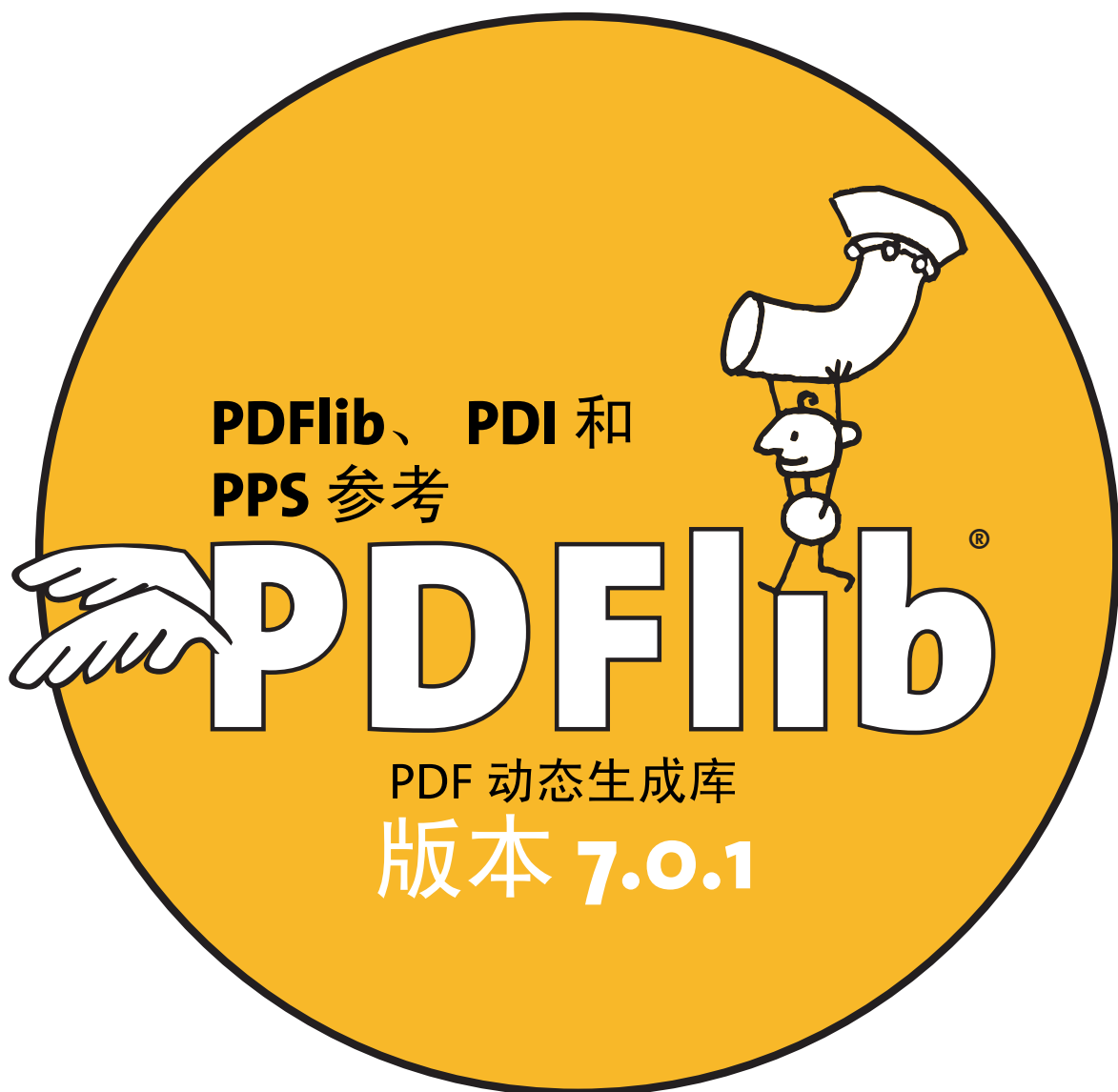


www.fanganwang.com 龙博方案网

PDFlib 7.0.1 控件使用手册



综合版本：

**Cobol、C、C++、Java、Perl、PHP、Python、RPG、
Ruby 和 Tcl**

版权所有 © 1997–2007 PDFlib GmbH 和 Thomas Merz。保留所有权利。
在此授予 PDFlib 用户许可，可以内部使用为目的复制本手册的印刷或数字副本。

PDFlib GmbH
Tal 40, 80331 München, Germany
www.pdflib.com
电话 +49 • 89 • 29 16 46 87
传真 +49 • 89 • 29 16 46 86

如果您有任何疑问，请查看 tech.groups.yahoo.com/group/pdflib 上的 PDFlib 邮件列表及相关存档。

许可联系人: sales@pdflib.com
商业 PDFlib 许可证持有人支持: support@pdflib.com (请提供您的许可证号)

本出版物及其所含信息“按原样”提供；如有更改，恕不另行通知，且不应视为 PDFlib GmbH 所做的承诺。PDFlib GmbH 不对存在的任何错误和不准确性承担任何责任，不对本出版物做出任何形式（明示、默示或法定）的保证，并明确声明：不对其适销性、特定用途的实用性以及有关第三方权利的无侵权做出任何形式的保证。

PDFlib 及 PDFlib 徽标是 PDFlib GmbH 的注册商标。在此授予 PDFlib 许可证持有者在其产品文档中使用 PDFlib 名称和徽标的权利。但是，并不要求使用 PDFlib 名称和徽标。

Adobe、Acrobat、PostScript 和 XMP 是 Adobe Systems Inc. 的商标。AIX、IBM、OS/390、WebSphere、iSeries 和 zSeries 是 International Business Machines Corporation 的商标。ActiveX、Microsoft、OpenType 和 Windows 是 Microsoft Corporation 的商标。Apple、Macintosh 和 TrueType 是 Apple Computer, Inc. 的商标。Unicode 及 Unicode 徽标是 Unicode, Inc. 的商标。Unix 是 The Open Group 的商标。Java 和 Solaris 是 Sun Microsystems, Inc. 的商标。HKS 是 HKS 品牌联名的注册商标：Hostmann-Steinberg、K+E Printing Inks、Schmincke。其他公司的产品和服务名称可能是其他公司的商标或服务标志。

软件应用程序或用户文档中显示的 PANTONE® 颜色可能不符合 PANTONE 确定的标准。有关准确的颜色，请参阅最新版的 PANTONE 颜色文献。PANTONE® 和其他 Pantone, Inc. 商标为 Pantone, Inc. 所有。版权所有 © Pantone, Inc., 2003。Pantone, Inc. 是颜色数据和 / 或软件的版权所有人，并已授予 PDFlib GmbH 使用许可，但仅限于同 PDFlib 软件一起分发。除非作为执行 PDFlib 软件的一部分，否则不得将 PANTONE 颜色数据和 / 或软件复制到另一此案或存储设备中。

PDFlib 包含下列经过修改的第三方软件组件：
ICLib，版权所有 © 1997-2002 Graeme W. Gill
GIF 图像解码器，版权所有 © 1990-1994 David Koblas
PNG 图像参考库 (libpng)，版权所有 © 1998-2004 Glenn Randers-Pehrson
Zlib 压缩库，版权所有 © 1995-2002 Jean-loup Gailly and Mark Adler
TIFFlib 图像库，版权所有 © 1988-1997 Sam Leffler，版权所有 © 1991-1997 Silicon Graphics, Inc.
Eric Young 编写的密码软件，版权所有 © 1995-1998 (eay@cryptsoft.com)
Independent JPEG Group 的 JPEG 软件，版权所有 © 1991-1998, Thomas G. Lane
密码软件，版权所有 © 1998-2002 The OpenSSL Project (www.openssl.org)
Expat XML 分析器，版权所有 © 1998, 1999, 2000 Thai Open Source Software Center Ltd

PDFlib 包含 RSA Security, Inc. 的 MD5 消息摘要算法。



作者: Thomas Merz
设计和插图: Alessio Leonardi
质量控制 (手册): Katja Schnelle Romaus、Kurt Stützer
质量控制 (软件): 软件开发组

目录

1 PDFlib 编程概念 5

- 1.1 数据类型 5
- 1.2 选项列表 5
- 1.3 函数范围 9

2 常规函数 11

- 2.1 参数处理 11
- 2.2 设置 13
- 2.3 文档函数 16
- 2.4 页面函数 22
- 2.5 PDFlib 虚拟文件系统 (PVF) 26
- 2.6 异常处理 28
- 2.7 日志 30

3 文本函数 31

- 3.1 字体处理 31
- 3.2 Type 3 字体定义 37
- 3.3 编码定义 39
- 3.4 简单文本输出 40
- 3.5 Unicode 转换函数 45

4 格式化函数 47

- 4.1 使用文本流的单行文本 47
- 4.2 使用文本流的多行文本 52
- 4.3 表格式化 64
- 4.4 匹配框 70

5 图形函数 73

- 5.1 图形状态 73
- 5.2 保存和恢复图形状态 76
- 5.3 坐标系统转换 77
- 5.4 显式图形状态 79
- 5.5 路径构建 81
- 5.6 路径绘制和剪切 84
- 5.7 图层 86

6 颜色函数 89

- 6.1 设置颜色和色彩空间 89
- 6.2 ICC 色彩特征描述文件 92
- 6.3 图案和着色 95

7 图像和模板函数 99

- 7.1 图像 100
- 7.2 模板 106
- 7.3 缩览图 107

8 PDF 导入函数 (PDI) 109

- 8.1 文档和页面 109
- 8.2 pCOS 函数 114
- 8.3 其他 PDI 处理 116
- 8.4 不常用的 PDI 参数 117

9 Personalization Functions (PPS) 119

10 交互功能 123

- 10.1 交互元素的参数 123
- 10.2 动作 123
- 10.3 命名目标 127
- 10.4 注释 129
- 10.5 表单域 134
- 10.6 书签 139

11 多媒体功能（3D 图片） 141

12 文档交换 143

- 12.1 文档信息域 143
- 12.2 XMP 元数据 144
- 12.3 标签 PDF 146

A 所有函数的列表 149

B 所有参数的列表 151

C 所有选项列表 153

D 修订历史记录 163

索引 165

1 PDFlib 编程概念

1.1 数据类型

本手册记录了用于各种语言绑定的函数 / 方法范例。各种语言绑定之间的主要差异在于：在面向对象的语言绑定中，PDFlib 方法的名称中不包含 `PDF_` 前缀，而在其他语言绑定中，`PDF_` 前缀却是所有函数名称的一部分。此外，PDF 上下文参数必须作为第一个自变量提供给非面向对象的语言绑定中的所有函数。相反，面向对象的语言绑定将 PDF 上下文隐藏在语言包装程序创建的对象中

表 1.1 详细说明了所有语言绑定使用的 PDF 文档类型以及字符串类型。有关文本和字符串处理的详细信息，请参见 *PDFlib 教程*。由于所有绑定中都有明显的对应数据类型，所以表中没有提及 **整型**、**长整型**和**双精度型**数据。

表 1.1 语言绑定中的数据类型

语言绑定	<i>p</i> 参数 ?	PDF_ 前缀 ?	字符串数据类型	二进制数据类型
C	是	是	<code>const char *</code> ¹	<code>const char *</code>
C++	否	否	<code>string</code> ²	<code>char *</code>
Cobol	是	否 ³	<code>STRING</code>	<code>STRING</code>
Java	否	否	字符串	<code>byte[]</code>
Perl	是	是	字符串	字符串
PHP	是	是	字符串	字符串
PHP 5（面向对象）	否	否	字符串	字符串
Python	是	是	字符串	字符串
RPG	是	是	<code>string</code> ，但必须添加 <code>x'oo'</code>	<code>data</code>
Ruby	否	否	字符串	字符串
Tcl	是	是	字符串	字节数组

1. 这里认为 C 语言 `NULL` 字符串值相当于空字符串。
2. `NULL` 字符串值不能用于 C++ 绑定。
3. Cobol 程序必须使用 *PDFlib* 函数的缩写名称。

1.2 选项列表

选项列表是一种强大且简便的 PDFlib API 函数调用控制方法。许多 API 方法都支持选项列表（简称 *optlists*），而不要求大量的函数参数。这些选项列表是字符串，可以包含任意数量的选项。选项列表支持各种数据类型以及数组等复合数据。在大多数语言绑定中，可通过连接所需关键字和值来简便地构建 *optlists*。C 语言编程人员也许需要使用 *sprintf()* 函数构建 *optlists*。表 1.2 列出了各种示例。

表 1.2 选项示例

选项类型	示例
浮点型	<code>opacityfill=0.75</code>
百分比	<code>leading=150%</code>
布尔型	<code>embedding</code> （等效于 <code>embedding=true</code> ）
布尔型	<code>nokerning</code> （等效于 <code>kerning=false</code> ）
字符串	<code>password { secret string }</code> （该字符串值包含三个空白）

