字符串表示一个 64 比特 <code>twos-complement</code> 整数, 最重要的字节优先● 如果表示为一个不同于电影自身的时间比例, 它被表示为一个两个值的矩阵: 一个整数或字符串表示开始时间, 如上, 后面跟上一个阐述单元每秒的时间比例整数。 <p>如果这个条目被省略了, 电影从开头播放。</p>
Duration	(various)	(可选的) 电影片段播放的持续时间, 指定为和 <code>Start</code> 相同的形式。如果这个条目被忽略了, 电影播放到结尾。
Rate	number	(可选的) 播放电影的最初速度。如果这个条目的值是负的, 电影往后播放, 直到开头和持续时间。默认值是 1.0。
Volume	number	(可选的) 播放电影的最初音量, 范围是 -1.0 到 1.0。更大的值表示更大的音量; 负值没有声音。默认值是 1.0
ShowControls	boolean	(可选的) 阐述播放时是否显示电影指示条的标

		识。默认值是 <code>false</code>
<code>Mode</code>	<code>name</code>	(可选的) 播放电影的播放模式: 播放一次 播放一次或停止 播放 播放且电影指示条显示 重复 重复播放, 从开头到结尾直到停止 <code>Palindrome</code> 连续向前或向后播放直到停止 默认值是: 播放一次
<code>Synchronous</code>	<code>boolean</code>	(可选的) 阐述是否同步或异步播放电影。如果值是 <code>true</code> , 电影播放器保留控制直到电影完整或被用户删除。如果值是 <code>false</code> , 播放器在开始播放后立即返回控制到视图应用程序。默认值是 <code>false</code>
<code>FWScale</code>	<code>array</code>	(可选的) 播放电影的放大倍数。这个条目的存在, 暗示了电影是在漂浮窗口中播放的。如果条目不存在, 电影在注释矩形中播放。 天幕的值是一个两个正数的矩阵, [分子 分母], 表示电影的一个合理的放大倍数。(参见 149 页附录 H 的执行注意) 最后的窗口尺寸像素是 $(\text{numerator} \div \text{denominator}) \times \text{Aspect}$ <code>Aspect</code> 的值是取自电影字典的地方。(参见表格 9.30)
<code>FWPosition</code>	<code>array</code>	(可选的) 为了在漂浮窗口播放电影, 屏幕上窗口相关位置。两个数字的矩阵的值是[水平 垂直] 0.0 到 1.0 之间的每一个数字, 表示了和屏幕匹配的电影窗口相关的水平和垂直位置。参见附录 H 执行注意 150。默认值: [0.5 0.5]。

9.4、候补介绍

以 PDF1.4 开头的 PDF 文档可能含有候补介绍, 它在可能被浏览的文档中阐述间隔的方式。文档名称字典(见表格 3.28)中, 可选的候补介绍条目(PDF1.4)包含一个名称树, 它为文档标出名称串给候补介绍。

注意: 因为 PDF 浏览工具不要求支持 `AlternatePresentations`, 包含

`AlternatePresentations` 文档的作者，应该例示文件使一些有用的和有意义的可以被显示和印刷。例如，如果文档包含候补介绍一系列图片的幻灯片，这些图片应该被无法玩幻灯片的浏览者以静态方式浏览。

正如 PDF1.5，`Alternate Presentation` 的唯一类型是幻灯片。幻灯片被 JavaScript 行为的方法（参见 668 页“JavaScript 行为”）特别地恳求，交互的格式元素上的用户行为所初步了解。附录 H 的执行注释 151 描述 acrobat 的幻灯片的执行。

下面表格展示了幻灯片字典的条目。

表格 9.32 幻灯片字典条目		
关键字	类型	描述
Type	name	(必需的; PDF1.4) 这个字典描述的 PDF 对象的类型。 必须是字典幻灯片的 <code>SlideShow</code> 。
Subtype	name	(必需的; PDF 1.4) 这个字典描述的 PDF 对象的子类型。必须是字典幻灯片的 <code>Embedded</code>
Resources	name tree	(必需的; PDF1.4) 标出名字串给被交替外观参考的对象的名字树。 注意：虽然 PDF 将名字树的字符串看作不带说明的字符串，幻灯片将它们诠释为 UTF-8 编码的统一码
startTeseource	string	(必需的; PDF1.4) 一个字符串，必须和 Resources 条目的一个字符串匹配。它定义了源对象给幻灯片外观。

资源名称树阐述幻灯片的一个实际的文件系统。它将字符串（“文件名称”）和阐述被幻灯片适用资源的 PDF 对象。例如，一个根 `stream` 可能参考一个文件的名称，这可能会在资源名称树中看到，相应的对象会被当作文件下载。（这个实际问价系统是统一的；即，没有办法参考 `subfolders`）

典型地，图片像 `image Xojects`（参见 4.8.4 “图片字典”）一样存储在文档中，因此允许它们在标准 PDF 表示法和幻灯片之间共享。其他媒体对象存储在或嵌入在文件流中（参见 3.10.3 “嵌入文件流”）。也可以参见附录 H 的执行注释 152。

为了允许浏览者核对内容与特别浏览所支持的特征，有一点是需要的，那就

是所有参考对象在它们字典中包含一个类型条目，甚至当这个类型条目是已给对象的正常选择。

下面的例子说明了可变外观幻灯片的用法。

```
1 0 obj
<</Type /Catalog
/Pages 2 0 R
/Names 3 0 R % Indirect reference to name dictionary
>>
...
3 0 obj      % The name dictionary
<</AlternatePresentations 4 0 R >>
endobj
4 0 obj      % The 候补介绍 name tree
<</Names [(MySlideShow) 5 0 R]>>
endobj
5 0 obj      % The slideshow definition
<</Type /SlideShow
/Subtype /Embedded
/Resources <</Names [(mysvg.svg) 31 0 R
(abc0001.jpg) 35 0 R (abc0002.jpg) 36 0 R
(mysvg.js) 61 0 R (mymusic.mp3) 65 0 R ]>>
/StartResource (mysvg.svg)
>>
...
31 0 obj
<</Type /Filespec % The root object, which
/F (mysvg.svg) % points to an embedded file stream
/EF <</F 32 0 R>>
>>
endobj
32 0 obj      % The embedded file stream
```

```
<</Type /EmbeddedFile
/Subtype /image#2Fsvg+xml
/Length 72
>>
stream
<?xml version="1.0" standalone="no"?>
<svg><!-- Some SVG goes here--></svg>
endstream
endobj
% ... other objects not shown
```

9.5、3D 艺术品

PDF1.6 介绍了 `three-dimensional` 对象的收藏能力，正如用于 CAD 软件，被嵌入在 PDF 文件中。这种收藏就是所谓的 3D 电影；在 PDF 的文本中，被归类为 3D 艺术品。3D 艺术品 PDF 结构具备下列特征：

- 3D 艺术品可以提供在一页内；不是作为一个分离的窗口或者用户界面元素
- 3D 艺术品多个例子会在页或文档中出现
- 明确说明的 3D 艺术品 `view` 可以获得，包含一个最初显示默认视图和其他可被选视图。视图可能有名称，在用户界面里出现。
- 2D 内容会被 3D 艺术品覆盖
- 包含 3D 艺术品的页可以打印
- 用户可以旋转和移动艺术品，这样就能从任何角度或方向仔细观察一个复杂的对象
- JavaScript 和其他软件可以有计划地操纵艺术品中的对象，创建活跃外观，对象可以移动、旋转、出现和消失。Acrobat JavaScript Scripting Reference（参见参考书目）在 3D 注解中描述了 JavaScript 界面下文部分描述与 3D 艺术品有关的主要 PDF 对象，而且它提供关于 3D 图形的背景信息
- 3D annotations 提供 `virtual camera`，通过 `virtual camera` 可以观看艺术品。（参见 9.5.1 “3D 注解”）

- **3Dstreams** 包含真实的 3D 艺术品片段（参见 9.5.2 “3D Streams”）的阐述。在 PDF1.6 中，这种规范和 3D 工艺论坛（<http://www.3dif.org>）发展的 U3D 格式是一致的。其他格式可能在将来会被支持
- **3D views** 阐述关于摄像头和 3D 艺术品之间关系的信息（参见 9.5.3 “3D views”）
- 3D 坐标系统在 9.5.4 中有描述，“3D 注解的坐标系统”

9.5.1、3D 注解

3D 注解（PDF1.6）意味着 3D 艺术品在 PDF 文档中有阐述。表格 9.33 给出了具体指定给 3D 注解字典的条目。

另外，对于那些条目，要求 3D 注解在他的 AP 条目（参见 570 页 8.11）提供外观流，这些条目有正常外观（579 页表格 8.15 N 条目）。这种外观会被不支持 3D 注释的应用程序和所有为注释的最初显示的应用所采用。

表格 9.33 3D 注释参考的额外条目

关键字	类型	值
Subtype	name	(必需的) 这个字典描述的注释的类型。对于一个 3D 注释必须是 3D。
3DD	stream or dictionary	(必需的) 一个 3D 流（参见 9.5.2 “3D 流”）或 3D 参考书目字典（参见 754 页 “3D 参考书目字典”），指定一个 3D 艺术品被标明
3DV	(various)	(可选的) 明确说明 3D 艺术品的初始视图的对象，这个 3D 艺术品在注释活跃的时候被使用。它可以是一个 3D 视图字典（参见 9.5.3 “3D 视图”）或下列说明一个 3D 流（参见 9.35）VA 矩阵元素的类型： <ul style="list-style-type: none"> ● 一个阐述索引到 VA 矩阵的整数 ● 在 VA 矩阵中视图之一中，匹配 IN 条目的文本串 ● 指明 VA 矩阵中第一个、最后一个或默认条目的名称

		默认值是 3DD 阐述的 3D 流对象中的默认视图
3DA	Dictionary	(可选的) 一个活跃字典 (参见表格 9.34)，它定义时刻包括注释应该被激活或无效，在那些时刻的 3D 艺术品的状态。默认值是为所有条目包含默认值的一个有效的字典。
3DI	boolean	(可选的) 一个布尔代数，指明 3D 注释的基本用法。如果是 <code>true</code> ，它表明有效的；如果是 <code>false</code> ，表明是无效的计划，如一个 JavaScript 活跃。视图应用程序可能出现不同用户界面控制给交互的 3D 注释 (例如旋转，平放或放大艺术品)，而不是给被脚本或其他机制管理的那些程序。 默认值是 <code>true</code>
3DB	array	(可选的) 3D 视图盒，绘制 3D 艺术品的矩形区域。它必须在矩形内，在注释 Rect 条目明确说明以及在注释目标坐标系统表示 (见下文)。 默认值：在目标坐标系统中表示的注释 Rect 条目。这个值是 $[-w/2 \ -h/2 \ -w/2 \ -h/2]$ ， w 和 h 分别是 Rect 的宽度和高度。

3DB 条目名曲制定 3Dview box,，一个出现 3D 艺术品的矩形。这个视图盒必须适合在注释矩形 (由它的 Rect 条目明确说明) 中。可能尺寸一样，或更小如果需要在注释中提供额外图画区域给增加的 2D 图形。

注意：虽然 U3D 视图节点会参考视图窗口尺寸，PDF 消费者忽略 3DB 条目提供的兴趣信息。

这个视图盒没有像注释矩形那样，在相同坐标系统中明确说明，但是可能在注释的目标坐标系统中，起源于注释矩形中心。在这个坐标系统里的单元，和默认用户空间单元是一样的。因此，目标坐标系统中的注释矩形的坐标是

$$[-w/2 \ -h/2 \ w/2 \ h/2]$$

其中 w 和 h 是矩形的宽度和高度

3DD 条目明确说明了包含出现在注释中的 3D 艺术品的 3D 流；3D 流在 9.5.2 有描述。3DD 条目可以直接阐述一个 3D 流；它也可以不直接阐述一个 3D 流，凭借 3D 参考字典 (见 754 页“3D 参考字典”)。

这些帮助控制注释是否要共享相同的艺术品的运行时间段。

3DV 条目明确说明 3D 艺术品的视图，当注释活跃时显示（如下段描述）。在 9.5.3 中阐述过的 3D 视图，描绘虚拟摄像头设置，就像位置、方向和映射类型。这个 3DV 阐述的视图可能是 3D 流或另外视图字典里 VA 条目列出的 3D 字典之一。

3DA 条目是一个活跃字典（参见 9.34），决定注释和相关艺术品的 state 怎样变化。这些状态反映的事实，是它经常在延迟处理或显示 3D 艺术品，直到由于一些包括操作的原因，用户选择与它相互影响方面令人满意。

3D 注释可能是下面两种状态之一：

- 不活跃的（默认原始状态）：注释电视注释的正常画面。

注意：虽然不要求，但正常画面很可能是呈现一个 3D 艺术品默认视图的位图。生产者应该为所有有意的文档用法提供合适的分辨率位图；例如，一个高分辨率的位图用于高质量印刷和屏幕分辨率位图用于屏幕上的视图。可选的内容（参见 4.10）会被用于选择用于每一种情况的恰当的位图。

- 活跃的：注释显示 3D 艺术品参考书目。这个参考书目被注释的 3DV 条目具体说明的。

表格 9.34 3D 活跃字典条目

关键字	类型	值
A	name	<p>(可选的) 阐述注释被激活的情况。有效值是：</p> <p>P0 一旦页面包含注释打开，注释应该被激活</p> <p>PV 一旦页面某部分包含的注释可见，注释应该被激活</p> <p>XA 注释应该保持停止直到被脚本或用户行为明确激活</p> <p>注意：在任何时刻，在观察应用程序中，只有一个页面时被认为是打开，虽然依赖于页面层多个页面是可见的。</p> <p>默认值是 XA</p> <p>注意：为了操作合理性，建议为看到网页浏览器的文档使用 XA。在非交互应用程序中，如打印系统或合计应用程序，P0 和 PV 指明当页面被打印或替换时，注释应该被激活；XA 指</p>

		明注释从不活跃和应该使用正常界面
AIS	name	<p>(可选的) 阐述关于注释激活的艺术品例子的状态的名称。有效的值是：</p> <p> 艺术品是举例说明的，但实时的脚本驱动是不能活跃的。</p> <p> 如果出现实时的脚本驱动可以活跃；如果没有，这个艺术品是无例示的</p> <p>默认值是 L</p> <p>注意：在非交互应用中，艺术品总是例示的和从不活跃</p>
D	name	<p>(可选的) 描述注释不活跃下循环的名称。有效值是：</p> <p>PC 注释应该不被激活，一旦页面被关闭</p> <p>PI 注释应该不被激活，一旦包含注释的页面变成不可见</p> <p>XD 注释保持活跃直到一个脚本或用户行为明确使其无效</p> <p>注意：在任何时刻，只有一个单一的页面在观察应用程序中被认为是打开的，即使多个页面可能是可见的，这取决于页面层叠。默认值是 PI</p>
DIS	name	<p>(可选的) 在注释失活下明确说明艺术品实例的状态的一个名称。有效值是 U (非例示)，I (例示) 和 L (活跃的)。默认值是 U。</p>

活跃字典 A 和 D 条目决定 3D 注释什么时候可能变成活跃或不活跃。AIS 和 DIS 条目阐述当注释是活跃的或不活跃的时候，相关艺术品应该在什么状态。3D 艺术品会处于三种状态之一：

- **Uninstantiated:** 艺术品被以任何形式适用前的初始状态。
- **Instantiated:** 艺术品已被阅读以及艺术品的运行时间段已被创建的状态。这种状态下，它会被给出，但是 `script-driven real-time` 不能被修改。
- **Live:** 艺术品是有例示的，正在被实时修改以获得一些有效的影响

注意：活跃的状态只有在交互观察器的支持 Java Script 的应用中是有效的。

如果 3D 艺术品在被举例说明后变得没例示，然后艺术品的使用需要重新例示（例示失去了，艺术品以他的原始形式出现）。由于这个原因，没例示实际上是不需要的，除非这个艺术品通过一些方式被修改；消费者可以因为效果的而选

择保持艺术品例示不变。

注意：在非交互系统，如打印系统，艺术品不能被改变。因此，应用程序可以不同地选择使注解无效和艺术品无例示，基于的因素有记忆语法、例示艺术品所需的时间，以及 D 和 DIS 条目可能被忽略。

多种多样的 3D 注解可以共享一个 3D 艺术品的例子，如 754 页定义的“3D 参考书目字典”。这种情况下，艺术品例示的状态是以以下方式决定的：

- 如果任一注解要求(通过它的行为字典)艺术品应该是 live, 它就是 live 的。
- 否则，如果任一注解要求艺术品应该被例示，它就是例示
- 否则，这个艺术品是无例示的

注意：如果任一注解参考了艺术品的行为，它必须不是无例示的。但艺术品可能是例示的或 live，如果所有参考了它的注解是不活跃的。

9.5.2、3D 流

这种 3D 艺术品规范在 3D 流中有包含。3D 流字典，在表格 9.35 列出它的全部条目，可能提供艺术品的前视图集，和默认视图一样。它可以提供脚本和资料，提供定制的行为或效果。

表格 9.35 3D 流字典增加的条目

关键字	类型	值
Type	name	(可选的) 字典描述的 PDF 对象的类型；如果出现，必须是适合于 3D 流的 3D
Subtype	name	(必需的) 阐述包含在流的 3D 的格式的名称。在 PDF1.6，唯一有效值是 U3D。
VA	array	(可选的) 一个 3D 视图字典的矩阵，每一个都阐述 3D 艺术品已命名的、预先设置的视图（参见 9.5.3 “3D 视图”）
DV	name tree	(可选的) 阐述 3D 艺术品默认视图的一个对象。它可以是 3D 视图字典（参见“3D 视图”）或下列类型之一： <ul style="list-style-type: none">● 一个阐述索引到 VA 矩阵的整数

		<ul style="list-style-type: none"> ● VA 矩阵中，匹配某一视图 IN 条目的文本字符串 ● VA 矩阵中，指明前或后条目的名称。如果 VA 出现，默认值是 0 (VA 矩阵的第一个条目)；否则，默认视图是 3D 流自己阐述的视图
Resources	(various)	(可选的) 为对象标出名称串的名称树，这个用于应用程序或修改 3D 艺术品默认视图的脚本这个名称树种的名称必须是文本串，所以他们可以从 JavaScript 被承认。
OnInstantiate	stream	(可选的) 一个 JavaScript 脚本，当 3D 流被读来创建一个艺术品例示的时候应该被执行

子类型条目明确说明了 3D 数据流的格式。在 PDF1.6 中，只有一个数值时有定义的，U3D，它指示了数据流和 U3D 文件格式规范是一致的。PDF 消费者应用应该有所准备，当子集遇到未知数值，并且适当恢复，通常意味着将注解放在它的不活跃状态，显示他的正常的界面。

注意：应用被鼓励跟从一条退回到正常界面的途径，凭借关于其他可能在将来版本 PDF 而不仅仅现在这种规范的，使用不同的类型或值字典条目。

VA 条目时一个矩阵，包含一个命名现有 3D 艺术品视图的列表。矩阵中每一条目是一个 3D 视图字典（参见 9.5.3 “3D 视图”），包含视图和显示视图所需信息的名称。矩阵元素的顺序是在用户界面显示的视图的顺序。DV 条目阐述用作 3D 艺术品初始视图的视图。

注意：默认视图可以在下面的优先顺序被具体说明：在注解字典中，在 3D 流字典中，或在含有 3D 流的 3D 艺术品。

3D 流包含可以被应用程序和做动画和其他计划性定义的行为的脚本信息使用，从改变方向视图来移动艺术品个别部分。OnInstantiate 条目阐述一个 JavaScript，应该被支持 JavaScript 的应用程序执行，当 3D 被阅读后创建一个 3D 艺术品的例子。这个特殊的 3DJavaScript 外观在 Acrobat JavaScript Scripting Reference 中有描述（参见参考书目）。

3D 参考书目字典

可能多于一个 3D 注解与同一个 3D 艺术品相关联。例如几个注解可能列出相

同对象的不同视图。这种注解会出现两种方法，在注解的 3DD 条目中有描述（参见 9.33）。

- 如果 3DD 条目明确指定一个 3D 流，这个注解有它自己的运行时间 3D 艺术品例子。艺术品的任何变化都只被反映到这个注解。其他参考相同流的注解有分开的运行时间例子。
- 如果 3DD 条目明确指定一个 3D 参考书目字典（它的条目在表格 9.36 中列出），这个注解共享运行时间 3D 艺术品例子，和所有其他明确指定这个相同参考书目字典的注解。艺术品的任何变化都反映在所有这样的注解中。

表格 9.36 3D 参考书目字典的条目		
关键字	类型	值
Type	name	(可选的) 字典描述的 PDF 对象的类型；如果出现，必须是适合于 3D 参考书目字典的 3Dref
3DD	stream	(必需的) 包含 3D 艺术品规范的 3D 流(参见 9.5.2“3D 流”)

例子 9.1 和图 9.1 通过图 9.3 列出用相同 3D 艺术品的 3 种注释。对象 100 (注解 1) 有它自己的运行时间 3D 流例子 (对象 200)；对象 101 (注解 2) 和对象 102 (注解 3) 共享一个运行时间例子，通过 3D 参考书目字典 (对象 201)。

例子 9.2

```

100 0 obj % 3D annotation 1
<< /Type /Annot
/Subtype /3D
/3DD 200 0 R % Reference to the 3D stream containing the 3D artwork
>>
endobj

101 0 obj % 3D annotation 2
<< /Type /Annot
/Subtype /3D
/3DD 201 0 R % Reference to a 3D reference dictionary
>>

```

```
endobj
102 0 obj % 3D annotation 3
<< /Type /Annot
/Subtype /3D
/3DD 201 0 R % Reference to the same 3D reference dictionary
>>
endobj
200 0 obj % The 3D stream
<< /Type /3D
/Subtype /U3D
... other keys related to a stream, such as /Length
>>
stream
... U3D data...
endstream
endobj
201 0 obj % 3D reference dictionary
<< /Type /3DRef
/3DD 200 0 R % Reference to the actual 3D artwork.
>>
Endobj
```

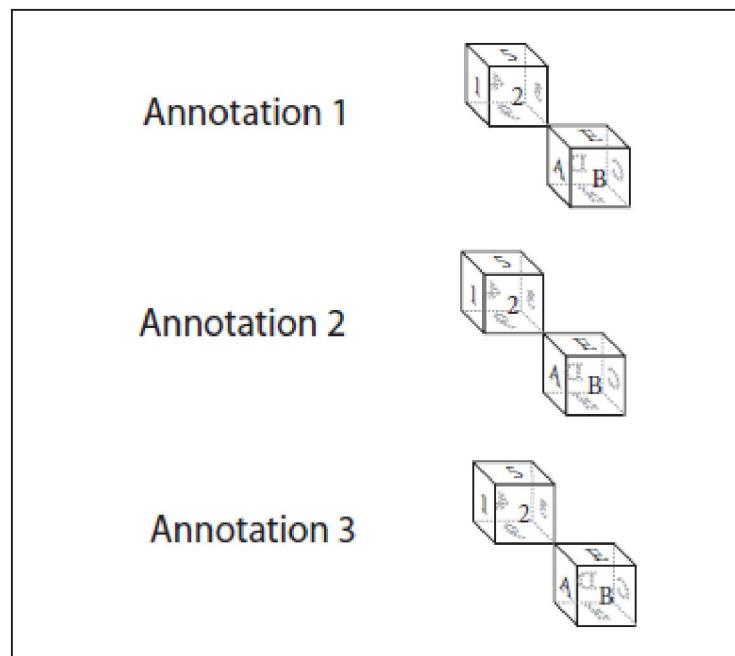


图 9.1 默认艺术品视图

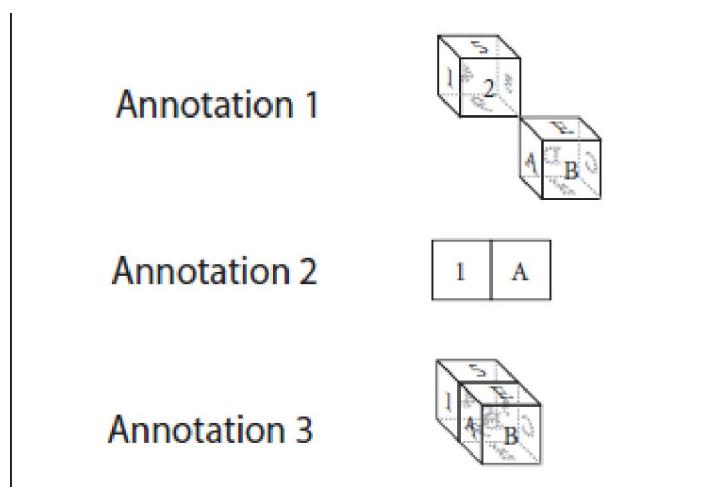


图 9.2 注释 2 被旋转

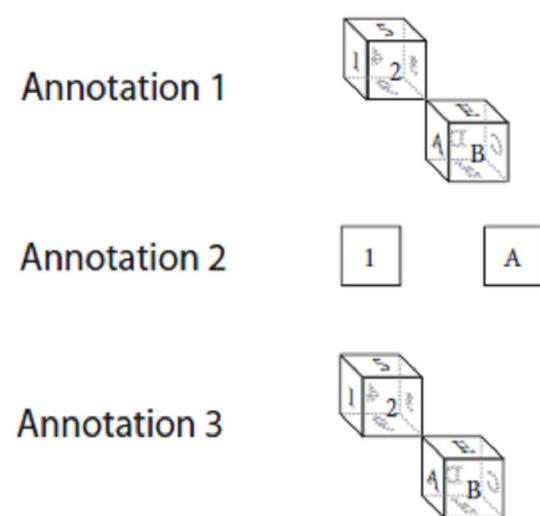


图 9.3 共享艺术品（注释 2 和 3）已修改

这些图表表明了 9.1 例子中的对象怎样会被用到。图 9.1 表明了三个注释中艺术品的相同初始视图。图 9.2 表明注释 2 中艺术品旋转视图的结果。图 9.3 表明了操控被注释 2 和注释 3 共享的艺术品结果：它们都反映艺术品的变化因为它们共享一个相同运行时间例子。注释 1 剩余部分不变，因为它有它自己运行时间例子。

注意：当多个注释参考相同 3D 艺术品例子，这个例子的状态被如 9.5.1 “3D 注释” 描述的决定。

9.5.3、3D 视图

一个 3D 视图（或简单视图）具体说明了应用到与 3D 注释虚拟摄像头相关的参数。这些参数可能包含在摄像头的方向或位置中，细节关于摄像头坐标进注释的目标坐标系统，和关于艺术品上的背景描述。

用户可以控制视图，通过演示互动业务，如自由循环和翻译。另外，3D 艺术品可以包含先前定义作者认为特别兴趣的视图集。例如，一个部分的一个机械图画，可能有描述表明对象的上面、下面、左边、右边、前面和后面的视图。

一个 3D 流会包含 3D 艺术品的已命名边框形式视图的列表，如是 3D 视图字典矩阵的 VA 条目明确说明的。这些 3D 视图字典中的条目在表格 9.37 中列出。

表格 9.37 3D 视图字典条目

关键字	类型	值
Type	name	(可选的) 字典描述的 PDF 对象的类型；如果出现，必须是适合于 3D 流的 3D
IN	text string	(必需的) 视图的额外名称，使用于再一个用户界面上显示
MS	name	(可选的) 视图的内部名称，用于从其他对象指向视图，例如转到三维视图操作（参见 632 页“转到三维视图操作”）
C2W	array	(如果 MS 的参数值是 U3D 是必需的，其他忽略) 一个 12 元素 3D 转换矩阵，在全局坐标系中说明摄像头位置和方向。

U3DPath	string or array	(如果 MS 的参数值是 U3D 是必需的，其他忽略) 一个或多个字符串系列用于接受 U3D 艺术品中的一个视图节点。这第一个字符是适合于根节点的 U3D 节点 ID，每一个随后的字符是适合于被前字符描述的字节点的节点 ID。每一个节点说明一个 3D 转换矩阵 (参见 9.5.4 “适合于 3D 注释的坐标系统”); 这一连串所有矩阵形成摄像头到全局矩阵。
C0	number	(可选; 只用于出现 MS) 一个非负数字指明从摄像头坐标系统沿着 z 坐标轴到视图轨道中心的距离; 参见下面的讨论。如果这个条目不出现，视图应用程序必须决定轨道中心
P	number	(可选的) 映射字典 (参见 760 页 “映射字典”), 定义了 3D 艺术品 (已经转换到摄像头坐标) 坐标到坐标注释的系统目标的映射。 默认值是一个 Subtype 的参数值是 Perspective 的映射字典, FOV 的值是 90, 所有其他条目使用它们的默认值。
O	stream	(可选的; 只有当 MS 和 P 出现时有意义) 一种 Xobject 形式, 用于在参考的 3D 艺术品的上面覆盖 2D 图形
BG	dictionary	(可选的) 一个背景字典 (参见 764 页的 “背景字典”), 定义关于哪个要绘制 3D 艺术品的背景。默认值是条目使用默认是的背景字典

对于任何的视图，文档作者可能提供 2D 内容，明确说明视图要被放在 3D 艺术品的上面。这个 O 条目明确说明了层叠在已提供的 3D 艺术品的 Xobject 格式。Xobject 格式的坐标系统定义成和摄像头坐标系统的 (x, y, 0) 坐标一样。(参见 9.5.4 “3D 注解坐标系统”)

O 条目明确说明的 Xobject 格式是服从于下面限制的：不能容忍和参考的 3D 图形层叠却不对准的结果：

- Xobject 格式和具体指定的视图时有关系的 (不是通过 3D 视图字典定义的摄像头位置)。当用户操纵使用 3D 视图时，而不是当用户通过指南恰好操纵相同方向的时候，它应该被取出。
- 如果这个面世艺术品模型已经被改变，它应该被取出

- 可能只在 3D 视图字典中有明确说明的，这个 3D 视图字典的面世摄像头模型（MS 和相关条目）和映射字典（P 条目）存在
CO 条目为 3D 视图阐述了从摄像头到轨道中心的距离，当演示轨道式导航的时候摄像头应该被旋转。图 9.4 指明了当沿着轨道中心轨道运行的时候，摄像头位置。

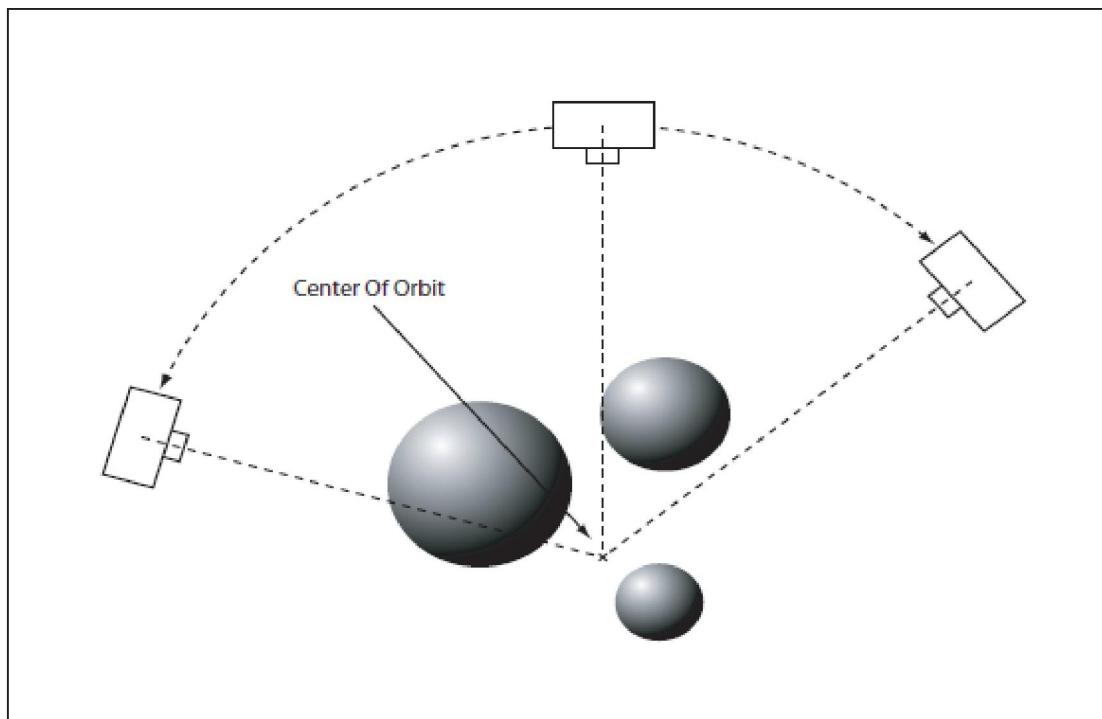


图 9.4 沿着轨道中心运行的轨道

映射字典

映射字典（参见表格 9.38）定义了 3D 摄像头坐标在注释的目标坐标系统上面。每一个 3D 视图可以明确指定一个映射字典，通过 P 条目的途径。

注意：虽然 U3D 视图节点能明确指定映射字典，PDF 阅读者会因为映射字典感兴趣的内容而忽略这点。

PDF1.6 支持 near/far clipping。这种片段类型定义了一个 near plane 和一个 far plane (762 页图 9.5 所示)。对象或对象的一些部分，超越 far plane 或比 near plane 近的摄像头，是不被取出的。3D 对象被映射到 near plane 上，然后用注释的如下描述的目标坐标系统上升、定位。

表格 9.38 映射字典条目

关键字	类型	值
Subtype	name	(必需的) 映射的类型。有效值是 0 (正交) 或 P (透视)