表 1.2 选项示例

选项类型	示例
字符串	password {weird\}string} （该字符串值包含一个右大括号）
单字符	replacementchar=space （等效于 replacementchar=0x20）
关键字	blendmode=overlay
矩形	cropbox={ 0 0 500 600 }
包含三个数字的列表	dasharray={ 11 22 33 }
包含两个关键字的列表	position { center bottom }
包含一个列表的列表	polylinelist={ {10 20 30 40 } }
色彩	backgroundcolor={ cmyk 0 1 0 0 }
包含单个动作的列表	action={ activate { 0 1 2 } }
包含三个动作的列表	action={ keystroke=0 format=1 validate=2 }
选项列表	metadata={ filename=info.xml }
包含一个选项列表的列表	fill={ { area=table fillcolor={rgb 1 0 0} } }
包含两个选项列表的列表	fill = { { area=rowodd fillcolor={rgb 0 1 0} } { area=roweven fillcolor={rgb 1 0 0} } }

选项列表语法 `optlist` 是一个字符串，包含一个或多个关键字 / 值对。
可用任意空白字符（例如空格、制表符、回车符、换行符等）或等号 “=” 分隔关键字和值，以及多个关键字 / 值对。另外，还可用等号 “=” 分隔关键字和值：

```
key value
key=value
```

如果相应值是个包含空白字符或等号的字符串，则必须用括号将该字符串括起来：

```
key={ multiple words }
key={ value=containing=equal=signs }
```

由于选项列表将由左至右计算，可以在同一列表内多次使用同一选项。这时，最后一次出现的选项将覆盖前面的选项。在下面的示例中，第一个选项赋值将被第二个覆盖；该选项列表处理完后，键值将为 2：

```
key=1 key=2
```

下文将更加详细地讨论选项列表支持的下列数据类型：

- ▶ 简单值：布尔型、字符串、内容 / 超文本 / 名称字符串、单字符、关键字、浮点型、整型、句柄
- ▶ 复合值：列表、矩形、动作列表、色彩

对于列表类型的选项，必须用大括号将该列表括起来，并且列表元素必须用空白字符分隔开来：

```
key={ value1 value2 value3 }
```

如果选项列表包含多个关键字 / 值对，则确保在左大括号前和右大括号后插入空白字符（也可以在选项名称和左大括号之间使用等号）：

```
key1={ ...values... } key2={ ...values... }
```

某些列表类型的选项接受 *option list* 或 *list of option lists* 类型。*option list* 类型的选项包含一个或多个从属选项。*list of option lists* 类型的选项包含一个或多个嵌套选项列表。处理嵌套选项列表时，指定适当数量的右大括号很重要。有关示例，请参见表 1.2。

简单值 简单值可以使用下列任一数据类型：

- ▶ 布尔型：*true* 或 *false*；如果省略布尔选项的值，则缺省值为 *true*。作为简略表示法，可使用 *noname* 替代 *name=false*。
- ▶ 字符串：这些字符串是 ASCII 纯文本字符串，通常用于不可本地化的关键字。包含空白字符或 “=” 的字符串必须用 { 和 } 括起来。可以使用 {} 构建空字符串。如果字符 { 和 } 是字符串的一部分，则必须在前面添加 \ 字符。
- ▶ 内容字符串、超文本字符串和名称字符串：这些字符串可以容纳各种格式的 Unicode 内容；有关这些字符串类型和字符串选项的编码选择的详细信息，请参见 *PDFlib 教程*。
- ▶ 单字符：这些字符串是单个的 Unicode 值，可支持多种语法变体：10 进制数值（例如 173）、带 x, X, ox, oX 或 U+ 前缀的 16 进制数值（xAD、oxAD、U+ooAD），以及不带 “&” 和 “;” 修饰符的数字引用、字符引用和字形名称引用（*shy*, #xAD, #173）。另外，也可以提供文字字符。单字符必须处于 0-65535 (0-oxFFFF) 的范围内。
- ▶ 关键字：预定义固定关键字列表中的一项。
- ▶ 浮点型和整型：10 进制浮点小数或整数；可使用小数点和逗号作为浮点值的小数分隔符。整型值可以 x、X、ox 或 oX 开头，指定 16 进制值。有些选项可直接在值后添加 % 字符，以支持百分比；相关文档中有具体说明。
- ▶ 句柄：包括各种类型的对象句柄，例如：字体、图像或动作句柄。从技术角度讲，这些值是整型值。

根据选项的类型和解释，可能还有其他限制。例如，整型或浮点型选项可能会限制在一定的值范围内；句柄必须对相应的对象类型有效，等等。各个函数说明中具体规定了各自的选项条件。

列表值 列表值由多个值组成，可以是简单值或依次排列的列表值。列表要用 { 和 } 括起来。

矩形 矩形是个包含四个浮点值的列表，指定矩形左下角和右上角的坐标。根据不同的选项，解释矩形坐标的坐标系统也不尽相同（如标准或用户坐标系统），将单独论述。

动作列表 动作列表可指定一个或多个动作。列表中的每一项都包括一个事件关键字（触发器）和一个动作句柄列表；该动作句柄列表必须用 *PDF_create_action()* 创建。各个动作将按照列表顺序执行。对于各个选项，将单独介绍允许的事件集（例如 *docopen*）和动作类型（例如 *JavaScript*）。

色彩 色彩选项是一个包含色彩空间关键字和具有不同数量浮点值的列表（具体取决于特定色彩空间）。色彩空间关键字等同于 *PDF_setcolor()*（参见第 89 页上的第 6.1 节“设置颜色和色彩空间”）。表 1.3 包含特定说明和示例。根据具体函数的详细说明，特定选项列表可能只提供上述关键字的一部分。

表 1.3 选项列表中色彩数据类型的关键字

关键字	其他值	示例
<i>gray</i>	灰度色彩空间的一个浮点值	{ gray 0.5 }
<i>rgb</i>	<i>RGB</i> 色彩空间的三个浮点值	{ rgb 1 0 0 }
<i>cmyk</i>	<i>CMYK</i> 色彩空间的四个浮点值	{ cmyk 0 1 0 0 }
<i>lab</i>	<i>Lab</i> 色彩空间的三个浮点值	{ lab 100 50 30 }
<i>spot</i>	专色句柄和指定色调值的浮点值	{ spot 1 0.8 }
<i>spotname</i>	专色名称和指定色调值的浮点值	{ spotname {PANTONE 281 U} 0.5 }
<i>spotname</i>	类似于上述简单格式的 <i>spotname</i> ，但是可以增加色彩值，以便为自定义专色指定替代颜色（即， <i>PDFlib</i> 内部未知的专色名称）。如果多个选项定义同一自定义专色名称，则所有定义必须一致（即，定义相同的替代颜色）。	{ spotname {PDFlib Blue} 0.5 { lab 100 50 30 } }
<i>iccbasedgray</i>	一个浮点值	{ iccbasedgray 0.5 }
<i>iccbasedrgb</i>	两个浮点值	{ iccbasedrgb 1 0 0 }
<i>iccbasedcmyk</i>	三个浮点值	{ iccbasedgray 0 1 0 0 }
<i>pattern</i>	图案句柄	{ pattern 1 }
<i>none</i>	指定色彩缺失的情况	none

1.3 函数范围

PDFlib 应用程序必须遵守某些易于理解的结构规则。例如，在关闭某个文档之前显然必须先打开它。由于 PDFlib API 几乎是完全仿效文档 / 页范例，所以用自然方式生成文档通常可得到标准格式的 PDFlib 客户端程序。

PDFlib 通过严格的范围界定系统强制推行正确的函数调用顺序。范围定义可以在表 1.4 中找到。图 1.1 说明了范围的嵌套。函数说明指定每个函数的允许范围。在允许范围以外调用函数会解发 PDFlib 异常。您可以使用 *scope* 参数查询当前范围。

表 1.4 函数范围定义

范围名称	定义
path	由 PDF_moveto()、PDF_circle()、PDF_arc()、PDF_arcn() 或 PDF_rect() 中的一个函数开始；由第 84 页上的第 5.6 节 “路径绘制和剪切” 中的任意函数终止
page	在 PDF_begin_page() 和 PDF_end_page() 之间，但在 path 范围之外
template	在 PDF_begin_template_ext() 和 PDF_end_template() 之间，但在 path 范围之外
pattern	在 PDF_begin_pattern() 和 PDF_end_pattern() 之间，但在 path 范围之外
font	在 PDF_begin_font() 和 PDF_end_font() 之间，但在 glyph 范围之外
glyph	在 PDF_begin_glyph() 和 PDF_end_glyph() 之间，但在 path 范围之外
document	在 PDF_begin_document() 和 PDF_end_document() 之间，但在 page、template、pattern 和 font 范围之外
object	在面向对象的语言绑定中：pdfobj 对象的生存期，但在 document 范围之外；在其他绑定中，在 PDF_new() 和 PDF_delete() 之间，但在 document 范围之外；
null	在 object 范围之外
任意	由于 null 范围内根本不存在 PDFlib 对象，因此当函数说明提及 » 任何 « 范围时，实际上是指除 null 以外的任何范围

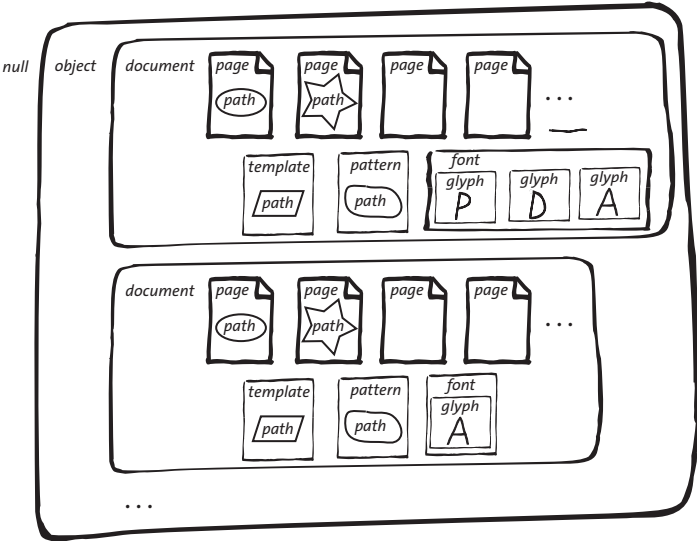


图 1.1 范围的嵌套



2 常规函数

2.1 参数处理

PDFlib 的操作可以被各种全局参数控制。有些字符串参数和数值可用于控制 PDFlib 的操作和 PDF 输出的外观。有四个函数用来设置和返回数字及字符串参数。在每节开头介绍了相关参数关键字名称和值；在附录 B 的“所有参数的列表”中是所有受支持参数的小结。

这些参数将在 PDFlib 对象的整个生存期内保留其设置，或者直到设置被客户端明确地更改。不过，在每页的开头将明确重设某些参数（在相应的说明中将提及这一点）。

C++ Java	<code>double get_value(String key, double modifier)</code>
Perl PHP	<code>float PDF_get_value(resource p, string key, float modifier)</code>
C	<code>double PDF_get_value(PDF *p, const char *key, double modifier)</code>

获取某些数字型 PDFlib 参数的值。

关键字 要查询的参数名。

modifier 应用到参数的可选修饰符。各参数表中将说明是否要求修饰符，以及与之对应的内容。如果不使用，修饰符必须为 0。许多参数要求作为修饰符传递句柄。

返回值 参数的数值。

范围 取决于 key。

C++ Java	<code>void set_value(String key, double value)</code>
Perl PHP	<code>PDF_set_value(resource p, string key, float value)</code>
C	<code>void PDF_set_value(PDF *p, const char *key, double value)</code>

设置某些数字型 PDFlib 参数的值。

关键字 要设置的参数名。

值 要设置的新参数值。

范围 取决于 key。

C++ Java	<code>String get_parameter(String key, double modifier)</code>
Perl PHP	<code>string PDF_get_parameter(resource p, string key, float modifier)</code>
C	<code>const char * PDF_get_parameter(PDF *p, const char *key, double modifier)</code>

获取某些字符串型 PDFlib 参数的内容。

关键字 要查询的参数名。

modifier 应用到参数的可选修饰符。各参数表中将说明是否要求修饰符，以及与之对应的内容。如果不使用，修饰符必须为 0。

返回值 作为超文本字符串的参数字符串值。可以在所处文档范围结束之前使用返回的字符串。如果没有信息，将返回空字符串。

范围 取决于 *key*。

绑定 C 和 C++: C 和 C++ 客户端不能释放返回的字符串。PDFlib 将在内部管理所有字符串资源。

C++ Java	<code>void set_parameter(String key, String value)</code>
Perl PHP	<code>PDF_set_parameter(resource p, string key, string value)</code>
C	<code>void PDF_set_parameter(PDF *p, const char *key, const char *value)</code>

设置某些字符串型 PDFlib 参数。

关键字 要设置的参数名。

值 （名称字符串）要设置的新参数值。

范围 取决于 *key*。

2.2 设置

表 2.1 和表 2.2 列出了用于 PDFlib 设置的相关参数和值关键字名称（参见第 11 页上的第 2.1 节“参数处理”）。

表 2.1 `PDF_get/set_parameter()` 中与设置有关的关键字

关键字	说明
任意资源类别名称	任意资源类别中的项。 <i>PDF_get_parameter()</i> : 修饰符包含项的索引（从 1 开始）。如果没有任何项，将返回空字符串。有关类别名称的列表，请参见 <i>PDFlib</i> 教程。范围：任意
<i>asciifile</i>	（只支持 <i>iSeries</i> 和 <i>zSeries</i> ）。要求采用 <i>ASCII</i> 编码的文本文件（ <i>PFA</i> 、 <i>AFM</i> 、 <i>UPR</i> 、编码）。默认值： <i>iSeries</i> 上为 <i>true</i> ， <i>zSeries</i> 上为 <i>false</i> 。范围：任意
<i>license</i> ¹	设置 <i>PDFlib</i> 、 <i>PDFlib+PDI</i> 或 <i>PPS</i> 的许可密钥。可在第一次调用 <i>PDF_begin_document()</i> 之前设置密钥（甚至可以多次设置，以进行累积密钥）。范围：对象
<i>licensefile</i>	设置包含许可密钥的文件的文件名。该文件在第一次调用 <i>PDF_begin_document()</i> 之前被设置一次。范围：对象
<i>nodemo-stamp</i>	如果为 <i>true</i> ，找不到有效的许可证密钥时将引发异常；如果为 <i>false</i> ，将在所有页上创建演示印章。必须在第一次调用 <i>PDF_begin_document()</i> 之前设置此选项。默认值： <i>false</i> 。范围：对象
<i>resourcefile</i>	<i>PDFlib UPR</i> 资源文件的相对或绝对文件名。将立即载入资源文件。将保留现有资源；如果再次设置，他们的值将由新值覆盖。范围：任意
<i>scope</i> ¹	当前范围的名称（参见表 1.4）。范围：任意
<i>SearchPath</i>	（不支持 <i>MVS</i> ）包含要读取文件的目录的相对或绝对路径名。可设置多个 <i>SearchPath</i> ；各项将累积，并首先使用最近设置的一项。如果字符串为空，将删除 <i>SearchPath</i> 列表中的所有项。 <i>PDF_get_parameter()</i> : 修饰符包含项的索引（从 1 开始）。如果没有任何项，将返回空字符串。返回的字符串将采用 <i>UTF-8</i> 编码。范围：任意
<i>string</i> ¹	返回由修饰中提供的字符串索引标识的字符串。在下次调用任何 <i>API</i> 函数前，返回的字符串将一直有效。范围：任意
<i>version</i> ¹	完整的 <i>PDFlib</i> 版本字符串，格式为 <i><major>.<minor>.<revision></i> ，可能附带 <i>beta</i> 、 <i>rc</i> 等附加限定符后缀。范围：任意、空 ²

1. 仅用于 *PDF_get_parameter()*
2. 可用 *PDF ** 的 *NULL* 或 *o* 自变量调用

表 2.2 `PDF_get/set_value()` 的与设置相关的关键字

关键字	说明
<i>compress</i>	压缩级别，从 <i>o</i> 代表无压缩， <i>1</i> 代表最快速度等至 <i>9</i> 代表最佳压缩。该参数不会影响以通过 (<i>passthrough</i>) 模式处理的图像数据。默认值： <i>6</i> 。范围：页面、文档
<i>major minor revision</i> ¹	分别为 <i>PDFlib</i> 的主要、次要或修订版本号。范围：任意、空 ²

1. 仅用于 *PDF_get_value()*
2. 可用 *PDF ** 的 *NULL* 或 *o* 自变量调用

Perl PHP *resource PDF_new()*
C *PDF *PDF_new(void)*

创建新的 PDFlib 对象。

详细说明 此函数使用 PDFlib 的内部缺省错误处理和内存分配例程创建新的 PDFlib 对象。

返回值 指向 PDFlib 对象的句柄，可用于后续的 PDFlib 调用。如果此函数由于内存不可用而导致失败，将返回 NULL（C 语言）或引发异常。

范围 空；此函数是对象范围的起点，必须与对应的 *PDF_delete()* 调用成对使用。

绑定 根据不同的语言绑定，用于不透明 PDFlib 对象句柄的数据类型会有所差异。由于 PDFlib 客户端所要做的就是作为第一个自变量将 PDF 句柄传递给所有函数，所以这实际上不会对客户端有任何影响。

C：为了在运行时动态地载入 PDFlib DLL，应用 *PDF_new_dl()*。*PDF_new_dl()* 将返回一个指向 *PDFlib_api* 结构的指针；该结构包含指向所有 PDFlib API 函数的指针。如果无法载入 DLL，或者检测到不匹配的主要 / 次要版本号，将返回 NULL。

C++、Java、PHP 5：由于此函数隐藏在 PDFlib 构造函数内，所以无法使用。

C *PDF *PDF_new2(void (*errorhandler)(PDF *p, int errortype, const char *msg),
void* (*allocproc)(PDF *p, size_t size, const char *caller),
void* (*reallocproc)(PDF *p, void *mem, size_t size, const char *caller),
void (*freeproc)(PDF *p, void *mem),
void *opaque)*

使用客户端提供的错误处理和内存分配例程创建新的 PDFlib 对象。

errorhandler 指向用户提供的错误处理函数的指针。*PDF_TRY/PDF_CATCH* 块中将忽略错误句柄。

allocproc 指向用户提供的内存分配函数的指针。

reallocproc 指向用户提供的内存重新分配函数的指针。

freeproc 释放用户提供内存的函数指针。

opaque 指向某些此后可用 *PDF_get_opaque()* 返回的用户数据。

返回值 指向 PDFlib 对象的句柄，可用于后续的 PDFlib 调用。如果此函数由于内存不可用而导致失败，将返回 NULL（C 语言）或引发异常。

详细说明 此函数使用客户端提供的错误处理和内存分配例程创建新的 PDFlib 对象。与 *PDF_new()* 不同，调用方可以选择提供自己的错误处理和内存分配过程。错误句柄的函数指针或内存过程可以为 NULL，或两者同时为 NULL。在这些情况下，PDFlib 将使用缺省例程。必须同时提供所有三个内存例程，或一个都不提供。

范围 空；此函数是对象范围的起点，必须与对应的 *PDF_delete()* 调用成对使用。调用此函数之后，不能调用其他具有相同 PDFlib 对象的 PDFlib 函数。

绑定 C++：可通过 PDF 构造函数间接使用此函数。由于将使用 NULL 作为缺省时的缺省值，所以并非所有的函数自变量都必须提供。所有提供的函数都必须为 »C« 函数，而非 C++。

Perl	PHP	<code>PDF_delete(resource p)</code>
C		<code>void PDF_delete(PDF *p)</code>

删除 PDFlib 对象并释放所有内部资源。

详细说明 此函数可删除 PDF 对象并释放所有文档相关的 PDFlib 内部资源。尽管生成单个文档不一定要求删除 PDF 对象，但强烈建议所有服务器应用程序在生成 PDF 后删除 PDF 对象。对于给定的 PDF 对象，此函数只能调用一次。在发生异常时，也应调用 `PDF_delete()`，以便执行清理操作。`PDF_delete()` 本身可保证不引发任何异常。如果将生成多个 PDF 文档，不必在建成每个文档后调用 `PDF_delete()`，只需在整个 PDF 文档序列完成后调用即可。

范围 任意；此函数是空范围的起点，即不允许调用其他 API 函数。

绑定 C：如果已经用 `PDF_new_dl()`，在运行时动态载入了 PDFlib DLL，应使用 `PDF_delete_dl()` 删除 PDFlib 对象。

C++：可通过 `PDF` 解构函数间接使用此函数。

Java：此函数由包装代码自动调用。但是，可以通过客户端代码显式调用，以克服 Java 终结器系统中的缺点。

PHP：PDFlib 对象超出范围时，面向对象的 PHP 5 接口将自动调用此函数。

2.3 文档函数

C++ Java	<code>int begin_document(String filename, String optlist)</code>
Perl PHP	<code>int PDF_begin_document(resource p, string filename, string optlist)</code>
C	<code>int PDF_begin_document(PDF *p, const char *filename, int len, const char *optlist)</code>
C++	<code>void begin_document_callback(size_t (*writeproc) (PDF *p, void *data, size_t size), string optlist)</code>
C	<code>void PDF_begin_document_callback(PDF *p, size_t (*writeproc) (PDF *p, void *data, size_t size), const char *optlist)</code>

根据各种选项创建新的 PDF 文档。

filename （名称字符串，但只在 Windows 上支持 Unicode 文件名）将生成 PDF 输出文件的绝对或相对文件名。如果 **filename** 为空，则将在内存中生成 PDF 文档，而不创建文件，因而只能从客户端通过 `PDF_get_buffer()` 函数来获取生成的 PDF 数据。特殊文件名 `»-«` 可用在 `stdout` 通道上生成 PDF。在 Windows 上，可以使用 UNC 路径或映射网络驱动器。

len （仅用于 C 语言绑定）UTF-16 字符串的 **filename** 长度（以字节为单位）。如果 **len=0**，则必须提供以 `null` 终结的字符串。

writeproc （仅用于 C 和 C++）C 回调函数，可由 PDFlib 调用，以提交生成的（部分）PDF 数据。

optlist 选项列表，根据表 2.3 指定文档选项。`PDF_end_document()` 中指定的选项优先于 `PDF_begin_document()` 中指定的相同选项。可以使用下列选项：
附件、*autoxmp*、*compatibility*、*destination*、*errorpolicy*、*flush*、*groups*、*hypertext-encoding*、*inmemory*、*labels*、*lang*、*linearize*、*master-password*、*metadata*、*moddate*、*openmode*、*optimize*、*pagelayout*、*pdfa*、*pdfx*、*permissions*、*recordsize*、*search*、*tagged*、*tempdirname*、*temp-filenames*、*uri*、*user-password*、*viewer-preferences*。

返回值	-1 （在 PHP 中：0）无错误，否则为 1。如果 filename 为空，此函数始终都能成功运行，从返回 -1 （在 PHP 中：0）错误值。
详细说明	此函数使用提供的 filename 创建新 PDF 文件。PDFlib 将尝试使用给定文件名打开文件，并在 PDF 文档完成时关闭文件。 <code>PDF_begin_document_callback()</code> 会在内存中打开新 PDF 文档，而不写入磁盘文件。作为提供的回调函数， writeproc 必须返回写入的字节数。如果返回值不匹配 PDFlib 提供的 size 自变量，将引发异常。 writeproc 的调用频率可由 <i>flush</i> 选项配置。
范围	对象；如果文件可以成功打开，此函数将为文档范围的起点；必须始终与对应的 <code>PDF_end_document()</code> 调用成对使用。
绑定	C、C++、Java、JavaScript：确保正确地转义反斜线路径分隔符。例如，以下路径指示网络驱动器上的一个文件： <code>\\\\malik\\rp\\foo.pdf</code> 。 <code>PDF_begin_document_callback()</code> 只能用于 C 和 C++。提供的 writeproc 必须为 C 语言函数，而非 C++。

C++ Java

void end_document(String optlist)

Perl PHP

PDF_end_document(resource p, string optlist)

C

void PDF_end_document(PDF *p, const char *optlist)

关闭生成的 PDF 文档并应用各种选项。

optlist 选项列表，根据表 2.3 指定文档选项。PDF_end_document() 中指定的选项优先于 PDF_begin_document() 中指定的相同选项。可以使用下列选项：

action、attachment-password 附件、autoxmp、destination、destname、hypertext-encoding、labels、metadata、moddate、openmode、pagelayout、search、uri、viewer-preferences。

详细说明 此函数完成 PDF 文档生成，释放所有文档相关的资源；如果已经用 PDF_begin_document() 打开了文档，此函数还可关闭输出文件。无论使用何种方法打开 PDF 文档，客户端完成页面生成操作时都必须调用此函数。

在内存中生成文档时（而不在文件中），调用此函数后仍将保留文档缓冲区（因此，可使用 PDF_get_buffer() 提取数据），并将在下次调用 PDF_begin_document() 或者 PDFlib 对象在 PDF_delete() 中的超出范围时释放文档缓冲区。

范围 文档；此函数终结文档范围，应始终与 PDF_begin_document() 或 PDF_begin_document_callback() 调用成对使用。

表 2.3 PDF_begin_document() 和 PDF_end_document() 的文档选项

选项	描述
action ¹	（动作列表；PDF/A 不支持）下列一个或多个事件的文档动作列表。默认值：空列表。 open 打开文档时执行的动作。由于在 Acrobat 中的执行顺序，文档级 JavaScript 不能用于 open 动作。 didprint/didsave/willclose/willprint/willsave (PDF 1.4) 将在打印 / 保存文档之后及关闭 / 打印 / 保存文档之前执行的 JavaScript 动作。
attachment-password ²	（字符串；PDF 1.6；如果设置了 userpassword 或 masterpassword 则会忽略此选项）将使用所提供的字符串作为密码对文件附件加密。文档的其余部分不加密。
附件	（选项列表的列表）指定文档级的文件附件（与限定于页面上特定位置的附件注释相对）。在 PDF_begin_document() 和 PDF_end_document() 中均可提供文件附件。支持的选项： filename （名称字符串；是必需的）文件名。支持 UTF-16 文件名。 description （超文本字符串；PDF 1.6）与文件关联的描述性文本。
autoxmp	（布尔型；在 PDF/A 模式下将强制为 true）如果为 true，PDFlib 将从文档信息域创建 XMP 文档元数据（参见第 144 页上的第 12.2 节“XMP 元数据”）。默认值：false
compatibility ²	（关键字）对于 Acrobat 4、5、6、7 或 8，将文档的 PDF 版本设置为以下字符串之一：1.3、1.4、1.5、1.6 或 1.7。如果使用 pdfx 或 pdfa 选项，将忽略此选项。默认值：1.6
destination	（选项列表；如果已指定打开动作，将忽略此选项）选项列表，根据表 10.3 指定文档打开动作。
destname ¹	（超文本字符串；如果已指定 destination 选项，将忽略此选项）目标名称，已使用 PDF_add_nameddest() 定义并将用作文档打开动作。
errorpolicy ²	（关键字）控制出现错误时的行为（参见第 28 页上的第 2.6 节“异常处理”）。
flush ²	（关键字；仅用于 PDF_begin_document_callback()）设置刷新策略。默认值：page。 none 只在文档结尾刷新一次 page 在每页的结尾刷新一次 content 在所有字体、图像、文件附件和页面后进行刷新 heavy 始终在内部 64 KB 文档缓冲区已满时刷新
groups ²	（字符串列表）定义文档中使用的页面组的名称和顺序。页面组使页面保持在一起（例如，对附加页标签很有用）；可以将页面指定到文档中定义的一个页面组，并在各自的组内引用。如果为文档定义了页面组，必须将所有页指定到一个页面组。
hypertext-encoding	（关键字）指定 destname 选项的编码方式（参见 PDFlib 教程）。空字符串相当于 unicode。默认值：全局 hypertextencoding 参数的值。