CS	name	(可选的) 片段类型。有效值是 XNF (明确近或远) 或 ANF (自动近或远)。默认值是 ANF
F	number	(可选的; 只当 CS 值是 XNF 时有意义) 远片段间隔, 在摄像头坐标系统中有表示。Z 坐标轴比这个条目的值大的对象没有被绘制部分。如果这个条目出现, 没有远片段出现。
N	number	(只当 CS 值是 XNF 时有意义; 如果要求的 Subtype 的值是 P) 近片段间隔, 在摄像头坐标系统中有表示。Z 坐标轴比这个条目的值大的对象没有被绘制部分。如果 Subtype 是 P, 这个值必须是正数; 如果 Subtype 值是 0, 这个值必须是非负, 默认值是 0
FOV	number	(如果 subtype 是 P, 则必需的; 其他忽略) 在 0-180 之间的一个数字, 包括一切阐述虚拟摄像头视图领域的度数。它在 3D 空间定义了一个以 z 坐标轴为中心, 和一个圆作为底部的圆锥体。这个圆, 沿着 PS 的值, 当在注释的 2D 平面给定的时候, 明确说明了映射艺术品的模型。
PS	number or name	(可选的; 只当 subtype 是 P 时有意义的) 一个明确说明当映射 3D 艺术品到注释的目的坐标系统时使用的刻度。它定义了由近平面和 FOV 阐述的圆锥体组成的圆的直径。这个值可能是下列之一: <ul style="list-style-type: none"> ● 一个正数, 明确说明直径作为注释目标坐标系统的距离。 ● 一个名称, 明确说明直径应该设置宽度 (W), 高度 (H), 注释 3D 视图盒的宽度和高度的最小值 (Min), 或注释 3D 视图盒的宽度和高度的最大值 (Max)。 默认值是 W
OS	number	(可选的; 只当 Substyle 是 0 才有意义) 一个正数, 当映射到注释目标坐标系统撒谎能够时, 明确说明刻度因素以应用到 x 和 y 坐标。(z 坐标已被声明)。默认值是 1

CS 条目定义了近的和远的平面怎样被确定的。XNF 的值意味了 N 和 F 条目分别地、明确地阐述了近和远平面的 z 坐标。CS 的 ANF 值意味了近和远平面是自动基于艺术品对象确定的。

Subtype 条目明确说明映射类型，用来确定对戏那个怎样映射到近平面上以及上升的。这些可能的值是 0 的 **orthographic projection** 和 P 的 **perspective projection**。

对于正交映射，一个给定的坐标 (x, y, z) 是映射在近平面上的，定义了一个 2D 坐标 (x_1, y_1)，使用了下面公式：

$$x_1 = x \times \frac{n}{z}$$

$$y_1 = y \times \frac{n}{z}$$

n 是近平面的 z 坐标。

通过视角映射上升，是比正交映射更复杂的。**FOV** 条目明确说明一个度数，定义了一个把摄像头坐标系统的 z 坐标轴中心的圆锥体。(参见图 9.5)。这个圆锥体和近平面交叉，在近平面上方形成一个圆区域。图 9.6 表明了这个圆和从摄像头位置来的图形。

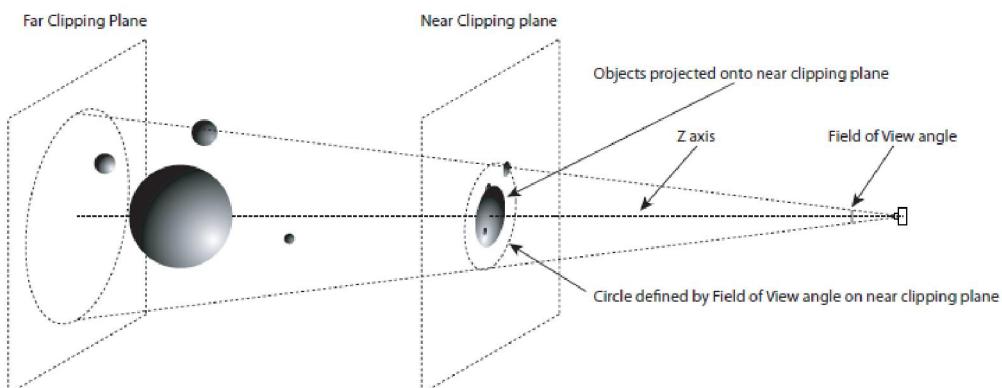


图 9.5 3D 映射在近平面上方的 3D 艺术品视角

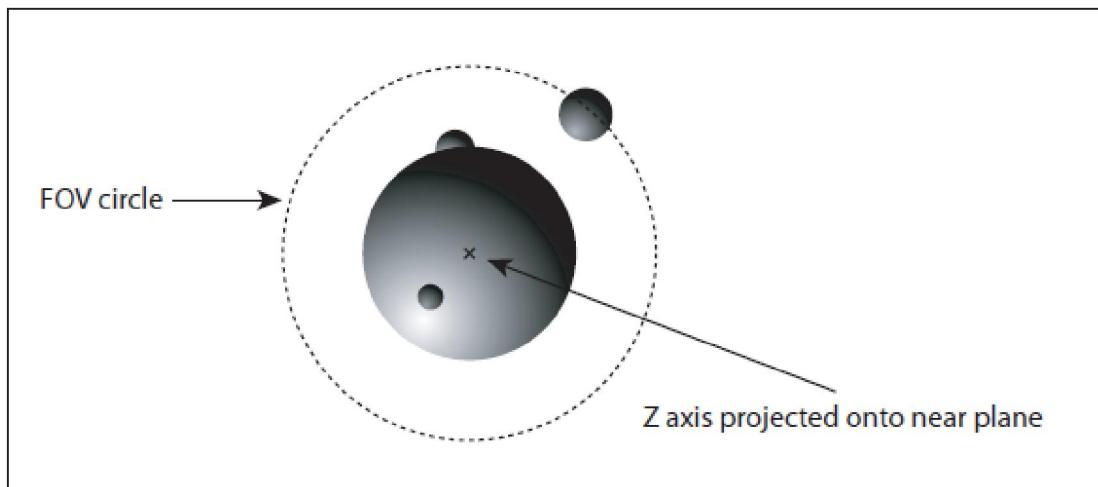


图 9.6 映射在进 clipping 平面上的对象，像从摄像头上看到的
PS 条目明确说明了当这个图形映射在近平面上的时候，这个圆会有的直径，
是参考了注释 3D 视图盒（参见图 9.7）。虽然这个圆的直径决定了模型的要素，
圆以外提供适合视图盒的图形也被显示，像在图中看到的。

图 9.8 表明了完整的 3D 注释。这种情况下，3D 视图盒是比注释的矩形小，
这个矩形也包含 3D 视图盒外的 2D 内容。

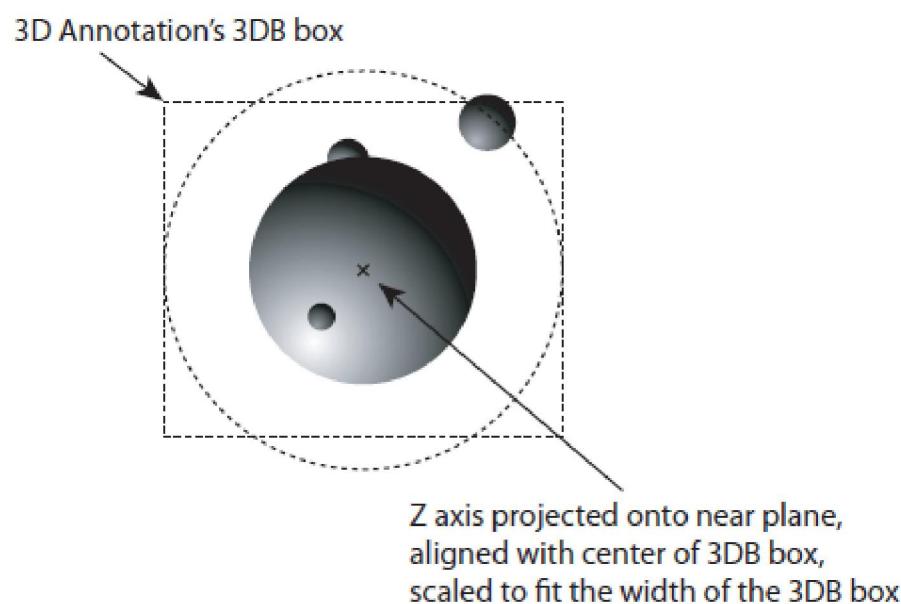


图 9.7 在注释 3D 视图盒上的近平面位置和模型

The Daily News

New Planet Discovered!

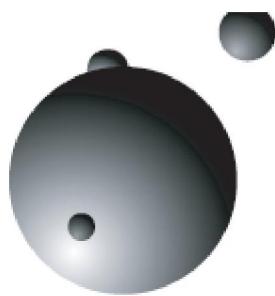


图 9.8 页上 3D 注释定位

背景字典

3D 背景字典定义了背景，通过绘制 3D 视图；背景字典中这些条目在表格 9.39 中列出。PDF1.6 只支持单一不透明颜色，使用 DeviceRGB 颜色空间。3D 艺术品可以包含透明的对象；但是，这些对象和注释下绘制的对象之间没有相互影响。实际上，3D 艺术品和它的背景形成一个透明体，平滑的结果会有 1 的不透明（参见第 7 章“透明度”）。

注意：当画面想要描述 3D 艺术品时，一个注释正常显示应该有相同涉及透明度行为。当画面用于其他目的，如作为不矛盾的警告信息时，这个建议不是必需应用的。

表格 9.39 3D 背景字典条目

关键字	类型	值
Type	name	(可选的) 这个字典描述的 PDF 对象的类型；如果出现，必须是适合于 3D 背景字典的 3DBG。
subtype	name	(可选的) 背景的类型。适合于 PDF1.6 唯一的有效值是 SC (纯色)，指示单一不透明的颜色。默认值是 SC
CS	name or array	(可选的) 背景的颜色空间。适合于 PDF1.6 唯一有效值是名称 DeviceRGB。默认值是 DeviceRGB 注意：PDF 阅读者应该准备好遇到其他可能被以后的版本

		所支持的值。
C	(various)	(可选的) 背景的颜色, 这个颜色在 CS 定义的颜色空间中。默认值是: 一个矩阵[1 1 1]表示白色, 当这个 CS 的值是 DeviceRGB
EA	boolean	(可选的) 如果是 true, 背景应该应用于全部的注释; 如果是 false, 背景应该只应用于在注释 3D 视图盒 (参见表格 9.33 的条目 3DB) 描述的矩形, 默认值是 false。

9.5.4、3D 注释坐标系统

3D 艺术品是对象的集合, 位置和几何是用三维坐标明确指定的。4.2 “坐标系统”, 谈论了二维坐标系统的概念, 它们的几何和转换。这部分扩展了那些概念, 包括第三个坐标轴。

如 4.2 部分描述的, 位置是按照笛卡尔平面 x 和 y 坐标对定义的。这个平面的原点阐述了 (0, 0) 位置; x 值向右增加和 y 值向上增加。对于三维图形, 第三个坐标轴 z 坐标轴, 这个 z 坐标轴是必需的。原点是在 (0, 0, 0); 正的 z 值向页里面增加。

在二维图形中, 转换矩阵可以变化在平面上的位置, 大小和方向。它是 3*3 矩阵, 只有 6 个元素可以被改变; 因此, 这个矩阵在 PDF 表示为一个 6 数字矩阵:

$$\begin{bmatrix} a & b & 0 \\ c & d & 0 \\ tx & ty & 1 \end{bmatrix} = [a \ b \ c \ d \ tx \ ty]$$

在 3D 图形中, 一个 4*4 矩阵是用来在三维坐标系统中, 转换对象的位置、大小和方向的。只有矩阵的前三列可以被改变; 因此, 这个矩阵在 PDF 表示为 12 个数字的矩阵:

$$\begin{bmatrix} a & b & c & 0 \\ d & e & f & 0 \\ g & h & i & 0 \\ tx & ty & tz & 1 \end{bmatrix} = [a \ b \ c \ d \ e \ f \ g \ h \ i \ tx \ ty \ tz]$$

3D 坐标转换表示为转换矩阵:

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \times \begin{bmatrix} a & b & c & 0 \\ d & e & f & 0 \\ g & h & i & 0 \\ tx & ty & tz & 1 \end{bmatrix}$$

运算这个乘法得到下列结果：

$$x' = a \times x + d \times y + g \times z + tx$$

$$y' = b \times x + e \times y + h \times z + ty$$

$$z' = c \times x + f \times y + i \times z + tz$$

3D 艺术品的位置和方向典型地涉及到转换（运动）和沿坐标轴旋转。这个虚拟摄像头阐述了艺术品的视图。摄像头和艺术品的关系可以看作两种途径：

- 3D 艺术品是固定在位置和方向中，摄像头可以移动到不同位置和方向
- 摄像头是固定位置的，3D 艺术品是转换和旋转的

两种途径都可以获得相同的视觉效果；实际上，3D 系统典型地使用两种结合。概念地，有三个限制坐标系统：

- 艺术品坐标系统。
- 摄像头坐标系统， z 坐标定位在 $(0, 0, 0)$ ，面向 z 坐标正方向， x 坐标向右和 y 坐标向上。
- 一个中间的系统叫做全局坐标系统。

两个 3D 转换矩阵在等价变换中使用：

- 面世艺术品矩阵在全局坐标系统中，明确说明了艺术品位置和方向。这个矩阵包含在 3D 流中。
- 面世摄像头矩阵在全局坐标系统中，明确说明了摄像头位置和方向。这个矩阵在 3D 视图字典条目 C2W 或 U3Dpath 中有阐述。

当在 3D 注释目标坐标系统中绘制 3D 艺术品时，下面转换会被使用：

1、艺术品坐标转换为全局坐标：

$$\begin{bmatrix} x_w & y_w & z_w & 1 \end{bmatrix} = \begin{bmatrix} x_a & y_a & z_a & 1 \end{bmatrix} \times aw$$

2、全局坐标转换到摄像头坐标：

$$\begin{bmatrix} x_c & y_c & z_c & 1 \end{bmatrix} = \begin{bmatrix} x_w & y_w & z_w & 1 \end{bmatrix} \times (cw^{-1})$$

前两步可以表示为一个单一的等式，如下：

$$\begin{bmatrix} x_c & y_c & z_c & 1 \end{bmatrix} = \begin{bmatrix} x_w & y_w & z_w & 1 \end{bmatrix} \times (aw \times cw^{-1})$$

3、最后，这个摄像头坐标被映射到二维中，消除了 z 坐标，然后再注释目的坐标系统中上升和定位。

第 10 章 文件交互

在这一章中描述的功能不影响文档的最终外观，相反，这些特性可以帮助文件，包括更高级别的信息，这些对应用程序之间文件的交互非常有用：

- 程序设置（10.1 节），定义 PDF 运算符的实施规则。
 - 元数据（10.2 节）由文件或文档部分的一般信息组成，如标题，作者，以及创建和修改日期。
 - 文件标识符（10.3 节）提供从一个 PDF 文件到另一个可靠的参考。
- Page-piece 词典**（10.4 节），允许应用程序在一个 PDF 文档资料中嵌入私人数据，为自己使用。
- 标明内容的运算符（10.5 节）确定部分内容流，使他们与额外的属性或指定外部对象相关联。

- 逻辑结构设施 (10.6 节) 实行分层次组织结构在文档内容上。
- 标记的 PDF (10.7 节), 使用公约标记的内容设置、逻辑结构和设施, 以方便提取和重用文档的内容作其他用途。
- 增加对残疾人用户的一份多种无障碍方式文件 (第 10.8 节), 包括自然语言识别, 它是被写 (如英语或西班牙语) 为文本到语音引擎的利好。
- Web 捕捉插件扩展 (10.9 节), 它从基于因特网或局部本地的 HTML, PDF, GIF, JPEG 和 ASCII 文本创建 PDF 文件。
- 设施配套印前生产流程 (10.10 节), 如
页边界规范和打印机的标志, 颜色分离, 输出意向, 携带物, 低分辨率代理
和高分辨率图像

10.1、程序集

用于内容流的 PDF 操作者被归类为叫做程序集的相关类别 (见表 10.1)。每个程序集对应到命名资源中包含在该程序的操作的实现设置。在内容流的资源字典 **ProcSet** 项 (见第 3.7.2, “资源词典”) 持有该数组的名称组成 使用该程序集的内容流。这些程序集是只用在 当内容流是打印到 PostScript 的输出设备。这个名字确定的 PostScript 程序集必须发送到该设备去解释内容流经营中的 PDF 操作者。这个数组的每个元素必须是表 10.1 所示的一个 预定义的名称。(见附录 H 的执行条例 153)

表 10.1 预定义的程序集

名字	OPERATORS 的类别
PDF	格式绘画与图形状态
文本	文字
ImageB	灰度图像或图像掩饰
ImageC	彩色图像
ImageI	索引 (颜色表) 图像

注: 从 PDF 1.4 开始, 这个功能被认为已经过时。出于与现有的消费类应用的兼容, PDF 格式制作的申请应继续指定 **procedure sets** (最好是在表 10.1 中

所列的一切，除非它是已知的需要较少的)。然而，消费者应用程序不应依赖于这些信息的正确性。

10.2、元数据

一个 PDF 文件可能包括的一般信息，诸如文件的标题，作者，创作和修改日期。这种文件的一般的信息(而不是其内容或结构)被称为元数据，有意协助编目和从外部文件中检索文件。文档的元数据还可以被用户或插件程序扩展添加或更改(见附录 H 实施条例 154)。从 PDF1.4 节开始，元数据也被认为是一个文档的独立组件。

元数据可以按以下方式存储在一个 PDF 文件中：

- 在一个与文档相关的文档信息目录(10.2.1 节)
- 在一个与文档或文档组件(10.2.2 节)相关的元数据流(PDF1.4)

10.2.1 文档信息词典

进入一个 PDF 文件拖车项里可选的信息(见第 3.4.4 节，“文件拖车”)

可容纳一个包含了文件元数据的文件信息字典；表 10.2 显示其内容。任何进入的未知的值应该从字典中被删除，而不是用一个空字符串被包含作为其值。

一些插件程序扩展可能选择允许搜索文件信息字典的内容。为了方便浏览和编辑，所有在该词典中的关键字是全拼的，而不是缩写。新的关键字应小心选取以便对用户有意义。

表 10.2 中没有具体提到任何与关键字相关的值必须是一个文本字符串。

注：虽然消费者应用程序可以在文件信息字典中存储文档的自定义元数据，但存储私人信息或结构信息在那里是不合适的。这些资料应保存在文档目录代替(见第 3.6.1 节，“文件目录”)。

表 10.2 文档信息字典之记入

关键字	类型	值
-----	----	---

标题	文本字符串	(可选; PDF1.1) 文件的标题
作者	文本字符串	(可选) 创建文档人的姓名
主题	文本字符串	(可选; PDF1.1) 文件的主题
关键词	文本字符串	(可选; PDF1.1) 与文件相关的关键字
创建者	文本字符串	(可选) 如果文件是从另一种格式转换到 PDF 格式的, 创建的原始文件的程序的名字 (例如, Adobe 公司的 FrameMaker)
转化者	文本字符串	(可选) 如果文件是从另一种格式转换到 PDF 格式的, 转化到 PDF 格式的应用程序 (例如, Acrobat Distiller)
创建日期	日期	(可选) 文档创建的日期和时间, 人类可读的形式 (见 3.8.3 节, “日期”)。
修改日期	日期	(如果 PieceInfo 在本文件目录, 则必填, 否则可选的, PDF 1.1) 文档最近被修改的日期和时间, 以人类 可读的形式 (见 3.8.3 节, “日期”)。
捕捉者	名称	(可选; PDF1.3) 一个名称指明文档是否被修改为包含 捕捉信息, (见第 10.10.5 节, 捕获支持 “)
True	该文件已全部被捕捉, 没有进一步的捕捉 需要。(这是名子 True, 而不是布尔值 true。)	
		False 文件尚未被捕获; 任何期望诱捕仍必须完成。 (这是名子 False, 而不是布尔值 false。)
		Unknown 它要么是不清楚的文件已被捕获或部分 被捕获, 但尚未完全被捕获, 部分附加 的捕捉可能仍然是必要的。
		默认值: Unknown。
		入口值可以被软件创建文档捕获信息自动设置, 或者 它被知道到人的操作或手动进入。

例 10.1 显示了一个典型的文档信息字典。

例 10.1

```
1 0 obj
<</标题 (PostScript 语言参考, 第三版)
/作者 (Adobe 系统公司)
/创建者 (Macintosh®的 Adobe ® FrameMaker® 5.5.3)
/转化者 (Acrobat®Distiller®3.01 for Power Macintosh)
/ 创建日期 (D:19970915110347-08'00')
/ 修改日期 (D:19990209153925-08'00')
>>
endobj
```

10.2.2 元数据流

元数据，为整个文件和文件的组件，可以存储在 PDF 流中的称为元数据流 (PDF1.4) 流。元数据流在文档信息字典中有以下优点：

- 基于 PDF 的工作流程往往在较大的文件中作为组件嵌入到定向元数据艺术品，元数据流为了审查下游提供了一个标准的方式维护这些组件元数据。
PDF-aware。应用程序应该能够从 PDF 文档自身获得定向元数据清单。

PDF 文件通常是在网上或其他环境中使用，这些地方有许多工具例行检查，遍目录，分类文件。这些工具应该能够理解文件的自我包含的说明，即使他们不懂 PDF。

除了常见的一般的流字典的条目(见 38 页表 3.4)，元数据流字典包含表 10.3 中列出的其他条目。

一个元数据流的内容是元数据而不是标记语言 (XML)。此信息作为到工具的纯文本，不是 PDF 格式的，是可见的，只有当元数据流是未过滤和未加密的。

表 10.3 元数据流字典中的其他条目

关键点	类型	值
Type (类型)	名字	(必须的) 字典所描述的 PDF 对象的类型，必须是 为了元数据流的元数据
Subtype (子类型)	名字	(必须的) 字典说描述的元数据的类型，必须是 XML

代表元数据的 XML 格式被定义为叫做可扩展元数据平台 (XMP) 框架的一部分并在 Adobe 文件 XMP 中描绘 (见参考书目)。这种框架提供了一种使用 XML 表示元数据描述文件和他的组件的方法，并拟被更广泛的应用类采用，而不仅仅是那些过程 PDF。它包含一种方在非 XML 数据文件中嵌入 XML 的数据的方法，在一个平台无关的格式中，通过简单的扫描，可以进行定位和存取，而不需要对文件进行解析。

元数据流可以通过文件目录中的元数据条目附加到文档中 (见第 3.6.1 节“文件目录”，并见附录 H 实施条例 155)。此外，大多数的 PDF 文档组件表示为一个流或字典可以有一个元数据条目 (见表 10.4)。

表 10.4 元数据组件的附加条目		
关键点	类型	值
Metadata (元数据)	流	(可选的; PDF1.4) 元数据流包含组件的元数据

一般来说，一个 PDF 流或元数据字典可以连接到它的元数据，只要 流或字典代表了实际的信息资源，而不是一个执行神器。有些 PDF 格式构造是考虑推行工作上，因此不能有关联的元数据。

至于其余的 PDF 格式构造，有时关于正确的其中的流或元数据字典应承担元数据条目。这种情况将被解决，以便元数据尽可能的紧密连接那些存储所描绘的实际数据对象。例如，元数据描述平铺模式是值得重视的模式流的字典，但阴影应该有附上阴影字典的元数据，而不是涉及到它的阴影样式字典。

同样，元数据描述的基于 ICC 的色彩空间应该附加到 ICC 概况流中描述它，字体元数据应该附加到字体文件流而不是字体词典上。

在这本书中描述文档组件的表中，元数据项只有在那些最有可能被使用的才可能被例举。请记住，然而，这个项目可能会出现在其他组件表示为流或字典。

此外，元数据也可以与内容流中相关的标记内容相关联。这种结合的被创建包括在列表中的项其关键是元数据字典，其价值是元数据流字典。由于这个结构参考了外部的内容流，属性列表必须被称为间接作为命名资源 (见第 10.5.1 节，“属性列表”)。

10.3、文件标识符

PDF 格式的文件可能包含其它 PDF 文件的引用(见 3.10 节“文件规范”)。简单地存储一个文件名，然而，即使在一个平台无关的格式，并不能保证该文件可以被发现。即使文件仍然存在，它的名字并未改变，不同的服务器软件应用程序可能以不同的方式来识别。例如，服务器运行在 DOS 平台上，必须转换所有的文件名为 8 个字符和一个 3 个字符扩展。不同的服务器可以使用不同策略转换长文件名为这种格式。

外部文件引用可以通过一个文件标识符变得更可靠，(PDF1.1)，用它增加正常的平台为基础的文件称号。同一个文件中匹配文件中引用标识符 确认是否正确的文件被找到。

文件标识符的定义是通过在 PDF 文件的拖车目录中可选的 ID 条目。(见第 3.4.4 节，“文件拖车”，又见附录 H 的实施条例 156)。此项的值是两个字符串排列。第一个字符串是基于文件内容的永久标识符，此时，它是原创的，当文件不断更新时，它并不会改变。第二个字符串是一个基于文件内容的不断变化的标识符，此时，该文件的内容是最近更新的。当一个文件第一次被写时，这两个标识符被设置为相同的值。

如果两个标识符匹配，当文件的引用得到解决，这很可能是正确的文件已被找到。如果仅有第一个标识符匹配，那么不同的版本正确的文件已被找到。

为了确保文件标识符的独特性，建议他们通过消息摘要算法计算如 MD5 (在 Internet 描述的 RFC 1321 中，MD5 消息摘要算法，见参考书目)，使用以下信息 (见附录 H 实施条例 157)：

- 当前时间
- 表示该文件的位置的字符串，通常是路径名
- 该文件的字节大小
- 在文件中的所有文件资料字典项的值 (见第 10.2.1，“文件信息词典”)

10.4、页片 (Page-Piece) 词典

一个页片词典 (PDF1.3) 可以用来保持私人应用程序数据。这些数据可以通过页面对象中的 PieceInfo 入口 (见页 119 表 3.27) 或形式字典 (见页 328 表 4.45) 与一个页面或 Xobject 相关联。从 PDF 1.4 开始，私人数据可能也通过文件目录中的 PieceInfo 入口与 PDF 文档相关联。(见第 114 页表 3.25)。

应用程序可以使用这本字典作为一个地方存储连接该文件，页，结构的私人数据。这种私人数据可以传递信息，这对产生它的应用程序是有意义的，（例如图形编辑器的对象分组上的信息或 Adobe Photoshop®的层信息），但经常被通用 PDF 浏览器应用程序忽略。

如表 10.5 所示，页面字典可能包含任何项目的数量，每次由一个独特的应用程序或一个被一个应用程序家族识别的众所周知的数据类型的名称标明为关键字。与每个键相关联的值是一个应用程序字典包含的由应用程序使用的私人数据。私人项可以是任何数据类型的值，但通常它是一个字典包含由应用程序所需要的所有私人数据，实际的文件的内容，页，或形式以外的。

表 10.5 页面词典条目

关键字	类型	值
任何应用程序名字 或知名数据类型	词典 (dictionary)	应用程序数据词典 (表 10.6)

表 10.6 应用程序词典条目

关键字	类型	值
最近修改	日期	(必须的) 最近一次由应用程序修改的文件的内容，页，形式的时间
私人数据	任意	(可选的) 任何与应用程序相关的私人数据，典型的是在一本字典中的形式

该 **LastModified** 条目表示该应用程序最后修改了页或表格的内容。如果页面 (page-piece) 字典包含几个应用程序数据字典，其修改日期可与那些页面对象或表格词典的相一致的目录比较 (见页 119 表 3.27，页 328 表 4.45)，或文件信息字典的 **ModDate** 项 (见表 10.2)，以确定哪个应用程序数据字典与内容或形式的当前页面一致。因为某些平台可能只是使用日期和时间近似值或没有正确处理不同时区，修改日期是比较平等，只没有顺序排序。

注：两个或多个应用数据字典可能有相同的修改日期是很可能的。应用程序可以使用此功能来定义相同数据格式的多个或延伸的版本。例如，假设一个早期版本应用程序使用的应用程序数据字典叫做 **PictureEdit**，同一个应用程序后来

版本延伸了先前没有用过的附加条目。原始数据可以继续保持在 `PictureEdit` 字典中，附加条目可以放置在一个新的字典名为 `PictureEditExtended` 的词典中。这使得应用程序的早期版本可以象以前那样继续工作，更高版本可以找到并使用扩展的数据项。

10.5、标记目录

标记目录操作符识别一部分 PDF 目录流作为特殊应用或者扩展插件的目录元素。标记目录的元素和操作符把目录分成两类：

`MP` 和 `DP` 操作符把单个标记目录点定义在目录流内。

`BMC`, `BDC` 和 `EMC` 操作符把对象的标记目录句子囊括在目录流内。值得一提是这里的句子不是目录流中简单的字节，而是完整的图像对象。每个对象在图形领域呈现出来的参数都是合格的。

例如，一个图形应用程序可以用标记目录来识别的一组相关的对象组成一个整体作为要处理的一个单元。一个文本处理应用用它使脚注标记文档的主体和相应的脚注文本的页尾保持关联。PDF 逻辑结构工具采用标记目录句子联系图形目录和结构元素（请参见 10.6.3, 结构目录）。表 10.7 对标记目录操作符做了总结。

所有的标记目录操作符除了 `EMC` 都带有标签操作数来指示标记目录元素对应用的作用或意义。所有这类标签都必须在 Adobe 系统中注册过，以避免在不同的应用中定义同一个目录流而发生冲突。除了标签操作数，`DP` 和 `BDC` 操作符指定一个属性列表包含更多跟标记目录相关联的信息。属性列表将在 10.5.1 被进一步讨论。

标记目录操作符也许只出现在标记目录流中两个图形对象之中。他们不可能出现在单个图形对象或者图形对象与它的操作数之间。标记目录句子也许会被嵌套，但是每个句子只能被一个目录流全部包括，比如它不能交叉页面边界。

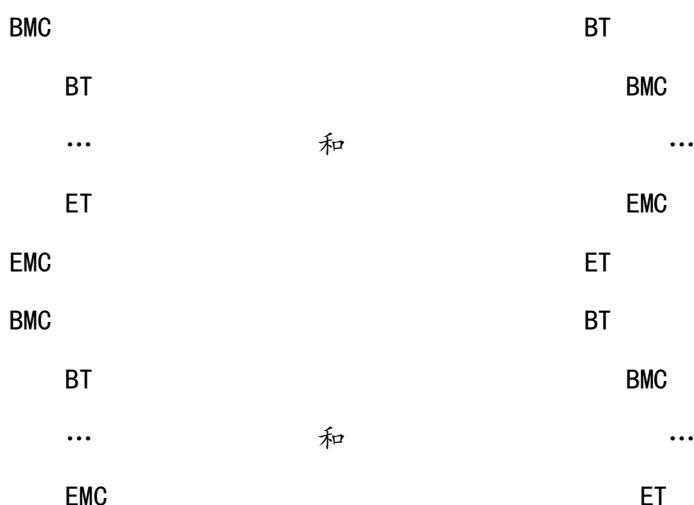
注意：目录是页面对象的入口（见 119 页“页面对象”），它有可能是单个流，也可能是一个流的阵列，但是它被认为是单个流式为了和标记目录句子相对应。

表 10.7 标记目录操作符

操作数	操作符	描述
标签	<code>MP</code>	指定一个标记目录点。标签是一个名字对象用来

		指示这个点的作用或意义。
标签属性	DP	指定一个标记目录点并带有属性列表。标签是一个名字对象用来指示这个点的作用或意义。属性要么是嵌入式的字典包括属性列表，或者是名字对象作为当前源字典的子字典（见 10.5.1，属性列表）。
标签	BMC	开始一个标记目录句子，被平衡 EMC 操作符中止。标签是一个名字对象用来指示这个点的作用或意义。
标签属性	BDC	开始一个标记目录句子并带有属性列表，被平衡 EMC 操作符中止。标签是一个名字对象用来指示这个点的作用或意义。属性要么是嵌入式的字典包括属性列表，或者是名字对象作为当前源字典的子字典（见 10.5.1，属性列表）。
—	EMC	结束一个以 BMC 或者 BDC 操作符开始的标记目录句子。

当标记目录操作符 BMC, BDC 和 EMC 跟文本对象操作符 BT, ET (见 5.3 节, “文本对象”) 混合使用时, 每一组相匹配的操作符 (BMC…EMC, BDC…EMC, 或者 BT…ET) 可以 (单独) 嵌套。因此, 句子是可以的, 但是是不允许的。



BT

EMC

10.5.1、属性列表

在目录流的标记目录元素中，标记目录操作符 **DP** 和 **BDC** 带有一个属性列表。属性列表是一个字典，其中包括对项目（应用或者扩展）解释的私有信息来创建标记目录。建议方案为项目采用一致的方式使用字典入口，比如，一个给定值的变量应该采用同一类型（或者低类型）。

如果属性列表中的值都是直接对象，那么字典在目录流中以嵌入的形式作为一个直接对象。只要有一个直接对象的参数在目录流之外，那么属性列表就要在当前源字典中被定义为一个子字典名字源所代替（见节 3.7.2，“源字典”），并且只能按名称被引用作为 **DP** 或者 **BDC** 操作符的操作数。

10.5.2、标记目录和剪切

一些 PDF 路径和文本对象被纯粹的定义为在当前剪切路径它们的影响，没有在页面上真正的标出实际对象。这一般发生在当路径对象被定义用操作句子 **W n** 或者 **W* n**（见节 4.4.3，“剪切路径操作符”）或者当一个文本对象被文本渲染模式 7 所渲染（见节 5.2.5，“文本渲染模式”）。这些剪切，没有渲染路径和文本对象就被称作剪切对象。当一个剪切对象在标记目录句子中出现，一般不认为它是句子的一部分，除非整个句子只有一个剪切对象。在例 10.2，标记目录句子标签剪切包括字符串（剪切我）但是不是定义了剪切边界的矩形路径。

例 10.2

```
/Clip BMC  
100 100 10 10 re W n      % Clipping path  
(Clip me) Tj             % Object to be clipped  
EMC
```

只有当一个标记目录句子包括整个剪切对象时，剪切对象才被认为是句子的一部分。在这个例子中，这个句子就被认为是一个标记剪切句子。这种句子可以被嵌套。在例子 10.3 中，文本的多行被用来剪切一连串的图形对象（在此例中是一个完整的路径）。文本的每一行都被包括在一个独立的标记剪切句子，带标签 Pgf。整个系列轮流的被外面的标记剪切句子包括，带标签 Clip。然而，值得一提的是带 ClippedText 标签的标记目录句子不是一个标记剪切句子，虽然它包括一个完整的矩形路径，但是它不是一个剪切对象。剪切对象属于 Clip 和 Pgf 句子，但是不是 ClippedText 句子。

例 10.3

```
/ClippedText BMC  
/Clip <<...>>  
BDC  
BT  
7 Tr                  % Begin text clip mode  
/Pgf BMC  
(Line 1) Tj  
EMC  
/Pgf BMC  
(Line) '  
( 2) Tj  
EMC  
ET                  % Set current text clip  
EMC  
100 100 10 10 re f % Filled path  
EMC
```

执行标记剪切句子的精确规则如下：

剪切对象是一路径对象并且是以 `W n` 或者 `W* n` 操作句子结束或者是一文本对象以文本渲染模式 7 渲染结束。

不可视图像对象是一路径对象并且是以仅 `n` 操作符结束（没有前缀 `W` 或者 `W*`），或者文本对象以文本渲染模式 3 渲染结束。

可视图像对象是一路径对象并且是不是以 `n` 操作符结束，或者文本对象不以文本渲染模式 3 或 7 渲染结束，或者任何 `XObject` 被 `D` 操作符引用。

一个空的标记目录元素是一个标记目录点或者是句子但是不包括图像对象。

标记剪切句子是一标记目录句子并且至少包括一个剪切对象但没有图形对象。

剪切对象和标记剪切句子作为内附标记目录句子的一部分，只有它是标记剪切句子。

不可视图形对象和空标记目录元素总是被认为是内附标记目录句子的一部分，不管它是否是一个标记剪切句子。

`q` 和 `Q` 操作符不能出现在标记剪切句子中。

例 10.4 说明这些规则的应用。标记目录句子 `S4` 是一句标记剪切句子是因为它包含一剪切对象（剪切路径 2）并且没有可视图形对象。因此剪切路径 2 被认为是句子 `S4` 的一部分。标记目录句子 `S1, S2` 和 `S3` 不是标记剪切句子，是因为它们至少包括一个可视图形对象。因此剪切路径 1, 2 不是这个三个句子的一部分。

例 10.4

/S1 BMC

/S2 BMC

/S3 BMC

0 0 m

100 100 l

0 100 l W n % Clipping path 1

0 0 m

200 200 l

0 100 l f % Filled path

EMC

/S4 BMC

0 0 m

300 300 l

0 100 l W n % Clipping path 2

EMC

EMC

100 100 10 10 re f % Filled path

EMC

在例 10.5 中，标记目录句子 S1 是一句标记剪切句子是因为唯一的图形对象包含一个剪切路径。因此，空标记目录句子 S3 和标记目录点 P1 都是句子 S2 的一部分，S2, S3 和 P1 都是句子 S1 的一部分。

例10.5

/S1 BMC

...Clipping path...