表 2.3 PDF_begin_document() 和 PDF_end_document() 的文档选项（续）

选项	描述
inmemory²	（布尔型；不能用于 <i>PDF_begin_document_callback()</i> ）如果为 true 且 linearize 或 optimize 选项也为 true ， <i>PDFlib</i> 不会为线性化处理创建任何临时文件，文件的处理将在内存中进行。这可能会大大提高某些系统上的性能（尤其是 <i>MVS</i> ），但要求两倍于文档大小的内存。如果为 false ，将为线性化和优化处理创建临时文件。默认值： false
labels	（选项列表的列表）包含根据表 2.4 提供的一个或多个选项列表的列表，指定符号页面名称。页面名称将在 <i>Acrobat</i> 的状态行中显示为页面标签（而非页码）。在一个文档中， style/prefix/start 值的组合必须是唯一的。默认值： none
lang²	（字符串；如果 tagged=true ，则要求此选项）将文档的自然语言设为由两个字符组成的 <i>ISO 639</i> 语言代码（例如： <i>DE</i> 、 <i>EN</i> 、 <i>FR</i> 、 <i>JA</i> ），后面可以跟一个连字符和由两个字符组成的 <i>ISO 3166</i> 国家 / 地区代码（例如： <i>EN-US</i> 、 <i>EN-GB</i> 、 <i>ES-MX</i> ）。不区分大小写。 在初始化时，需为文档的语言规范统一赋值，随后可在结构树的各个级别上用新值覆盖。
linearize²	（布尔型；不能用于 <i>PDF_begin_document_callback()</i> ）如果为 true ，将对输出文档进行线性化处理。在 <i>MVS</i> 系统上，此选项不能用于将生成的 <i>PDF</i> 输出到内存（即，空 <i>filename</i> ）。默认值： false
master-password²	（字符串；不能用于 <i>PDF/A</i> 和 <i>PDF/X</i> ）文档的许可口令。如果此选项为空，将不应用许可口令。默认值： empty
metadata	（选项列表； <i>PDF 1.4</i> ）提供文档的元数据（参见第 144 页上的第 12.2 节“ <i>XMP</i> 元数据”）。
moddate	（布尔型）如果为 true ，将创建 <i>ModDate</i> （修改日期）文档信息关键字，以符合某些 <i>preflight</i> 工具的要求。默认值： false
openmode	（关键字）设置文档打开时的外观。默认值：如果文档包含任何书签，则为 bookmarks ；否则为 none 。 none 打开时不显示任何其他面板。 bookmarks 打开时显示书签面板。 thumbnails 打开时显示缩览图面板。 fullscreen 以全屏模式打开（不适用于浏览器）。 layers (<i>PDF 1.5</i>) 打开时显示图层面板。 attachments (<i>PDF 1.6</i>) 打开时显示附件面板。
optimize²	（布尔型）如果为 true ，将在生成输出文档后单独对输出文档进行优化处理。优化处理将通过消除多余的重复对象来减小文件大小。在 <i>MVS</i> 系统上，此选项不能用于将生成的 <i>PDF</i> 输出到内存（即，空 <i>filename</i> ）。默认值： false
pagelayout	（关键字）打开文档时使用的页面版面。默认值：默认值。 default <i>Acrobat</i> 浏览器的缺省设置。 singlepage 一次显示一页。 onecolumn 在一列中连续显示多页。 twocolumnleft 在两列中显示多页，奇数页在左侧。 twocolumnright 在两列中显示多页，奇数页在右侧。 twopageleft (<i>PDF 1.5</i>) 一次显示两页，奇数页在左侧。 twopageright (<i>PDF 1.5</i>) 一次显示两页，奇数页在右侧。
pdfa²	（关键字）设置 <i>PDF/A</i> 规范等级，例如 <i>PDF/A-1a:2005</i> 、 <i>PDF/A-1b:2005</i> 或 none 。如果值为“ <i>PDF/A-1a:2005</i> ”，将自动启用标签 <i>PDF</i> 模式。默认值： none
pdfx²	（关键字）设置 <i>PDF/X</i> 规范等级，例如 <i>PDF/X-1a:2001</i> 、 <i>PDF/X-1a:2003</i> 、 <i>PDF/X-2:2003</i> 、 <i>PDF/X-3:2002</i> 、 <i>PDF/X-3:2003</i> 或 none 。默认值： none

表 2.3 PDF_begin_document() 和 PDF_end_document() 的文档选项（续）

选项	描述
permissions²	（关键字列表；不能用于 PDF/A 和 PDF/X）输出文档的访问权限列表。它包含任意数目的下列关键字（默认值：空）。 noprint Acrobat 将阻止打印文件。 nomodify Acrobat 将阻止用户进行除 noannots、noforms 和 noassemble 所允许之外的任何其他更改。 nocopy Acrobat 将阻止复制或提取文本或图形；辅助工具接口将由 noaccessible 控制。 noannots Acrobat 将阻止添加或更改注释、填充表单域以及添加或更改表单域。 noforms (PDF 1.4) Acrobat 将阻止填充表单域。 noaccessible (PDF 1.4) Acrobat 将阻止提取文本或图形，以限制辅助工具（如读屏器程序）。 noassemble (PDF 1.4) Acrobat 将阻止插入、删除或旋转页面和创建书签和缩览图。 nohiresprint (PDF 1.4) 如果没有设置 noprint，则只能使用“打印为图像”功能；这将打印以低分辨率呈现的页面。 plainmetadata (PDF 1.5) 保持 XMP 文档元数据不加密；即使加密文档，也是如此。
recordsize²	（整型；仅用于 MVS）输出文件（以及为 linearize 和 optimize 选项所创建的必要的临时文件）的记录大小。默认值：0（无阻断输出）
search	（选项列表）指示 Acrobat 在打开文档时附加搜索索引。支持下列子选项： filename （名称字符串；是必需的）包含搜索索引的文件名。索引的文件名可能是相对文档的，不过用户应负责提供正确的索引文件名。 indextype （名称字符串）索引的类型；对于 Acrobat，必须为 PDX。默认值：PDX
tagged²	（布尔型；PDF 1.4）如果为 true，则生成加标签的 PDF 输出。必须由客户端在加标签的 PDF 模式下，提供正确的结构信息（参见第 146 页上的第 12.3 节“标签 PDF”）。如果 pdfa 选项的值为“PDF/A-1a:2005”，此选项将自动强制为 true。默认值：false
tempdirname²	（字符串；不能用于 PDF_begin_document_callback()）将在其中创建 linearize 和 optimize 选项所需临时文件的目录的名称。如果为空，PDFlib 将在当前目录内生成临时文件。如果提供了 tempfilenames 选项，则将忽略此选项。默认值：empty
temp-filenames²	（两个字符串的列表；仅在 MVS 上用于 PDF_begin_document()）linearize 和 optimize 选项所需的两个临时文件的完整文件名。如果为空，PDFlib 将生成唯一的临时文件名。用户负责在 PDF_end_document() 之后删除临时文件。如果提供了此选项，filename 参数不能为空。默认值：empty
uri	（字符串）设置文档的基本 URL。将具有指向其他文档的相对 Web 链接的文档移至不同位置时，此选项非常有用。设置到 »old« 位置的基本 URL 可确保相对链接仍然可以使用。默认值：none
user-password²	（字符串；不能用于 PDF/A 和 PDF/X）文档的打开口令。如果为空，则不应用文档打开口令。默认值：empty
viewer-preferences	（选项列表）选项列表，根据表 2.5 指定各种查看器首选项。默认值：empty

1. 仅用于 PDF_end_document()
2. 仅用于 PDF_begin_document() 和 PDF_begin_document_callback()

表 2.4 PDF_begin/end_document() 中 labels 选项和 PDF_begin/end_page_ext() 中 label 选项的子选项

选项	描述
group	（字符串；仅用于 <i>PDF_begin_document()</i> ；如果文档使用页面组，则要求此选项；否则，不允许使用此选项）标签将应用于指定组以及应用新标签之前的所有后续组中的所有页面。必须已经用 <i>PDF_begin_document()</i> 中的 <i>groups</i> 选项定义了组名。
hypertextencoding	（关键字）指定 <i>prefix</i> 选项的编码方式。空字符串相当于 <i>unicode</i> 。默认值：全局 <i>hypertextencoding</i> 参数的值。
pagenumber	（整型；仅用于 <i>PDF_end_document()</i> ；如果文档不使用页面组，则要求此选项，否则不允许使用此选项）标签将应用于指定页面以及应用新标签之前的所有页面。
前缀	（超文本字符串）处于该范围内的所有标签的标签前缀。默认值： <i>none</i>
start	（整型 ≥ 1 ）该范围内第一个标签的数值。该范围内的后续页面将从该值开始，依次编号。默认值： <i>1</i>
style	（关键字）要使用的编号样式。默认值： <i>none</i> 。 <i>none</i> 无页码；标签将只包含前缀。 <i>D</i> 十进制阿拉伯数字（ <i>1</i> 、 <i>2</i> 、 <i>3</i> 、...） <i>R</i> 大写罗马数字（ <i>I</i> 、 <i>II</i> 、 <i>III</i> 、...） <i>r</i> 小写罗马数字（ <i>i</i> 、 <i>ii</i> 、 <i>iii</i> 、...） <i>A</i> 大写字母（ <i>A</i> 、 <i>B</i> 、 <i>C</i> 、...、 <i>AA</i> 、 <i>BB</i> 、 <i>CC</i> 、...） <i>a</i> 小写字母（ <i>a</i> 、 <i>b</i> 、 <i>c</i> 、...、 <i>aa</i> 、 <i>bb</i> 、 <i>cc</i> 、...）

表 2.5 PDF_begin_document() 和 PDF_end_document() 中 viewerpreferences 选项的子选项

选项	描述
centerwindow	（布尔型）指定是否将文档窗口置于屏幕中央。默认值： <i>false</i>
direction	（关键字）阅读文档的顺序，将影响双页面视图中的滚动顺序。（缺省为 <i>l2r</i> ）： <i>l2r</i> 从左至右 <i>r2l</i> 从右至左（包括直排书写系统）
displaydoctitle	（布尔型）指定是否在 <i>Acrobat</i> 的标题栏 (<i>true</i>) 或是文件名 (<i>false</i>) 中显示 <i>Title</i> 文档信息域。默认值： <i>false</i>
fitwindow	（布尔型）指定是否将文档窗口的大小调整为第一页的大小。默认值： <i>false</i>
hidemenubar	（布尔型）指定是否隐藏 <i>Acrobat</i> 菜单栏。默认值： <i>false</i>
hidetoolbar	（布尔型）指定是否隐藏 <i>Acrobat</i> 工具栏。在浏览器中查看 <i>PDF</i> 时， <i>Acrobat 5</i> 会忽略此设置。默认值： <i>false</i>
hidewindowui	（布尔型）指定是否隐藏 <i>Acrobat</i> 的窗口控件。默认值： <i>false</i>
nonfullscreenpagemode	（关键字；仅用于 <i>openmode</i> 选项设为 <i>fullscreen</i> 的情况）指定如何在现有全屏模式下显示文档。默认值： <i>none</i> 。 <i>bookmarks</i> 显示页面和书签窗格 <i>thumbnails</i> 显示页面和缩览图窗格 <i>layers</i> 显示页面和图层窗格 <i>none</i> 只显示页面
printscaling	（关键字； <i>PDF 1.6</i> ）打开文档的打印对话框时将选择的页面缩放选项。默认值： <i>appdefault</i> 。 <i>none</i> 无页面缩放；这对于按页面内容的实际大小打印很有帮助。 <i>appdefault</i> 使用 <i>Acrobat</i> 中指定的当前打印缩放。
viewarea	（关键字）在屏幕上查看或打印文档时，代表要显示或剪贴的页面区域的页面边界框类型。 <i>Acrobat</i> 会忽略此设置，但可能对其他应用程序很有用。默认值： <i>crop</i> 。
viewclip	
printarea	<i>art</i> 使用作品框
printclip	<i>bleed</i> 使用出血框 <i>crop</i> 使用裁剪框 <i>media</i> 使用媒体框 <i>trim</i> 使用裁切框

C++	<code>const char *get_buffer(long *size)</code>
Java	<code>byte[] get_buffer()</code>
Perl PHP	<code>string PDF_get_buffer(resource p)</code>
C	<code>const char * PDF_get_buffer(PDF *p, long *size)</code>

获取 PDF 输出缓存区的内容。

size （仅用于 C 和 C++ 语言绑定）指向内存位置的 C 语言指针；将在其中存储返回数据的长度（以字节为单位）。

返回值 充满二进制 PDF 数据的缓存区，供客户端使用。根据表 1.1 返回二进制数据的语言特定数据类型。调用任何其他 PDFlib 函数之前，返回的缓存区必须供客户端使用。如果希望在调用其他 PDFlib 函数时使用，切记要复制数据（尤其是在调用 `PDF_create_pvf()` 创建包含该数据的 PVF 文件之前）。

详细说明 提取全部或部分包含生成 PDF 数据的缓存区。如果此函数在页面描述之间调用，它将返回迄今为止生成的 PDF 数据。如果在内存中生成 PDF，此函数至少须在 `PDF_end_document()` 之后调用，并将返回 PDF 文档的剩余部分。可以在此之前调用此函数，以提取部分文档数据。如果此函数只在 `PDF_end_document()` 之后调用一次，则返回的缓存区肯定会在连续缓存区内包含整个 PDF 文档。

由于 PDF 输出包含二进制字符，客户端软件必须准备接受包括空值在内的不可打印字符。

范围 对象、文档（换言之：在 `PDF_end_page_ext()` 之后和 `PDF_begin_page_ext()` 之前，或者在 `PDF_end_document()` 之后和 `PDF_delete()` 之前。此函数只能在已确保向 `PDF_begin_document()` 提供了空的 filename 的情况下，才只能使用。

如果 `PDF_begin_document()` 中的 `linearize` 选项已设为 `true`，则范围将限制在对象内；即：只能在 `PDF_end_document()` 之后调用此函数。

绑定 C 和 C++: `size` 参数只能用于 C 和 C++ 客户端。

其他绑定：将返回适当长度的对象，且必须省略 `size` 参数。

2.4 页面函数

表 2.6 和表 2.7 列出了本节的相关参数及相应的值关键字名称（参见第 11 页上的第 2.1 节“参数处理”）。

表 2.6 `PDF_get/set_parameter()` 中与页面相关的关键字

关键字	说明
<code>topdown</code>	如果为 <code>true</code> ，将假定页面、图案或模板开头坐标系统的原点位于页面左上角， <code>y</code> 坐标值向下增加；否则，将使用缺省坐标系统。详情见 <i>PDFlib</i> 教程。范围：文档。默认值： <code>false</code>

表 2.7 `PDF_get/set_value()` 中与页面相关的关键字

关键字	说明
<code>pagewidth</code> <code>pageheight</code>	获取当前页面的页面大小（ <i>MediaBox</i> 的尺寸）。范围：页面、路径

C++ Java	<code>void begin_page_ext(double width, double height, String optlist)</code>
Perl PHP	<code>PDF_begin_page_ext(resource p, float width, float height, string optlist)</code>
C	<code>void PDF_begin_page_ext(PDF *p, double width, double height, const char *optlist)</code>

向文档添加一个新页面并指定各种选项。

width, height `width` 和 `height` 参数以点数为单位（或以用户单位，如果已指定 `userunit` 选项）指定新页面的尺寸。可用具有相同名称的选项覆盖这两个参数（这种情况下，可为参数使用虚值 `0`）。表 2.8 列出了常用的页面格式。有关更多详细信息（`width` 和 `height` 选项），另请参见表 2.9。

表 2.8 以点为单位的常见页面大小尺寸¹

格式	<code>width</code>	<code>height</code>	格式	<code>width</code>	<code>height</code>	格式	<code>width</code>	<code>height</code>
<code>a0</code>	2380	3368	<code>a4</code>	595	842	<code>letter</code>	612	792
<code>a1</code>	1684	2380	<code>a5</code>	421	595	<code>legal</code>	612	1008
<code>a2</code>	1190	1684	<code>a6</code>	297	421	<code>ledger</code>	1224	792
<code>a3</code>	842	1190	<code>a5</code>	501	709	<code>11x17</code>	792	1224

¹ 有关 *ISO*、*Japanese* 和 *U.S.* 标准格式的更多信息可以在下面的 URL 上找到：
home.inter.net/eds/paper/papersize.html、www.cl.cam.ac.uk/~mgk25/iso-paper.html

optlist 根据表 2.9 提供的选项列表。这些选项的优先级低于 `PDF_end_page_ext()` 中指定的相同选项。可以使用下列选项：

`action`、`artbox`、`bleedbox`、`cropbox`、`defaultcmyk`、`defaultgray`、`defaultrgb`、`duration`、`group`、`height`、`label`、`mediabox`、`metadata`、`pagenumber`、`rotate`、`separation-info`、`taborder`、`topdown`、`transition`、`trimbox`、`userunit`、`width`

详细说明 此函数将新页面的所有文本、图形和颜色状态参数重设为他们的缺省值。

范围 文档；此参数是页面范围的开头，必须与对应的 `PDF_end_page_ext()` 调用成对使用。

C++ Java

void end_page_ext(String optlist)

Perl PHP

PDF_end_page_ext(resource p, string optlist)

C

void PDF_end_page_ext(PDF *p, const char *optlist)

完成页面并应用各种选项。

optlist 根据表 2.9 提供的选项列表。`PDF_end_page_ext()` 中指定的选项优先于 `PDF_begin_page_ext()` 中指定的相同选项。可以使用下列选项：
`action`、`artbox`、`bleedbox`、`cropbox`、`defaultcmyk`、`defaultgray`、`defaultrgb`、`duration`、`group`、`height`、`label`、`mediabox`、`metadata`、`rotate`、`taborder`、`transition`、`trimbox`、`userunit`、`width`

范围 页面；此函数终结页面范围，必须与对应的 `PDF_begin_page_ext()` 调用成对使用。

表 2.9 `PDF_begin_page_ext()` 和 `PDF_end_page_ext()` 的选项

选项	描述
action	（动作列表）下列一个或多个事件的页面动作列表（默认值：空列表）： open 打开页面时执行的动作。 close 关闭页面时执行的动作。
artbox bleedbox cropbox	（矩形）更改当前页面的页面框参数。坐标在缺省坐标系中指定。默认值：无方框项
defaultgray defaultrgb defaultcmyk	（ICC 句柄）根据提供的配置文件句柄，设置页面的缺省灰色、RGB 或 CMYK 色彩空间。
duration	（浮点型）如果 <code>openmode=fullscreen</code> ，则以秒为单位，设置当前页面的显示时间（参见表 2.3）。默认值：1
group¹	（字符串；如果文档使用页面组，则要求此选项；否则，不允许使用此选项。）页面所属页面组的名称。此名称可使页面保持在一个页面组内，并可用 <code>PDF_resume_page()</code> 确定页面地址。必须已经用 <code>PDF_begin_document()</code> 中的 <code>groups</code> 选项定义了组名。
height	（浮点型或关键字；如果 <code>topdown</code> 选项或参数为 <code>true</code> ，则不允许此选项）新页面的尺寸，以点数为单位（或以用户单位，如果已指定 <code>userunit</code> 选项）。要生成横向页面，应使用 <code>width > height</code> 或 <code>rotate</code> 选项。 <code>PDFlib</code> 使用 <code>width</code> 和 <code>height</code> 选项构建页面的媒体框，但也可以使用 <code>mediabox</code> 选项显式设置媒体框。 <code>width</code> 和 <code>height</code> 选项将覆盖名称相同的参数。 可在下列符号页面大小名称后附加 <code>.width</code> 或 <code>.height</code> ，将其用作关键字（例如： <code>a4.width</code> 、 <code>a4.height</code> ）： <code>a0</code> 、 <code>a1</code> 、 <code>a2</code> 、 <code>a3</code> 、 <code>a4</code> 、 <code>a5</code> 、 <code>a6</code> 、 <code>b5</code> 、 <code>letter</code> 、 <code>legal</code> 、 <code>ledger</code> 、 <code>11x17</code>
label	（选项列表）根据表 2.4 提供的选项列表，指定符号页面名称。页面名称将在 <i>Acrobat</i> 的状态行中显示为页面标签（而非页码）。指定的编号方案将用于当前和后续页面，直至再次更改为止。在一个文档中， <code>style/prefix/start</code> 值的组合必须是唯一的。
mediabox	（矩形；如果 <code>topdown</code> 选项或参数为 <code>true</code> ，不允许使用 <code>mediabox</code> 选项）更改当前页面的媒体框。坐标在缺省坐标系中指定。缺省情况下，将只使用 <code>width</code> 和 <code>height</code> 参数创建媒体框 (<i>MediaBox</i>)。 <code>mediabox</code> 选项将覆盖 <code>width</code> 和 <code>height</code> 选项和参数。
metadata	（选项列表； <code>PDF 1.4</code> ）提供页面的元数据（参见第 144 页上的第 12.2 节“XMP 元数据”）。
pagenumber¹	（整型）如果用值 <code>n</code> 指定此选项，则将在 <code>group</code> 选项指定的页面组内现有第 <code>n</code> 页前插入页面（如果文档不使用页面组，则在文档中插入页面）。如果未指定此选项，将在页面组末尾插入页面。
rotate	（整型）页面旋转值。旋转将影响页面显示，但不会修改坐标系。其可选值为 <code>0</code> 、 <code>90</code> 、 <code>180</code> 、 <code>270</code> 。缺省值： <code>0</code>

表 2.9 PDF_begin_page_ext() 和 PDF_end_page_ext() 的选项（续）

选项	描述
separation-info¹	（选项列表）包含当前页面的详细分色信息的选项列表。 <i>Acrobat</i> 中将忽略此选项，但第三方软件可能会用它在预分色工作流程中确定并正确预览分色页： pages （整型；一组分色页的第一个页面要求此选项，但同一组的后续页面不允许使用此选项）属于同一组分色页的页面数量；分色页包含单个复合页的颜色数据。同一组内的所有页面必须在文件中依次显示。 spotname （字符串；除非提供了 <i>spotcolor</i> ，否则要求使用此选项）当前页面的色料名称。 spotcolor （专色句柄）描述当前页面色料的颜色句柄。
taborder	（关键字； <i>PDF 1.5</i> ）指定页面上表单域和注释的跳位顺序的关键字（默认值：none）： column 从顶到底逐列访问注释，其中各列按 <i>PDF_begin/end_document()</i> 的 <i>viewerpreferences</i> 选项的 <i>direction</i> 子选项指定的顺序排列。 none 不指定跳位顺序。 structure 按注释在结构树中的显示顺序访问注释。不指定不包括在结构树中的注释的顺序。 row 从顶行开始逐行访问注释，其中行内按 <i>PDF_begin/end_document()</i> 的 <i>viewerpreferences</i> 选项的 <i>direction</i> 子选项指定的方向。
topdown¹	（布尔型）如果为 true ，将假定页面开头坐标系统的原点位于页面左上角， <i>y</i> 坐标值向下增加；否则，将使用缺省坐标系统。默认值：false
transition	（关键字）如果 <i>openmode=fullscreen</i> ，设置当前页面的页面转换，以实现特殊效果；在 <i>Acrobat</i> 的全屏模式下，以演示文稿的形式显示 <i>PDF</i> 时，这个选项会很有用（参见表 2.3）。默认值：replace split 两行在屏幕上掠过，显示整个页面 blinds 多行在屏幕上掠过，显示整个页面 box 在框中显示页面 wipe 单行掠过屏幕，显示整个页面 dissolve 旧页面分解，逐渐显示页面 glitter 分解效果从屏幕的一个边缘移动至另一边缘 replace 简单地用新页面替代旧页面 fly (<i>PDF 1.5</i>) 新页面飞入旧页面。 push (<i>PDF 1.5</i>) 新页面将旧页面推出屏幕 cover (<i>PDF 1.5</i>) 新页面滑入屏幕，覆盖旧页面。 uncover (<i>PDF 1.5</i>) 旧页面滑出屏幕，显示新页面。 fade (<i>PDF 1.5</i>) 在旧页面中逐渐显示新页面。
trimbox	（矩形）指定当前页面的裁切框。坐标在缺省坐标系统中指定。默认值：无裁切框项。
userunit	（浮点型或关键字； <i>PDF 1.6</i> ）一个介于 1..75 000 之间的数字，指定用户单位的大小，以点或以下关键字之一为单位表示：mm、cm 或 m（缩放至相应的单位）。用户单位不会更改页内容；它们仅用于提示 <i>Acrobat</i> 在打印页面或使用测量工具时采用何种单位。默认值：1（即单位为一个点）
width	（浮点型或关键字；如果 <i>topdown</i> 选项或关键字为 true ，则不允许此选项）参见上述 <i>height</i> 选项。

1. 仅用于 *PDF_begin_page_ext()*

C++ Java void suspend_page(String optlist)

Perl PHP PDF_suspend_page(resource p, string optlist)

C void PDF_suspend_page(PDF *p, const char *optlist)

暂停当前页面，以便可以在稍后恢复。

optlist 供将来使用的选项列表。

详细说明 将在内部保存当前页面的完全状态（图形、颜色、文本等）。稍后可使用 *PDF_resume_page()* 恢复页面，以添加更多内容。必须先恢复暂停的页面，然后才能将其关闭。