/S2 BMC

/S3 BMC

EMC

/P1 DP

EMC

EMC

在例 10.6 中，标记目录句子 S1 和 S4 都是标记剪切句子是因为唯一的对象包括一剪切路径。因此剪切路径是 S1 和 S4 的一部分，S3 是 S2 的一部分，S2, S3, S4 都是 S1 的一部分。

例10.6

/S1 BMC

/S2 BMC

/S3 BMC

EMC

EMC

/S4 BMC

…Clipping path…

EMC

EMC

10.6、逻辑结构

PDF 的逻辑结构设备（PDF 1.3）提供一设备能够把文档目录的结构信息纳入到 PDF 文件中。这些信息可能包括章节的组织，特殊元素的识别，比如图像，表格和脚本。逻辑结构设备是可扩展的，允许这样应用使产生 PDF 文档时选择什么样的结构信息并且怎么去实现它，同时消费者在阅读 PDF 文档时不用知道制作者的结构转换信息。

PDF 逻辑结构跟标准的文档标记语言，比如 HTML, SGML 和 XML，共享基本的特征。一个文档逻辑结构用结构元素的等级来表示，每个等级用字典对象表示。像其他的标记语言一样，PDF 结构元素有内容和属性。在 PDF，文档目录的呈现

在 HTML, SGML 和 XML 之上。

一个 PDF 文档的逻辑结构跟它的可视内容分开来存储，用点来彼此联系。这种独立结构允许逻辑元素的顺序和位置能够独立于文档页面中图形对象的顺序和位置。

文档目录（见节 3.6.1，“文档目录”）的 `MarkInfo` 入口指定一标记信息字典，表 10.8 显示它的入口。它提供了额外的信息关于 PDF 文档结构的特殊用途。

表10.8 标记信息字典的入口

键值	类型	数值
标记	布尔	(可选) 标志位指示文档是否符合带标签的 PDF 规则。默认值：否。 注意：如果 <code>Suspects</code> 是对的，这个文档也许不完全符合带标签的 PDF 规则。
用户属性	布尔	(可选, PDF1.6)，标志位指示结构元素的存在是否包括用户的性能属性（见 804 页“用户性能”）。默认值：否。
怀疑	布尔	(可选, PDF1.6)，标志位指示标签怀疑的存在（见 817 页“页面内容顺序”）。默认值：否。

10.6.1 结构层次

文档的逻辑结构用对象的层次所谓的结构层次或者结构树来描述。在层次的根部是字典对象称为结构树根部，在文档目录（见节 3.6.1，“文档目录”）中位于名字为“`StructTreeRoot`”的入口。表 10.9 展示结构树根部字典的入口。K 入

口指定当前结构树根部的子结构，称为结构元素。

结构元素也用字典来展示，它们的入口在表 10.10 中描述。K 入口指定结构元素的子结构，它可以是 0 或者更多的以下种类项目：

其他结构元素

参考内容项目，要么是标记内容句子（见节 10.5.，“标记内容”）或完整 PDF 对象比如 XObjects 和注释。这些内容项目代表跟结构元素关联的图形内容。内容项目将在节 10.6.3 “结构内容” 中详细讨论。

表10.9 结构树根的入口

键名	类型	数值
Type	名字	(必需) 字典描述的PDF对象的类型；一结构树根必须定义为 StructTreeRoot
K	字典或数组	(可选) 在结构层次中当前结构树根的子结构。它的值可以是一字典单个结构元素或者很多字典的阵列。
IDTree	名字树	(如果任何一个结构元素都有元素身份，则必需) 名字树指出结构元素的身份（见表 10.10）。

ParentTree	数字树	(如果任何结构元素都包含内容项目，则必需) 数字树(见节3.8.6，“数字树”)被用来内容项目属于哪一个结构元素。每一个在数字树中的整形数都跟文档中的单页或者是有权利处理内容的独立的对象(比如注释或者XObject)相联系。整形值在对象(见797页“从内容项目中查找结构元素”)中被赋予StructParent或者StructParents的值。相关的值的形式取决于对象的自然属性:
		如果一个对象有权利处理自己的内容项目，那么数值间接参考该对象的父元素(结构元素包括它作为一个内容项目)。
		如果页面对象或者内容流包括内容项目的标记内容句子，这个数值就是一数组参考标记内容句子的父元素。
		进一步讨论见797页“从内容项目中查找结构元素”。
ParentTreeNextKey	整形	(可选) 在父树种一个整形比任何键都伟大，通常被用来作为下一个加入书的键。
RoleMap	字典	(可选) 应用在文档的标准结构类型设置中标出其最近似等同的结构类型名字。(见节10.7.3，“标准结构类型”)。
ClassMap	字典	(可选) 作为一字典，映射对象指定属性名称到相应的属性对象或类的属性的对象数组(见802页“属性类”)。

表10.0 结构元素字典入口

键	类型	数值
Type	名字	(可选) 字典描述的PDF对象的类型; 若存在, StructElem一定是结构元素。
S	名字	(必需) 结构类型, 一个名字对象确定结构的自然属性和它在文档中的作用, 比如章, 段或者脚注(见节10.6.2, “结构类型”)。结构类型的名字必须跟附录E描述的方针相符合。
P	字典	(必需, 必须是一个间接引用) 在结构层次中结构元素的现在的父结构。
ID	字符串	(可选) 元素标识符, 用字符串来标识结构元素。在文档结构层次中这些字符串都是唯一的。结构树根(见表10.9)的IDTree入口定义了元素标识符和结构元素表示的对应关系。
Pg	字典	(可选; 必须是间接引用) 页对象表示用K入口来表示的一些或者整个内容项目的页。
K	(变量)	(可选) 结构元素的子结构。入口的数值是下面对象的一种或者一个数组: 一个结构元素字典代表另一个结构元素 一个整形标记内容标识符代表一个标记内容句子 一个标记内容引用字典代表一个标记内容句子 一个对象引用字典代表一个PDF对象

上述对象除了第一个(结构元素字典)被认为是一个内容项目; 见节10.6.3, “结构内容”对这些代表形式有进一步的讨论。

注意: 如果K的值是一字典你过去不是type入口, 那么它被认为是一结构元素字典。

键	类型	数值
---	----	----

A	(变量)	(可选) 跟结构元素相关的单个属性对象或者是数组属性对象。每个属性对象要么是一字典，要么是一束流。如果入口的值是一数值，那么数组的属性对象被一个整形代表其版本数（见803页节10.6.4，“结构属性”，和“结构版本数”）。
C	数组或名称	(可选) 跟结构元素相关的单个属性类名或者是数组属性类名。每个属性对象要么是一字典，要么是一束流。如果入口的值是一数值，那么数组的类名被一个整形代表其版本数（见803页节10.6.4，“结构属性”，和“结构版本数”）。注意：如果A和C两个入口都存在，并且属性都被定义，那么只有A入口生效。
R	整形	(可选) 结构元素的当前版本数（见803页“属性版本数”）。它的数值只能是无符号整形数。默认值：0
T	文本字符串	(可选) 结构元素的标题，在人类可读的形式中用文本字符串来表示它。这个题目能够描述结构元素的特点，比如第1章，而不仅仅是一个普通的元素类型，比如章。
Lang	文本字符串	(可选；PDF1.4) 整个结构元素的文本的语言除了嵌入结构元素或者标记内容的重写（见节10.8.1，“自然语言规范”）。如果这个入口缺省，那么文档目录的语言将被应用。
Alt	文本字符串	(可选) 用人类可读形式补充描述结构元素及子类型，当抽取文档目录辅助支持用户不可用或者其他目的时很有用（见节10.8.2，“辅助描述”）。
E	文本字符串	(可选；PDF1.5) 缩写的扩展方式。
ActualText	文本字符串	(可选；PDF1.4) 准确更换结构元素及子结构的文本。这个更换文本（应该应用与尽可能少的内容）对于抽取文档目录辅助支持用户不可用或者其他目的时很有用（见节10.8.3，“更换文本”）。

10.6.2、结构类型

每一个结构元素都有一个结构类型，一个名字对象确定结构元素的自然属性和在文档中的作用（比如章，段或者脚本）。为方便PDF应用内容的转换，Adobe定义了一系列标准的结构类型：见节10.7.3，“标准结构类型”。然而，应

用不需要去选用它们并且可以使用结构类型的任何名字。

当标准的名字都用完了，在结构树根中 `map` 将被使用，它在文档中找到最接近标准的名字来标记。例如，文档中的名字为 `Section` 的结构类型名称将被标记成标准类型 `Sect`。这个等价不需要太精确；`role map` 仅仅在不同类型之间用最接近的类型来指示，除了用合理的方式建立一个文档来处理非标准的结构元素外都可以应用。

注意：同一结构类型在 `role map` 时也许会作为一个键和数值，并且联系的循环链被允许。单角色地图可以额外定义双地图。在使用角色地图时应该遵循链的联系直到它找到一个可以识别的结构类型或者返回一个已经遇到过的类型。

注意：在 PDF 1.5 以前的版本，标准元素类型不能被重地图。从 PDF1.5 版本开始，元素名字总是在测绘图中被测绘跟它相应的名字，如果有一个，即使原始的名字是标准类型。这就允许元素代表一个带有同样名字的标签作为一个标准的角色，甚至它跟标准角色不同。

10.6.3、结构内容

结构元素通常跟图像内容联系在一起，组成一个或多个内容项目。内容项目是在文档中独立于结构树存在的图像对象但是跟结构元素相联系就像下面所描述的。

内容项目有两种类型：

内容流中的标记内容句子（见“标记内容句子作为内容项目”）

完整的 PDF 对象比如注释和 XObjects（见“PDF 对象作为内容项目”）

在结构元素字典中（见表 10.10）的 `K` 入口指定结构元素的子结构，它能包含任何内容项目的数字，就跟子结构元素轮流的有它自己的内容项目一样。

从概念来讲，内容项目必须是结构树的叶子节点，也就是说，他们不能在其中嵌套其他内容的逻辑结构的目的项目。结构元素之间的层次关系是完全代表由结构元素的 `K` 字典项，由嵌套不相关的内容项目。因此，有下面的限制：

标记内容句子划定结构内容项目不能有另外一个嵌套有内容项目的标记内容句子（虽然无结构的标记内容是允许的）

结构内容项目不能缓引（用 Do 操作符）本身就是一个结构内容项目的 XObject。

标记内容句子作为内容项目

在一个内容流中图像操作符的系列以以下方式被指定为结构元素的内容项目：

操作符必须被作为一个标记内容句子在 BDC 和 EMC 操作符所包含（见节 10.5，“标记内容”）

注意：虽然跟标记内容句子相关联的标签不是直接跟文档逻辑结构相对应，但是它应该跟相关联的结构元素的结构类型一样。

标记内容句子必须有一个属性列表（见 10.5.1，“属性列表”）包括一个 MCID 入口，这个 MCID 入口时一个整形标记内容识别码，能够在自己的内容流中唯一识别标记内容句子，如下例所示：

例10.7

```
2 0 obj % Page object
<< /Type /Page
/Contents 3 0 R % Content stream
...
>>
endobj

3 0 obj % Page's content stream
<< /Length ... >>
stream
...
/P << /MCID 0 >> % Start of marked-content sequence
BDC
...
(Here is some text) Tj
...
EMC % End of marked-content sequence
```

```
...
endstream
endobj
```

注意：这个及以下的例子忽略在作为内容项目的对象中必须的 StructParents 入口（见 797 页“从内容项目中查找结构元素”）。

一个结构元素字典能包含一个或多个标记内容句子作为内容项目，通过它的 K 入口能够找到它们（见表 10.10）。这类引用由两种形式：

一是所谓的标记内容引用的字典对象。表 10.11 描述了这种能够指定标识内容指示符的字典类型的内容，和其他信息指定包括句子的流一样。例 10.8 说明标记内容引用例 10.7 中的标记内容句子的应用。

指定标记内容指示符的整形。在通常情况下这是可以的，当页面的内容流包含标记内容句子时，这些页面能够被结构元素字典的 Pg 入口所指定。例 10.9 描述了一个含义三个子结构的结构元素：一个标记内容句子被一个标记内容指示器所指定，跟其他两个结构元素一样。

例10.8

```
1 0 obj % Structure element
<< /Type /StructElem
/S /P % Structure type
/P ... % Parent in structure hierarchy
/K << /Type /MCR
/Pg 2 0 R % Page containing marked-content sequence
/MCID 0 % Marked-content identifier
>>
>>
endobj
```

表10. 11 标记内容引用字典的入口

键	类型	数值
Type	名字	(必需) 字典所描述的PDF对象的类型; 为标记内容引用时必须用MCR。
Pg	字典	(可选; 必须是一个间接引用) 页面对象指定含有标记句子被嵌套的图像对象的页面。这个入口重写在结构元素中包含标记内容引用的Pg入口; 如果结构元素中没有此入口时必须有一个。
Stm	流	(可选; 必须死一个间接引用) 内容流包括标记内容句子。这个入口要有除非标记内容句子嵌套在内容流比如XObject(见节4.9,“XObject形式”)或者注解间隙流(见节8.4.4,“间隙流”)。如果这个入口缺省, 页面中的内容流通过Pg来指定标记内容句子(在标记内容引用字典中或者是父结构元素中)。
StmOwn	(任一)	(可选; 必须是一间接引用) PDF拥有Stm指定的流, 比如注解属于间隙流一样。
MCID	整形	(必须) 标记内容句子的标记内容指示符包括它的内容流。

例10.9

```
1 0 obj % Containing structure element
<< /Type /StructElem
/S /MixedContainer % Structure type
/P ...
/Pg 2 0 R % Page containing marked-content sequence
/K [ 4 0 R % Three children: a structure element
0 % a marked-content identifier
5 0 R % another structure element
]
>>
endobj
```

```
2 0 obj % Page object
<< /Type /Page
/Contents 3 0 R % Content stream
...
>>
endobj

3 0 obj % Page's content stream
<< /Length ... >>
stream
...
/P << /MCID 0 >> % Start of marked-content sequence
BDC
(Here is some text) Tj
...
EMC % End of marked-content sequence
...
endstream
endobj
```

内容流除了页面内容外能够包括结构元素的内容项目的标记内容句子。
XObjects 形式的内容被以下形式的结构元素所纳入：

Do 指示符能够把 **XObjects** 形式渲染成跟结构元素（见例 10.10）相关的标记内容句子的一部分。在这个例子中，整个 **XObjects** 形式被认为是结构元素内容的一部分，就相信它在 **Do** 操作符的指点下被插入标记内容句子。**XObjects** 形式不能轮流包括任何跟其他结构元素有关联的标记内容句子。

XObjects 形式的内容流能够包含一个或多个跟其他结构元素有关联的标记内容句子（见例 10.11）。**XObjects** 形式能有任意的子结构，包括跟逻辑结构元素相连的任何数值的标记内容句子。然而，任何用来渲染 **XObjects** 形式 **Do** 指示

符应该不能是逻辑结构内容项目的一部分。

注意：采用多次 Do 指示符调用来渲染的 XObjects 形式只能通过第一种方法来嵌入到文档的逻辑结构中，每个 Do 调用个别的跟结构元素相关联。

例10.10

```
1 0 obj % Structure element
<< /Type /StructElem
/S /P % Structure type
/P ... % Parent in structure hierarchy
/Pg 2 0 R % Page containing marked-content sequence
/K 0 % Marked-content identifier
>>
endobj

2 0 obj % Page object
<< /Type /Page
/Resources << /XObject << /Fm4 4 0 R >> % Resource
dictionary
>> % containing form XObject
/Contents 3 0 R % Content stream
...
>>
endobj

3 0 obj % Page's content stream
<< /Length ... >>
stream
...
/P << /MCID 0 >> % Start of marked-content
sequence
BDC
```

```
/Fm4 Do % Paint form XObject
EMC % End of marked-content sequence
...
endstream
endobj

4 0 obj % Form XObject
<< /Type /XObject
/Subtype /Form
/Length ...
>>
stream
...
(Here is some text) Tj
...
endstream
endobj
```

例10.11

```
1 0 obj % Structure element
<< /Type /StructElem
/S /P % Structure type
/P ... % Parent in structure hierarchy
/K << /Type /MCR
/Pg 2 0 R % Page containing marked-content sequence
/Stm 4 0 R % Stream containing marked-content sequence
/MCID 0 % Marked-content identifier
```

```
>>  
>>  
endobj  
2 0 obj % Page object  
<< /Type /Page  
/Resources << /XObject << /Form4 4 0 R >> % Resource  
dictionary  
>> % containing form XObject  
<< /Contents 3 0 R % Content stream  
...  
>>  
endobj  
3 0 obj % Page's content stream  
<< /Length ... >>  
stream  
...  
<< /Form4 Do % Paint form XObject  
...  
endstream  
endobj  
  
4 0 obj % Form XObject  
<< /Type /XObject  
/Subtype /Form  
/Length ...  
>>  
stream  
...  
<< /P << /MCID 0 >> % Start of marked-content sequence
```

```
BDC
...
(Here is some text) Tj
...
EMC % End of marked-content sequence
...
endstream
endobj
```

PDF 对象作为内容项目

当一个结构元素的内容包括整个 PDF 对象，比如 XObject 或者一注解，跟页面有联系但是不是这季节包括在页面的内容流里面，那么这个对象通过对象引用字典（见表 10.12）被结构元素 K 入口所指定。请注意这种形式的引用只应用于整个对象。如果引用内容形式指示对象内容流的部分，那么它被当作标记内容句子处理，就像前面一节所描述的那样。

表10.12 对象引用字典入口

键	类型	数值
Type	名字	(必需) 字典描述的PDF对象的类型；只能用OBJR引用对象。
Pg	字典	(可选；必须是间接引用) 页面对象代表有对象的页面。这个入口重写包括对象引用的结构元素中的任何Pg入口；如果结构元素中没有这入口，那么它是必须的。
Obj	(任一)	(必需；必须是间接引用) 对象引用。

注意：如果引用对象出现在多页面中，那么每次出现都要求一个独立的对象引用。然而，如果它只是多次在同一页面中出现，那么只需要一个对象引用来代表它们。（它会变得重要如

果是用来区分多次出现在同一页面的同一个 XObject，它们应该通过包括特殊系列的 Do 操作符来处理而不是通过对对象引用)。

从内容项目中查找结构元素

因为流不能包括对象引用，所以对标记内容句子直接跟它们的父结构元素(作为内容项目属于父结构元素)相对应的内容项目就没有办法。取而代之的是一种不同的机制，父树结构就应运而生了。为求一致，整个 PDF 对象的内容项目，比如 XObjects，也可以用父树链接到它们的父结构元素。

父树是一个数字树(见节 3.8.6，“数字树”)，从一个文档的根结构树(786 页表 10.9) 来访问父树。这树为每个至少包含一个结构元素的内容项目的对象和至少包括作为内容项目的一个标记内容句子的内容流都设置了一个入口。每个入口的键值都是一个整形数，作为 StructParent 和 StructParents 在对象中的入口的数值。这些入口的数值如下：

由于一个对象通过对象引用(见 796 页“PDF 对象作为内容项目”)而作为内容项目，这个数值是一个间接对应于父结构元素。

若一个内容流包含作为内容项目的标记内容句子，这个数值就是间接对应于句子父结构元素的一个数组。这个数值元素通过句子的标记内容指示符作为数值的 0 目录来对应每个句子。

注意：由于标记内容指示符作为父结构树中数组的指数，其指定的值应该尽可能的小以此来包括数组的空间。

结构树跟中 ParentTreeNextKey 入口有着比当前在父结构树中大得多的整形数值。当一个新的入口被加进父树，那么它将当前的 ParentTreeNextKey 作为它的键。然后，它的数值将递增为了下一个新的入口的增加。

要找到相关父树的入口，在树种出现的对象或者是内容流必须包含一个特殊的字典入口，**StructParent** 或者 **StructParents**（见表 10.13）。取决于内容项目的类型，这个入口也许出现在包含标记内容句子的页面对象中，出现在 **XObject** 的形式或者图片的字典流中，在注解字典中，或者在其他的包含在结构元素中充当内容项目的对象字典类型中。它的数值是一整形键出现在父结构树入口对应于对象的地方。

表10.13 为结构元素处理而增加的字典入口

键	类型	数值
StructParent	整形	(必需对于结构内容项目的对象；PDF1.3) 在父结构树中对象入口的整形键。
StructParents	整形	(必需对于包含标记内容句子的所有内容流；PDF1.3) 在父结构树中对象入口的整形键。

注意：在给定的对象中大部分都会出现两个入口的一个。对象可以是它的整体的一个内容项目，或者是内容项目的标记内容句子的综合，但不是都包含。

对于用对象引用来表示的内容项目，因此父结构元素能够通过在项目对象字典中使用 **StructParent** 入口的数值作为父结构树（在结构根树的父树入口找到）的检索键被找到。从父树得到的对应的数值检索跟父结构元素相对应（见例 10.12）。

例10.12

```
1 0 obj % Parent structure element
<< /Type /StructElem
...
/K << /Type /OBJR % Object reference
/Pg 2 0 R % Page containing form XObject
/Obj 4 0 R % Reference to form XObject
>>
>>
endobj

2 0 obj % Page object
<< /Type /Page
/Resources << /XObject << /Fm4 4 0 R >> % Resource dictionary
>> % containing form XObject
/Contents 3 0 R % Content stream
...
>>
endobj

3 0 obj % Page's content stream
<< /Length ... >>
stream
...

```

```
/Fm4 Do % Paint form XObject
...
endstream
endobj
4 0 obj % Form XObject
<< /Type /XObject
/Subtype /Form
/Length ...
/StructParent 6 % Parent tree key
>>
stream
...
endstream
endobj

100 0 obj % Parent tree (accessed from structure tree root)
<< /Nums [ 0 101 0 R
1 102 0 R
...
6 1 0 R % Entry for page object 2; points back
... % to parent structure element
]
>>
endobj
```

如果一个内容项目是一标记内容句子，那么检索方法是类似的但要复杂很多。因为标记内容句子不是拥有自己权利的对象，它的父树键在页面对象的 **StructParents** 入口或者其他有句子存在的内容流被发现。检索到父树的数值不是一个父结构元素自身的参数而是这些参数的一个数组，对每一个标记内容句子的数组都包含在内容流中。给定句子的父结构元素通过使用句子的标记内容指示符作为数值的指针才能找到（见例 10.3）。

例10.13

```
1 0 obj % Parent structure element
<< /Type /StructElem
...
/Pg 2 0 R % Page containing marked-content sequence
/K 0 % Marked-content identifier
>>
endobj

2 0 obj % Page object
<< /Type /Page
/Contents 3 0 R % Content stream
/StructParents 6 % Parent tree key
...
>>
endobj
```

```
3 0 obj % Page's content stream
<< /Length ... >>
stream
...
/P << /MCID 0 >> % Start of marked-content sequence
BDC
(Here is some text) TJ
...
EMC % End of marked-content sequence
...
endstream
endobj
100 0 obj % Parent tree (accessed from structure tree root)
<< /Nums [ 0 101 0 R
1 102 0 R
...
6 [1 0 R] % Entry for page object 2; array element at index 0
... % points back to parent structure element
]
>>
Endobj 能够
```

10.6.4、结构属性

一种应用或者扩展使得逻辑结构能够连接附加的信息给任何的结构元素，我们称之为属性。属性信息由一个或多个跟结构元素有关的属性对象所组成。属性对象是一个字典或流包括 0 入口（见表 10.14）指示应用或者扩展自己拥有的属性信息。其他入口代表这些属性：键代表属性名称，数值代表相关的属性数值。为了方便在 PDF 应用中内容的互换，Adobe 已经定义了一系列标准结构属性通过特殊的标准拥有者所识别；见 10.7.4，“标准结构属性”。此外，PDF1.6 介绍了一种属性的使用代表用户属性（见 804 页“用户属性”）。

表10. 14 所有属性对象字典的常见入口

键	类型	数值
0	名字	(必需) 拥有自己属性数据的应用或者扩展的名字。这些名字必须跟附录E中描述的方针相符合。

任何应用都能把属性连接到任一结构元素，甚至被另外应用创立的应用。对一个应用能够把属性连接到相同的结构元素。在结构元素字典中 A 入口能够拥有单个属性的对象或者这类对象的数组，连同版本数虽然有不同的所有者协调制定的属性（见 803 页“属性版本数”）。为一个结构元素创建或者摧毁第二个属性对象的应用有责任把 A 入口的数值由单个对象转换成数组，反之亦然，以及保持版本数的完整性。没有继承顺序用 A 数值来定义属性对象，但是最好是在数组的末尾增加新的对象以至于第一个数组元素从属于原始建立结构元素的应用。

属性类

如果很多结构元素共享同一套属性值，那么它们可以定义为属性类来分享相同的属性对象。结构元素通过类的名字相关联。类名和属性对象的关系通过一个所谓的类图来定义，它被保存在结构树根的 ClassMap 入口（见 786 页表 10.9）。在类图中的每一个键都是一个名字对象，它代表一个类的名字。相应的数值是一个属性对象或者这类对象的数组。

注意：PDF 属性类跟 Java 和 C++ 等面向对象的编程语言中的类的概念是没有关系的。属性类严格地只是存储更多压缩形式属性信息的机制；它们不像真正面向对象类所具有的继承属性。

结构元素字典（见 787 页表 10.10）中 C 入口包含类名或者类名的数组（通常伴随着修改版本数；见下面的“属性版本数”）。C 入口中的每一个类名，相关

属性对象被认为是给定的结构元素相联系，伴随着在元素的 A 入口被识别。如果 A 和 C 入口都出现并且给定属性都被二者指定，那么用 A 入口指定的生效。

属性版本数

当一项应用改变结构元素或者它的内容，这种改变可能影响其他应用相关的结构元素的属性信息的正确性。版本数的系统运行这些应用检测到这些改变并且升级它们自己相应的属性信息，就像这节所描述的。

一个结构元素有一个版本数，它存储在结构元素字典的 R 入口 9(见 787 页表 10.10)。最初，这个版本数为 0 (如果没有 R 入口的缺省值)。如果一项应用改变结构元素或者任何它的内容项目，它通过递增版本数来示意这种改变。

注意：版本数跟间接对象提到的普通数是没有关系的（见节 3.2.9，“间接对象”）。

跟一个结构元素相连的每一个属性对象都有一个相关的版本数。这个版本数本存储在跟数组中用来连接属性对象和结构元素：

在结构元素中的 A 数组中的每个属性对象都用一对数组元素来表示，第一组包含自身的属性对象，第二组包含跟结构元素有关的整形版本数。

结构元素的 C 数组对每个属性类都包含一对元素，第一个元素包含类名，第二元素包含相关的版本数。

注意：版本数可以是 A 数组或者是 C 数组。不能跟在整形数组元素的后面的一个属性对象或者类名有一个版本数 0。

注意：版本数在属性对象中不是被直接存储，因为一个信息属性对象可能跟大于一个的结构元素相关联（它们的版本数不一样）。

当一个属性对象被建立或修改，它的版本数被设置成结构元素的 R 入口的当前值。通过比较结构元素属性对象的版本数，一个应用被决定

是否属性对象的内容还是当前的或者是否由于在底层结构元素的很多最近更改应该而变得过时。

注意：一个属性对象的改变不要改变跟结构元素相关联的版本数，它们只有在结构元素本身或者它的内容项目被更改的是被改变。

有时，一项应用可能产生对一个结构元素的大量更改，这有可能是以前所有与此相关的属性信息都失效。在这种情况下，取代递增结构元素的版本数的是，应用将选择删除所有从 A 到 C 数组未知的属性对象。这两个动作时相互排斥的：应用要么递增结构元素的版本数，要么删除它们的属性对象，不能两者兼有。值得注意的是任何一项应该创建属性对象是必须为其他应用在任何时候删除它的可能性做准备。

用户属性

大多数的结构属性（见节 10.7.4，“标准结构属性”）指定反映元素外观的信息；例如 `BackgroundColor` 或者 `BorderStyle`。然而一些 PDF 产品，比如 CAD 应用，也许会用一个具有标准外观的对象，每一个对象都包含没有图形信息来区分彼此。比如，很多晶体管具有相同的外观但是具有不同的属性比如类型和部件数。

用户属性 (PDF1.6) 用来包含这些信息。任何一个跟结构元素相对应的图形对象也许会跟用户属性相关联，通过一个属性对象字典和 0 入口的 **UserProperties** 的数值来指定 (见表 10.15)。

表10.15 在属性对象字典中为用户属性增加的入口

键	类型	键值
0	名字	(必需) 属性拥有者，一定是UserProperties
P	数组	(必需) 字典的数组，每一个都代表一个用户属性 (见表10.16)。

P 入口是一指定用户属性的数组。在数组中的每一个元素都是一个用户属性字典代表一个独立的属性 (见表 10.16)。数组元素的顺序是很重要的，允许制造者指定属性的重要顺序。

表10.16 用户属性字典的入口

键	类型	键值
N	文本	(必需) 用户属性的名字
V	任何	(必需) 用户属性的数值
F	文本字符串	(可选) 是V数值的格式化表示，当特殊格式是被使用；比如“(\$123.45)”作为数字-123.45。如果这个入口缺省，应用将使用默认的格式。
H	布尔	(可选) 如果为真，则属性隐藏；也就是说，它在任何使用者的

代表对象属性的界面元素不被显示。默认值：假。

包含用户属性的 PDF 文档都必须在文档标记信息字典中提供一个数值为真的 **UserProperties** 入口（见表 10.8）。这个入口允许消费者应用快速决定是否有必要为包含用户属性的元素去搜索结构树。

例 10.14 展示了一个结构元素包括一个称为 **Part Name**, **Part Number**, **Supplier**, 和 **Price** 的用户属性。

例10.14

```
100 0 obj
<< /Type /StructElem
/S /Figure % Structure type
/P 50 0 R % Parent in structure tree
/A << /O /UserProperties % Attribute object
/P [ % Array of user properties
```

```
<< /N (Part Name) /V (Framostat) >>
<< /N (Part Number) /V 11603 >>
<< /N (Supplier) /V (Just Framostats) /H true >> % Hidden attribute
<< /N (Price) /V -37.99 /F ($37.99) >>      % Formatted value
]
>>
>>
endobj
```

10.6.5、逻辑结构例子

例 10.15 展示了 PDF 文件的一部分带有一个简单的文档树。结构树根（对象 300）包含结构类型 Chap（对象 301）和 Para（对象 304）类型的元素。这个被标为第一章的 Chap 元素包含 Head1（对象 302）和 Para（对象 303）类型的元素。