范围 页面；此参数是文档范围的开头，必须与对应的 *PDF_resume_page()* 调用成对使用。不能在标签 *PDF* 模式下使用此函数。

C++ Java void resume_page(String optlist)

Perl PHP PDF_resume_page(resource p, string optlist)

C void PDF_resume_page(PDF *p, const char *optlist)

恢复页面，以向其添加更多内容。

optlist 根据表 2.10 提供的选项列表。可以使用下列选项：
group、*pagenumber*

详细说明 必须已经使用 *PDF_suspend_page()* 暂停页面。页面将再次打开；这样，便能够添加更多内容。关闭之前，必须先恢复所有已暂停的页面；即使未添加更多内容，也是如此。

范围 文档；此参数是页面范围的开头，必须与对应的 *PDF_suspend_page()* 调用成对使用。

表 2.10 *PDF_resume_page()* 的选项

选项	描述
group	（字符串；如果文档使用页面组，则要求此选项；否则，不允许使用此选项）恢复的页面所属页面组的名称。必须已经用 <i>PDF_begin_document()</i> 中的 groups 选项定义了组名。
pagenumber	（整型）如果提供了此选项，将恢复 group 选项中选择的页面组内、具有指定页码的页面（如果文档不使用页面组，则在文档中恢复）。如果缺少此选项，将恢复该组中的最后一页。

2.5 PDFlib 虚拟文件系统 (PVF)

C++ `void create_pvf(string filename, const void *data, size_t size, string optlist)`

Java `void create_pvf(String filename, byte[] data, String optlist)`

Perl PHP `PDF_create_pvf(resource p, string filename, string data, string optlist)`

C `void PDF_create_pvf(PDF *p,
const char *filename, int len, const void *data, size_t size, const char *optlist)`

用内存中提供的数据创建命名的只读虚拟文件。

filename （名称字符串）虚拟文件的文件名。这是个任意字符串，稍后可用来引用其他 PDFlib 调用中的虚拟文件。

len （仅用于 C 语言绑定）UTF-16 字符串的 **filename** 长度（以字节为单位）。如果 **len=0**，则必须提供以 **null** 终结的字符串。

data 对虚拟文件的数据的引用。在 C 和 C++ 语言中，这是个指向内存位置的指针。在 Java 中，这是个字节数组。在 Perl、Python 和 PHP 中，这是个字符串。

size （仅用于 C 和 C++）包含数据的内存块的长度，以字节为单位。

optlist 根据表 2.11 提供的选项列表。可以使用下列选项：**copy**

详细说明 可以为任何使用输入文件的 API 函数提供虚拟文件名（虚拟文件不能用于生成的 PDF 输出，可在 `PDF_begin_document()` 中使用空文件名，以实现这一目的）。其中一些函数可能会在虚拟文件上设置锁定，直至数据不再需要为止。虚拟文件将一直保存在内存中，直至使用 `PDF_delete_pvf()` 显式删除或在 `PDF_delete()` 中自动删除为止。

每个 PDFlib 对象都将维护自己的 PVF 文件组。虚拟文件不能在不同 PDFlib 对象之间共享，但可以通过虚拟文件，使用相同的 PDFlib 对象创建多个文档。处理不同 PDFlib 对象的多个线程无需同步 PVF 的使用。如果 **filename** 引用现有虚拟文件，将引发异常。此函数不检查一般磁盘文件是否正在使用 **filename**。

除非已经提供了 **copy** 选项，否则在相应地成功调用 `PDF_delete_pvf()` 之前，调用方不能修改或释放（删除）提供的数据。如不遵守此规则，很可能会导致崩溃。

范围 任意

表 2.11 `PDF_create_pvf()` 的选项

选项	描述
copy	（布尔型）PDFlib 会立即创建所提供数据的内部副本。在这种情况下，调用方可以在此调用后立即删除提供的数据。在 COM 、 .NET 和 Java 绑定中， copy 选项将自动设为 true （其他绑定的缺省设置： false ）。在其他语言绑定中，除非提供 copy 选项，否则不会复制数据。

C++ Java	<code>int delete_pvf(String filename)</code>
Perl PHP	<code>int PDF_delete_pvf(resource p, string filename)</code>
C	<code>int PDF_delete_pvf(PDF *p, const char *filename, int len)</code>

删除命名的虚拟文件，并释放它的数据结构（但不释放内容）。

filename （名称字符串）提供给 `PDF_create_pvf()` 的虚拟文件文件名。

len （仅用于 C 语言绑定）UTF-16 字符串的 **filename** 长度（以字节为单位）。如果 **len=0**，则必须提供以 `null` 终结的字符串。

- 返回值

-1（在 PHP 中：0）其条件是，对应的虚拟文件存在，但已被锁定；否则，为 1。
- 详细说明

如果文件没有锁定，PDFlib 将立即删除与 **filename** 关联的数据结构。如果 **filename** 未引用有效的虚拟文件，则此函数不执行任何操作，也不发出任何提示。成功调用此函数后，可以重复使用 **filename**。所有虚拟文件都将在 `PDF_delete()` 自动中删除。

具体语义取决于是否为对应的 `PDF_create_pvf()` 调用提供了 **copy** 选项：如果提供了 **copy** 选项，将同时释放文件的管理数据结构和实际的文件内容；否则，由于假定会由客户端完成此动作，所以不会释放内容。
- 范围

任意

2.6 异常处理

表 2.12 列出了本节的相关参数关键字名称（参见第 11 页上的第 2.1 节“参数处理”）。

表 2.12 *Exception-related keys for PDF_get/set_parameter()*

关键字	说明
errorpolicy	（关键字）控制出现错误时各种函数的行为。许多函数的 errorpolicy 选项都可能覆盖此设置，此选项用作同名选项的默认值。支持的关键字（默认值： legacy ；范围：任意）： legacy 函数的行为与 <i>PDFlib 6</i> 中的相同（受各种在 <i>PDFlib 7</i> 中已不常用的 *warning 参数和选项的控制）。通过此设置实现了与使用 <i>PDFlib 6</i> 或早期版本开发的应用程序的兼容性。 return 如果发生错误，函数将返回。可以返回错误代码的函数（例如 <i>PDF_load_image()</i> ）将返回 -1 （在 <i>PHP</i> 中： o ）。应用程序开发人员必须检查返回值是否为 -1 （在 <i>PHP</i> 中： o ）以便检测错误情况。如果出现错误，可以使用 <i>PDF_get_errmsg()</i> 查询问题的详细描述。将忽略所有 *warning 参数和选项。对于新应用程序以及现有应用程序的更新，建议使用此设置。 exception 如果发生错误，函数将引发异常。（部分） <i>PDF</i> 输出文档将不可用。
warning	不常用；警告将不再触发异常，而只能通过日志进行访问（参见第 30 页上的第 2.7 节“日志”）

C++ Java int get_errnum()
Perl PHP int PDF_get_errnum(resource p)
C int PDF_get_errnum(PDF *p)

获取上次引发的异常的编号或者函数调用失败的原因。

返回值 异常的编号，或者最近调用失败并返回错误代码的原因代码。

范围 在 *PDFlib* 引发的异常与 *PDFlib* 对象终止响应之间。另外，可以在某一函数返回错误代码 **-1**（在 *PHP* 中：**o**）之后，但在本节中所列函数之外的任何其他函数之前调用此函数。

绑定 在 C++、Java、和 *PHP 5* 中，还可以在 *PDFlibException* 对象中将此函数用作 *get_errnum()*。

C++ Java	<i>String</i> <i>get_errmsg()</i>
Perl PHP	<i>string</i> <i>PDF_get_errmsg(resource p)</i>
C	<i>const char *PDF_get_errmsg(PDF *p)</i>

获取上次引发的异常的文本或者函数调用失败的原因。

- 返回值** 包含上次引发的异常描述或者最近一次函数调用失败并返回错误代码的原因的文本。
- 范围** 在 PDFlib 引发的异常与 PDFlib 对象终止响应之间。另外，可以在某一函数返回错误代码 -1（在 PHP 中：o）之后，但在本节中所列函数之外的任何其他函数之前调用此函数。
- 绑定** 在 C++、Java、和 PHP 5 中，还可以在 *PDFlibException* 对象中将此函数用作 *get_errmsg()*。

C++ Java	<i>String</i> <i>get_apiname()</i>
Perl PHP	<i>string</i> <i>PDF_get_apiname(resource p)</i>
C	<i>const char *PDF_get_apiname(PDF *p)</i>

获取引发上次异常或失败的 API 函数名。

- 返回值** 引发异常的函数名，或最近一次函数调用失败并返回错误代码的函数名。
- 范围** 在 PDFlib 引发的异常与 PDFlib 对象终止响应之间。另外，可以在某一函数返回错误代码 -1（在 PHP 中：o）之后，但在本节中所列函数之外的任何其他函数之前调用此函数。
- 绑定** 在 C++、Java、和 PHP 5 中，还可以在 *PDFlibException* 对象中将此函数用作 *get_apiname()*。

C++	<i>void *get_opaque()</i>
C	<i>void *PDF_get_opaque(PDF *p)</i>

提取 PDFlib 中存储的 opaque 应用程序指针。

- 返回值** 在 PDFlib 中存储的、在 *PDF_new2()* 调用中提供的 opaque 应用程序指针。
- 详细说明** PDFlib 从不使用 opaque 指针，但会将其提供给客户端，而不做任何更改。可用于多线程应用程序，在 PDFlib 对象内存储专用的线程特定数据。它对处理线程特定的异常处理特别有用。
- 范围** 任意
- 绑定** 仅用于 C 和 C++ 语言绑定。

2.7 日志

日志文件的内容可能对调试很有用，PDFlib GmbH 技术支持人员可能也需要这些内容。
表 2.13 列出了日志功能的参数关键字名称。（参见第 11 页上的第 2.1 节“参数处理”）

表 2.13 PDF_set_parameter() 的与日志相关的关键字

关键字	说明
logging	选项列表，根据表 2.14 包含日志选项
logmsg	字符串，将被复制到日志文件

- 可以下列方式提供日志选项：
- ▶ 作为 PDF_set_parameter() 的 logging 选项的选项列表，例如：
`p.set_parameter("logging", "filename=debug.log remove")`
 - ▶ 以名为 PDFLIBLOGGING 的环境变量的形式 这样做将启动日志输出，从对任一 API 函数的第一次调用开始。

表 2.14 logging 参数的选项

关键字	说明
(空列表)	启用日志输出
disable	(布尔型) 禁用日志输出
enable	(布尔型) 启用日志输出
filename	(字符串) 日志文件的名称 (stdout 和 stderr 将被视为特殊名称)。输出将被附加到任何现有内容之后。默认值： <div><div>pdflogMVS 上</div><div>PDFlib.logMac 和 iSeries 上</div><div>\\PDFlib.logWindows 上</div><div>/tmp/PDFlib.log所有其他系统上</div></div> 还可以在环境变量 PDFLIBLOGFILE 中提供日志文件名。
flush	(布尔型) 如果为 true，每次输出后将关闭日志文件，然后重新打开以用于下次输出，以便确保将实际刷新输出。这在跟踪程序崩溃而日志文件会被截短时可能很有用，但会显著延缓处理速度。如果为 false，日志文件将只打开一次。默认值: true
remove	(布尔型) 如果为 true，在写入新输出之前将删除现有日志文件。默认值: false
stringlimit	(整型) 每行的字符数限制，或者为 0，表示无限制。默认值: 0
classes	(选项列表) 包含类型整数的选项的列表，其中每个选项描述一个日志类型，对应的值描述细分级别。级别为 0 禁用日志类型，级别为正数将启用日志类型。级数越高，输出越详细。提供以下选项 (默认值: {api=1 warning=1})： <div><div>api记录所有 API 调用及其函数参数和结果。如果 api=2，将在所有 API 跟踪行前面创建一个时间戳。</div><div>filesearch记录与通过 SearchPath 或 PVF 查找文件有关的所有尝试。</div><div>resource记录所有通过 Windows 注册表、UPR 定义查找资源的尝试以及资源搜索的结果。</div><div>user用户指定的日志输出（由 logmsg 参数提供）。</div><div>warning记录所有 PDFlib 警告，即可以忽略或内部修复的错误条件。如果 warning=2，将记录未引发任何异常而是借助挂钩发出消息文本以便通过 PDF_get_errmsg() 检索的函数的消息，以及所有打开文件时失败的尝试。</div></div>

3 文本函数

3.1 字体处理

表 3.1 和表 3.2 列出了本节的相关参数及相应的值关键字名称（参见第 11 页上的第 2.1 节“参数处理”）。

表 3.1 `PDF_get/set_parameter()` 中与字体相关的关键字

关键字和说明
<i>Encoding</i> 、 <i>FontAFM</i> 、 <i>FontPFM</i> 、 <i>FontOutline</i> 、 <i>HostFont</i> 如 UPR 文件中相关类别所示的对应资源文件行。可由多个调用向内部列表添加新条目。 （另请参见表 2.1 中的 <i>resourcefile</i> ）。范围：任意
<i>fontwarning</i> 不常用，使用 <i>errorpolicy</i>
<i>ascenderfaked</i> 、 <i>capheightfaked</i> 、 <i>descenderfaked</i> 、 <i>fontencoding</i> 、 <i>fontname</i> 、 <i>fontstyle</i> 、 <i>xheightfaked</i> 不常用，使用 <i>PDF_info_font()</i>
<i>autocidfont</i> 、 <i>autosubsetting</i> 、 <i>unicodemap</i> 不常用，使用 <i>PDF_load_font()</i> 中的对应选项