这些元素都在带标签的 PDF（见节 10.7.3，“标准结构类型”）中被测绘成标准结构类型，通过在结构树根中的角色图中被指定。对象 302 通过 304 有相连的属性（见节 10.6.4，“结构属性”和节 10.7.4，“标准结构属性”）。

这个例子也说明父树结构（对象 400）测绘内容项目反馈给它们的父树结构元素和 ID 树（对象 403）测绘元素指示符给代表的结构元素。

例10.15

```
1 0 obj                         % Document catalog
<< /Type /Catalog
/Pages 100 0 R                  % Page tree
/StructTreeRoot 300 0 R          % Structure tree root
>>
endobj
```

```
100 0 obj % Page tree
<< /Type /Pages
/Kids [ 101 1 R % First page object
        102 0 R % Second page object
]
/Count 2 % Page count
>>
endobj

101 1 obj % First page object
<< /Type /Page
/Parent 100 0 R % Parent is the page tree
/Resources << /Font << /F1 6 0 R % Font resources
/F12 7 0 R
>>
/ProcSet [/PDF /Text] % Procedure sets
>>
/MediaBox [0 0 612 792] % Media box
/Contents 201 0 R % Content stream
/StructParents 0 % Parent tree key
>>
endobj

201 0 obj % Content stream for first page
<< /Length ... >>
stream
1 1 1 rg
0 0 612 792 re f
BT % Start of text object
/Head1 << /MCID 0 >> % Start of marked-content sequence 0
BDC
```

```
0 0 0 rg
/F1 1 Tf
30 0 0 30 18 732 Tm
(This is a first level heading. Hello world: ) Tj
1.1333 TL
T*
(goodbye universe.) Tj
EMC % End of marked-content sequence 0

/Para << /MCID 1 >> % Start of marked-content sequence 1
BDC
/F12 1 Tf
14 0 0 14 18 660.8 Tm
(This is the first paragraph, which spans pages . It has four fairly
short and concise sentences. This is the next to last ) Tj
EMC % End of marked-content sequence 1
ET % End of text object
endstream
endobj
102 0 obj % Second page object
<< /Type /Page
/Parent 100 0 R % Parent is the page tree
/Resources << /Font << /F1 6 0 R % Font resources
/F12 7 0 R
>>
```

```
/ProcSet [/PDF /Text] % Procedure sets
>>
/MediaBox [0 0 612 792] % Media box
/Contents 202 0 R % Content stream
/StructParents 1 % Parent tree key
>>
endobj
202 0 obj % Content stream for second page
<< /Length ... >>
stream
1 1 1 rg
0 0 612 792 re f
BT % Start of text object
/Para << /MCID 0 >> % Start of marked-content sequence 0
BDC
0 0 0 rg
/F12 1 Tf
14 0 0 14 18 732 Tm
(sentence. This is the very last sentence of the first paragraph.) Tj
EMC % End of marked-content sequence 0
/Para << /MCID 1 >> % Start of marked-content sequence 1
BDC
/F12 1 Tf
```

```
14 0 0 14 18 570.8 Tm
(This is the second paragraph. It has four fairly short and concise
sentences. This is the next to last ) Tj
EMC % End of marked-content sequence 1
/Para << /MCID 2 >> % Start of marked-content sequence 2
BDC
1.1429 TL
T*
(sentence. This is the very last sentence of the second paragraph.)
Tj
EMC % End of marked-content sequence 2
ET % End of text object
endstream
endobj
300 0 obj % Structure tree root
<< /Type /StructTreeRoot
/K [ 301 0 R % Two children: a chapter
    304 0 R % and a paragraph
]
/RoleMap << /Chap /Sect % Mapping to standard structure types
/Head1 /H
/Para /P
>>
/ClassMap << /Normal 305 0 R >> % Class map containing one attribute class
/ParentTree 400 0 R % Number tree for parent elements
/ParentTreeNextKey 2 % Next key to use in parent tree
/IDTree 403 0 R % Name tree for element identifiers
>>
endobj
301 0 obj % Structure element for a chapter
<< /Type /StructElem
```

```
/S /Chap
    /ID (Chap1)           % Element identifier
    /T (Chapter 1)         % Human-readable title
    /P 300 0 R             % Parent is the structure tree root
    /K [ 302 0 R
        303 0 R           % Two children: a section head
                           % and a paragraph
    ]
>>
endobj

302 0 obj           % Structure element for a section head
<< /Type /StructElem
    /S /Head1
    /ID (Sec1. 1)          % Element identifier
    /T (Section 1.1)        % Human-readable title
    /P 301 0 R              % Parent is the chapter
    /Pg 101 1 R             % Page containing content items
    /A << /0 /Layout
                           % Attribute owned by Layout
        /SpaceAfter 25
        /SpaceBefore 0
        /TextIndent 12.5
    >>
    /K 0                   % Marked-content sequence 0
>>
endobj

303 0 obj           % Structure element for a paragraph
<< /Type /StructElem
    /S /Para
    /ID (Para1)            % Element identifier
    /P 301 0 R              % Parent is the chapter
    /Pg 101 1 R             % Page containing first content item
```

```
/C /Normal           % Class containing this element's attributes
/K [ 1               % Marked-content sequence 1
    << /Type /MCR      % Marked-content reference to 2nd item
    /Pg 102 0 R       % Page containing second item
    /MCID 0           % Marked-content sequence 0
    >>
]
>>

endobj

304 0 obj           % Structure element for another paragraph
<< /Type /StructElem
    /S /Para
    /ID (Para2)        % Element identifier
    /P 300 0 R         % Parent is the structure tree root
    /Pg 102 0 R         % Page containing content items
    /C /Normal          % Class containing this element's attributes
    /A << /0 /Layout
        /TextAlign /Justify % Overrides attribute provided by classmap
    >>
/K [1 2]           % Marked-content sequences 1 and 2
>>

endobj

305 0 obj           % Attribute class
<< /0 /Layout        % Owned by Layout
    /EndIndent 0
    /StartIndent 0
    /WritingMode /LrTb
    /TextAlign /Start
>>

endobj
```

```
400 0 obj % Parent tree
<< /Nums [ 0 401 0 R % Parent elements for first page
              1 402 0 R % Parent elements for second page
            ]
>>
endobj

401 0 obj % Array of parent elements for first page
[ 302 0 R % Parent of marked-content sequence 0
  303 0 R % Parent of marked-content sequence 1
]
endobj

402 0 obj % Array of parent elements for second page
[ 303 0 R % Parent of marked-content sequence 0
  304 0 R % Parent of marked-content sequence 1
  304 0 R % Parent of marked-content sequence 2
]
endobj

403 0 obj % ID tree root node
<< /Kids [404 0 R] >> % Reference to leaf node
endobj

404 0 obj % ID tree leaf node
<< /Limits [ (Chap1) (Sec1.3) ] % Least and greatest keys in tree
  /Names [ (Chap1) 301 0 R % Mapping from element identifiers
           (Sec1.1) 302 0 R % to structure elements
           (Sec1.2) 303 0 R
           (Sec1.3) 304 0 R
         ]
>>
endobj
```

10.7、标记 PDF

标记的 PDF (PDF 格式 1.4)，是一个独具风格，建立在 10.6 节“逻辑结构”所描述的逻辑结构框架基础之上的 PDF。它定义了一套标准的结构类型和属性，让网页内容（文本，图形，和图像）可以被提取及用作其他用途。它作为工具来使用时执行如下形式的操作：

- * 简单的文字和图形的提取粘贴到其他应用程序
- * 与原来设计布局相比，自动回流的文字和相关图形，以适应所采取的不同页面的大小
- * 处理文本作为搜索，索引、拼写检查 等用途，
- * 将文件的结构与基本样式信息保存并转换到其他常见的文件格式，如 HTML，XML 和 RTF 格式)
- * 使有视觉（直觉？）障碍的用户可以理解和接受内容（见第 10.8 节，“无障碍支援”）

标记的 PDF 文件符合下列约定：

- * 网页内容（第 10.7.1 节，“标记的 PDF 和网页内容”）。标记的 PDF 定义了用于网页内容文字的一套规则，使字符，单词和文本的顺序可以可靠地确定。所有文本是呈现一种可以转换为 Unicode 的形式。字符代表明确。实际内容区别于布局和分页。内容按照确定的创作应用程序给出了它的网页上出现的顺序。
- * 一个基本的布局模型（第 10.7.2，“基本布局模式”）。一整套规则为描述页面上的结构元素的安排。
- * 结构类型（第 10.7.3，“标准结构类型”）。一整套结构类型定义的结构元素的标准，如段落，标题，文章和表格。
- * 结构属性（第 10.7.4，“标准结构属性”）。标准结构属性用布局的网页内容中的创作风格应用程序来保存资料。

一个标记的 PDF 文件也必须包含真正标记项的值的标记信息字典（见表 10.8），

注：PDF 定义的型式和标记属性的目的是提供一整套的可靠标准及最低保证后备角色属性，让消费者应用程序执行诸如上述的操作。只要生产者应用提供了额外的与同等标准类型最相近的结构类型，他们就可以对其自由定义。可见 10.6.2 节中描述的“结构类型。”同样，生产者应用程序可以使用任何可用的扩展机制来定义其它的结构属性。

10.7.1、标记 PDF 和网页内容

同所有的 PDF 文件一样，一个标记的 PDF 文件包含一个序列的 selfcontained 页，其中每个都由一个或多个网页的内容流来描述（包括任意附属流，如 XObjects 形式和注释亮相。）

标记的 PDF 进一步定义了一些组织和标记惯例的内容流，以得到更多的信息：

- * 区分作者的原始内容和布局进程（见“真正内涵和文物” 813 页）
- * 如果网页内容必须回流，必须列举一个内容秩序来便指导布局过程中（见 817 页“网页内容顺序”）
- * 代表文字的形式，它的一个 Unicode 表示和信息关于字体特征可以明确地得出（见 819 页“提取属性”）
- * 字符间停顿明确（见 822 页“字符间停顿的识别”）
- * 文本标记使视觉障碍的用户能够访问信息（见第 10.8 节，“无障碍支援”）

真实内容和文物

在一个文档中的图形对象可以分为两类：

- * 作者介绍的原来文件物质材料组成的文档真实内容；
- * 文物图形对象，不属于作者的原创内容的一部分，而是在 PDF 格式的生产应用过程中产生的布局，或其他严格的机械过程。

文件的逻辑结构包含了所有的图形对象，这些图形对象制作了真正的内容，并介绍这些对象是如何彼此关联的。它不包括图形对象的布局和生产过程中产品的纯粹的产物。

一个文件的实际内容，不仅包括网页内容流和附属形式 XObjects，而且包含了满足以下所有条件的相关注释：

- * 标注有一个载有正常 (N) 的外观的外观流（见第 8.4.4 节，“外观 流”）。
- * 标注的隐藏标志（见第 8.4.2 节，“注释标记”）未设置。
- * 标注已包含在文档的逻辑结构（见第 10.6 节，“逻辑结构”）。

文物的详细说明

衍生物可明确与实际内容区分开来的地方在围绕它的标签工件上的标记内容顺序：

/衍生物/衍生物性质清单

BMC BDC

... 或...

电磁兼容 电磁兼容

第一种形式是用来识别一个通用的构件，第二种形式是用来识别有相关性质清单的构件。表格 10.7 中显示了可能出现在性质清单中的一些性质。

注：在文本流中，建议尽可能将这些衍生物用相应的性质清单区分开。文物缺乏指定边界框可能会在回流过程中丢失。

表 10.17 衍生物的性质明细清单

项	类型	值
Type	名字	(可选) 的工件类型, 这个属性的列表介绍, 必须作为名称分页, 布局, 或是页面类型之一而存在。
Bbox	矩形	(可选) 一和四个成员的队列在默认用户空间中提供的单位数目阵列, 左, 下, 右和顶部边缘, 分别边界框工件 (完全包围的可见的矩形的程度)
Attached	数组	(可选, 只是分页文物) 一列名字对象含有一对上, 下, 左, 右对象的数组, 指定页面的边缘, 如果有的话, 哪些是逻辑上的工件连接。页边定义裁剪框的网页(见第 10.10.1, 页边界-)。这些名称的顺序在无关紧要的数组内。包括左, 右或顶部 底部分别显示一个全角或全高的工件。

以下类型的文物, 可以指定该类型项目:

- * 分页项目。辅助功能, 如页面和开本运行 (页码)。
- * 布局项目。纯装饰性印刷或设计元素, 如脚注惯例或背景画面。
- * 佩奇项目。与物体本身产生目的无关, 如剪切标志和颜色标记。

标记的 PDF 消费者应用程序可能对于何种内容相关有自己的看法。例如, 文本-语音引擎可能在打开页面时不需要说出运行开头或者是页码。一般来说, 消费者应用程序可以做到如下:

- * 忽视网页内容中不太重要的元素（例如，特定类型的工件）
 - * 将一些页面元素作为不再继续检验的终端元素对待（例如，将一个图解当成是一个用作回流的单位）
 - * 更换一个备用文本元素（见第 10.8.2 节，“候补说明”）不同的消费应用程序可以根据他们的目标，在此方面作出不同的决定。
- 标记的 PDF 目的不是要规定消费者应用程序应该做什么，而是要提供足够的声明和描述 信息，以使在继续进行决策时作出合理的选择。

注：为了给残疾人士提供无障碍的用户及消费者应用，标记的 PDF 文件必须使用在第 10.8 节“无障碍支持”中所述的自然规范的语言，补充说明（ALT），替换文字（ActualText）和缩写扩展文本等方式。

附带文物

除了明确标记为文物和排除对象文档的逻辑结构，页面的运行文本也可能包含与文件实际内容没有逻辑关系的其它元素部分。但在铺设过程中将这些内容仅仅看成是偶然的结果文件。它们可能包括以下内容：

* 断字。经由文本布局介绍了文物是连字符标记在一行的末尾一个单词附带表决。在标记的 PDF 格式，这种附带字师必须用一个软连字符，而 Unicode 的映射算法（见 820 页“标记的 Unicode 映射 PDF 格式”）转换为 Unicode 值 U+00AD。（此字符有别于普通的硬盘连字符，它的 Unicode 值是 U+002D）

一个标记的 PDF 文件制作人必须明确区分软连字符与硬连字符，以使消费者不用再猜测给定的字符表示类型。

注：在某些语言中，情况更为复杂：可能有多个连字符和断字可能改变单词的拼写。详见 872 页的 10.24 示例。

* 文字间断。该网页的运行文本，如网页内容顺序（见下“网页内容顺序”），可能会有正常地方的文本出现间断的现象。例如，页面可能包含两个独立的文章的开头（见 8.3.2 节，“细则”），每一个都联系到文件的下一页。第一篇文章的

结尾与第二篇文章的开始不能重叠。

消费者应用程序可以通过检查文件的逻辑结构来识别这种不连续性。

* 隐藏网页元素。由于各种各样的原因，元素的一个文档的逻辑内容可能在网页上不可见：它们可能已被剪辑，它们的颜色可能趋近背景颜色，或者它们可能被其它物体遮挡重叠。

消费类应用仍能够识别和处理这类隐藏元素。例如，以前看不见的元素可以在页面回流时可见；或者文本到语音引擎可以对一个视力障碍的读者选择发音的文本。

标记的 PDF 的目的是，网页内容默认为含有全部的文字和插图，无论这些内容在显示和打印的时候是否可见。

页面内容顺序

在处理网页上逐页的基础材料，一些标记的 PDF 消费者应用过程中的网页内容元素，取决于在一个文本对象中内页的内容流和字符序列的图形对象，而不是在逻辑结构顺序中整理定义 该网页的逻辑结构，层次结构。两个序列在逻辑上是不同的，有可能或不可能重合。特别是，任何文物的页面都可能包含有在网页内容的顺序，但不是逻辑结构秩序，因为它们并不作为文件逻辑结构的部分。

一个标记的 PDF 文件的作者不仅负责建立适合每一页的内容中的页面顺序，而且负责建立对应的整个文档的适当的逻辑结构层次。

由于网页内容的主要要求是保持正常阅读序列中元素的回流，它通常应（为西部 书写系统）自上而下地（在一个多列布局），并对应其正确的衍生物而进行。

一般来说，一篇文章中网页上的所有显示部分应一并保存，即使它在页面上流向分散的地点。插图或脚注，可穿插于文本的相关文章或出现在内容结束处（或以脚注的形式显示在整个页面的逻辑内容结尾）。

在某些情况下，生产者意图产生标记的 PDF，可能由于一些文档的内容而无法产生正确的网页内容顺序。这很常见，例如，如果内容是从另一个应用程序中提取，或者如果

有不清楚的地方或文本输出中丢失了信息。在这种情况下就可以使用疑问标签（PDF 格式 1.6）生产者可以通过疑问标签来识别有疑问的内容。（见第 10.5 节，“标记的内容”）

标记的内容必须有一个属性字典，此字典必须含有疑问标签，名称是

TagSuspect, 其功能就是命令, 指示, 可以显示封闭的不适合标记的 PDF 格式规范要求的内容。

例 10.16

```
/TagSuspect <</TagSuspect /Ordering>>  
BDC  
.... % 问题页面内容  
EMC
```

文件包含的疑问标签必须包含一个在信息字典里具有价值的疑问条目(见表 10.8)。消费类应用在遇到此条目时, 应该在标记的 PDF 的使用中依靠疑问标签标记的内容来进行后续工作。

序列注解

与页面相关的注释不交织在网页的内容上, 而是在其网页中放置对象 **Annots** 阵列流(见 119 页“页面对象”)。因此, 一个页面中的内容标注的正确位置秩序不是很明显, 取决于文档的逻辑结构。

这两个网页的内容(标明内容序列)和说明可以被视为从内容结构元素引用的项目(见第 10.6.3, “结构内容”)。类型阿诺(PDF 格式 1.5), 链接, 或形态构成要素(见“内联级别结构元素”和 834 页上的“插图元素”第 841 页)明确指定标记之间的内容序列协会和一个相应的注释。在其他情况下, 如果

结构元素

对应于注释立即之前或之后（在逻辑 结构顺序）一结构元素对应一个显着内容序列， 注释被认为是之前或按照标明的内容序列， 分别在网页内容的顺序。

注：如有必要，一个标记的 PDF 生产者可能会推出一个空标记内容序列，并仅仅作为一个定位为邻接结构元素注释在网页内容的顺序。

反向阶显示字符串

在书写系统中默认的顺序是从右向左（如阿拉伯语或希伯来语）。这使得字体的自行在右下角的起源和它们的宽度（向右水平位移）变得可以预期，指定为阴性。

各种技术和历史的原因，很多这样的字体是按照上述公约为西方书写系统设计的，是具有标志符号的起源。在左下角的和积极的宽度，如 364 页图 5.4 所示。因此，在这种显示从右到左的书写系统文本要求为每个单独的定位字形（这是繁琐和昂贵的）或代表字符串文本（见“组织和字体使用”358 页），字符代码给出了相反的顺序。当使用后一种方法时，字符代码的正确网页内容的秩序与它们字符串显示的顺序相反。

标记的内容标签 `ReversedChars` 告知标记的 PDF 消费者应用。这标志着在内容顺序显示字符串包含字符与该网页内容顺序相反。如果该序列包含多个显示字符串，每个字符串只是在个别字符中相反，那么这些字符是自然的阅读顺序。例如，序列

/ ReversedChars

BMC

(olleH) Tj

-200 0 的 Td

(.d\row) Tj

EMC

代表文本

Hello world

显示字符串可以在开头或结尾用空格来表示字符（见“822 页字识别符”），但可能不包含内部的空格。这种限制并不严重，因为空间提供了调整没有显著效果的字体的这样的一个机会，并且提供了一个限制文字处理反向应用的有价值的处理。

提取的特征特性

标记的 PDF 文档的一个要求是：字符代码可以明确地转换为 Unicode 值代表的文本信息内容。这样做有几种方法，一个标记的 PDF 文件，必须符合其中至少一项的要求（见下面所述“标记的 PDF Unicode 的映射”）。

此外，标记的 PDF 文件必须允许一些相关的特征字体推断（见 820 页“字体特征”）。这些 Unicode 的价值观和字体特征可以被用来 cutand – 粘贴编辑，检索，文本到语音转换，并转换到其他应用程序或文件格式。

Unicode 的映射标记的 PDF

标记的 PDF 要求每个字符在文档中的代码可以被映射到相应的 Unicode 值。

Unicode 定义为世界上中的最标值，及一个应用程序特定的字符私人使用领域。

有关 Unicode 信息可以在 Unicode 标准发现，Unicode Consortium（见参考文献）。

映射到 Unicode 字符代码值的方法介绍 第 5.9.1 节“映射到 Unicode 的字符代码值。”标记的 PDF 制作应确保 PDF 文件包含足够的信息以使 Unicode 字母代码能映射到 PDF 文件。

ALT 键, ActualText, 或 E 项中指定的一个或一个结构元素字典 标记内容性质清单 (见第 10.8.2 节, “替换说明,” 10.8.3, “替换文本” 和 10.8.4 节, “扩大缩略语”), 可能影响一些消费者的实际标记的 PDF 的字符流的使用。例如, 一些消费者可能会选择使用 Alt 或 ActualText 值, 而忽略所有文字和其他内容相关的结构元素及其衍生物。

有些使用的标记的 PDF 需要的字符, 可能无法适用于所有的字体, 如一些软连字符 (816 页 “附带文物”)。这些字符可以表示为通过将它们添加到字体的编码的方式或 CMAP 和 使用 ToUnicode 将它们映射到适当的 Unicode 值, 或使用 ActualText 在关联的结构元素提供替代字符。

字体特征

除了 Unicode 值, 内容流中的每个字符代码有一个关联, 设置字体的特点。这些特征是有用的当输出字符到另一个应用程序或文件格式, 有一个有限的乐曲和有用的形式

表 10.18 列出了与 CSS 及 XSL 相关的一系列字体特征。每一特征可以从信息中推导出字体描述的标志项 (请参阅 5.7.1 节, “字体描述项”)。

表 10.18 推导字体特征

特征	类型	起源
Serifed	布尔	在字符描述标志条例中的 Serifed 标志的值
Proportional	布尔	在字符描述标志条例中 FixedPitch 标志的补充
Italic	布尔	在字符描述标志条例中 Italic 标志的值
Smallcap	布尔	在字体描述标志条例中的 SmallCap 标志的值

注：本表中的特征表现为字符代码只适用于存在在内容中显示的字符串流。他们不存在于替代说明文字 (ALT) 的，替换文字 (ActualText)，或缩写扩展文本 (e) 项。

注：标准 14 Type 字体，字体描述可能会丢失；

标记的 PDF 1.5 PDF 格式定义了更宽层次的字体特征，这些字体特征为将 PDF 转换成类似 HTML, XML 和 OEB 格式提供了需要的信息，并保证了内容的可及性和回流。表格 10.19 列出了这些字体选择属性，并显示他们的值是如何产生的。

注：如果 fontFamily 中，FontWeight 和 FontStretch 字段不是在字体描述符，这些值在一个执行定义方式中产生于字体名称。

字体选择属性表 10.19

属性	描述
----	----

FontFamily	一个字符串，指定首选字体系列名称。来自 fontFamily 中的字体描述项（见表 5.19 -426 页）。
GenericFontFamily	分类的通用字体，在 fontFamily 未找到时使用。下面的值提供支持；有两个例外，它们可以来自字体描述的标志项： <ul style="list-style-type: none">* 衬线：所选如果衬线标志设置和 FixedPitch 和脚本没有设置标志* SansSerif：所选如果 FixedPitch, 脚本和 Serif 标志都没有设置* 行草：所选如果脚本标志设置和 FixedPitch 标志未设置* 等宽：所选如果 FixedPitch 标志被设置* 装饰：不能得出* 标志：不能得出
FontSize	字号的字体大小：一个明确的定点数字指定的字体高度。这来自当前 A, B, C 和 D 文本域矩阵。
FontStretch	字体的伸展值，来自字体描述中的 FontStretch（见 426 页表格 5.19）。
FontStyle	字体 italicization 值。如果倾斜标志设置在标志字段字体的描述，它被设置为斜体。否则，它被设置为正常字体。
FontVariant	小型股价值的字体。如果小型股指标的设置在标志领域的字体描述，它被设置为小型股。否则，它被设置为正常字体。
FontWeight	重量（厚度）字体的价值。它来自字体描述符中的 FontWeight（见 426 页表 5.19）。

字符识别

文档的文本流定义，不仅存在于网页的文本字符，也同样存在在文字中。不像一个字，一个词的概念不好单独准确定义，而是取决于该文本正在处理的目的。工具需要一个回流以确保其可以分解成线运行的文本；一个文本到语音引擎识别可以发音的文本；以及拼写检查工具和其它应用程序，都有自己的想法。标记的 PDF 文件能否制定一个单纯，明确的定义文字来满足客户的要求并不重要。重要的是每个客户在作出决策时确保可以获得足够的信息。

标记的 PDF 文件使用者通过系统地检验 Unicode 字符流来找寻文字。这可能与“替换文本”（带 ActualText）有冲突（见第 10.8.3 节，“替换文本”）。消费者不用根据信息在页面上的字形定位、字体变化或字体大小来猜测字符间的停顿，而值得肯定的是确保自保字符间距能够出现在纯文本或是标记的 PDF 文档来分割文字。

值得注意的是，文字由什么组成与文本如何组织起来显示字符串无关。由其值得注意的是，在一个字符串结尾时，如果有断字的需要，那么空格和文字断字符仍然需要。

注：某些应用程序单纯根据字符空格来识别文字。某些其他程序会稍微复杂，通过字号或者是一些标点符号来区分文字，例如将连字号或者标点来作为字的分隔。还有一些应用，可以从现有的 Unicode 之一通过使用类似识别算法的换行（在 Unicode 标准附件 #29，文字边界，）来识别。（见参考文献）。

10.7.2、基本布局模型

标记 PDF 的标准结构类型被解释为一个基本布局模型的内容，描述了结构元素在页面上安排的框架。这种模式是为了捕捉文件的基本结构，并不一定对应一个实际的页面布局使用的应用程序创建的文件。（PDF 内容流指定确切的外观。）我们的目标是提供足够的资料，标记的 PDF 消费者做出自己的决定，同时保持紧密的布局作为自己的布局模式的应用。

注：该标记 PDF 的布局模型类似标记语言中使用的那些 HTML, CSS, XSL 和 RTF，但与其中任何一个都不完全一致。该模型定义是有意让松散的结构元素和属性的解释在合理的纬度转换为其他文件格式。在生成的布局有些程度上变化从一个格式到另一种的方法是可以预料的。

基本布局模式开始于一个参考范围的概念。这是一个矩形区域，被一个布局应用作为一个框架或指导在其中放置文件的布局的应用程序。一些标准的结构和属性，如 StartIndent 和 EndIndent (见 BLSEs 的布局属性 - 851 页)，是由该地区的边界参考布局属性测量。参考地区没有规定明确但从上下文推断。这些利益通常是列在一般地区或文本布局方面，外包围盒的一个表和其组成单元，和一个插图或其他浮动元素的边界框。

标准的结构类型根据他们扮演的角色在页面布局中分为四大类：

分组元素 (见分组元素- 827 页) 组序列或其他成层次结构的元素，但对内容没有直接的布局影响。

块级的结构元素 (BLSEs) (见 829 页“块级的结构元素”) 描述了页面内容的总体布局，在块的进展方向迈进。

内联级别的结构元素 (ILSEs) (见 834 页“内联结构元素”) 描述了含有 BLSEs 内容的布局，在直列进展方向上。

插图元素（见插图元素 - 841 页）是内容紧凑顺序的，在页面内容上，这被认为是关于页面布局单一对象。一个例子可以被视为任何一个 BLSE 或 ILSE。

该条款的意义块进展方向和直列进展方向取决于所使用的文字系统，作为指定的标准属性 `WritingMode`（见“总体布局的属性” - 846 页）。在西方的书写系统，块方向自上而下和内联的方向是从左至右。其他书写系统使用了内容不同的方向。

由于进化的方向可以有不同的书写系统，边缘区和页面上的指示，必须有确定的条款方面的进展为中性，而不是由熟悉的词语，如向上，下，左，右。由前至后的收益座布局，内联从开始到结束。因此，例如，在西方的书写系统，在顶部和底部，分别为前，后的参考区域边缘，开始和结束的边缘在左，右。另一个术语中，转移方向（即转变为上标方向），指的是相反的方向进展，即由后到前（在西方文字系统从下到上）。

BLSEs 用参考地区被堆积在块进展秩序中。一般来说，第一个 BLSE 放在对的参考范围前的边缘。随后 BLSEs 对前面的堆积，对以后的边缘前进，直到没有更多的 BLSEs 在参考地区适合。如果溢出 BLSE 允许自己被分裂，例如可以文字之间的分裂“的部分段落”可以包含在当前的参考区域，其余结转到以后的参考区域（或者其他地方一样或在另一页的文件的页面）。一旦内容，在确定参考区域适合内容，个人 BLSEs 展示位置可以调整，以纠正偏差，在边缘上，中间，或参考地区边缘之间的间距之后，BLSEs 可以调整，以填补参考区的全部情况。

注：BLSEs 可以嵌套，与堆放在与 BLSEs 同样的方式在一个父 BLSE 参考区域内的孩子 BLSEs。除了在少数情况下注意到下面（即 `BlockAlign` 和 `InlineAlign` 元素），这样的 BLSEs 不会导致参照区筑巢，一个单一的参考面积为嵌套 BLSEs 各级为准。

在 BLSE 里，子 ILSEs 被装成线。（直接的内容项目“那些都是 BLSE 直接的子元素，而不包含在子 BLSE 里”——被含蓄地视为包装用途 ILSEs。）每一行被视为是一个综合 BLSE，在父 BLSE 里堆积。线可能混杂着其他领域内的父 BLSEs。该生

产线的建设过程是类似与 BLSE 堆叠内的参考面积，除了它的直列进展收益，而不是块进展的方向：一个线是挤满了含有 BLSE，从包含了 ILSEs 的边缘开始，持续到年底达到边缘和饱和。充足的 ILSE 可能使自己在语言上决定或明确标明破发点，(如一个字的断点内)，剩下的片段是结转到下一行。

注：一个元素的放置属性的一定值从正常的包或正常堆积中删除了元素，并允许它不是浮动到一个封闭的职权范围或父 BLSE，见 846 页的总体布局属性作进一步讨论。

内附两个矩形，与每个 BLSE 和 ILSE 相关(包括直接的内容，被视为隐含的 ILSEs 项目)：

矩形的内容是从封闭的内容和形状得来，并定义为任何包含子元素的布局使用的范围。

矩形的分配包括任何额外的边界或间隔周围的元素，影响到它是如何定位方面的相邻元素含量和封闭矩形或参考区域。

这些矩形的定义是由相关的布局确定的属性与结构元素，见第 8 页“内容和分配矩形”，作进一步讨论。

10.7.3、标准结构类型

标记 PDF 的标准结构类型描绘了文档内部的内容元素，并与标准文件属性相结合的作用。(详见第 10.7.4，标准结构属性-) 这些内容如何在页面上阐述。正如第 10.6.2 节所述，“结构类型”逻辑结构元素的结构类型，被结构元素词典里的 S 条目指定。被认为是标准的结构类型，该值必须是下面二者之一：

如下说描述的一个标准的结构类型说明之一。

一个任意名，通过文件作用图映射到标准名称之一（见第 10.6.2 节，结构类型），很可能通过映射的多重水平。

注：从 PDF1.5 开始，元素名称总是映射到与它相对应的作用图中的名字，如果有一个元素名，即使原来的名称是标准类型之一。这样做是为了让元素，例如，表示一个标签，用相同的名称作为标准作用，即使它的使用不同于标准作用。

一般来说，结构要素拥有的标准结构类型用相同的方法处理，不论类型是直接表达的或是由角色图间接决定的。然而，有些消费者应用程序可能归节另外的语义到非标准结构类型。甚至通过角色地图用标准的结构类型与他们关联。例如，当输出到一个如 XML 的标记语言时，S 项的实际值可以被使用，当转化到诸如 HTML 或 RTF 的表示形式，或为了残疾人的无障碍使用时，相对应的映射作用值可以被使用。

注：大多数标准元素类型首先被设计成为铺设的文字；使用的术语反映了这一点。然而，布局实际上可以包含任何类型的内容，如路径或图像对象。与结构相关的内容项目元素，载列于一页，就像他们是文本块（用于 BLSE）或是一个文本行字符（ILSE）

分组元素

分组元素仅用于分类其他结构元素，他们不直接与内容项目相关。表 10.20 描述了这个类中的标准结构类型。

对于大多数内容提取格式，该文件必须是一个单个顶层元素的树；结构树的根（被文档目录中的 StructTreeRoot 项验证）必须只有一个孩子在其 K（孩子）

队列。如果 PDF 文件包含一个完整的文档，结构类型文件被推荐为逻辑结构层次中的顶层元素。如果文件包含格式良好的文档片段，结构类型 Part, Art, Sect, or Div 其中之一可以用来替代。

表 10.20 分组元素的标准结构类型

结构类型	描述
Document	(Document) 一个完整的文档。这是任何包含了多个 part 和多个 article 的结构树的根元素。
Part	(Part) 大规模的文档分组。元素类型对分组文章和章节是适当的。
Art	(Article) 一个单叙述或论述的文字构成的相对自足的客体。文章应该是不相交的，也就是说，他们不应当含有其他文章构成要素。
Sect	(Section) 与内容元素相关分组容器。例如，一个章节可能包含一个标题，几个引言，两个或更多的嵌套章节作为他的子章节。
Div	(Division) 一个普通的块级元素或元素组。
BlockQuote	(Block quotation) 一个或多个段落组成的文本部分，归因于某个人而不是周围文本的作者。
Caption	(Caption) 用来描述一个表或图形的简短的一部分。