表 3.2 `PDF_get/set_value()` 中与字体相关的关键字

关键字	说明
<i>fontmaxcode</i> 、 <i>capheight</i> 、 <i>ascender</i> 、 <i>descender</i> 、 <i>xheight</i> 、 <i>monospace</i> 不常用，使用 <i>PDF_info_font()</i>	
<i>subsetlimit</i> 、 <i>subsetminsize</i> 不常用，使用 <i>PDF_load_font()</i> 中的对应选项	

C++ Java	<code>int load_font(String fontname, String encoding, String optlist)</code>
Perl PHP	<code>int PDF_load_font(resource p, string fontname, string encoding, string optlist)</code>
C	<code>int PDF_load_font(PDF *p, const char *fontname, int len, const char *encoding, const char *optlist)</code>

搜索字体，并准备稍后使用。

fontname （名称字符串）字体的真实名称或别名。用于查找字体数据。区分大小写。

len （仅用于 C 语言绑定）UTF-16 字符串的 *fontname* 长度（以字节为单位）。如果 *len = 0*，则必须提供以 *null* 终结的字符串。

encoding 要用于字体的编码（区分大小写）：

- ▶ 用于基于 Unicode 的寻址的 *unicode* ；
- ▶ 一种预定义 8 位编码，*winansi*、*macroman*、*macroman_apple*、*ebcdic*、*ebcdic_37*、*pdfdoc*、*iso8859-X*、*cpXXXX* 或 *U+XXXX* ；
- ▶ 用于自动选择编码的 *host* 或 *auto* ；
- ▶ 从文件载入的用户定义编码或通过 *PDF_encoding_set_char()* 定义的编码的名称；
- ▶ 用于 CJK 代码页的 *cp932*、*cp936*、*cp949*、或 *cp950* （在 Windows 上将使用系统代码页，在所有其他系统上，对应的 CMap 必须可用）；
- ▶ 用于字形 ID 寻址的 *glyphid* ；
- ▶ *builtin* 选择字体的内部编码（大多数情况下用于符号字体）；
- ▶ 标准 CMap 的名称；
- ▶ 用于标准 CJK 字体和 OpenType CID 字体的 CID 寻址的 *Identity-H* 或 *Identity-V* ；必须为每个字形提供 2 个本地字节顺序的字节；这些编码主要用于创建 CJK 字符集表；
- ▶ 操作系统已知的编码名称（并非所有平台上都可以使用）。

编码必须兼容选择的字体。表 3.3 列出了允许的编码和字体类型组合。

表 3.3 允许对不同字体类型进行编码

字体格式	<i>unicode</i>	8 位编码	CMap、 <i>cp936</i> 等 ¹	内建	字形
<i>PostScript Type 1</i>	是 ²	是	—	是	—
<i>Type 3</i>	是 ²	是	—	—	—
具有 <i>PostScript</i> 轮廓的 <i>Western OpenType (SID)</i>	是 ³	是 ³	—	是	是 ³
带有 <i>TrueType</i> 轮廓的 <i>TrueType</i> 和 <i>OpenType</i>	是 ³	是 ³	是 ^{3,4}	仅限 <i>Symbol</i> 字体	是 ³
具有 <i>PostScript</i> 轮廓的 <i>CJK OpenType (CID)</i>	是	—	是	—	是
无字体嵌入的标准 <i>CJK</i> 字体	是	—	是 ⁵	—	—

1. 字体必须支持所选的 CMap 或代码页，否则将被拒绝。由于文本输出需要使用 CMap，必须配置 CMap 访问。
2. 最多可寻址 256 个不同字形。
3. 在默认情况下将嵌入字体，但可通过设置 *embedding=false* 阻止嵌入字体。
4. 在表单域中无法使用此字体。
5. 如果 *keepnative=false*，则可以在文本流中使用带非 *Unicode CMap* 的标准 *CJK* 字体；如果 *keepnative=true*，则可以在表单域中使用该字体。绝不能在文本流和表单域中同时使用带非 *Unicode CMap* 的标准 *CJK* 字体。

optlist 根据表 3.4 提供的选项列表。可以使用下列选项：
autocidfont、*auto-subsetting*、*capheight*、*descender*、*embedding*、*errorpolicy*、*fontstyle*、*keepnative*、 *Kerning*、*linegap*、*metadata*、*monospace*、*replace-mentchar*、*subsetlimit*、*subsetminsize*、子集化、*unicodemap*、*vertical*、*xheight*

返回值 字体句柄，供稍后与 *PDF_info_font()*、文本输出函数和各种函数的 *font* 选项一起使用。默认情况下，如果无法加载请求的字体 / 编码组合并且不引发异常，此函数将返回错误代码 -1（在 PHP 中：o）。不过，可以使用 *errorpolicy* 参数或选项更改此行为。如果该函数返回 -1（在 PHP 中：o），可以使用 *PDF_get_errmsg()* 请求失败原因。
除用作字体句柄外，返回的数值对用户没有任何意义。例如，在不同文档中请求相同的字体 / 编码组合可能会导致不同的字体句柄。

详细说明 此函数准备字体，以供稍后使用。将从内存、系统（对于宿主字体）或者（虚拟或基于磁盘）的规格文件载入字体规格。如果由于配置问题而不能使用请求的字体 / 编码组合（例如，无法找到字体、规格和编码文件，或检测到匹配错误），将返回错误代码或引发异常（取决于 *errorpolicy*）。如能使用，此参数返回的值可以用作调用其他字体相关函数的字体句柄。

除非提供了不同的 *encoding* 参数，或 *fontstyle*、*monospace* 或 *vertical* 选项之一不相同，否则再次使用相同字体名称调用此函数时，将返回与第一次调用相同的字体句柄。

冲突的选项：（直接通过 *PDF_load_font()* 或者间接通过 *PDF_add/create_textflow()* 或 *PDF_fill_textblock()*）载入请求字体，而不使用 *embedding*、*subsetting* 或 *subsetting* 时，在稍后再次载入相同字体时，将忽略这些选项。

范围 字体、文档、页面、路径、图案、模板、字形

参数 参见表 3.1 和表 3.2

表 3.4 *PDF_load_font()* 的选项

选项	描述
<i>ascender</i>	（介于 -2048 到 2048 之间的整型）强制将对应的排版属性设为指定值。这将覆盖字体中找到的任何值，并在字体（例如 <i>Type 3</i> 字体）不包含任何此类信息时尤其有用。默认值：如果存在，则为字体中的值，否则估计值（可使用 <i>PDF_info_font()</i> 查询）
<i>autocidfont</i>	（布尔型）如果为 true，具有除 <i>winansi</i> 、 <i>macroman</i> 、 <i>builtin</i> 之外的 8 位编码的 <i>TrueType</i> 字体以及没有字形名称的 <i>OpenType</i> 字体将自动存储为 <i>CID</i> 字体。这样可防止某些 <i>winansi</i> 编码之外的字形无法访问的问题。默认值：true
<i>auto-subsetting</i>	（布尔型）动态地决定是否对字体进行子集化处理，具体取决于 <i>subsetlimit</i> 和 <i>subsetminsize</i> 选项以及实际使用的字形。如果已经提供了 <i>subsetting</i> 选项，将忽略此选项。默认值：true
<i>capheight</i>	（介于 -2048 到 2048 之间的整型）参见上面的 <i>ascender</i> 。
<i>descender</i>	（介于 -2048 到 2048 之间的整型）参见上面的 <i>ascender</i> 。
<i>embedding</i>	（布尔型；对于 <i>PDF/A</i> 和 <i>PDF/X</i> 必须为 true）控制是否嵌入字体。如果要嵌入字体，除规格信息外，还必须提供字体轮廓文件（ <i>TrueType</i> 和 <i>OpenType</i> 字体则无此要求），并且， <i>PDF</i> 输出内必须包括实际的字体轮廓定义。如果不嵌入字体， <i>PDF</i> 输出中只包括字体的常规信息。 默认值：通常为 false，但对于 <i>encoding=unicode</i> 以及采用其他导致转换为 <i>CID</i> 编码的 <i>TrueType/OpenType</i> 字体，为 true（参见表 3.3）。虽然 <i>PDFlib</i> 将自动嵌入此类字体，可通过将 <i>embedding</i> 设置为 false 阻止字体嵌入。在这种情况下，必须在浏览或打印 <i>PDF</i> 文档的系统上安装该字体。此选项对 <i>Type 3</i> 字体没有任何影响。
<i>errorpolicy</i>	（关键字）控制出现错误时的行为（参见第 28 页上的第 2.6 节“异常处理”）。
<i>fontstyle</i>	（关键字）控制人工字形的生成。可能的关键字包括 <i>normal</i> 、 <i>bold</i> 、 <i>italic</i> 、 <i>bolditalic</i> 。对于未嵌入的 <i>TrueType</i> （非 <i>TTC</i> ）和 <i>OpenType</i> 字体，将由 <i>Acrobat</i> 创建伪字体样式，否则由 <i>PDFlib</i> 创建（使用与 <i>fakebold</i> 参数或选项中相同的增强方法）。在后一种情况下，可以使用 <i>italicangle</i> 参数或选项控制倾斜角度。如果对核心字体之一应用此选项，将不创建伪字体样式，而是选择适当的粗体、斜体或粗斜变体。如果这些字体不可用（例如对 <i>Times-Bold</i> 应用 <i>bold</i> ），将忽略此选项。默认值： <i>normal</i>
<i>fontwarning</i>	不常用，使用 <i>errorpolicy</i>
<i>keepnative</i>	（布尔型；仅与带非 <i>Unicode CMap</i> 的标准 <i>CJK</i> 字体相关；对于其他字体将忽略此选项）如果为 false，在 <i>embedding=true</i> 的情况下将强制为 false）如果为 false，在创建 <i>PDF</i> 输出时，将此字体的文本转换成 <i>Unicode</i> （使用 <i>glyphid</i> 寻址和 <i>Identity-H</i> 编码）。在创建 <i>PDF</i> 输出时，将此字体的文本转换成 <i>Unicode</i> （使用 <i>glyphid</i> 寻址和 <i>Identity-H</i> 编码）。这不影响提供给 <i>API</i> 函数的文本，这些文本仍将匹配所选择的 <i>CMap</i> （例如 <i>Shift-JIS</i> ）。不过，可以在文本流和所有简单文本输出函数中（表单域除外）使用此字体。 如果为 true，根据指定的 <i>CMap</i> 将此字体的文本以其本地格式写入 <i>PDF</i> 输出。可以在表单域和所有简单文本输出函数中使用此字体，文本流除外。默认值：在 <i>embedding=false</i> 的情况下为 true，在 <i>embedding=true</i> 的情况下为 false
<i>kerning</i>	（布尔型）控制是否从字体中读取字距调整值。默认值：false

表 3.4 PDF_load_font() 的选项（续）

选项	描述
linegap	（介于 -2048 到 2048 之间的整型）参见上面的 ascender。
metadata	（选项列表； PDF 1.4）提供字体的元数据（参见第 144 页上的第 12.2 节“XMP 元数据”）。
monospace	（介于 1 到 2048 之间的整型； PDF/A 除外）强制字体中的所有字形使用指定的宽度（在字体坐标系中：1000 单位等于字体大小）。对于 Type 3 字体，将修改所有不同于 o 的字形宽度。只建议为标准 CJK 字体使用此选项，不支持核心字体；如果是嵌入字体，则忽略此选项。默认值：absent（使用字体的规格）
replacementchar	（Unichar；只有在 glyphcheck=replace 时才相关）使用指定的 Unicode 值（或在 builtin 编码的情况下使用指定的代码）替换在选择的字体中不可用和无法替代的字形。可以使用 U+0000 指定字体的“缺失字形”符号。默认值：无替换字符
subsetlimit	（浮点型或百分比；对于 Type 3 字体将忽略此选项）如果文档中使用的字形与字体中字形总数的百分比超出提供的百分比，将禁用自动字体子集化。默认值：100%
subsetminsize	（浮点型；对于 Type 3 字体将忽略此选项）如果原始字体文件的大小小于提供的 KB 值，将禁用自动字体子集化。默认值：50
子集化	（布尔型）控制是否子集化字体。Type 3 字体的子集化要求两次定义字体（参见 PDFlib 教程），并且必须在第一次调用 PDF_load_font() 时提供 subsetting 选项。默认值：false
unicodemap	（布尔型；对于 pdfa=PDF/A-1a:2005 必须设为 false）控制是否生成 ToUnicode CMap。标签 PDF 模式将忽略此选项。默认值：true
vertical	（布尔型；仅用于 TrueType 和 OpenType 字体；如果指定了预定义 CMap，将忽略此选项；如果字体名称以 @ 开头，将强制为 true）如果为 true，将针对垂直书写模式准备字体。
xheight	（介于 -2048 到 2048 之间的整型）参见上面的 ascender。

C++ Java double info_font(int font, String keyword, String optlist)

Perl PHP float PDF_info_font(resource p, int font, string keyword, string optlist)