TOC	(Table of contents) 一个列表项内容组成 (struc-TOCI; 见下文) 和/或其他内用表元素。一个内容表因此是潜在分层次的。理想情况下，层次与文件中的主体的结构到结构的层次相对应。 注：数字和表格清单，以及书目，可视为内容表结构为目的的标准类型。
TOCI	(Table of contents item) 一个内容表的单个成员，其子女可能包括一个标签 (结构类型标签，见 831 页表元素) 参考的标题和页码 (结构类型参考，见页 834 内联级结构的有限元) NonStruct 元素 (包装一个领导者的材料，例如，见 827 页“组元素”)，说明文件 (见结构类型 P，见 830 页 Paragraphlike Elements)

结构类型	描述
Index	(Index) 包含了识别文件的项顺序伴随着参考元素 (结构类型参考，见 834 页“在线层结构元素”) 指出了制定文件发生在文件的主体。
NonStruct	(Nonstructural element) 一个分组元素没有内在结构上的意义，它完全是为了分组的目的。这个元素类型不同于分组 (Div 结构类型，见上面) 在这它不是解释或出口到其他文件格式，但它的后代是正常处理

Private	(Private elements) 一个分组元素包含的私人内容属于生成它的应用程序，这种类型的元素结构的意义不明，是由生产应用程序决定。既不是私人因素，也不是它的后代，是任何解释或输出到其他文件格式。
---------	---

块层结构元素

一个块级结构元素 (BLSE) 是任何文本区域或其他内容，在块进展方向布局，如段落，标题，列表项目或脚注。一个结构元素是一个 BLSE，如果其结构类型（在角色映射之后，如果有的话）是在表 10.21 所列之一。所有其他的标准结构类型被视为 ILSEs，但下列情况除外：

TR (表列)，TH (表头)，TD (表数据)，THead (表头)，TBody (表体)，和TFoot (表页脚)，那些用于表中族元素，并被认为是既不是 BLSEs 也不是 ILSEs。

与安置的属性元素（见 846 页“总体布局的属性”）而不是内联的默认值。

表 10.21 块层结构元素

种类	结构类型		
象段元素	P	H1	H4
	H	H2	H5
列表元素	L	Lb1	
	LI	Lbody	
表元素	Table		

很多情况下，一个 BLSE 显示为一个紧凑，连续的一块页面内容，在其他情况下，它是不连续的。后者的例子包括了一个 BLSE，它横跨了一个页面边界或者在页面内容次序中被其它的打断，嵌套的 BLSE 或直接包含脚注。必要时，标记 PDF 格式的消费者应用程序可以从逻辑结构识别这种片段的 BLSEs 并用这种信息去重组他们和合理的裁员。

象段元素

表 10.22 描绘了由运行文本和在其它常规段落形式指定的内容组成的像段元素的结构类型。(相对于更专业的布局如列表和表格)。

表 10.22 象段元素的标准结构类型	
结构类型	描述
H	(Heading) 一个文件的内容细分标签，它应该是他带领的细分的一个孩子。
H1-H6	特定层次的标题，在应用中使用，不能分级构筑起他们的章节并因此不能从他们构筑的层次决定头层次
P	(Paragraph) 文本的底层划分。

列表元素

表 10.23 说描绘的结构类型被用来组织列表内容。

表 10.23 列表元素的标准结构类型	
结构类型	描述
L	(List) 条目的喜欢的意义和次序的重要性。它的直接孩子应该是一个可选的标题（结构类型标题，见第 827 页中元素分组）随之是一个或多个列表项（结构类型 LI；见下面。）
LI	(List item) 一个列表的单个成员，它的子女可以是一个标签，团体名单，或者两者（结构类型 Lbl 或 Lbody，见下面）
LBl	(Label) 一个名字或数字，在同一个列表或其它像项目组里从其它的地方辨别出一个给定条目。在一个词典列表，例如，它包含了被定义的期限，在一个项目符号或编号的列表中，它包含项目符号字符或列表数字及相关标点符号。
LBody	(List body) 一个列表项描述的内容。在词典列表，例如，它包含了这个名词的定义。它可以直接包含内容，或有其他 BLSEs，也许包括嵌套列表，比如子女。

表元素

10.24 表描绘的结构类型被用来组织表内容。

注：严格来说，表元素是 BLSE，在此表中的其他既不是 BLSEs 也不是 ILSEs。

表 10.24 表元素的标准结构类型

结构类型	描述
Table	(Table) 矩形数据单元的二维布局很可能有一个复杂的基础，它包含一个或多个表列（结构类型 TR，见下面）作为子女，或者一个可选的表头（结构类型 Thead；见下面）随之是一个或多个表体元素（结构类型 Tbody，见下面）和一个可选的表页脚，（结构类型 TFoots，见下面），此外，一个表可以有一个可选的标题（结构类型标题，见 827 页“组元素”）作为它第一个或最后一个子女。
TR	(Table row) 表格中的标题或数据行，它可能包含表头单元和表数据单元（结构类型 TH 和 TD，见下面）

TH	(Table header cell) 一个表单元包含了一个描绘一个或多个表的行或列的头文本。
TD	(Table data cell) 一个包含了数据的表单元，这个数据是表内容的一部分。
Thead	(Table header row group; PDF 1.5) 构成表头的一个行组，如果表是从多个页面分裂的，这些行可能会在每个表的片段上面重画。(虽然只有一个 Thead 元素)
Tbody	(Table body row group; PDF 1.5) 表的主体部分组成的一个行组，如果该表是从多个页面分裂的，主体可能破裂从行列边界分离。一个表可以有多个 Tbody 元素，以便画一组行的边境或背景。
Tfoot	注：(Table footer row group; PDF 1.5) 表的注脚组成的一个行组，如果该表是从多个页面分裂的，这些行可能会在每个表的底部重绘。(虽然只有一个 TFoot 元素)

注：在与行和数据列标题的关联典型的是由应用程序探索式地决定。这种探索可能由于复杂的表格而失败，表 10.35 所示的表的标准属性可以被用来制作详细的关联。

块级结构的使用准则

由于不同的消费应用程序用不同的方法使用 PDF 的逻辑结构设施，标记的 PDF 不执行任何有关该命令的严格规定并且元素嵌套使用标准的结构类型。此外，每一个输出格式有它自己的逻辑结构习惯。然而，坚持某些一般准则有助于

在不同标记的 PDF 消费者中实现最稳定的和可预见的准则。

正如 827 页“组元素”说描述的，一个标记的 PDF 文档可以有一个或多个分组元素层，例如 Document, Part, Art(Article), Sect (Section), and Div (Division)。这些的后代是 BLSEs，例如 H (标题)，P (段) 和 L (表)，他们持有的实际内容。他们的后代，反过来，要么是内容项目，要么是 ILSEs，进一步描述了内容。

注：如前所述，元素与结构类型，被普遍的认为是 ILSEs 的，可以作为一个 Placement 属性（见 846 页的“总体布局属性”，）这将导致他们被视为 BLSEs 代替。这些元素可以作为 BLSEs 归类，以象标题和段落同样的方式。

块层结构可以跟随两个主要模式之一：

坚固结构。分组元素的筑巢尽可能多层次的反映到文章，节，小节材料等等的组织。在每个层面，分组元素的子元素包括标题 (H)，一个或多个段落 (P)，在这一水平内容 (P)，或许一个或多个组元素的嵌套小节内容。

弱结构。该文件是较平坦的，有可能只有分组元素的一两个级别的，所有的标题，段落，及其他 BLSEs 作为其直接子女。在这种情况下，材料的组织不是反映在逻辑结构上，但它可以通过与具体层的标题的使用表达出来 (H1-H6)。

这种坚固的结构模式是被一些基于 XML 的富裕的文件为使用。弱结构模式是典型的以 HTML 为代表的文件。

列表和表格应使用特定的组织结构类型描述，831 页的目录元素和表元素。同样地，

内容表和索引应该象根据 872 页“分组元素”为描述 TOC 和索引结构类型那样的方式被构造。

内联级结构要素

一个内联级别的结构元素 (ILSE) 包含文字或其他内容的部分，他们在文档

中有具体的造型特点或在文档中发挥了特定作用。在一个段落或含有 BLSE 定义的其它块，连续的 ILSEs——可能与其他内容混合项目，那些是父 BLSE 直接的孩子——在内联进程方向连续载列。(在西方书写系统从左到右)。由此产生的内容可能被分裂为多行，这反过来又是在块的方向进展堆积。一个 BLSE 很可能反过来包含一个 BLSE，这被视为一个在内联方向单一的项目布局。表 10.25 列出了 ILSEs 的标准的结构类型。

表 10.25 内联结构元素的标准结构类型

结构类型	描述
Span	<p>(Span) 一个文本的通用的内嵌部分没有特别的固有特征的文本。例如，它可用给定的设置属性的样式划定文本的范围。</p> <p>注：不是所有的内联样式的变化都需要跨度确定。文本颜色和字体的变化（包括修改者如粗体，斜体和小盘）不用那么明显，因为这些可以从 PDF 的内容派生出来（见页 820 “字体特征”）。然而，有必要使用跨度申请明确的布局属性，如 LineHeight，BaselineShift，或 TextDecorationType（见 855 页 ILSEs 的布局属性）。</p> <p>注：标记内容序列所有的标签跨度也被用来带有一定的属性（ALT，ActualText，Lang，和 E，见 10.8 节“无障碍支持”）。这种序列缺乏 MCID 性质并与任何结构元素不关联。这种使用标明内容标记标签与它作为一个结构类型的使用是不同的。</p>

结构类型	描述
Quote	<p>(Quotation) 内嵌的文字归因与某个人，而不是周围文本的作者。</p> <p>注：引用文字被包含在单个段落。这不同于块层元素 BlockQuote（见 827 页“组元素”）这组成了一个或多个完整段落（或其他元素表现为好像他们为完整的段落。）</p>
Note	<p>(注) 解释说明文字项，如脚注或尾注，是参考了文档的正文处。它可能有一个孩子标签（结构类型 LBI，见 831 页列表元素）。该注明可作为结构的孩子在正文中引用它，或者它可能被包括在其他地方，（如在第一个尾注）和访问的一种手段（结构类型的引用，见下文）。</p> <p>注：标记的 PDF 并没有规定页内容顺序中注脚的位置。他们可以是内联或在网页的末端，在生产应用程序的谨慎处</p>
Reference	<p>(Reference) 文档中一个引用到其它地方的内容</p>
BibEntry	<p>(Bibliography entry) 一个参考文献确定的一些引用内容的外部参考源</p> <p>它可能包含一个标签（结构类型 LBI，见 831 页“列表元素”）作为它的子女。</p> <p>注：虽然书目项可能包括组件部分识别了引用内容的作者，工作，出版商等等，没有标准的结构类型定义在这个细节层次上，在这个出版时间上。</p>
Code	<p>(Code) 一个计算机程序的文本片段</p>
Link	<p>(Link) 一个关联在 ILSEs 的内容的部分和相应的链接注释或说明（见 587 页链接注解）。其子女是一个或多个内容项目或孩子 ILSEs 和一个或多个对象引用（见 796 页“PDF 对象为内容项”，）确定相关链接注释。</p>

见“链接元素”，下文作进一步讨论。

结构类型	描述
Annot	(Annotation; PDF 1.5) ILSEs 内容的部分和相应的 PDF 的注释之间的关联（见 8.4 节注解）注释被用于所有的 PDF 注释说明，除了连接注释（见上面的链接元素段）和小元件说明（参见 841 页 10.27 表中的表格元素）。见 839 页“注释元素”作进一步讨论。
Ruby	(Ruby; PDF 1.5) 写在一个较小的文本上的边界说明（注释），放在旁边的基地文本。它是用来在日语和中文来描述不平常的发音，去描绘这样的条目作为编写和标志。一个 Ruby 元素可能同样包含 RB, RT 和 RT 元素，见 839 页“Ruby and Warichu Elements”作更

详细了解。

Warichu (Warichu; PDF 1.5) 在一个较小的文字大小上的注释或批注和在两个较小的线路上格式化的高度上包含行文本和放在以下（内嵌）基础文本。它是指在日语中描述意见和 ruby 注解文字，这样被过长美观地格式化。一个 Warichu 元素可能也包含 WT 和 WP 元素，见 839 页“Ruby 和 Warichu 元素”作详细了解。

链接元素

链接元素（像所有 PDF 链接注释说明）与页面的几何元素相关联，而不是内容流中的特定客体。任何链接和内容之间的联系是纯粹的基于视觉外观而不是一个明确指定的关联。基于这个原因，单独的链接注释对有视觉障碍的用户是没有用的，或者对必要的应用程序去决定哪些内容可以被激活去调用一个超文本链接。

标记的 PDF 链接元素（结构型链接）使用 PDF 的逻辑结构设施去建立项目内容和链接说明之间的联系，提供功能相当的的 HTML 超文本链接。下列项目可以是一个链接元素的子元素：

一个或更多的内容条目或其它 ILSEs (除了其它链接)
对象引用（见 796 页“内容项目对象”）对一个或更多的与内容相关的链接注释。

一个链接元素可以包含多个链接注释，如果几何内容需要它。例如，如果一个文本跨度从一行结束另一行开始，单独的链接注释可能需要覆盖两种文本部分。所有的子链接注释都必须具有相同的目标和行动。为了保持内容和注释之间

的几何关联，这些是逻辑关联的。所有的链接元素的内容必须被其子女的链接注释的联合覆盖。

作为一个例子，考虑一下的 HTML 代码片段，它产生一个含有超文本链接文本行。

```
<html>
  <body>
    <p>
      Here is some text <a href="http://www.adobe.com">with a link</a> inside.
    </p>
  </body>
</html>
```

例 10.17 显示了一个相当的使用链接元素的 PDF 片段，其文本将用蓝色显示并强调。例如 10.18 显示了一个从逻辑结构层次相关联的摘要。

例 10.17

```
/P<< /MCID0 >>          % 标记内容序列 0(段落)
  BDC                      % 开始标记内容序列
    BT                      % 开始文本对象
      /T1_01 Tf            % 设置文本字体和大小
      14 0 0 14 10.000 753.976 Tm  % 设置文本模型
      0.0 0.0 0.0 rg          % 设置 nonstroking 颜色为黑色
      (Here is some text )Tj        % 显示文字链接
    ET                      % 结束文本对象
  EMC                      % 结束标记内容序列

/Link<< /MCID1 >>        % 标记内容序列 1 (链接)
  BDC                      % 开始标记内容序列
    0.7 w                  % 设置线宽
    [ ] 0 d                % 固体短跑模式
```

```
111.094 751.8587 m % 移动到强调的开始
174.486 751.8587 l % 画下划线强调
0.0 0.0 1.0 RG % 设置 stroking 颜色为蓝色
S % Stroke 强调
```

```
BT % 开始文本对象
14 0 0 14 111.094 753.976 Tm % 设置文本模型
0.0 0.0 1.0 rg % 设置 nonstroking 颜色为蓝色
(w with a link)Tj % 显示链接文本
ET % 结束文本对象
EMC % 结束标记内容顺序
```

```
/P << /MCID 2 >> % 标记内容顺序 2 (段落)
BDC % 开始标记内容顺序
BT % 开始文本对象
14 0 0 14 111.094 753.976 Tm % 设置文本矩阵
0.0 0.0 1.0 rg % 设置 nonstroking 为黑色
( inside.)Tj % 显示文本链接
ET % 结束文本对象
EMC % 结束标记内容顺序
```

例 10.18

```
501 0 obj % 段落的结构元素
<< /Type /StructElem
/S /P
...
/K [ 0 %三个子女: 标记内容顺序 0
502 0 R % 链接
2 %标记内容顺序 2
```

```
]  
>>  
Endobj  
  
502 0 obj %链接的结构顺序  
<< /Type /StructElem  
/S /Link  
...  
/K [ 1 % 两个子女: 标记内容顺序 1  
503 0 R % 对象引用链接注释  
]  
>>  
Endobj  
  
503 0 obj %对象引用链接注释  
<< /Type /OBJR  
/Obj 600 0 R % 链接注释 (不显示)  
>>  
Endobj
```

注释元素

标记的 PDF 注释内容（结构类型注释，PDF1.5）使用 PDF 的逻辑结构设施，以建立内容项目和 PDF 注释之间的关联。注释元素是用于说明注释类型而不是链接（见链接元素，836 页）和表格（见页 841 表 10.27）。

下列项目可以是一个注释元素的子元素：

对象引用（见 PDF 对象为内容项，796 页），以一个或更多的注解词典。

或者，一个或多个内容的项目（如标明内容序列）或其他与相关注释的 ILSEs（除其他说明）。

如果一个注释没有子女以外的其他对象引用，它的译文是由注释的外观定义的，它的文本内容被视为好像它是一个 Span 元素。它可能有一个可选的 BBox

属性；如果有提供的，这覆盖了注解词典指定的矩形属性。

如果注释元素有是内容的项目子女，这些子女代表了显示的注释形式，以及相关的注释的外观，还可以应用（例如，有一突出显示注释）。

是可以有不同的说明对象引用多个孩子，受约束的说明必须是相同的，除了矩形项。这是许多作为是为完成相同的链接元素，它允许一个注解与不连续块的内容，如自动换行的文本。

Ruby 和 Warichu 元素

Ruby 的文本是边注，用一个较小的书面文字写的，放在旁边的它说指向的基础文字。这是用日语和中文来描述不寻常的单词的发音或描述作为缩写和标志等项目。

Warichu 文字是注释或批注，写在一个较小的文本大小和上两个小线条格式内包含文本行和高度放在以下（内嵌）的它说指向的基础文本。在日本，它是用来描述意见和 Ruby 注释文本，这是难以完美地格式化。

表 10.26 Ruby 和 Warichu 的标准结构类型

结构类型	描述
Ruby	(Ruby) 围绕着整个 ruby 组装的包装。它包含一个 RB 元素跟随着一个 RT 元素或 3 元素小组，组成的 RP，RT 和 RP。Ruby 元素及其含量元素可能不会被多行打跨。
RB	(基于文本的 Ruby) 全尺寸文本的 Ruby 注释应用。RB 可以包含文字，其他内联元素，或两者的混合物。这可能有 RubyAlign 属性。
RT	(Ruby annotation text) 较小尺寸的文本放在与 ruby 基本文本相邻的地方。它可以包含文字，其他内联元素，或两者的混合物。它可能有 RubyAlign 和 RubyPosition 属性。

RP	(Ruby punctuation) Ruby 注释文本周围的符号，我们只用当 ruby 注释不能是一个格式正确的 ruby 风格，取而代之的是作为一个正常的评论，或当它作为一个 warichu 格式。它包含文本（通常是一个单一的开放或关闭插入，或类似的包围括号字符）
Warichu	(Warichu) 围绕着整个 warichu 装配包装。它可能包含由 WP, WT 和 WP 组成的 3 个元素组。Warichu 元素（及其内容元素）可以跨越多行包，根据日本工业标准（JIS）X 4051-1995 中 warichu 突破规则的描述。
WT	(Warichu text) 一个 warichu 评论的小尺寸文本，格式化为两行之间，围绕 WP 元素放置。
WP	(Warichu punctuation) WT 文本周围的标点符号。它包含文本（通常是一个单一的开放或关闭，或括号类似的括号字符）。据日本 JIS X 4051-1995 标准，包围了 warichu 的括号可能会被转换为空格（名义上宽度的 1 / 4EM）在格式化决定。

插图元素

标记的 PDF 定义了一个任何结构元素，其结构类型（角色映射后，如果有的话）是在表 10.27 所列的其中一个例证元素。插图内容必须由一个或多个完整的图形组成。它可能不会出现在 BT 与 ET 间划定一个文本对象（见第 5.3 节，文本对象）。它可能包括裁剪，只在一个包含标记剪辑序列的形式，如第 10.5.2 节定义的，标签 PDF 中标记的内容和剪辑，所有这些标记必须进行剪辑序列标记的内容标记剪辑。

表 10.27 插图元素的标准结构类型

结构类型	描述
Figure	(Figure) 一个图形内容的项目。其位置可以指定为安置布局属性（见总体布局的属性 - 846 页）。
Formula	(Formula) 一个数学公式。 注：这种结构类型是有用的，只是确定一个公式作为整个内容元素。 没有标准定义的结构类型，在公式中确定各个组件。从格式的角度来看，该公式被视为类似于一个数字（结构类型图；见上文）。
Form	(Form) 小装置注释代表了互动的窗口域。（见第 8.6 节，互动形式 - ）。 它的唯一的子元素是一个对象，并未重新提及（见 PDF 对象的内容项，796 页）。注释外观流（见第 8.4.4 节，外观流）定义了表单元素的渲染。

一个插图可能有逻辑子结构，包括其他插图。为了回流，但是，它被作为一个单元移动（调整大小）而没有研究其内部内容。为了获得回流有用，它必须有一个 **BBox** 属性。它或许还有安置，宽度，高度和 **BaselineShift** 属性（见布局属性 - 845 页）。

常常是一个例证在逻辑上是一个段落或一个文档的其他元素一部分的，或至少重视。任何此类遏制或附件是代表通过使用图的结构类型。该图显示了元素的附着点，并且其放置属性描述了该附件的性质。一个没有画像元素的安置属性被

视为一个 ILSE 或阐述内联。

注：对于无障碍残疾人和其他文本提取用途的用户，一个例子元素应该始终有一个 ALT 键进入或 ActualText 项（或两者）的结构元素词典（见第 10.8.2，交替说明，和第 10.8.3 节，替换文字）ALT 键是一个插图的说明，而 ActualText 给出了一个图形化的确切文字，同等画像，有文字的外观。

10.7.4 标准结构属性

除了标准结构类型，带标签的 PDF 为其他类型的结构元素定义了标准层和式样属性。这些属性使得比如把 PDF 内容的输入输出到其他文档格式在操作中有可预见的格式被应用。

就像节 10.6.4 “结构属性” 中所讨论的，属性在属性对象中被定义。它们是字典或者流在下面两种方式中跟结构元素相联系：

结构元素字典中的 A 入口识别属性对象或者是这类对象的数组。

结构元素字典中的 C 入口给出属性类或者这类名字数值的名字。这些类名能够在类图中按顺序被查找到，一个字典在结构树根中用 ClassMap 入口中被识别，跟随一个属性对象或者跟类相关的对象数组。

除了标准结构元素，下面还有很多其他可选的入口——Lang, Alt, ActualText, 和 E 将在节 10.8，“辅助功能支持”，但是对其他的 PDF 文档也很有用。它们出现在 PDF 的下列位置（而不是在属性字典中）：

在结构元素字典中作为入口（见 787 页表 10.10）

在跟带有跨度标签的标记内容句子有关的属性列表中的作为入口（见节 10.5，“标记内容”）

例 10.15 说明了标准结构属性的应用。

标准属性的拥有者

每一个属性对象都有一个被对象的 0 入口指示的拥有者，它决定在对象字典中属性的阐述。多个拥有者也许会定义类似名字的属性，但是带有不同数值类型或阐述。带标签的 PDF 定义一些的标准属性拥有者，见表 10.28。

表 10.28 标准属性拥有者

拥有者	描述
Layout	内容层的属性管理
List	列表数的属性管理
Table	表格中单元格组织的属性管理
XML-1.00	转换成 SML 的附加属性管理，版本 1.00
HTML-3.20	转换成 HTML 的附加属性管理，版本 3.20
HTML-4.01	转换成 HTML 的附加属性管理，版本 4.01
OEB-1.00	转换成 OEB 的附加属性管理，版本 1.0
RTF-1.05	转换成微软复文本格式的附加属性管理，版本 1.05
CSS-1.00	转换成用 CSS 格式的附加属性管理，版本 1.00
CSS-2.00	转换成用 CSS 格式的附加属性管理，版本 2.00

当一个属性对象属于一个特殊的导出格式比如 XML-1.00，那么它只能在导出 PDF 内容为那种格式的情况下使用。这个特殊格式属性能覆盖任何有 Layout, List, 和 Table 拥有的相关的属性。这有可能增加特殊格式属性；一系列有可能的属性是开放式的，并且没有被带标签的 PDF 指定或限制。

属性数值和继承

一些属性被定义成可继承的。可继承属性传播结构树；也就是说元素指定的属性能被应用于所有在结构树中元素的子节点除非这个子元素为属性指定一个明确的数值。

注意：在本节中每个标准属性的描述都指定这个数值是否可继承。

为了传递它的数值给子元素，指定一个可继承的属性是允许的，甚至这个属性对父元素来说是没有意义的。不可继承的属性只有对元素有意义才被指定。

下面的列表给出设置属性数值的优先级。一个处理应用设置属性数值给列表中的第一个项目必须遵循：

- 1、在元素的 A 入口，属性的数值通过导出格式（比如 XML, HTML-3.20, OEB-1.0, CSS-1.00, CSS-2.0 和 RTF）被指定，如果存在并且输出成那种格式
- 2、属性的数值在元素 A 的入口被指定，被 Layout, table 或者 List，如果存在
- 3、属性的数值在跟元素的 C 入口有联系的类图中被指定，如果说有的话

- 4、如果属性是可继承的，父结构元素的定值
- 5、属性的默认值，如果有的话

注意：`Lang`, `Alt`, `ActualText` 和 `E` 属性不要再属性字典中出现。应用管理原则将在节 10.8，“辅助支持” 中讨论。

属性的明确指定和可继承性没有语义差别。逻辑上讲，结构树有属性完全束缚每个元素，甚至一些事从原先的元素继承过来的。这是跟属性的表现是一致的（比如字体），它们不是结构属性所指定的，但是从内容中得到的。

属性布局

布局属性指定布局处理的参数用来产生有 PDF 文档内容描述的外观。目录中的属性在属性对象被定义，它的 `O`（拥有）入口有 `Layout` 数值（或者是在 843 页表 10.28 列表中特殊格式拥有者的名字之一）。目的是这些参数能够用于跟随内容后面或者导出成其他文档格式至少保留原先类型的基本。

表 10.32 总结了标准布局属性和应用的结构元素。下面的部分描述这些属性的意思和用途。

注意：属性名字后面带星号 (*) 表示这个属性是可以继承的。就如在 844 页“属性数值和继承”所描述的，一个可继承属性也是被指定为任何元素传递给子元素，不管它对这个元素是否有意义。

表 10.29 标准布局属性

结构元素	属性
任何结构元素	<code>Placement</code>
	<code>WritingMode*</code>
	<code>BackgroundColor</code>
	<code>BorderColor*</code>

	BorderStyle
	BorderThickness*
	Color*
	Padding
任何带 Placement 的 BLSE ILSEs 除了内嵌	SpaceBefore
	SpaceAfter
	StartIndent*
	EndIndent*
BLSEs 包括文本	TextIndent*
	TextAlign*
说明元素（数字，公式，格式）表格	BBox
	Width
	Height

结构元素	属性
TH (表头)	Width
TD (表数据)	Height
	BlockAlign*
	InlineAlign*
	TBorderStyle*
	TPadding*
任何 ILSE BLSEs 包括 ILSEs 或者包括直接或嵌套内容项目	LineHeight*
	BaselineShift
	TextDecorationType
	TextDecorationColor*
	TextDecorationThickness*

整体元素 Art, Sect 和 Div	ColumnCount
	ColumnWidths
	ColumnGap
垂直文本	GlyphOrientationVertical*
红宝石文本	RubyAlign*
	RubyPosition*

一般的布局属性

表 10.30 所描述的布局属性能应用在模板水平 (BLSEs) 和内嵌水平 (ILSEs) 的任何标准类型的结构元素中。

表 10.30 所有标准结构类型共有的标准布局属性

键	类型	数值
Placement	名字	(可选; 不可继承) 元素的位置关于囊括参考区和其他内容: 模块 堆放在块累计方向在内附参考区或者父 BLSE。 内嵌 封装在内嵌累计方向在内附 BLSE 中。

键	类型	数值
---	----	----

以前 放置以便在元素的分配矩形的边缘之前（见 859 页，“连接于分配矩形”）恰逢在最近的内附参考区。如果有需要，元素可以浮动以得到特殊放置（见下面的注解）。元素被当做一块在内嵌方位中内附参考区占领的完整范围。其他内容被堆放以此开始在元素的分配矩形的后面边缘。

开始 放置以便在元素的分配矩形的开始边缘（见 859 页，“连接于分配矩形”）恰逢在最近的内附参考区。如果有需要，元素可以浮动以得到特殊放置（见下面的注解）。被介绍进元素的分配矩的其他内容作为一个圈子被放置。

结尾 放置以便在元素的分配矩形的结束边缘（见 859 页，“连接于分配矩形”）恰逢在最近的内附参考区。如果有需要，元素可以浮动以得到特殊放置（见下面的注解）。被介绍进元素的分配矩的其他内容作为一个圈子被放置。

当在 ILSE 中应用，任何数值除了内嵌使得元素被当做一个 BLSE。默认值：内嵌。

注意：元素带有 Before, Start 或者 End 的 Placement 数值在正常的堆放或者封装过程中被删除并且允许悬浮在内附参考区或者父 BLSE 的特殊边缘。像这种多浮动元素也许彼此相邻，为了反对参考区的特殊边缘或者串行反对边界放置，在遇到顺序时。复杂情况比如相互干涉或者不在同一页面安装的浮动元素在不同的布局应用中会采用不同的处理。带标签的 PDF 只是指定这些元素浮动并且只是它们相应的位置。

键	类型	数值
WritingMode	名字	<p>(可选; 可继承) 布局进展的方向为了 ILSEs (内嵌进展) 的封装和 BLSEs (块进展) 的堆放:</p> <p>LrTb 内嵌进展从左到右; 块进展从上到下。这是一个西方写字系统的经典书写模式。</p> <p>RITb 内嵌进展从右到左; 块进展从上到下。这是一个阿拉伯和希腊写字系统的经典书写模式。</p> <p>TbRI 内嵌进展从上到下; 块进展从左到下右。这是一个中国和日本写字系统的经典书写模式。</p> <p>这特殊布局方向应用于给定结构元素和所有它的子元素在如何嵌套水平。默认为: LrTb。</p> <p>由于元素产生多列, 书写模式定义列在参考区处理的方向: 内嵌方向决定列的堆放方向和从列到列的文本的默认流动顺序。</p> <p>对于表格, 书写方式控制行和列的布局: 表行 (结构类型 TR) 被堆放在块方向, 行当中的单元 (结构类型 TD) 在内嵌方向。</p> <p>注意: 被书写模式指定的内嵌进展的方向取决于在文本被搁置的本地覆盖, 就像在 Unicode 标准附件#9 里所描述的, 来自 Unicode 协会的双向算法 (见书目)。</p>
BackgroundColor	数组	(可选; 不可继承; PDF1.5) 颜色被用来填充表格单元或者任

一元素内容矩形的背景 (通过 `Padding` 属性可以调节)。这个数值是一组在矩形 0.0 到 1.0 中的 3 个数的数组，代表红，绿和蓝数值，分别为 RGB 颜色空间。如果这个属性没有被指定，元素就被认为是透明的。

键	类型	数值
<code>BoardColor</code>	数组	(可选；可继承；PDF1.5) 在表格单元或者任何元素的内容矩形中被画的边界颜色 (用 <code>Padding</code> 属性可以调节)。这个数值是一组在矩形 0.0 到 1.0 中的 3 个数的数组，代表红，绿和蓝数值，分别为 RGB 颜色空间。有两种形式： 三个数值的单个数值代表 RGB 数值应用于四个边界。

一个包括四个数组的数组，每个数值为边界的每个边沿（前，后，开始，结束边沿）指定 RGB 数值。如果任何的边沿的数字是 null，表明没有被涂颜色。

如果这个属性没有被指定，那么这个元素的边界颜色就是跟内容相联系的刚开始的当前文本填充颜色。

BoardStyle 数组或者 (可选；不可继承；PDF1.5) 元素边界的样式。指定每个表格单元或者任何元素的内容矩形边沿的笔画模式(可以用 **Padding** 属性进行调节)。有两种形式：

下面列表中的名字指定应用于四个边沿的边界样式。

四个入口的数组，每个入口指定边界的一个边沿的样式，遵循前，后，开始，结束边沿。如果任何的边沿的数字是 null，表明没有被涂颜色。

None 无边界。强制 **BoarderThickness** 的计算机的数值为 0。

Hidden 有些跟 **None** 类型，除了依据表格元素的冲突解决。

Dotted 边界是一系列点组成。

Dashed 边界是一系列短线组成。

Solid 边界是一系列单一的线组成。

Double 边界是双实线。两条线的总和和之间的空间等价于 **BoarderThickness** 的数值。

Groove 边界看起来像是被雕刻在画布上。

Ridge 边界看起来像是从画布上跑出来。(跟 **Groove** 相反)

键	类型	数值
		<p>Inset 边界使整个盒子看起来像是嵌入在帆布中。</p> <p>Outset 边界使整个盒子看起来像是从帆布中出来。（跟 Inset 的相反）</p> <p>默认值：0</p> <p>注意：所有边界都被涂在盒子的背景的最上面。边界的颜色图画的 Groove, Ridge, Inset 和 Outset 取决于结构元素的 BorderColor 属性和背景被涂的背景的颜色。</p> <p>注意：符合 HTML 应用也许说明 Dotted, Dashed, Double, Groove, Ridge, Inset 和 Outset 变成 Solid。</p>
BoarderThickness	数 值	<p>(可选；可继承；PDF1.5) 在表格单元或者任何元素的内容矩形中被画的边界的厚度（用 Padding 属性可以调节）。每个边沿数组的数值是正数在默认的用户空间单元代表边界的厚度（数值为0 代表边界没有被涂画）。这有两种形式</p> <p>数值代表所有四个边沿的边界厚度。</p> <p>四个入口的数组，每个入口指定边界的一个边沿的厚度，遵循前，后，开始，结束边沿。如果任何的边沿的数字是 null，表明没有被涂颜色。</p>
Padding	数 字	<p>指定偏移以在元素的内容矩形和周边边界取得独立空间（见 859 页“内容和分配矩形”）。一个正数值放大背景区域；一个数组负数值减小背景区域，允许边界跨越元素的内容或者图形。</p> <p>要么这个数值是单个数代表在默认用户空间单元中焊盘的宽度，应用于四个边界；或者是 4 个入口数组分别内容矩阵的代</p>

表前, 后, 开始, 结束边沿的焊盘宽度。默认值: 0

Color	数组 (可选; 可继承; PDF1.5) 颜色被用来填充文本, 默认值为表格边界和文本修饰的颜色。这个数值是一组在矩形 0.0 到 1.0 中的 3 个数的数组, 代表红, 绿和蓝数值, 分别为 RGB 颜色空间。如果属性没有被指定, 这个元素的边界颜色是在相连内容的开始就影响的当前的文本填充颜色。
-------	--

BLSEs 的布局属性

表 10.31 描述了只用于模块水平结构元素 (BLSEs) 的布局属性。

注意: 带有一个 Placemet 属性的内嵌水平结构元素 (ILSEs) 除了 Inline 的默认值外像 BLSEs 一样对待, 因此这里的属性描述同样适用于它。

表 10.31 附加的标准布局属性具体到模块水平结构元素

键	类型	数值
SpaceBefore	数字 (可选; 不可继承)	在 BLSE 的前边沿额外空间的数量, 在模块过

程的默认用户空间单元中被测量。这个数值可以被增加到任何调节被 ILSEs 的 LineHeight 的属性和在 BLSE 的第一行包括（见 855 页“ILSEs 的布局属性”）。如果前 BLSE 有一个 SpaceAfter 属性，两个属性属性的较大者被使用。默认值：0。

注意：这个属性在给定参考区的第一次 BLSE 放置中被忽略。

SpaceAfter 数字
(可选；不可继承) 在 BLSE 的后边沿额外空间的数量，在模块过
程的默认用户空间单元中被测量。这个数值可以被增加到任何调节
被 ILSEs 的 LineHeight 的属性和在 BLSE 的最后一行包括（见 855
页“ILSEs 的布局属性”）。如果前 BLSE 有一个 SpaceBefore 属性，
两个属性属性的较大者被使用。默认值：0。

注意：这个属性在给定参考区的最后 BLSE 放置中被忽略。

键	类型	数值
StartIndent	数字	<p>(可选; 可继承) 从参考区的开始边沿到 BLSE 的开始边沿的距离, 由在内嵌过程方向的默认用户空间单位被测量。这个属性只应用于带有块或开始的 Placement 属性的结构元素 (见 864 页 “一般布局属性”)。这个属性在当元素带有其他 Placement 数值是被忽略。默认值: 0。</p> <p>注意: 这个属性的负值是把 BLSE 的开始边沿放置在参考区的外面。这个结果有执行依赖并且不一定被所有带标签的 PDF 用户应用或者其他格式所支持。</p> <p>注意: 如果一个结构元素带有一个 StartIndent 属性将被放置在带有开始的 Placement 属性的浮动元素旁边, 对元素的开始缩进的准确值是它自己的 StartIndent 属性或者相邻元素的浮动程度, 无论哪个更大。若有的话, 这个值通过元素的 TextIndent 的属性来进一步调整。</p>
EndIndent	数字	<p>(可选; 可继承) 从 BLSE 的结束边沿到参考区的结束边沿的距离, 由在内嵌过程方向的默认用户空间单位被测量。这个属性只应用于带有块或结束的 Placement 属性的结构元素 (见 864 页 “一般布局属性”)。这个属性在当元素带有其他 Placement 数值是被忽略。默认值: 0。</p> <p>注意: 这个属性的负值是把 BLSE 的结束边沿放置在参考区的外面。这个结果有执行依赖并且不一定被所有带标签的 PDF 用户应用或者其他格式所支持。</p> <p>注意: 如果一个结构元素带有一个 EndIndent 属性将被放置在带有结束的 Placement 属性的浮动元素旁边, 对元素的结束缩进的准确值是它自己的 EndIndent 属性或者相邻元素的浮动程度, 无论哪个更大。</p>
TextIndent	数字	<p>(可选; 可继承; 就下面所描述的只应用于一些 BLSEs) 附加的距离, 在内嵌过程方向中默认用户空间单位被测量, 从 BLSE 开始边沿到文本的第一行边沿, 被 StartIndent (上述的) 所指定。一个负数值指</p>

示悬挂缩进。默认值：0

这些属性只应用于图形像 **BLSEs** 和 **LbI**（标签），**LBody**（列表实体），**TH**（表头）和 **TD**（表数据），被证实他们包含内容除了嵌套 **BLSEs**。

关键域	类型	取值
块对齐	名称	(可选, 可继承, 仅针对单元格) 单元格中内容的方向在各级块中的对齐
	前	在为第一个子项分配的的方框边缘之前与表格的内容框对齐
	中	在表格单元格内子项的中心。为子项分配的方框的前边缘到表格内容方框的距离和为子项分配的方框的后边缘到表格内容方框的距离相通。
	后	为最后一个子项分配的方框的后边缘与表格内容方框的后边缘对齐
	正好	子项和表格内容方框的前端都要对齐。第一个子项像上文所描述的一样放在前面，最后一个子项像上文所描述的一样放在后面，两个子项之间的间距是相等的，如果只有一个子项，只许于内容方框的前端对齐即可。
		这些属性只适用于 TH 和 TD 结构中的元素，他们控制着所有给定元素子项的所有 BLSE 的位置。表格

的内容框成了所有子项的参考区。默认值是：前。

线内对齐 名称 (可选，可继承，仅针对单元格) 单元格中内容在线性发展方向的对齐：

开始 子项所分配方框的起始边缘与单元格内容框的起始边缘对齐。

中心 子项所分配方框的起始边缘到单元格内容框的距离与其结束边缘到单元格内容框的距离相同

结束 子项所分配方框的结束边缘与单元格内容框的结束边缘对齐。

这些属性只适用于 TH 和 TD 结构中的元素，他们控制着所有给定元素子项的所有 BLSE 的位置。表格的内容框成了所有子项的参考区。默认值是：开始。

关键域	类型	取值
TBorderStyle	名称或阵列	(可选，可继承 PDF1.5) 边框的样式画在单元格的每个边缘上，可能的取值和为 BorderStyle 指定的取值相同，如果所有的 TBoederStyle 和 BorderStyle 都适用于给定的单元格，BorderStyle 将取代 TBorderStyle。默认值：无。

TPadding	整形或阵列	(可选, 可继承 PDF1.5) 指定一个偏移来统计单元格内容框和周围边框 间的距离。如果所有的 TPadding 和 Padding 都适用于给定的单元格 , Padding 将取代 TPadding。一个正值将使背景面积增大, 负值将使其缩小, 可能允许边框将元素的文字或者图像重叠。该取值可能是一个有符号数, 它代表着填充区域的宽度, 该填充区在用户空间单元且适用于单元格的四个边缘, 或者是一个有着四个入口的阵列, 分别代表着内容边框填充区前后左右边缘的宽度。默认值为零。
----------	-------	--

ILSES 的布局属性

表 10.32 所描述的属性适用于线内级别的结构元素。他们也可能被指定给块级别的元素而且适用于立即子项的任何内容条目。

表 10.32 线内级别结构元素的标准布局属性

关键域	类型	取值
BaseLineShift	数值	(可选, 不可继承) 在默认的用户空间, 基线的移动距离和其父项相关。其移动方向和当前写模式属性所指定的块进展方向相反。因此, 正值使基线朝参考区域的前边缘移动而负值使其朝后边缘移动。默认值为零。 被移动的元素可能是一个超级脚本, 子脚本, 或者线内图像。这种移位适用于元素以及元素的内容, 及其所有子项。任何适用于子项元素的基线的进一步移动都是用来相对的测量父项基线移位的。

关键域	类型	取值
LineHeight	数值或名字	(可选, 可继承) 元素的首先高度是在块进展方向的默认用户空间被测量的。线的高度是被完全或者局部 ILSE 所包含的最大线高所决定的。
如果用一个正常或者自动的名字代替数值则说明没有明确的高度限制。元素的高度可以更加字体大小合理设置:		
正常		调整线的高度, 使他能包含为 BaseLineShift 所指定的每一个非零值
自动		不为 BaseLineShift 的值做什么调整
默认值:		正常

此属性适用于所有元素子项的 ILSE 或者其嵌套的 ILSE, 不适用于嵌套的 BLSE

注意: 当朝导出格式转化时, 如果指定了自动或者正常的值, 而他们又在目标格式中可用, 可以直接使用这些指定的值。上文所提到的术语“合理的值”

是留给消费应用来决定的。可以先假设线高为字体高度的 1.2 倍，但是这个值又严格依赖导出的格式。在没有明确的线高值的时候，一个合理的方法通过一个标记的 PDF 文件中的信息来计算线高值是寻找相关联字体上身和下降值的不同，然后将其从字形空间映射到默认的用户空间，然后使用最大的值。

TextDecorationColor 矩阵（可选择，可继承）是用于绘制文本的装饰颜色。该值是一个三个数字组成的举证，取值范围为 0.0 到 1.0，他们分别代表着红、绿、蓝、三种 RGB 空间中的颜色。如果没有指定这个属性，这个元素的边框颜色在目前实施的是在其填写相关内容的初始颜色。

TextDecorationThickness 数值（可选，可继承）每一行画出的厚度也是装饰的一部分，其取值是一个处于默认用户空间的非负的数，它表示着厚度。如果没有指定这个属性，它是来自于目前中风厚度实际上在元素的相关内容展开，并被转换到默认用户空间中。

关键域	类型	取值
TextDecorationColor	名称	(可选, 不可继承) 文本装饰, 如果有的话, 他适用于该元素的文本。

None	无文本装饰
Underline	文本下面的线
Overline	文本上面的线
LineThrough	贯穿于文本中间的线
默认值:	无

如果某个元素有此属性, 它适用于该元素子项的所有文本内容的条目或者该元素嵌套的 ILSEs。该属性不适用于嵌套的 BLSEs 和除文本之外的其他内容条目。

注意:所有子项装饰的颜色, 位置, 厚度必须统一, 不管他们在字体, 颜色, 字形上有什么变化。

RubyAlign	名称	(可选, 可继承)
	Start	其内容在直接进展方向的开始边缘对齐

Center 其内容将在直列进展方向的中心

End 其内容在直接进展方向的结束边缘对齐

Justify 其内容将被扩大，用以填补在直列进展方向的现有宽度

Distribute 其内容将被扩大，用以填补在直列进展方向的现有宽度。然而，一些空间也会被插入文本的开始或者结束边缘。正常情况下，这些空间会以 1:2:1(开始: 插入: 结束) 的比例分布。如果 ruby 出现在文本的起始端，这个比例会变成 0: 1: 1，如果出现在文本的末端，这个比例会变成 1: 1: 0。

关键域	类型	取值
		<p>默认值: Distribute</p> <p>此属性有可能在 RB 或者 RT 中指定。当 Ruby 被设置为特定格式的时候，此属性将适用于这两个元素中较短的线。(如果 RT 元素比 RB 元素窄，则 RT 元素将作为指定的 RubyAlign 属性被对齐。)</p>
RubyPosition	名称	(可选，可继承) 在 Ruby 的组成中，RT 结构的位置和 RB 结构相关。
	Before	RT 中的内容要在元素前端对齐。

After	RT 中的内容要在元素的后端对齐。
Warichu	RT 和相关的 RP 元素将在 RB 元素之后被转换成 Warichu 格式
Inline	RT 和相关的 RP 元素将在 RB 元素之后被转换成插入注释的括号
默认值: Before	
GlyphOrientationVertical	名称 (可选, 可继承) 当内线的进展方向是自下向上或者自上向下时, 需要指定字形方向。
此属性可能取如下值:	
Angle	此数字代表字形顶部到参考区的顶部沿顺时针方向的度数。必须在 -180 度到 360 度之间。
Auto	为文本指定默认的方向, 这取决于该文本是否是全宽的 (高和宽相等), 对于全宽的拉丁或者全宽的表意文本 (包括表意的标点符号) 被设置为零度的角。表意的标点符号和其他具有备用水平方向和垂直方向格式的使用字形的垂直合格。非全满文本被设置为 90 度角。
默认值: Auto	

关键域	类型	取值
		<p>这个属性是用来区分最常见的首选的字母（非表意文字）在垂直方向的书面文字日本文件（自动或 90）和方向表意文字和/或字母（非表意文字）西文标志和广告（90）。</p> <p>它影响到的字形的对准和宽度。如果一字形垂直于垂直基准线，其水平对齐点对齐于该字形的脚本对齐基准线。该区域的宽度是决定字形从横向宽度字型特征的标志符号。</p>

内容和分配矩形

如 10.7.2 “基本布局模块所定义”，元素内容方框是由元素内容的形状所派生来的内容方框，它定义了所有包含子元素所定义的边界。该分配矩形包括围绕在元素周围任意的附加边界或空间，它将影响到该元素将如何定位，以使他能够与周围的元素和内附内容框以及参考区域更加协调。

内容方框的准确定义要依赖元素结构的类型：对于单元表格（TH 或 TD 结构类型），它的内容方框由该表格中所有图形对象的边界框所决定（比如 XObject 对象中的 BBox 条目）。这意味着大小可以被表格单元的高度和宽度属性所覆盖。

表格的高度会被进一步调整，以使他和该行所有表格的最大宽度相等，同时表格的宽度也会被调整，以使其等于该列的最大宽度。

对于所有其他的 BLSE，内容方框的高度是其所包含的所有的 BLSE 高度的总和再加上这些元素之间的宽度调整。

对于包含文本的 ILSE，内容方框的高度由 LineHeight 属性来设置。其宽度由所有包含的字符的宽度之和决定，同时会针对所有的缩进，字母间距，字间距或线端条件做适当调整。

对于一个包含插图和表格的 ILSE，内容方框是由该内容中所有图像对象的边界箱决定的，同时还要考虑所有明确的边界箱（比如 XObject 对象中的 BBox 条目）这意味着大小可以被表格单元的高度和宽度属性所覆盖。

对于包含混合元素的 ILSE，内容的矩形的高度取决于调整的子对象相对于彼此的在他们的文本的基线（文字 ILSEs）或结束边缘的非文本（ILSEs），沿任何适用 BaselineShift 属性（对于所有的 ILSE），并寻找及顶端和底部的所有颜色。

注意：有些应用程序可能适用于这一进程块内的所有内容，而其他的可能只适用于基于逐行逐行的应用。

分配方框派生自内容方框，他也依赖于结构的类型：对于 BLSE：分配方框的前后边缘和矩形方框相等，它会被元素的 SpaceBefore 和 SpaceAfter 属性所调整，而且起始和结束边缘不能被改变。

对于所有的 ILSE，分配方框和内容方框相同。

注意：新版本的标记 PDF 可能会包含新的属性，使他可以调整所有 BLSE 和 ILSE 的分配方框的四个边缘。

插图属性

某些额外的限制出现在和插图特定使用的连接中。

当一个插图元素在一个块中有一个位置属性，他必须有一个明确数值的高度属性。这个数值是插入图像在块进展方向的唯一识别信息。

当一个插图元素拥有线内的位置属性时，它必须要有一个具有明确数值的宽度属性（非自动）。该数值是插入图像在块进展方向的唯一识别信息。

当一个插图元素拥有线内、起始或者结束的位置属性时，`BaseLineShift` 的属性值用来觉得此插图的后边缘到文本基线的距离。位置的其他值都会忽略掉 `BaseLineShift`（一个拥有 `Start` 位置值的插图元素可以节省资源，而拥有 `Inline` 位置值的插图元素却很浪费资源）

列属性

表 10.33 所描述的属性仅适用于 `Art`、`Sect`、`Div` 三种群元素。他们在群元素被分划分成列的时候会用到。

表 10.33 标准列属性

关键域	类型	取值
<code>ColumnCount</code> <code>integer</code>		(可选，不可继承) 群元素中内容的列数。 默认值：1
<code>ColumnGap</code>	数值或矩阵	相邻列所需要的空间是由内线进展方向中默认用户空间的单位来衡量的。如果该值是一个数字，他指定所有列之间的距离。如果该值是一个矩阵，它需包含 <code>ColumnCount-1</code> 个数字，分别代表第一行和第二行，第二行和第三行之间的距离，以此类推。如果少于 <code>ColumnCount-1</code> 个数字，最后一个数字将指定所有剩余的空间，超过的元素将被忽略。
<code>ColumnWidth</code>	数值或矩阵	相邻列所需要的宽度是由内线进展方向中

默认用户空间的单位来衡量的。如果该值是一个数字，他指定所有列的宽度。如果该值是一个矩阵，它需包含 ColumnCount 个数字，分别每一列的宽度。如果少于 ColumnCount 个数字，最后一个数字 将指定所有剩余列的宽度，超过的元素将被忽略。

列表属性

表 10.34 描述的列表属性，是有 L 元素带来的，但是它控制值列表中 LI 元素的 LBI 元素的解释。此属性是在属性对象包含 0 条目的属性对象中定义的。

表 10.34 标准列表属性

关键域	类型	取值
ListNumbering	名称	(可选, 可继承) 用于产生 LBI 中内容的数值系统在一个自动标号的列表中，而用于标识每个条目的符号存在于一个没有标号的列表中。
	None	无自动标号, LBI 元素包含任意文本, 不含任何编号方案
	Disc	封闭环形弹头
	Circle	开放环形弹头
	Square	封闭方形弹头
	Decimal	十进制阿拉伯数字 (1-9910-99、、、、、)
	UpperRoman	大写罗马数字 (I, II, III, IV、、、、)
	LowerRoman	小写罗马数字 (i, ii,

iii, iv、、、)

UpperALpha 大写字母 (A, B,
C、、、、)

LowerAlpha 小写字母 (a, b,
c、、、、)

默认值：无

注意：用于 UpperALpha 和 LowerAlpha 的字母是由当时的 Lang 条目定义的。

注意：这些可能的数值集合可能会被扩大为 Unicode 来标识额外是数值系统

注：这个属性是用来允许内容提取工具自动为列表编号的。然而，表中的 Lbi 元素还是应该明确的包含数值结果，这样一个文档才能在不需要自动编号的情况下能被正常打印。

表属性：

表 10.35 中描述的属性只适用于单元格。此类属性是在 0 条目具有 Table 值的属性对象中定义的。

表 10.35 标准表属性

关键域	类型	取值
RowSpan	整型	(可选, 不可继承) 内附表中的行数是跨越表格的。 这些表格会被块进展方向中 WritingMode 属性所指定的附加的行来扩展。默认值：1
ColSpan	整型	(可选, 不可继承) 内附表中的行数是跨越表格的。这些表格会被块进展方向中 WritingMode 属性所指定的附加的列来扩展。默认值：1
Headers	矩阵	(可选, 不可继承) 一个由字符串组成的矩阵，每

一个字符串都是一个 TH 结构元素的标识，它是相关表格的头。

此属性可适用于头表格和数据表格。因此，与表格相关联的头是头矩阵加上头矩阵中所有的 TH 表格。

关键域	类型	取值
Scope	名称	(可选, 不可继承) 是一个取值 Row、Column 或者 Both 的名称。此属性仅使用于 TH 元素，这意味着不管头表格是否适用于该行中剩余的表格，过着所有的行或者列中的表格。

10.8、辅助功能支持

PDF 有一些特性可以帮助残障人士来使用文档，通常，一些有视觉障碍的用户会使用屏幕阅读器去阅读文档。为了能让用户无论是使用屏幕阅读器或者直接调用文档到声音的引擎都能够正常发声，PDF 支持如下特性：

指定 PDF 文档中使用的自然语言——比如英语或者西班牙语

对于那些不能直接自然地转换成文字的图像或者条目，或者能够转换但转换方式比较特殊的，要提供文字描述。

指定缩写或者缩语的全称

这中支持的核心在于能够决定 PDF 文档的内容逻辑顺序。独立的文件结构或者布局，通过逻辑结构或者标记的 PDF，像 817 页所描述的页内容顺序一样。一种辅助应用程序可以通过遍历层次结构和每个节点，来提取文档内容以供残障人士使用。基于这种原因，PDF 文件的制作者必须确保文档的信息可以通过访问文档的层次结构来获取，并且强烈推荐他们使用本段所描述的特性。

注：可以从标记的 PDF 文档中提取文本，然后将其针对非残障人士以外的群体进行重用或者检查。

10.8.1、自然语言规范

文档中用于文本的自然语言是由一种分级风格决定的，它是基于在目前的几地点中是否有一个可选的 `Lang` 条目。在最高层，文档的默认语言可以被文档目录中的 `Lang` 条目指定。在下面，语言可以被如下条目指定：

- 任何类型的结构元素，可以通过结构元素词典中的 `Lang` 条目来指定。
- 结构等级中不存在的标记内容序列，可以通过利用 `Span` 标记和标记内序列绑定的资源序列的 `Lang` 条目来指定。

以下各节提供了 `Lang` 条目的详细值和分层方式，在这里面讲决定着文本的语言。

注：以 `Unicode` 编码的文本字符串可能会包含一些转义序列或者用来说明文本语言和重载当前 `Lang` 条目的语言标记。

语言标识

文件目录、结构元素字典或者资源列表中的 `Lang` 条目的值是一个字符串，他通过一个语言标识符标记语言，该语句定义于英特网 `RFC3066` 语言标识标记中。这个语句，将在下面提到，根据 可扩展标记语言 `W3C` 文档 1.1，他在 `XML` 文件中也会被用来标记语言。可以查看“书目”章节的内容以获得该文档的更多信息。一个空字符说明该语言是未知的。

语言标识符可以基于国际化标准组织在 `ISO639` 和 `ISO3166` 中定义的代码或者通过互联网编号分配机构来注册（`IANA`，其网页是 `http://iana.org`），或者他们可以包含用于私人用途的代码。一个语言标识符是由可选的初级代码和一个或多个自带吗构成。初级代码可以使下面的任意一种：

- 一个两个字符的 `ISO 639` 语言代码，比如，`en` 代表英语，`es` 代表西班牙语。
- 字母 `i`，指定一个 `IANA` 注册的标识符
- 字母 `x`，代表私人用处。

第一个自带吗可以是一个两个字符的 `ISO 3166` 国家代码，例如 `en` 代码美国，或者 `3-8IANA` 注册的子代码字符，例如科克尼中的 `en`，切诺基中的 `i`。

第一个超出的自带吗可以使任何一个被注册的 IANA .

语言规范层次

文档中的目录指定了文件中除为结构元素指定语言所覆盖的所有文本的自然语言，或者被标记而 又不在结构目录中的内容序列。本节的例子说明了分层方式，以及在这里面，文本的语言是如何被确定的。

例 10.19 展示了一个语言是如何作为一个完全可以被重载的内容标记序列指定给一个文档的，它独立于任何逻辑结构。因此，文档目录中的 Lang 条目拥有 en-US 值，表示美国英语，他会被和标记内容序列绑定的 Lang 资源重载。Lang 资源用值 es—MX 标识标记内容序列的语言，es—MX 的意思是墨西哥西班牙语。

例 10.19

```
2 0 obj % 页对象
<< /Type /Page
/Contents 3 0 R % 内容流
...
>>
endobj
3 0 obj % 页的内容流
<< /Length ... >>
stream
BT
(See you later, or as Arnold would say, ) Tj
/Span << /Lang (es-MX) >> % 标记内容序列的开始
BDC
(Hasta la vista .) Tj
EMC % 标记内容序列的结束
ET
endstream
endobj
```

当在一个文档中有逻辑结构描述的时候，此文档目录中的 Lang 条目被设置为默认值。

下面是：任何一种满足这种结构等级规格的语言，以如下顺序提供：

● 一个结构元素的语言规范

注：如果一个结构元素没有 Lang 条目，它将从任何一个拥有该条目的父辈元素中继承语言。

● 在一个结构元素内部，一个嵌套的结构元素或标记内容序列。

在例 10.20 中，结构元素字典中的 Lang 条目适用于标记内容序列，该序列在现实网页的内容流里面，他拥有一个为 0 的 MCID 值。然后，嵌套在标记内容序列中的是另外一个，他的 Lang 资源和 Span 标记绑定，并且重构了结构元素的语言规范。

注：这个例子和下一个被忽略的例子需要 StructParent 条目在对象中用于内容条目。

例 10.20

```
1 0 obj % 结构元素
<< /Type /StructElem
/S /P % 结构类型
/P ...
/K << /Type /MCR
/Pg 2 0 R % 含有标记内容序列的页
/MCID 0 % 标记内容识别符
>>
/Lang (en-US) % 该元素的语言规范
>>
endobj
2 0 obj % 页对象
<< /Type /Page
```

```
/Contents 3 0 R          % 内容流
...
>>
Endobj

3 0 obj                  % 页的内容流
<< /Length ... >>
stream
BT
/P << /MCID 0 >>      % 标记内容序列的开始
BDC
(See you later, or as Arnold would say, ) Tj
/Span << /Lang (es-MX) >>      % 嵌套内容序列的开始
BDC
(Hasta la vista .) Tj
EMC                      % 嵌套内容序列的结束
EMC                      % 标记内容序列的结束
ET
endstream
endobj
```

如果网页内容的一部分包含在结构层次和结构化的内容是嵌套在非结构化内容的不同语言规范适用，结构元素的语言规范 优先。在示例 10.21 中，该网页的内容流包含一个标记内容序列，通过 Span 标记利用 Lang 资源指定西班牙语作为其语言。嵌套在里面的是结构元素内容的一部分，语言规范适用于字母的是结构元素。

例 10.21

```
1 0 obj                  % 结构元素
<< /Type /StructElem
/S /P                      % 结构类型
```

```
/P ... % 结构等级中的父元素  
/K << /Type /MCR  
/Pg 2 0 R % 含有标记内容的页  
/MCID 0 % Marked-content identifier  
>>  
/Lang (en-US) % 该元素的语言规范  
>>  
Endobj
```

```
2 0 obj % 页对象  
<< /Type /Page  
/Contents 3 0 R % 内容流  
...  
>>  
endobj  
3 0 obj % 页的内容流  
<< /Length ... >>  
stream  
/Span << /Lang (es-MX) >> % 标记内容序列的开始  
BDC  
(Hasta la vista, ) Tj  
/P << /MCID 0 >> % 结构标记内容序列的开始  
BDC % 结构元素语言所适用的  
(as Arnold would say. ) Tj  
EMC % 结构标记内容序列的结束  
EMC % 标记内容序列的结束  
endstream  
endobj
```

换句话说，和标记内容序列绑定的语言标识符指定了该序列中除嵌套标记内容和语言规范所重构的之外的所有文本的语言。

多语言文本矩阵

多语言文本矩阵允许指定多种文本字符串，每一种都和一种文本语言标识符相关联。该矩阵包含如下一对字符串：

- 每一组的第一个字符串是语言标识符。一个给的的语言标识符可能只在一个矩阵中出现一次，所有不能被识别的语言标识符都会被忽略。当在该矩阵中找不到匹配的语言标识符的时候，一个空字符串会指定一个默认文本以供使用。
- 第二个字符串是一个和语言相关的文本。

例 10.22

```
[ (en-US) (My vacation) (fr) (mes vacances) () (default text) ]
```

当一个消费应用在一个多语言矩阵中寻找一个特定语言的文本的时候，它需要在给定语言标识符和矩阵中的语言标识符之间寻找一个准确的匹配。如果没有发现准确的匹配，将进行首字母匹配：如果一个给定的标识符匹配率最高，就表明匹配成功，不区分大小写，该矩阵中标识符字符串的一部分，矩阵中第一个字符串是一个连字符号。比如：给定的标识符 en 和矩阵中的标识符 en-US 匹配，但是给定的识别符 en-US 即不和 en 也不和 en-GB 匹配。如果没有找到准确的或者部分匹配，将使用默认的文档。

10.8.2、备用说明

通过为图像、公式或者其他不易自然地转换成文本的条目提供备用说明可以增强 PDF 文档的性能。备用说明是人类可读的文本，比如，通过文本到语音转换引擎，将其转换成声音，为视力有障碍的用户提供方便。

一个备用的描述符可以通过下面条目来指定：

- 一个结构元素，通过结构元素字典中的 Alt 条目定制。
- 一个标记内容的序列，通过一个资源清单中的 Alt 条目利用 Span 标记标记内容序列绑定。
- 没有一个由文本标识的任何类型的注解，通过注释字典中的 Content 条目对于那些正常显示文本的注释，文本是声音目标的自然资源。对于那些不显示文本的注释类型，可以包含 Content 条目来指定一个备用说明。由于声音注释默认就被语音划了，因而不需要语音目的的备用注释，它可以包含一个 Content 条目来指定一个在弹出窗口显示的注意，这样对视觉有障碍的人有帮助。此外，一个

备用名称可以被指定为一个交互式表单字段，用于实际字段名称必须被标识的用户接口。这个备用名称，是别改字段中可选的 TU 条目的，对于声音目的的使用很有用。

备用描述符是文本字符串。它可能会被以 PDFDocEncoding 或者 Unicode 来编码。像 3.8.1 章节所描述的一样，“文本字符串”，Unicode 定义了一个转义字符来说明文本的语言。这中机制使备用描述从 Lang 条目所指定的语言开始变化。当应用到结构元素，文本被认为是一个词或短语以取代当前元素。例如，如果两个（或更多）的元素序列中有一个 ALT 键在字典项，应视作他们之间出现了文字破坏，这个字同样适用于连续的标记内容序列。

注：资源列表中的 Alt 条目可以和其他条目结合起来，如例 10.23 所示

例 10.23

```
/Span << /Lang (en-us) /Alt (six-point star) >> BDC (☆) Tj EMC
```

10.8.3、代替文本

正如备用描述可以被提供给不能被正常转化成文本的图像或者其他条目，替代文本可以被指定为那些可以被转换成文本而方式不标准的内容。这些非标准的描述可能包括，例如，字形的连字或自定义字符，图形或内嵌在相应的字母扫描手稿或者下降的首部。

代替文本可以有以下条目来指定：

- 一个结构元素，通过结构元素字典中可选的 ActualText 条目来确定。
- 一个标记内容序列，通过一个和标记内容序列绑定的资源列表的 ActualText 条目来确定。ActualText 的值不是对内容的描述而是对他的替代，只提供一个相当于读者阅读时用眼光看得到的文本，相对于被认为一个字或者替代短语的 Alt 的值，ActualText 的值被看作是结构元素或者标记内容序列的替代。因此，如果两个或者更多的连续结构或者标记内容序列具有 ActualText 条目，就 应当认为他们之间没有坏字。

下面的例子展示里替代文本的用法，说明了当断字改变了单词的拼写的时候，如何提示正确的字符。

例 10.24

```
(Dru) Tj
/Span
<</Actual Text (c) >>
BDC
(k-) Tj
EMC
(ker) '
```

像备用描述符一样，替代文本如果用 **Unicode** 来编码，它会包含一个转义字符来说明文本的语言，同时会重构当前的 **Lang** 条目。

10.8.4、缩略语的全写

缩略语可能会给文本到语音引擎造成一个问题，经常缩略语的全写可以避免这一问题。例如，字典搜索可能会发现，“大道。”读“大道”，以及“夫。”读“的途径。”然而，一些缩写难以解决，如在句子“博士 Healwell 123 工业工程博士”出于这个原因，缩写或缩写扩展可以指定以下项目：

- 标记内容序列：通过和 **Span** 标记一起和序列绑定的资源列表中的 **E** 资源确定，如例 10.25 所示
- 结构元素，通过结构元素中的 **E** 条目。

例 10.25

```
BT
/Span << /E (Doctor) >>
BDC
(Dr. ) Tj
EMC
(Healwell works at 123 Industrial ) Tj
/Span << /E (Drive) >>
BDC
(Dr. ) Tj
EMC
```

ET

E 值被看做是具有标签文本的一个字或者替代短语，因此应该被认为有一个坏字使他和周围的文本分开。像其他字符串文本一样，如果扩展文本用 Unicode 进行编码，它可以包含一个转义字符来表示文本的语言。

按常规，一些缩略语没有被扩展成一个字。例如，对于“CBS”文本，既不需要给他提供扩展，也不需要处于安全考虑，为他指定扩展。

10.9、网页捕获

网页捕获是 PDF1.3 中的特性，它能够将基于网页或者本地 HTML 的 PDF、GIF、JPEG 或者 ASCII 文本文件导入到 PDF 文件中，此特性通过一个网页捕获延期插件在 Acrobat4.0 和后来的阅读器中开始实施。网页捕获中数据结构的信息允许阅读器可以支持如下操作：

- 保持当地或者从网页获取的资源的视觉外观
- 检索网络上的附件资源，并将其加到 PDF 文件中
- 跟新或者修改当前从网页上捕获的资源
- 寻找要从网上捕获的资源的源，比如 URL
- 从 PDF 文件中寻找所有给定 URL 产生的资源
- 从 PDF 文件中寻找所有和给定数字标识符相匹配的资源
- 要执行这些操作所需要的信息保存在 PDF 文件中的两个数据结构中：
- 网页捕获信息词典包含和网页捕获相关的文档级别信息
- 网页捕获内容数据库持续跟踪被网页捕获所检索的资源和他的来源，确保避免使网页捕获器下载 PDF 文件中已经存在的资源。

下面章节提供了这些结构的详细描述，可以在附录 C 中查询和网页捕获实施相关的限制信息。

注：虽然网页捕获在其他格式的文件中也能很好使用，下面的讨论是基于 HTML 和 GIF 文件的。

10.9.1、网页捕获信息词典

文档目录中可选的 **SpiderInfo** 条目包含了一个可选的网页捕获信息词典，它里面包含的内容是和网页捕获相关的文件级别信息。表 10.36 显示了这个词典中所包含的内容：

表 10.36 网页捕获信息词典中的条目

关键域	类型	取值
V	数字	(必须的) 网页捕获版本号, 对于 PDF1.3, 其版本是 1.0。 注: 这个值是一个有符号实数, 不是一个主次版本号。例如, 版本号 1.2 要比版本号 1.15 大。
C	矩阵	(可选的) 一个包含对网页捕获信息词典间接引用的矩阵, 它描述了在创建一个 PDF 文件时将要使用到的命令, 这些命令按照他们创建 PDF 文件时被执行的顺序存储在该矩阵中。

10.9.2、内容数据库

网页捕获 将坚持 URL 中的 HTML 文件, 然后将他们转换成 PDF 文件。产生的 PDF 文件中可能包含有多个 HTML 网页中的内容。反过来, 由于 HTML 页没有一个固定的大小, 一个 HTML 页可能需要数页 PDF 文件才能装的下。为了持续跟踪这些信息, 网页捕获器维护着一个内容数据库, 它将 URL 和数字识别符映射到页或者 XObjects 之类的 PDF 对象中。通过查询数据库中的数字识别符, 网页捕获可以判断刚下载的内容是否和以前从其他 URL 中检索到的内容重复。这样, 就可以执行优化操作, 例如, 对于一个多个 HTML 文件到应用到的图像, 我们只做一个备份就可以了。网页捕获内容数据库被组织到内容集中。每一个内容集都是一个

字典，它包含着相同数据源所产生的和一群 PDF 对象相关的信息。内容集由两个子类型组成：页集和图像集。比如看，当网页捕获器将一个 HTML 文件转换成 PDF 页的时候，它将生成一个页集来包含页信息的集合。同样，当它将一个 GIF 格式的图片转换成一个或者多个 XObjects 的时候，它将产生一个图像集来描述这些 XObject。

内容集被分配给一个给的数据源，他可以通过两种途径来访问。

- 通过他被检索的 URL 访问
- 通过数据源自己产生的数字标识符

PDF 文件名称词典中的 URLs 和 IDS 包含一个名称树，它将数字标识和 URL 映射到网页捕获内容集中。图 10.1 展示了一个简单的例子。一个从 URL <http://www.adobe.com/> 检索过来的 HTML 被转换成一个三页的 PDF 文件，URLS 中名称树的 URL 条目指向包含这三页的页集。同样的，IDS 名称树中叶包含一个指向相同页集的条目，它和 HTML 源计算出的数字标识符有关。

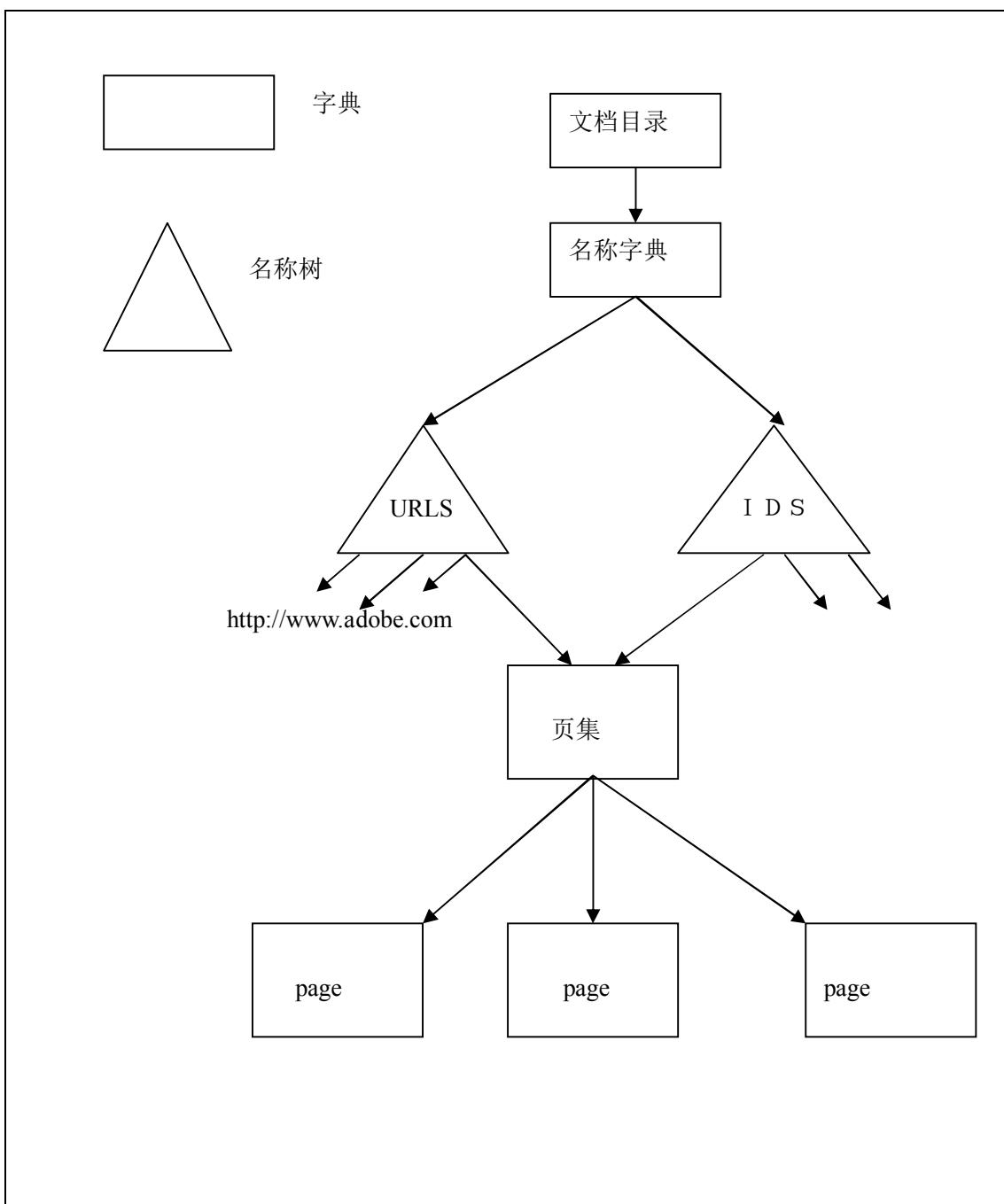


图 10.1 简单网页捕获文件框架

URLS 和 IDS 的名称树种的条目可能会引用一个内容集合中的矩阵来替代一个单一的内容集合。内容集合不一定需要相同的子类型，但是有可能同时包含页集合和图像集合。例如，在图 10.2 中，一个从 <http://www.adobe.com/getacro.gif> 检索的 GIF 文件被转换成一页 PDF。如图 10.1 所示，会产生一个页集合来存储新页的信息。然而，由于这个检索同时会产生一个新图像的 XObject，所以会生成一个图像集。

与直接指向一个单一的内容集不同，URLS 和 IDS 指向一个同时包含页集合和图像集合的矩阵。

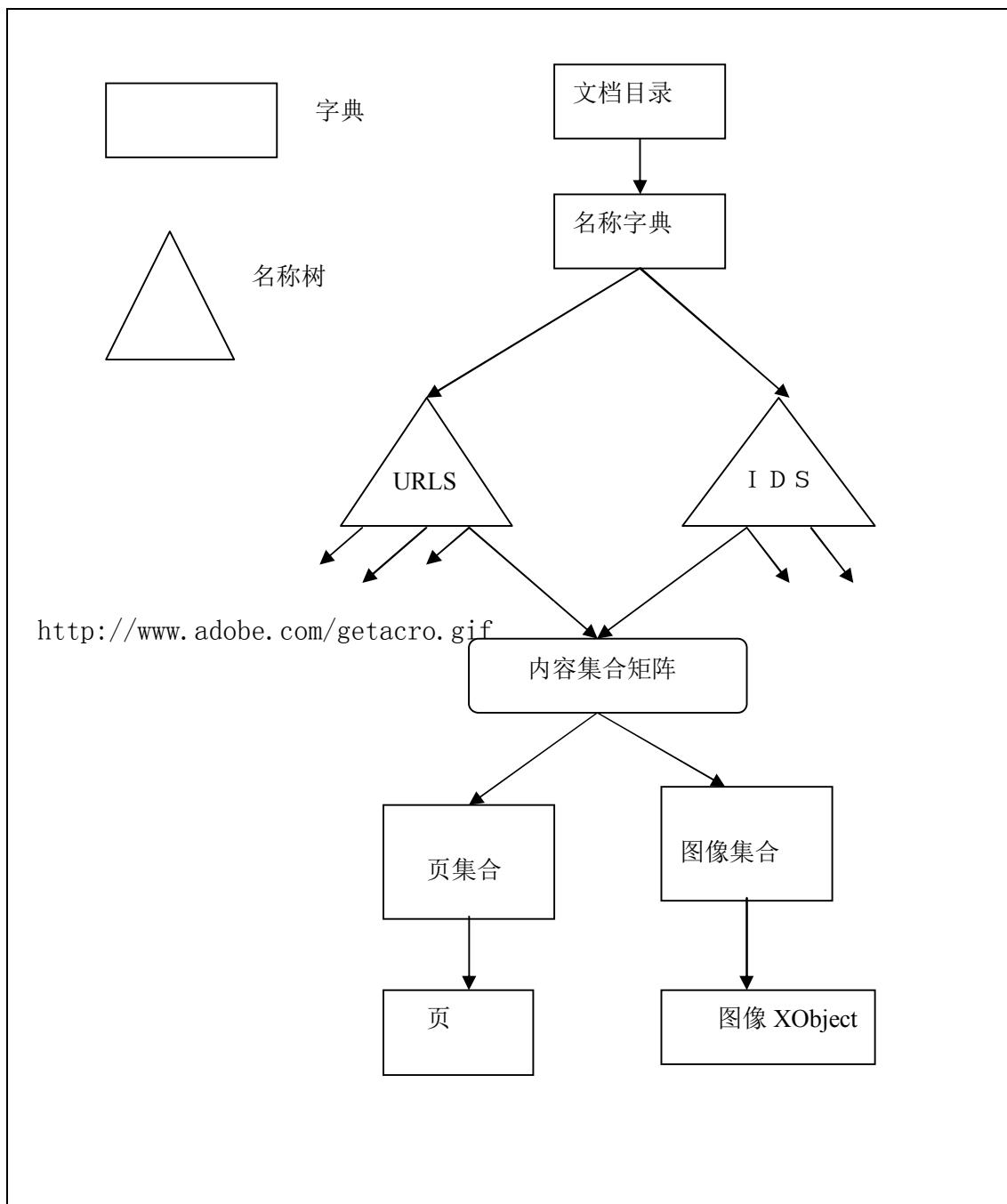


图 10.2 复杂网页捕获文件框架

URL 字符串

与网页捕获相关的 URLs 在被用于 URLs 的名称树的密匙之前必须被减少成为一种可预见的关键的格式。接下来的步骤将会使用来自因特网 RCF1738，统

一资源定位器和 1808 的相对统一资源定位器的术语来说明如何进行这种缩减。该算法是检索相关的 HTTP, FTP 和文件的 URL:

1. 如果 URL 是相对的，使其绝对化。
2. 如果 URL 包含一个或者多个数字符号字符 (#)，去除最左边的数字符号和之后的任何字符。
3. 将方案却转换成小写的 ASCII 码。
4. 如果有一个端口段而且此端口是给定协议的默认端口 (HTTP 的 80 端口或者 FTP 的 20 端口)，缩减此端口段。
5. 如果路径中包含点 (.) 或者两点 (..) 的序列，按照第四章 RFC1808 转换此序列。

注：因为更加 R F C 1738，百分比符号 (%) 式不安全的，而且对于未编码的字符来说，他又是一个转义字符，所以一般是无法区分一个编码和未编码的 URL 的。比如，你无法判断一个像%00 这样的序列是一个有符号的非编码空字符或者是三个编码的字符。所以在一个 URL 中，所有编码和非编码的序列，都是不安全的。事实证明，嵌在 HTML 中的 URL 具有不安全的字符编码，而 WEB 服务器就像一个编码的通行证。因此典型的网址假定发生了一个且只有一个编码通过。其最初的 URL 编码状态是已知可以安全地把一个经历了网址转换只一个编码通过。

数字标识符

数字标识符是通过 IDS 名称树和网页捕获内容集合相关联的，它是通过 MD5 信息摘要算法产生的。确切的数据传递给该方法取决于内容设置的类型和该数字标识被 计算出来的性质。

对于一个页面设置，数据源先会被传递给 MD5 算法，其次是字符串代表任何辅助数据文件如图像（数字标识符）中引用的数据源，在该命令中，他们首先被引用。

这将产生一个复合标识符代表的视觉外观在页面设置的网页。两个 HTML 源文件是相同的，但对其中所引用的图像包含不同的数据，例如，如果他们已生成的脚本或相对 URL 所指向的，不产生同样的标识符。

注：当源数据是取自一个 PDF 文件时，只从该文件生成标识符，不生成辅助数据。

一个页面设置也可以有一个文本标识符，运用 MD5 算法计算仅仅呈现在当前数据源文本。对于一个 HTML 文件，例如，文本标识符是完全基于标记标签之间的文本，没有用图像中使用的算法。

对于一个图像集合，数字标识符的计算方法是通过数据源原始图像的 MD5 算法。例如，一个图像标识是从创建一个 GIF 图像时由 GIF 的内容计算出来的。

唯一名称的代

在从数据源生成 PDF 格式的网页，网页捕捉转化项目，如超文本链接和 HTML 表单域成相应的指定目的地和交互式表单域。这些项目的名称不能与现有项目的名称发生冲突。此外，当更新文件时，网页捕捉可能需要找到所有目的地为给定的页面设置构造域。因此，每个目表或域被赋予一个唯一的名称是来自它的建成使原来的名字，但它避免了在其他网页设置中类似的命名冲突。

注：像这类所使用的，项目的名称指向一个字符串，而不是一个名称对象。

唯一的名称是由一个编码的形式附加在页面设置的数字标识符字符串到目的地或现场原来的名字。标识符字符串必须被编码，用来移除那些在目标或者在域中含有特别意义的字符。比如，由于点在互动表格字段名中被用作分隔符，所以它在唯一名称的标识符中不能出现。因此，在内部使用两个字节编码，92 或者 112 去对应标识\p 的 ASCII 码。要注意的是，由于在字符串对象语句中，反斜杠 (\) 具有特殊的意义，所以当把反斜杠写入 pdf 文件中的时候，要在他的前面再加一个反斜杠。例如，如果原始的数字识别符是 alpha.beta，他可能在内部会被编码成 alpha\beta。然后想这样 (alpha\\beta) 被写入到 PDF 文件中。同样的，空字符（字符码是 0）在内部被编码成两个字节的 92 和 48，他们对应 \0 的 ASCII 码，如果原始的数字识别符字符串是

AlphaΦbeta

会在内部被编码成

Alpha\\0beta

然后像如下的样子被写入到 PDF 文件中

(Alpha\\0beta)

最后，反斜杠自己在内部被编码成两字就的 92 和 92，代表 ASCII 码中的 \\。在输入模式下，他们的前面都要加入一个反斜杠。因此，数字识别符字符串

Alpha\beta

会在内部被编码成

Alpha\\beta

然后像如下的样子被写入到 pdf 文件中

(Alpha\\beta)

如果名称被用作一个互动的格式的域，有一个额外的编码来确保唯一性和互动形式的兼容性。源字符串中每一个字节，将像上文所述进行编码，会被目标字符串中的两个字节来替代。每对字符串的第一个字节是 65（相当于 ASCII 字符 A）加上源字节的高四位，第二个字节是 65 加上源字节的低四位。

10.9.3、内容集合

一个网页捕获的内容集合是对由相同源数据生成的 PDF 对象集合进行描述的字典。它可能包含这个集合中的对象所共同的信息以及有关的设置本身。表 10.37 显示了这种类型的字典的内容。

页集合

页面集合是一个包含由通用源所生成的 PDF 对象的内容集合。这些页面将以他们被加入到文件中相同的顺序被列出在 0 矩阵中。一个单一的页对象只属于一个单一的页集合。表 10.38 显示了被指定给这种内容的集合的内容集合字典。

可选的 TID 条目可以被用来存储由这种页集合产生的文本标识符。这中标识符可以被用于，比如说看，判断一个文档中的文本是否发生了改变。文本标识符可能不适用于某些页面集合，在这些情形中他们应该被删除。

表 10.37

网页捕获内容集合中所有通用的条目

关键域	类型	取值
Type	名称	(可选) 此字典所描述的 PDF 对象的类型, 网页捕获内容集合中如果有此项, 一定是 SpiderContentSet
S	名称	(必须的) 此字典所描述的内容集合的子类型: SPS (蛛网型页集合) 一种页面集合 SIS (蛛网型图像集合) 一种图像集合
ID	字符串	(必须的) 内容集合的数字标识, 如果内容集合是通过 URLs 的名称书来定位的, 这就能在和他相关的 IDS 内容树的条目被发现。
O	矩阵	(必须的) 一个间接应用属于内容集合对象的矩阵。通常, 该矩阵中对象的排列顺序是未定义的, 但是可以通过指定内容集合的子类型来限制。
SI	字典或矩阵	(必须的) 内容类型, 是一个来标识内容集合中的对象是从哪里被创建的字符串。这个字符串必须符合因特网 RFC2045 中描述的内容类型。比如: 对于一个由 HTML 文件生成的数页 PDF 所组成的页面集合, 它的内容类型必须是 text/html。
TS	日期	(可选) 一个显示内容集合是何时创建的视觉戳。

表 10.38 指定网页捕获页面集合的附件条目

关键域	类型	取值
S	名称	(必须的) 本字典所描述的内容几乎的子类型, 对于一个页面集合, 必须是 SPS
T	文本字符串	(可选) 页面集合的标题, 一个人类可读的字符串
TID	字符串	(可选的) 该页面集合的文本所生成的文本标识。

图像集合

图像集合是一个包含一个组由共同源所产生的一组图像对象的内容集合, 比如多帧动画的 GIF 图像。一个单一的 XObject 只属于一个唯一的图像集合。表 10.39 展示了为此种类型的内容集合所指定的内容集合字典条目

表 10.39 为网页捕获的图形集合所指定的附加条目

关键域	类型	取值
S	名称	(必须的) 该字典所描述的内容集合的子类型, 对于图像集合, 他必须是 S S
R	整形或矩阵	(必须的) 图像集合的图像 XObject 的引用次数计数器。 对于一个含有单一 XObject 的图像集合, 它的取值是一个记录 XObject 应用次数的简单整数。如果这个图像集合包含有多个 XObject, 该取值是一个矩阵。这就是说, R 矩阵中的每一个元素存储这个图像 XObject 在对于的 O 矩阵中被引用的次数。

图像集合中的每一个图像 XObject 都有一个引用计数, 他表示 PDF 页中和这个 XObject 相关的数字。当网页捕获生成一个新的页面并且应用了 XObject 的时候, 这个引用计数就会加一, 当这个页面被毁掉的时候, 引用计数相应的减 1。如果引用计数为零, 则表示, 当前存在的页面中都没有引用 XObject, 而且它可以从图像集合 O 矩阵中移除。

10.9.4、源信息

内容集合词典中的 SI 条目标识一个或者多个包含元数据从哪里被检索的源信息词典。表 10.40 展示了这种类型字典的内容。

表 10.40

源信息字典中的条目

关键域	类型	取值
AU	字符串或者字典	(必须的) 一个字符串或者别名字典, 标识从哪里获检索数据的 URL
TS	日期	(可选) 这是一个时间戳, 他指定内容集合中的内容在什么时间会被更新到元数据一样。
E	date	(可选) 过期印章, 它指定内容集合中的内容在什么时候会过期。
S	数值	(可选) 一个表示提交表格类型的代码, 如果有的话, 他表示源数据时通过什么途径获得的。 0 提交的表格不能获取 1 通过 HTTP GET 请求获取 2 通过 HTTP POST 请求获取 只有当源信息字典和页面集合相关联的时候这个条目才能出现。默认值: 0
C	字典	(可选, 但必须直接引用) 数据字典, 它描述索引数据源的命令。只有当源信息字典和页面集合相关联的时候这个条目才能出现。默认值: 0

在简单情形下, 内容集合的 SI 条目只包含单一的源信息字典。然而, 如果通过两个或者多个 URL 来访问一个相同的数据源, 这是不正常的。当网页捕获检测到这个情形的时候, 他会从数据源中产生一个单一的内容集合, 它只包含相关 PDF 页或者图像的 XObject 的一个备份, 但是会产生不同的源信息字典来分别描述取得原始数据的方法。然后他会产生一个包含多种信息源字典的矩阵, 就像内容集字典中的 SI 条目的值一样。

信息源字典 AU 条目标识源数据时从哪里索引的。如果只有一个 URL，用一个简单的字符串作为这个条目的值就足够了。如果多个 URL 网站都通过重定向映射到同一个地方，侧代替他们的 AU 的值是 一个 URL 的别名字典。

注：为了使文件的大小跟节省空间，建议整个网址别名词典（不包括 URL 字符串）被表示为一个直接的对象，因为它的内部结构不应该被共享或外部引用。

在 TS (时间戳) 项允许每个源的位置与内容相关的设置有它自己的时间戳。这是必要的，因为在时间邮票内容设置字典仅仅指的是创造日期内容的设置。一个假设的“更新内容设置”命令可能会重置在源信息字典为当前时间，如果时间戳

发现源数据并没有因为时间的最后一个时间戳而改变。

E (过期) 项指定为每个源位置到期日期与内容相关的设置。如果当前日期和时间晚于那些指定的内容设置应考虑的原始源的内容已经过时了。

网址别名词典

如果通过 HTTP 协议访问一个网址，它会是收到一个应答的信息头来表明该请求时一个不同的网址。这个进程可能会被重定向到一个新网址上，而且很有可能还会继续下去。不难发现，通过重定向技术，多个网址最终会被定向到同样目标上。表 10.41 展示了这种类型的字典的内容。

表 10.41 网址别名字典中的条目

关键域	类型	取值
U	字符串	(必须的) 所有 C 条目指定的链接的目的地址
C	矩阵	(可选的) 一个由一个或者多个矩阵字符串组成的矩阵，分别代表由 U 条目所指定的通用目标的链接

如果网址别名字典纸包含一个网址，C 条目应该被省略。如果存在 c，他的值是一个多维矩阵的一维，每一个代表一个指向通用目标的网络连接。在每一个连接内部，网址是以字符串的形式按他们被重定向的顺序存储的。通用目标应该被取消，因为他已经被 U 条目所标识。

命令字典

网页捕获命令字典代表被网页捕获器所执行的命令，这些命令被用来索引创建新网页或者修改已经存在的 PDF 所需要的数据。本词典中的条目代表着用户所请求的被捕获的网页的内容时所指定的原始参数。这些信息被存储起来，然后这些命令可以被快速的执行来更新捕获内容。表 10.42 展示了这种类型的词典中的内容。

表 10.42 网页捕获命令字典中的条目

关键域	类型	取值
URL	字符串	(必须的) 获得源数据的初始化网址
L	整形	(可选的) 从初始化网页中检索到的页面的等级数。 默认值: 1
F	整形	(可选) 指定各种明天特性的标志的集合。默认值: 0
P	字符串或流	(可选的) 发送给 URL 的数据。
CT	字符串	(可选的) 一个描述发送给网址数据的内容类型。 默认值: application/x-www-form-urlencoded
H	字符串	(可选) 发送给网址的附件 HTTP 协议请求头
S	字典	(可选的) 一个包含在转换过程中所用到的设置 的命令设置字典

URL 条目为恢复命令指定初始化网址。L 条目指定了从该网页检索的页面的等级数。如果 L 条目被取消，它的值仍被假设为 1，表明只检索初始网址。命令字典的 F 条目是一个 32 位的无符号整数，包含的标志位指定了命令的各个特定。这个字中的标志位从低位到高位按 1 到 32 编号。表 10.43 说明了这些标志位的意义。所有未定义的标志位都是保留的，而且必须把他们的值设置为 0。

表 10.43 网页捕获命令标志

位	名称	意义
1	SameSite	如果被置位，只检索初始化网址中指定的主机中的网页
2	SamePath	如果被置位，只检索初始化网址指定路径中的

页面

3	Submit	如果被置位，这个命令代表一个提交格式。
---	--------	---------------------

SamePath 标志，如果被置位，则表明该页面只有在初始化网址中指定了相同的路径时才会被检索。如果一个页面的策略或者网络配置化组件和初始化网址的策略或者网络配置化组件想匹配，则认为他们在同一个路径中。其路径组件在最初网址匹配直至并包括最后一个斜杠 (/) 字符。比如，下面网址

<http://www.adobe.com/fiddle/faddle/foo.html>

被认为和如下初始化网址处于同一路径中

<http://www.adobe.com/fiddle/initial.html>

比较是个案的计划和网络位置不敏感元件和区分大小写的路径组件。

如果提交标志被置位，该命令表示一种提交格式。如果没有 P 条目，提交的数据将会在网址中被编码。如果有 P 条目，该命令代表一个 HTTP POST 请求。在这种情况下，提交标志的值将会被忽略。如果被发送的数据足够的小，该数据有可能被一个字符串替代。对于大量的数据，推荐用流，因为它可以被压缩。

CT 条目仅和 POST 请求有关。它像因特网 RFC2045，多用途互联邮件扩展第一部分所描述的那样，描述了发送数据的内容类型。

H 条目指定了发送给网址的请求信息的附加 HTTP 请求报头。字符串中的每一个报头以回车换行符作为借宿的标志，如下例所示：

(信息源：<http://frumble.com\015\012From:veebie@frotz.com\015\012>)

HTTP 请求头的格式是在英特网 R F C 2616，超文本传输协议—HTTP1.1 中指定的。

S 条目指定了命令设置字典，存储着指定转换引擎的设置。如果此项被忽略，将假设一个默认值。建议命令设置字典和任何具有相同设置的命令设置字典共享。

命令设置

命令字典中的 S 条目包含一个命令设置字典。它里面存储了将命令结果转换成 PDF 的转换引擎的设置。表 10.44 展示了这种类型的字典的内容。

表 10.44 网页捕获命令设置字典中的条目

关键域	类型	取值
G	字典	(可选的) 包含和所有转换引擎相关的转换引擎的全局设置。如果缺少这一项, 将使用默认的设置。
C	字典	(可选) 指定转换引擎的设置。这个字典中的每一个关键字都是一个转换引擎的内部名字。关联的值是一个字典含有与该转换引擎相关的设置。如果在该字典中找不到某一特定转换引擎的相关设置, 将使用默认设置。

C 字典中的每一个关键字都是一个转换引擎的内部名称, 它应该是一个如下格式的名称对象。

/公司: 产品: 版本: 内容类型

公司是制造转换引擎的公司的名称。产品是次转换引擎的名称。这一栏可以为空, 但是最后的冒号必须要有。版本是该转换引擎的版本。内容类型是相关联设置的内容类型的标识符。因为有些转换引擎可能包含几种转换类型, 因此这一项是必须的。

比如:

/ADBE: H2PDF: 1.0: HTML

注意内部名称的所有域都要区分大小写。公司名称必须符合附录 E 中的命名规则。对于其他域的取值, 除了不能包含冒号外, 没有其他的限制。

注: 在不明确该设置可能包含的设置信息的时候, 必须有可能对该字典做一个深度拷贝的副本。为了保证此性能, 基于该命令设置字典的方向图必须是完全独立的, 这就是说, 他必须不包含任何从其他 PDF 文件中引用的对象。

10.9.5、和网页捕获相关的对象的属性

一个给的的页对象和图像 XObject 最多只能属于一个网页捕获内容集合，调用它的父内容集合。然而，该对象没有直接指向父内容集的指针。这种指针会在那些跟踪一个对象的所以指针以确定该对象依靠什么资源的应用中造成麻烦。相反，该对象的 ID 条目包含有父内容集合的识别符，它可以通过文档内容字典中的 IDS 名称树来定位父内容集合。

在由 HTML 文件创建 PDF 页面的过程中、网页捕获器周期性地扫描下载的内容，以使他适应固定页面的大小。页对象中的 PZ 条目指定了一个放大因子，通过它网页还可以扩展到撤消缩小并查看页面在其原来的大小。也就是说，当以首先的放大效果浏览该页面的时候，一个默认用户空间的单元对于一个原始的象素。

10.10、打印前支持

本节描述 PDF 支持打印前流程的特性：

- 印前页面边界的修整
- 配套的设施包括打印机标记，比如登记目标，灰度调整，颜色工具条，以及生产过程中的剪切工具。
- 为文档中页面生成区分颜色的信息。
- 在输出设备上或者打印环境中，为了匹配文档颜色特性的输出特性，
- 支持生产陷阱以在多种调色剂之间尽量减少重合失调的视觉效果

10.10.1、页面边界

一个 PDF 页面可能是为一个准备完成的媒介所准备的，比如表格，或者作为打印前的一部分，在这里页面会被放在一个内部媒介上，比如电影或者强加复制盘。就后者而言，重要的是区分中间页和完成页。中间页可能会包括一些附件的产品相关的内容，比如剪切或者打印标记，他们都在完成页的边框之外。要处理这种情况，可以为 PDF 页面定义多达 5 个独立的界限，以控制各方面的成像过程：

- 页面边框定义了将要打印页面的物理媒介的边框。
- 为了打印标记，流血边框，这样或者那样的目的，可能在完成边界的周围扩展出很多区域。由于输出设备物理特性的限制，它也可能包括一些接近不能包

括的边缘的区域。边界之外的内容可以被丢弃，而且不会影响 PDF 文件的意义。

●当显示或者打印的时候，裁剪框定义的该区域的页面的内容将被剪辑。不像其他的工具框，裁剪框没有定义几何物理页面或拟用语的含义，而只是规定在网页上的内容剪辑。然而，在缺少附加信息的时候，裁剪框将确定如何将页面的内容将被定位到输出介质。默认值是页的媒体盒。

●裁剪边框定义了在产品输出环境中页面内容应该被修剪的区域。这可能需要一些额外的修剪区域来容纳那些受物理限制的折叠和修整设备。实际的打印页可能会包含一些落在打印框外的打印标记。默认值是也的裁剪框。

●修饰框第一了经过修整后的完成页面的预定尺寸。他可能比媒体盒子要小，以便跟随产品相关的内容，比如印刷仪器，切口痕迹和彩条。默认值是页的裁剪框。

●艺术框 (PDF 格式 1.3) 定义了页面的有意义的内容范围。默认值是页的裁剪框。

这些边界是由页对象字典中的媒体框，裁剪框，对象边框，修饰框分别决定的。他们都是在默认用户空间单元中表示的方框，修饰、裁剪、艺术边框不能超过媒体边框的边缘。如果超过了，他们将有效减少与媒体框的相交。

如果这样，它们就会被有效缩小到它们与页面边框的交叉点。图 10.3 阐明了这些边界之间的关系。(下图没有表示出修剪边框，因为他没有定义和其他任何边界的关系。)

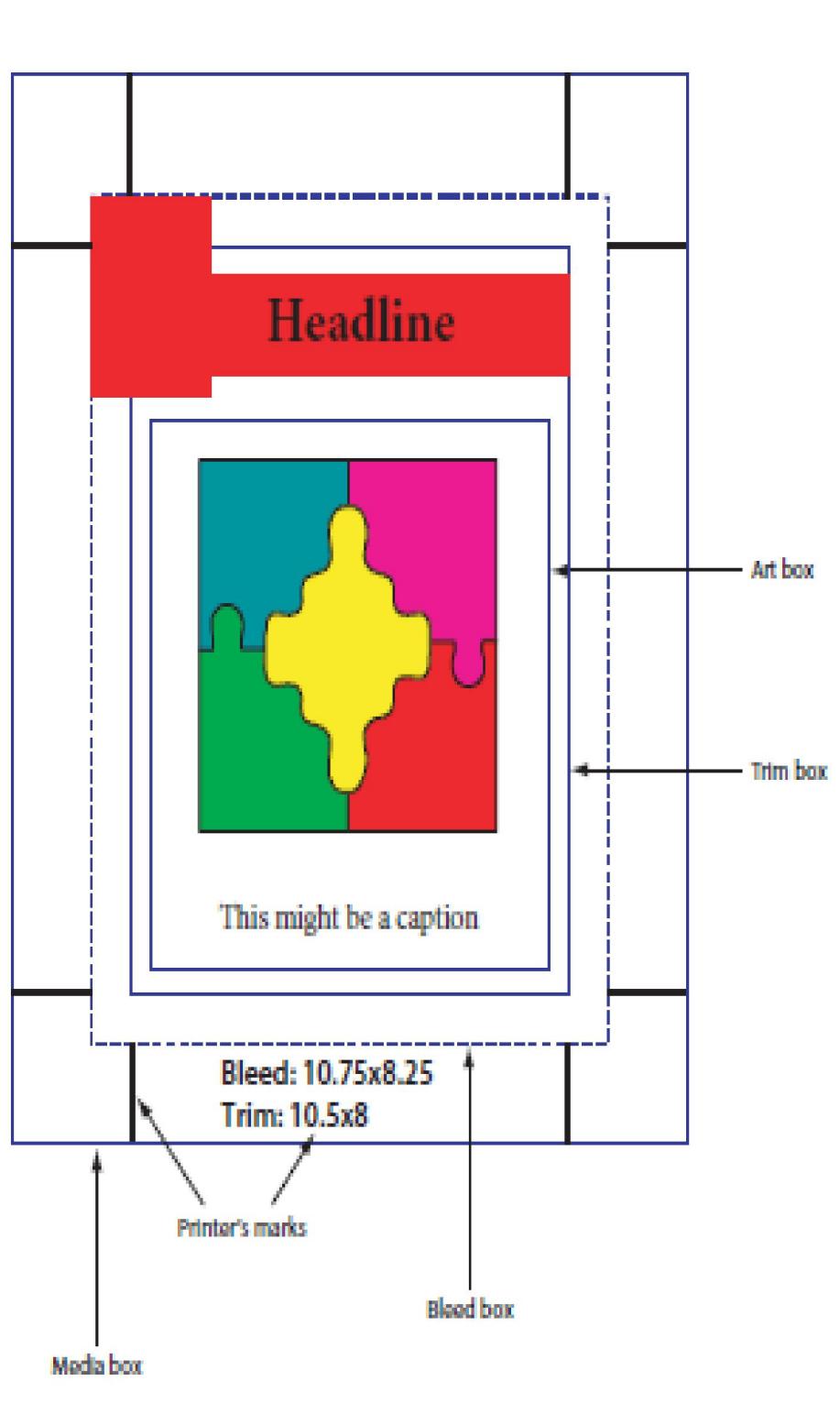


图 10.3 页面边界

不同边界的使用方法，取决于页面的用途。下面就是几种典型的用途：

●在另一个应用中置入一个页面的内容。内容边框决定了将被放入应用中内容的边界。根据使用的用法公约，被置入的内容可能被剪入内容边框或出血边框。（例如，一个将被放在一个杂志页面的四分之一页面广告，可能被别在朝向页面中间部分的广告的两边的内容边框和出血在页面边缘两边的出血边框。）页面边框和剪裁边框被忽略。

●打印一个完成的页面。这个情况是个典型的桌面或共享页面打印机，其中页面的内容定位在最终输出介质。内容边框和出血边框被忽略。页面边框可被用作选择合适大小媒体的参考。修剪边框和剪裁边框如果被表现出来，应该是和页面边框一样的。（见附录 H 的执行情况说明 161）。

●打印一个中间页以便在打印前的过程中使用。内容边框被忽略。出血边框定义了被成像的内容的边界。剪裁边框使媒介中内容的位置特殊化；它可能被用来在出血边框外产生剪切或折叠标志。页面边框内和出血边框外的内容有可能也有可能不被形象化，取决于使用的具体生产过程。

●在一张出版社表格中建立一个多页的排版。内容边框被忽略，出血边框定义了要进行成像的内容的剪辑边界；出血边框外的内容被忽略。剪裁边框使排版内页面的内容的定位具体化。剪切和折叠标记总体上通常是由排版而生成。

●在上面的情景介绍中，为某种目的解释出血，剪裁，内容边框的应用程序典型地改变了裁剪框，以施加剪辑这些箱子规定。

页面边界显示

为了用户的方便，用户应用程序提供了在屏幕上为不同的页面边界显示指引的能力。页面对象中可选的“边框颜色信息”输入（见 119 页的“页面对象”）保持了使使用于这种显示具体化的颜色和其他可见信息的边框颜色信息词典（PDF1.4）。表性地提供了可以让用户交互的设置这些特征的用户界面。请注意这些信息使专门针对页面，并可以从一个页面到另一个页面变化。

10.10.2、打印机的商标

打印机商标是添加到页面以协助协助生产工作人员在确定一个多片组件和维护工作在生产过程中的输出一致的图形符号和文本。在这些印刷行业中常用的例子包括：

- 对准板的登记目标
- 测量颜色和幽默密度的灰色坡道和颜色条。

● 显示其中输出媒介要修剪的切口痕迹

尽管 PDF 格式制作的应用，通常包括某个文档内容流中的这些标记。它们在逻辑上是分开的网页内容，通常出现在定义内容范围的边界的外部（修剪边框，剪裁边框，内容边框）。（见区域 10.10.1，“页面边界”）。

打印机的标志说明（PDF 格式 1.4）提供一个机制纳入打印机的标记到页面的 PDF 格式的代表性，同时保持它们分开，从实际网页内容。 PDF 文件中的每个网页可能包含任何这些数字说明，每一个都代表一个打印机的商标。

注意：由于打印机的标记一般超出页面内容的边界，每个标记必须被表示为一个单独的注解。否则，如果，例如，在网页上的四个角落的切口痕迹被定义在一个注解，注释矩形将包括本页面的全部内容和会干扰用户的能力来选择内容或与其它说明网页上。单独的注释定义打印机的标记也有利于在指定他们执行一个拖和拖放用户界面。

表 10.45 盒子颜色信息词典

关键词	类型	值
剪切边框	词典	(可选) 框样式词典（见表 10.46）指明的视觉特征用于显示的页面的裁剪框的指导方针。此项目被忽略，如果没有剪切是定义在页面对象。
出血边框	词典	(可选) 框样式词典（见表 10.46）指明的视觉特征用于显示该网页的出血框准则。此项目被忽略，如果没有出血框定义在页面对象。
剪裁边框	词典	(可选) 框样式词典（见表 10.46）指明的视觉特征用于显示该网页的装饰框的指导方针。此项目被忽略，如果不修剪框是定义在页面对象。
内容边框	词典	(可选) 框样式词典（见表 10.46）指明的视觉特征用于显示该网页的艺术中的指导方针。此项目被忽略，如果没有艺术框是定义在页面对象。

表 10.45 盒子颜色信息词典		
关键词	类型	值
C	阵列	(可选), 3 号在 0.0 到 1.0 范围数组, 代表组件在色彩 DeviceRGB 色彩空间是用来显示指导方针。默认值: [0.0 0.0 0.0]。
W	数字	(可选) 该指引的默认用户空间单位的宽度。默认值: 1。
S	名字	(可选) 的指引风格:
		S (固体) 一个坚固的矩形
		D (虚线) 虚线矩形。破折号模式指定为 D 项 (见下文)。
		其他准则风格可以在将来定义, 默认值: S
D	陈列	(可选) 定义一个数组破折号和差距模式, 用于绘图虚线准则 (准则上文 D 型)。破折号数组中指定的默认用户空间的单位, 在冲线在该模式的参数相同的格式图形状态 (见“行短跑模式”187 页)。在冲刺阶段没有指定并假定为 0。例如, [3 2] 和 D 项指定指引得出 3 点 2 点的差距交替破折号。默认值: [3]。

打印机商标的视觉表现是由一种 XObject 形式作为在 N (正常) 项的外观流打印机的商标标注的外观词典 (见第 8.4.4, “外观流”) 所定义的。不止一种外观可以为同一种商标而定义, 已满足不同地区和生产设施的要求。在这种情况下, 外观词典的 N 入口持有 subdictionary 包含候补外表, 每个被一个任意键标志。在注释词典的 AS (外表状态) 入口, 指定他们其中一个要被显示或者打印的。

注: 打印机的商标标注的外观字典可能包括 R (过渡) 或 D (下) 项, 但是在其中任一入口定义的外观, 永远不会被显示或打印。像所有的注释, 一台打印机的商标是指由一个注解注释字典 (见第 8.4.1 节, “诠释词典”) 所定义; 它的注释类型是 “PrinterMark” 。AP (外观) 和 F (标记) 条目 (通常是可选项) 必须存在, 如果外观字典包含不止一个外观流, 那么 AS (外观状态) 条目也必须存在。F 条目下的打印和只读标记必须被设置并且其他的都清除。(见第 8.4.2 节, “注释标记”)。表 10.47 显示了一个针对这种注释的额外的注释词条。

表 10.47 针对打印机商标注释的额外词条

关 键词	类 型	值
Sub type	名 字	(必选), 这个字典描述的这种类型的注释, 必须为 PrinterMark 为打印机的商标注释。
MN	名 称	(可选) 一个任意的识别打印机商标的名称, 例如 ColorBar 或 RegistrationTarget。

表 10.48 针对打印机商标形式字典的额外条目

关 键 词	类 型	值
MarkS tyle	文 本字符 串	屏幕上显示的可读的适合为用户显示的代表打印机商标的文本字符串
Color ants	字 典	(可选); PDF 1.4) 一个识别单个着色剂与打印机商标的词典, 例如一个颜色条。对字典中的每个条目, 关键词是一个元素的名称, 值是为该元素定义的一个分离着色空间的阵列。(参见“分离颜色空间”, 页 234)

10.10.3、分离字典

在高端印刷流程中，页面最中都会被处理成一套分离集，每个元素对应一个（见 234 页的分离颜色空间）。一般情况下，pdf 文档中的每一页都被视为一个用所有的程序元素或者是点元素绘制图形对象的复合页面。换句话说，一个页面的所有分离元素产生于该网页的一个单一的 PDF 说明。

在一些工作流程中，网页在生成 PDF 文档之前预先分离。在一个预先分离的 PDF 文档中，网页的分离被描述成分离的页面对象，每个对象绘制一个单独的元素（通常在指定 DeviceGray 色彩空间）。一旦这样，就需要额外的信息来标识与实际着色剂和相关的各分离组一起在代表所有的特定网页离职页对象。这个对象包含在每个网页对象“**SeparationInfo**”条目下的分离字典（PDF1.3）（见“**页对象**”119 页）。图 10.49 表示出了这种类型词典的内容。

表 10.49 分离字典的条目

关键字	类 型	值

网页	阵列	(必填) 一个以代表离职页间接引用的数组对象同一文件的页面。数组中的页面对象之一必须分离与本词典是相关的，和他们都必须已分离字典 (SeparationInfo 项)，其中包含的页面阵列相同的这一个。
DeviceColorant	名称或字符串	在把这表示为例如 Cyan 或者 PANTONE 35CV 分开使用的(所需要的)装置着色剂的名字.
ColorSpace	阵列	(可选) 数组定义一个分离或 DeviceN 色彩空间(参见“分离色彩空间”和“DeviceN 234 页 238 页上的颜色空间”)。它提供更多有关规定 DeviceColorant-in 的颜色特定的、交替的色彩空间、色彩转换功能, 将被用来代表着色剂作为一个过程的颜色这个信息使浏览程序来预览分离在颜色接近的装置着色剂。DeviceColorant” 时间的价值, 就必须与空间的色度的名字(如果它是一个或者是分离的空间)的空间的色度的名字(如果它是一个 DeviceN 空间)。

10.10.4、输出意向

输出意图提供了以个匹配 PDF 文档的颜色特征和文档打印的目标输出设备和生产环境的途径。这个文件目录的可选的 `outputintents` 条目（见 3.6.1 部分，“文件目录”），包含一个输出意图字典的阵列，每一个描述可能输出设备和生产条件的颜色复制特征。这些字典的内容会因为设施和条件的改变而改变。这本字典的 `S` 条目，制定了以个决定剩余条目的格式和意义的输出意图亚型。

这种多输出意向的生产工艺的使用可以允许生产过程自定义为预期的工作流程，并可使特定的工具可用。例如，一种生产设施可能会处理文件，使它符合公认的格式或者标准例如 PDF/X-1，而另一个利用 Acrobat 插件扩展在网络上生产文件分布的 RGB 输出。其中每一个工作流都会要求不同的配套的输出意向信息。多重输出意图也让同一个 pdf 文件不做修改地分发到多个输出设备。在给定的生产环境下选择使用哪种输出意图是一个事关买方和供应商之间就生产服务达成一致的问题。PDF 文档故意不包含在 pdf 文档中选择特定输出意图的选择区域。在出版时，只有一个已定义的输出意向亚型，`GTS_PDFX`，对应于 PDF / x 的格式标准（可在网络上<<http://www.npes.org/standards/>>）。表 10.50 表示出了这种类型输出意向字典的内容。其他亚型可在将来添加；任意额外亚型的名字必须符合附录 E 命名指南的描述

PDF/X 输出意向字典条目

关键词	类 型	值
<code>TYPE</code>	名 称	(可选) 这个字典描述的 PDF 类型，如果存在，必须为一个输出意向字典的 <code>OutputIntent</code>
<code>S</code>	名 称	(必选) 输出意向的亚型；必须为以个 PDF/X 输出意向的 <code>GTS_PDFX</code>
<code>OutputCondition</code>	文本 字符 串	(可选) 一个文本字符串用易读的方式简洁的识别了有意的输出设备和生产条件。这是一个对用户来说定义字符串的首选方

		法。
OutputConditionIdentifier	字符串	(必选) 一个用人或机器可读的形式区分输出设备和生产条件的字符串。如果人类可读的, 这个字符串可以用来代替在 one OutputCondition 字符串提交给用户。 这个条目的典型值是一个保持在工业标准注册表的生产条件的名称比如国际刑事法院的特征数据注册表 (见参考文献)。如果在生产时, 那在效果上匹配指定的生产条件, 就行在注册表定义的那样, 制作软件是负责提供相应的 ICC 配置文件。 如果有意的生产状况是不被认可的标准, 该条目的值可能是定制, 或由应用程序指定, 机器可读的名字。该 DestOutputProfile 条目定义 ICC 配置文件, Info 条目用于进行进一步的人类可读的鉴定。
RegistryName	字符串	一个字符串 (传统的统一资源标识符, 或 URL) 标志由 OutputConditionIdentifier 指定的字符已经被定义的注册表
Info	文本字符串	(如果 OutputConditionIdentifier 并未指定标准生产条件则必选, 否则可选) 可读的包含附加信息的文本字符串, 或有关有意目标设备或者生产条件的评论。
DestOutputProfile	流	(如果 OutputConditionIdentifier 并未指定标准生产条件则必选否则可选) 一个定义从 PDF 文档的来源颜色向输出设备元素的 ICC 配置文件流 配置文件流的格式和用在指定一个 ICCbased 色彩空间的格式相同 (见 222 页)

	的 ICCbased 色彩空间)。输出转换使用配置文件的“从智库“资料(国际商会 BT0A 术语),(向智库)信息可以选择用来重新映射源颜色值的一些其他目的地的色彩空间。如屏幕预览或硬拷贝打样。(查看附录 H 162 执行情况的说明)。
--	--

注：PDF 格式/ X 是实际上是代表不同的系列标准的一致性程度。对于一个给定的标准一致性程度可能进一步限制规定的用法和含义在输出意向词典的条目。任何此类限制接管表 10.50 所描述的优先级。

在一个输出意向字典补充，而不是在 ICCbased 文件中或默认彩色空间代替

该输出意向字典（参见 222 页“ICCBased 颜色空间”和 227 页上的”默认颜色空间“）。这些机制是专门用于描述颜色元件值的特点。输出的意图，可用于与他们一起源转换为一个特定的颜色生产条件或要求那些使显示或预期的输出打样。

输出意向字典的数据时专门用作信息的，PDF 用户程序可以自由无视它。尤其地，不要期望 PDF 产品工具在产生输出之前自动地将同一个色彩空间表现出的颜色转换为指定的目标空间。（在某些工作流程中，这种转换事实上是不可取的。当同用以特性描述为目的 ICC 源配置文件标记的 CMYK 源颜色合作时，从四个这样的颜色转换三个组成部分和背部是不必要的，并会在造成损失的保真度黑部分的价值；看到“隐智库基于彩色转换空间的“228 页上用作进一步讨论。）在另一方面，当源颜色信息在不同的基础颜色空间中表示时，例如，当结合分别生成的图像在同一 PDF 页面，有可能（尽管不要求）使用目标配置文件中指定的输出意向词典来转换源颜色到同一目标空间。（见附录 H163 上的执行情况说明）

例 10.26 表示除了一种基于 ICC 表征数据注册表的工业标准生产条件的 PDF/X 输出意向字典。例 10.27 表示除了一种定制生产条件。

例 10.26

《类型/输出意向 %输出意向字典

```
/S /GTS_PDFX
/输出条件(CGATS TR 001 (SWOP ))
/输出条件鉴别(CGATS TR 001)
/注册表名称(http://www.color.org)
/DestOutputProfile 100 0 R
>>
100 0 obj % ICC 配置文件流
<</N 4
/Length 1605
/Filter /ASCIIHexDecode
>>
流
00 00 02 0C 61 70 ... >
endstream
endobj
```

Example 10.27

```
<</Type /输出意向 % 输出意向字典
/S /GTS_PDFX
/输出条件 (Coated)
/OutputConditionIdentifier (Custom)
/信息 (Coated 150lpi)
/DestOutputProfile 100 0 R
>>
100 0 obj % ICC 配置文件流
<</N 4
/Length 1605
/Filter /ASCIIHexDecode
>>

stream
00 00 02 0C 61 70 ... >
```

endstream

endobj

10.10.5、捕获支



在最后两个案例中，捕获过程，由两个指定两种信息的捕获说明控制

1. 内部可以创建捕获的捕获区域。
2. 指定每个区域本质的捕获参数。

捕获区和捕获参数在 6.3.2 和 6.3.3 分别进行了充分的讨论，这时其它语言参考第三版。捕获指令并不是直接在一个 PDF 档案中指定（就像在附言文件）。相反，他们是由一份带有工作票的 PDF 档案中指定或者是嵌入了它。各种标准因工作票的不同格式而存在；两种典型的格式，JDF（工作定义格式）和 PJTF（便携式工作票格式），在 CIP4 文档 JDF 说明书和 adobe 科技注解#5620，可移植工作票格式（见参考书目）。

当捕获在生产的最终输出产生之前表现，捕获结果放在 PDF 文档中做后续的使用，产生的陷阱是在 PDF 档案为后续的使用。捕获结果本身被描述成捕获网络注解的内容流（见下文）。流字典可能包括描述用来产生捕获和其他表象信息的方法的额外条目。

捕获网络的注解

它是一种包含图形对象 `XObject` 绘画所需的陷阱。一页可以有多个陷阱网络基于不同捕获指令，大概是专为不同的输出设备。所有的陷阱网络对于一个给定的页面都被包含在一个单一的陷阱网络注释(见第 84, “注释”)。最多只能有一个陷阱网络注释每页，这是必须的在网页的最后一个元素 `Annots` 阵列(参见“页面”页上的物体 119)。这保证了陷阱网络打印出来后，所有的页面的其他内容。(看到在附件 H 实施注 164)。

这个表格 `XObject` 定义一个陷阱网络都被指定为一个外表流在 `N`(正常的)进入网络注释的外表陷阱字典。(见第 8.4.4、外观溪流) 如果超过一个陷阱网络定义为同一页，氯进入持有 `subdictionary` 包含备用陷阱网络，每一个由一个武断的关键。作为 `AS`(外表的国家)中输入注释字典选定其中一种电流陷阱网络被显示或打印。

在一个指定的捕获说明下，为一个指定页面而产生的一套完整的捕获被称为捕获网络 (PDF1.3)。它是一个包含在一个网页上绘制必要捕获的图形对象的 `XObject` 形式。一个网页可能包括不止基于不同捕获说明的捕获网络，为不同的输出设备而设置。给定页面的所有捕获网络包含在一个单一的捕获网络注解中(见 119 页的页面对象)。这可以确保捕获网络在所有网络打印之前被打印。(见附 H 164 页的执行情况说明)

定义一个捕获网络的 `Xobject` 形式被指定为捕获网络注解下的表象字典下的 `N` 条目下的一个表象流。(见第 8.4.4 流，“外观流”)。如果为同一个页面定义不同的捕获网络，那么 `N` 条目下含有一个包含由任意关键词鉴别的可选捕获网络的亚字典。注解字典下的 `AS` 条目指定其中之一作为要被打印或显示的当前的捕获网络。

注意：捕获网络表象字典可能包括 `R` (翻转) 和 `D` (下拉) 条目，在其中任意条目中定义的外观从来不会被打印。

像其他所有的注解一样，捕获网络注解是由一个注解字典定义的(见第 8.4.1 部分，“注解字典”)；它的注解类型是 `TrapNet`。`AP` (外观) 和 `AS` (外观状态)，和 `F` (标志) 条目(通常可选)必须被呈现，打印和只读按钮必须被设

置，其他设置要被清除（见 8.4.2，“注解字典”）。表 10.51 表示除了针对这种注解的额外注解字典条目。

这个 **version** 和 **annotstates** 条目，如果被表现出来，是用来探测一个可能需要注解网络的页面的内容的变化。这个 **version** 阵列决定了可能被编辑应用程序改变的页面内容元素并且因此使捕获网络无效。因为一个捕获网络注解最多只有一个 **version** 阵列，任何产生一个新的捕获网络的应用程序，现有的有效性进行了验证捕获网络，列举了物体辨识阵列和验证结果完全符合阵列的当前目录。任何被发现是无效的捕获网络必须重新产生。（见附录 H 165 和 166 页的参考注解）。

LastModified 条目从 PDF1.4 开始，并且可能用来代替 **version** 阵列，来追踪一个网页捕获网络的变化。（捕获网络注解必须包括 **lastmodified** 条目 **version** 和 **annotstates** 的结合，不是三者都有）。如果页面对象的 **lastmodified** 条目下的修改时间比捕获网络注解中的那个晚，页面的捕获网络不可用并且必须重新产生。但是，注意，并不是所有的编辑程序和扩展插件正确获取了这些修正信息。这些追踪捕获网络修改日期的方法，只可能被可信赖地用在修改日期的完整性被确保的可控制的工作流环境下。

表 10.51 指定捕获网络注解的额外条目

关键词	类型	值
Subtype	名称	这种字典描述的注解类型；对一个捕获网络注解来说必须是 TrapNet
LastModified	日期	如果没有 <i>Version</i> 和 <i>AnnotStates</i> 则必选；如果 <i>version</i>

		和 <i>annotstates</i> 存在则必须空缺; PDF1.4) 日期和时间当捕获网络是最新修改的
Version	阵列	<p>(必填如果 <i>AnnotStates</i> 存在, 必须空缺如果 <i>LastModified</i> 存在)</p> <p>对所有对象在网络上的捕获被产生, 而且, 如果改变目前的无序时间页面描述数组, 可能会影响页面的外观。</p> <p>如果存在, 数组必须包括以下对象:</p> <ul style="list-style-type: none"> • 全部内容流确定在页面对象的内容项 (见 119 页“页对象”) • 所有资源对象在页面的资源字典 (比其他程序集) (见 3.7.2 节, “资源词典”) • 所有资源对象在页面上的任何形式 <i>XObjects</i> 资源字典 (比其他程序集) (见 4.9 节, “表格 <i>XObjects</i>”) • 与所有的 <i>OPI</i> 在页面上 <i>XObjects</i> 相关字典 (见第 10.10.6, “开放印前界面 (<i>OPI</i>)”)
AnnotStates	阵列	<p>(必填, 如果目前的版本是, 必须为缺席如果 <i>LastModified</i> 存在) 的</p> <p>数组名称代表的外观状态 (作为值对象项), 与该页面相关说明。外观国家必须将其作为在网页的 <i>Annots</i> 阵列相同的顺序列出的说明 (见“页对象”119 页)。对于一个没有作为入注解, 相应的数组元素应该为空。任何国家都不应出现的捕获网络包括注解本身</p>
FontFauxing	阵列	(可选) 代表说, 在被 <i>fauxed</i> 捕获网络的页面生成 (由替换字体取代) 字体字体字典数组。

捕获网络外观

每个条目在 *N*(正常的) *subdictionary* 的捕获网络注释的外观词典拥有一个外表流定义一个捕获网络 *associated* 与给定的页面。像所有的外表, 一个捕获网

络流对象定义表格 `XObject` (见第 4.9”，形成 `XObjects`”)。身体的图形对象包含了需要画捕获的捕获网络。它的字典条目包括,除了标准的参赛表格字典,额外的条目显示在表 10.52

表 10.52 针对捕获网络外观流的额外条目

关键词	类型	值
<code>PCM</code>	名称	(必填) 进程的颜色模式, 是假设当这个捕获网络创建的名称, 相当于 PostScript 页面器件参数 <code>ProcessColorModel</code> (见组, PostScript 语言参考, 第三版 6.2.5)。有效值是 <code>DeviceGray</code> , <code>DeviceRGB</code> , <code>DeviceCMYK</code> , <code>DeviceCMY</code> , <code>DeviceRGBK</code> 和 <code>DeviceN</code> 。
<code>SeparationColorNames</code>	阵列	(可选) 以一个数组的形式返回名称识别, 当这种假设料网络;相当于邮报-脚本页装置参数的名字(见第 6.2.5 相同的语言, 第三版参考跋)。着色剂所暗示的过程中颜色模型可自动和 PCM 不需要明确宣布。如果这个入口, 在假设 PCM 所隐含的料。
<code>TrapRegions</code>	阵列	(可选) 向 <code>TrapRegion</code> 对象确定页的诱捕区和间接引用的数组相关的捕捉参数, 如 Adobe 技术描述注意#5620, 便携式作业传票格式。这些引用的对象包括部分的 PJTF 工作票是在 PDF 文件中嵌入。当捕捉区和参数是由一个外部工作票(或以其他方式, 如 JDF 定义), 此项目是没有的。
<code>TrapStyles</code>	文本字符串	(可选) 一个记录的文本字符串的程序可以用它来形容这个捕获网络使用者(例如, 允许切换捕获网络)。

注:预先分离的 PDF 文件(见第 10.10.3, ”分离的字典”),不能被捕获,因为捕获在不同颜色之间的边界定义,预分离文件只使用一种颜色。因此预分离必须在被捕获后发生,不会提早。一个应用程序预分离的 PDF 文档,负责为分离的捕获网络计算新版本阵列。

10.10.6、开放的打印页面

打印之前的工作流程经常会牵涉到很多应用，比如图像设计，页面布局，文字处理，图片处理，以及文件资源处理。由于最后文件中的很多部分会从一个中移动到另外一个，将那些高分辨率的图像，在某些情形下他们可能会非常大，可能会和剩下的文档的内容一样大，将他们从原文档中分开是十分有用的。开发印前接口是最初由奥尔德斯公司开发的低分辨率占位符或者代理机制，因为这种高清晰度的图像。代理服务器通常由一组完整的分辨率 `downsampled` 版本形象，以用于屏幕显示和校对。在一个文档打印之前，他会通过 OPI 服务进行过滤，它用原来高分辨率的图像代替这种代理。

在 PostScript 编程中，OPI 代理是有特别的 OPI 注释所围绕的 PostScript 代码定义的。它指定了位置、图像裁剪、大小的调整，翻转、颜色以及其他属性等信息。在 PDF 中，代理被作为一个图像或者一个和包含与 PostScript 中 OPI 注释所传递的相同信息的 OPI 字典相关联的表格 XObject。支持两种版本的 OPI，版本 1.3 和版本 1.0。在 1.3 版本的 OPI 中，一个代理由一个单一的图像组成，它不会改变图像的状态，可能用一个 XObject 来代替，否则，它必须是一个表格 XObject。在 OPI2.0 中，代理经常需要改变图像的状态，因此他必须由一个表格 XObject 表示。

一个代表 OPI 代理的 XObject，在他的图像或者表格字典中，必须包含 OPI 选项。该选项的值是一个 OPI 版本字典，该版本字典该代理所对应的 OPI 版本。该字典由一个单一的选项组成，他的名称是 1.3 或者 2.0，他的取值是定义代理 OPI 属性的 OPI 字典。

表 10.53 OPI 版本字典中的选项

关键域	类型	取值
Version number	字典	(必须的，PDF1.2) 一个指定代理属性的 OPI 字典。该项的关键域必须是名称 1.3 或者 2.0，它标识与该代理对应的 OPI 版本。

注：和在其他任何一个 PDF 字典中一样，OPI 字典中的关键域必须是一个名称对象。该 OPI 版本字典应该以如下任意一种格式写在 PDF 文件中。

<< /1.3 d 0 R >> % OPI 1.3 字典

or

<< /2.0 d 0 R >> % OPI 2.0 字典

d 是相应 OPI 字典的对象号

表 10.54 和表 10.55 分别描述了 OPI1.3 和 OPI2.0 OPI 字典中的内容和相应的 PostScript OPI 注释。字典选项以与相应的 OPI 注释应该在 PostScript 程序中出现的顺序被列出。这些选项在 OPI 服务上详细的意义和效果可以在开放式印前接口 1.3 和 Adobe 技术手册中#5600，开发印前接口 2.0 中找到。

表 10.54 OPI 字典 1.3 中的选项

关键域	类型	OPI 注释	取值
Type	名称		(可选的) 该字典所描述的 PDF 对象的类型，如果有，必须是指定 OPI 字典的 OPI
Version	数字		(必须的) 该字典所指 OPI 的版本，必须是 1.3 的号码。
F	文件规范	%ALDImageFilename	(必须的) 包含与代理相对印的图片的外部文件

ID	字符串	%ALDImageID	(可选的) 一个表明图像的识别字符串
Comment	文本字符串	%ALDObjectComments	(可选的) 一个人类可读的注释, 一般包含有指定 OPI 服务操作或者图像处理的指令或者建议。

关键域	类型	OPI 注释	取值
Size	数组	%ALDImageDimensions	(必须的) 一个针对表格由两个整形元素组成的数组, 格式如下 [<i>pixelsWide pixelsHigh</i>] 指定图像像素的尺寸。
Croprect	方框	%ALDImageCropRect	(必须的) 一个由四个整形元素组成的数组, 格式如下 [<i>left top right bottom</i>] 用以指定将要使用的图像的一部分。
CropFixed	数组	%ALDImageCropFixed	(可选的) 一个拥有和 Croprect 相同格式和意义的数组, 但是是用实数表示的。默认值:

Croprect 的值

Position 数组 %ALDImagePosition (必须的) 一个由八个元素组成
的数组, 格式如下

[/ /x / /y u/x u/y urx ury /rx /ry]

用以指定被裁减图像在页面上的位置。 $(/ /x, / /y)$ 表示用户空间左下角的坐标。 $(u/x, u/y)$ 表示用户空间左上角的坐标。 (urx, ury) 表示用户空间右上角的坐标。 $(/rx, /ry)$ 表示用户空间右下脚的坐标。指定坐标必须定义一个平行四边形。这就是说他们必须满足以下条件:

$$u/x - / /x = urx - /rx$$

和

$$u/y - / /y = ury - /ry$$

Position 和 Croprect 联合决定了
图像的缩放, 旋转, 反射,
和偏差。

Relusotion 数组 %ALDImageResolution (可选的) 一个由两个数字组成的数
组, 格式如下 [horizRes vertRes]
指定样例中图像的分辨率

关键域	类型	OPI 注释	取值
ColorType	名称	%ALDImageColorType	(可选的) 由 Color 选项指定的颜 色的类型。合法的取值是:

Process、Spot、Separation。

默认值: Spot

Color 数组 %ALDImageColor (可选的) 一个由四个数字和一个字符串组成的数组, 格式如下

[C M Y K colorName]

指定该图像将要呈现的颜色的名称和取值。C、M、Y, K 的取值必须在 0.0 到 1.0 范围之类。默认值是 [0.0 0.0 0.0 1.0 (Black)].

Tint 数字 %ALDImageTint (可选的) 一个在 0.0 到 1.0 范围内的数字。它指定被 Color 项所指定的讲要显示的图像的颜色的浓度。默认取值: 1.0。

Overprint 布尔型 %ALDImageOverprint (可选的) 一个根据其他分隔符上的标记觉得该图像时该被嵌套打印(true)或者取消(false)的标志。默认值: false

ImageType 数组 %ALDImageType (可选的) 一个由两个整数组成的数组。格式如下
[samples bits]
指定每个像素中样品的数量和

图像中样品的位数。

GrayMap 数组 %ALDImageGrayMap (可选的) 一个含有 2 的 n 次方

一个整形元素的数值，取值范围是 0~65535，由于记录图像亮度和对比度的变化。

Transparency 布尔 %ALDImageTransparency (可选的) 一个指定是否在图像中将白色像素视作透明的标志。
默认值: true

Tags 数组 %ALDImageAsciiTag<NNN> (可选的) 一对数组，格式如下

[tagNum1 tagText1 … tagNumn tagTextn]
每一个 tagnum 代表一个 TIFF 标记号，每一个 tagText 是一个代表相应 ASCII 标记值的字符串。

表 10.55 2.0 版本 OPI 中的字典选项

关键域	类型	OPI 注释	取值
Type	名称		(可选的) 该字典所描述的 PDF 对象的类型，如果使用该选项，必须是一个针对 OPI 字典的 OPI。
Version	数字		(可选的) 该字典所指定的 OPI 的版本。必须是号码 2 或者 2.0
F	文件规范	%%ImageFilename	(必须的) 包含低分辨率像素的

外部文件

MainImage	字符串	%%MainImage	(必须的) 包含与代理相对应的全分辨率的图像的文件的路径名称或者唯一标示全分辨率
-----------	-----	-------------	--

图像的标识字符串。

Tags	数组	%%TIFFASCIITag	(可选的) 双元素数组, 格式如下:
------	----	----------------	--------------------

$[tagNum1 tagText1 \dots tagNumn tagTextn]$

每一个 `tagNum` 代表一个 TIFF 标记号, 每一个 `tagText` 是一个代表相应 ASCII 标记值的字符串。

Size	数组	%%ImageDimensions	(可选的, 看下面的注释) 一个含有两个元素的数组。格式如下 $[width height]$ 指定像素中图像的大小。
------	----	-------------------	--

关键域	类型	OPI 注释	取值
CropRect	方框	%%ImageCropRect	(可选的, 参考下面注释) 一个四个元素的数组, 格式如下, $[left top right bottom]$ 用于指定将要使用的图像的一部分。 注: Size 和 CropRect 项应该同时存在或者不存在。如果存在, 他们必须满足如下条件:

$$0 \leqslant left < right \leqslant width$$

和

$$0 \leq top < bottom \leq height$$

请注意，在此坐标空间，正 Y 轴垂直向下延伸，因此要求 `top < bottom`

OverPrint 布尔 `%%ImageOverprint` (可选的) 一个根据其他分隔符上的标记觉得该图像时该被嵌套打印 (`true`) 或者取消 (`false`) 的标志。默认值: `false`

Lnks 名称或数组 `%%ImageLnks` (可选的) 一个指定为该图像提供的着色剂的图像和数组。它的取值可能是一个全色名称、注册或者下面格式数组

`[/monochrome name1 tint1 ... namen tintn]`
每一个名称都是一个代表调色剂的字符串。

每个色彩是一个实数范围在 0.0 到 1.0 指定的浓度这种着色剂应适用。

IncludedImageDimensions

数组 `%%IncludedImageDimensions` (可选的) 一个由两个整形数组组成的数组，格式如下 [`pixelsWide pixelsHigh`] 指定像素中包含图像的大小。

IncludedImageQuality

数字 `%%IncludedImageQuality` (可选的) 表示包含的图像的质量的数字，合法的取值是 1、2、3