C double PDF_info_font(PDF *p, int font, const char *keyword, const char *optlist)

查询有关已加载字体的详细信息。

字体 PDF_load_font() 返回的字体句柄。

keyword 根据表 3.5 指定请求的信息的关键词。

optlist 根据表 3.5 提供的选项列表。可以使用下列选项：
ascender、capheight、cidfont、code、descender、encoding、fontfile、fontname、fontstyle、
glyphid、glyphname、hostfont、italicangle、kerningpairs、linegap、maxcode、metricsfile、
numcids、numglyphs、replacementchar、standardfont、supplement、symbolfont、unicode、
unicodefnt、vertical、weight、willembd、willssubset、xheight

返回值 通过 keyword 以及在某些情况下通过辅助选项请求的一些字体属性值。对于未指定的关键字和选项组合，将返回 -1。某些关键字将通过返回其字符串索引来间接返回字符串。可以通过 PDF_get_parameter() 和 string 参数检索对应的字符串（参见表 2.1）。

范围 任意，对象除外

表 3.5 PDF_info_font() 的关键字和选项

keyword	说明
ascender	ascender 的规格值。支持的选项（默认值：fontsize=1000）： faked （布尔型）如果该值因在字体或规格文件中不可用而必须估计，则为 1；否则为 0 fontsize （浮点型）值必须缩放至指定字体大小
capheight	capheight 的规格值。参见 ascender。
cidfont	如果字体将以 CID 字体形式嵌入，为 1；否则为 0
code	（仅用于 8 位编码的字体）介于 0...255 之间的数字，指定一个编码插槽，该插槽根据下列选项之一指定特征；或者，如果找不到此类槽，则 -1： unicode (Unichar) Unicode 字符 glyphname （字符串）具有指定字形名称的插槽
descender	descender 的规格值。参见 ascender。
encoding	字体的编码或 CMap 名称的字符串索引。支持的选项（默认值：actual）： api （布尔型）如果为 true，编码名称与 API 中指定的相同 actual （布尔型）如果为 true，则为字体所使用的实际编码名称
fontfile	字体轮廓文件的路径名称的字符串索引，或者为 -1（如果不可用）
fontname	字体名称的字符串索引，或者为 -1（如果不可用）支持的选项（默认值：acrobat）： api （布尔型）如果为 true，字体名称与 API 中指定的相同 full （布尔型）如果为 true，则为 PDF 字体说明符中的 /FontName 项 acrobat （布尔型）如果为 true，字体名称将与 Acrobat 中显示的相同
fontstyle	fontstyle 选项的值的字符串索引（normal、bold、italic 或 bolditalic）。支持的选项： faked 如果由 PDFlib 实现字体样式，则为 1，如果由 Acrobat 实现字体样式则为 0
glyphid	（用于 8 位编码的字体、encoding=builtin 的符号字体和 encoding= unicode 的字体）介于 0...65535 之间的数字，指定字体内部字形 id (GID)（使用下列选项之一指定特征）；或者为 -1（如果找不到此类字形）： code （介于 0...255 之间的数字；仅用于 8 位编码的字体和 encoding=builtin 的符号字体）编码插槽 unicode （Unichar；仅用于 encoding=unicode 的字体）具有指定 Unicode 字符的插槽
glyphname	（用于除 encoding=builtin 的符号字体和具有标准 CMap 的字体之外的所有字体）由下列选项之一指定的字形名称的字符串索引，或者为 -1（如果找不到此类字形）： code （介于 0...255 之间的数字；仅用于 8 位编码的字体）编码插槽 glyphid （介于 0...65535 之间的数字；仅用于 encoding=unicode 的 TT/OT 字体和 encoding=glyphid 的字体）内部字形 id unicode ¹ (Unichar) 具有指定 Unicode 字符的插槽
hostfont	如果字体为宿主字体，为 1；否则为 0
italicangle	字体的倾斜角度（PDF 字体说明符中的 /ItalicAngle）
kerneingpairs	字体中的字距调整对的数目
linegap	linegap 的规格值。参见 ascender。
maxcode	字体编码的最高代码值
metricsfile	字体规格文件（AFM 或 PFM）的路径名称的字符串索引，或者为 -1（如果不可用）
numcids	如果字体使用标准 CMap，为 CID 数；否则为 -1
numglyphs	字体中字形的数目
replacement char	字体的替换字体的 Unicode 值（对于兼容 Unicode 的字体）或代码值（对于符号字体）
standardfont	如果字体为 PDF 核心字体或标准 CJK 字体，为 1；否则为 0
supplement	对于具有标准 CJK CMap 的字体，为字符集的补充数目；否则为 0

表 3.5 PDF_info_font() 的关键字和选项（续）

keyword	说明
symbolfont	如果字体为符号字体，为 1；否则为 0（在 PDF 字体说明符中的符号标志；基于字体数据判断）
unicode	（用于除 encoding=builtin 的符号字体之外的所有字体）由下列选项之一指定的字形的 Unicode UTF-32，或者为 -1（如果找不到 Unicode 值）： cid （数字；仅用于具有标准 CMap 的字体）字形的 CID 值 code （介于 0...255 之间的数字；仅用于 8 位编码的字体）编码插槽 glyphid （介于 0...65535 之间的数字；仅用于 encoding=unicode 的 TT/OT 字体和 encoding=glyphid 的字体）内部字形 id 字形名称 ¹ （字符串；仅用于 8 位编码的字体）为其确定 Unicode 值的字形的名称
unicodefont	如果使用允许 Unicode 文本的编码（builtin、glyphid、和非 Unicode CMap 不允许 Unicode 文本）加载字体，为 1；否则为 0
vertical	如果字体为垂直书写模式，为 1；否则为 0
weight	字体粗细度，介于 100...900 之间；400 代表常规，700 代表粗体
willembed	如果将嵌入字体（通过 embedding 选项或强制的字体嵌入），为 1；否则为 0
willsubset	如果将创建字体子集（如果 autosubsetting=true，必须达到 subsetlimit 才能激活子集化），为 1；否则为 0
xheight	xheight 的规格值。参见 ascender。

1. 可以使用关键字 glyphname 和选项 unicode（反之亦然）确定带 8 位编码的字体的字形名称与 Unicode 值之间的关系。不过，也可以使用此组合基于 PDFlib 的内部映射表独立于任何特定的 8 位编码确定这些映射。对于这种情况，必须提供不带任何 8 位编码的字体的字体句柄。

3.2 Type 3 字体定义

C++ Java	<code>void begin_font(String fontname, double a, double b, double c, double d, double e, double f, String optlist)</code>
Perl PHP	<code>PDF_begin_font(resource p, string fontname, float a, float b, float c, float d, float e, float f, string optlist)</code>
C	<code>void PDF_begin_font(PDF *p, char *fontname, int reserved, double a, double b, double c, double d, double e, double f, const char *optlist)</code>

开始 Type 3 字体定义。

fontname （名称字符串）用来注册字体的名称，稍后可用于 `PDF_load_font()`。

reserved （仅用于 C 语言绑定）保留，必须为 0。

a, b, c, d, e, f （在第二次处理子集字体时将被忽略）字体矩阵的元素。此矩阵定义将在其中绘制字形的坐标系统。这六个值以 PostScript 和 PDF 中相同的方式构成坐标系统（参见相关参考）。为了避免转换退化， $a*d$ 不能等于 $b*c$ 。1000 x 1000 坐标系统典型的字体矩阵是 `[0.001, 0, 0, 0.001, 0, 0]`。

optlist （在第二次处理子集字体时将被忽略）根据表 3.6 的选项列表。可以使用下列选项 `colorized`、`familyname`、`stretch`、`weight`、`widthsonly`

详细说明 此函数将所有文本、图形和颜色状态参数重设为他们的缺省值。字体可以包含任意数量的字形，但只能通过编码访问 256 个字形。可以在当前文档范围结束之前使用字体。

范围 文档、页面；此函数是字体范围的起点，必须与对应的 `PDF_end_font()` 调用成对使用。在第二次处理子集化字体时，仅允许 `document` 范围。

表 3.6 `PDF_begin_font()` 的选项

选项	描述
colorized	（布尔型）如果为 <code>true</code> ，字体可以显式指定具体字符的颜色。如果为 <code>false</code> ，将使用（使用字体时，而非定义字体时的）当前颜色绘制所有字符，字形定义不能包含除蒙版外的任何其他色彩运算符或图像。默认值： <code>false</code>
familyname¹	（字符串；PDF 1.5）字体系列的名称
stretch¹	（关键字；PDF 1.5）字体的拉伸值。关键字： <code>ultracondensed</code> 、 <code>extracondensed</code> 、 <code>condensed</code> 、 <code>semicondensed</code> 、 <code>normal</code> 、 <code>semiexpanded</code> 、 <code>expanded</code> 、 <code>extraexpanded</code> 、 <code>ultraexpanded</code> 。默认值： <code>normal</code>
weight¹	（整型或关键字；PDF 1.5）字体粗细度。数字和等同关键字： <code>100=thin</code> 、 <code>200=extralight</code> 、 <code>300=light</code> 、 <code>400=normal</code> 、 <code>500=medium</code> 、 <code>600=semibold</code> 、 <code>700=bold</code> 、 <code>800=extrabold</code> 、 <code>900=black</code> 。默认值： <code>normal</code>
widthsonly	（布尔型）如果为 <code>true</code> ，只定义字体和字形的规格。将忽略 <code>PDF_begin_glyph()</code> 和 <code>PDF_end_glyph()</code> 之间的 <code>API</code> 函数调用。必须在单独的字体范围内定义实际的字形轮廓，并循环遍历所有字形。这使 <code>PDFlib</code> 能够对 Type 3 字体执行子集化；因为在第一次执行中将忽略字形定义，客户端应用程序可以在两次执行中都提供相同的字形定义。默认值： <code>false</code>

1. 在创建标签 PDF 时强烈建议使用这些选项，否则将忽略它们。

C++ Java

void end_font()

Perl PHP

PDF_end_font(resource p)

C

void PDF_end_font(PDF *p)

终结 Type 3 字体定义。

范围 字体；此函数终结字体范围，必须与对应的 `PDF_begin_font()` 调用成对使用。

C++ Java

void begin_glyph(String glyphname, double wx, double llx, double lly, double urx, double ury)

Perl PHP

PDF_begin_glyph(resource p, string glyphname, float wx, float llx, float lly, float urx, float ury)

C

void PDF_begin_glyph(PDF *p,
char *glyphname, double wx, double llx, double lly, double urx, double ury)

开始 Type 3 字体的字形定义。

glyphname 字形的名称。此名称必须用于将与该字体一起使用的任何编码。字体内的字形名称必须是唯一的。

wx （在对子集字体的第二次执行中将被忽略）字形坐标系统内字形的宽度，具体由字体的矩阵指定。

llx、**lly**、**urx**、**ury** （在对子集字体的第二次执行中将被忽略）如果字体的 *colorized* 选项为 *false*（缺省值），这些值可指定字形定界框左下角和右上角的坐标。为避免出现 *PostScript* 打印问题，定界框值必须是正确的。如果字体的 *colorized* 选项为 *true*，所有四个值都必须为 0。

详细说明 字体中的字形可使用文本、图形和图像函数定义。但是，如果字体的 *colorized* 选项为 *true*，或者已使用 *mask* 选项打开了图像，则只能使用图像。强烈建议使用内嵌图像功能定义 Type 3 字体中的位图。

因为 *colorized* 选项为 *true* 时，字形定义将继承所处页面的整个图形状态，所以字形定义应显式设置图像状态与字形定义相关的各个方面（例如 *linewidth*）。

范围 页面，字体；此参数是字形范围的开头，必须与对应的 `PDF_end_glyph()` 调用成对使用。如果 `PDF_begin_font()` 中的 *widthonly=true*，将忽略 `PDF_begin_glyph()` 和 `PDF_end_glyph()` 之间的所有 API 函数调用。

C++ Java

void end_glyph()

Perl PHP

PDF_end_glyph(resource p)

C

void PDF_end_glyph(PDF *p)

终结 Type 3 字体的字形定义。

范围 字形；此函数终结字形范围，必须与对应的 `PDF_begin_glyph()` 调用成对使用。

3.3 编码定义

C++ Java	<code>void encoding_set_char(String encoding, int slot, String glyphname, int uv)</code>
Perl PHP	<code>PDF_encoding_set_char(resource p, string encoding, int slot, string glyphname, int uv)</code>
C	<code>void PDF_encoding_set_char(PDF *p, const char *encoding, int slot, const char *glyphname, int uv)</code>

向自定义 8 位编码添加字形名称和 / 或 Unicode 值。

encoding 编码的名称。此名称必须用于 `PDF_load_font()`。编码名称必须不同于任何内置编码以及所有以前使用过的编码。

slot 要定义的字符位置，`0 <= slot <= 255`。在给定的编码中，特定插槽只能填充一次。

glyphname 字符的名称。

uv 字符的 Unicode 值。

详细说明 只有必须与非标准 8 位编码一起工作的特殊应用程序才要求此函数。此函数可多次调用，最多可在编码内定义 256 个字符插槽。在第一次使用之前，可向特定编码添加更多的字符；否则，将引发异常。并非所有代码点都必须指定，未定义的插槽将用 `.notdef` 和 `U+0000` 填充。

字形名称和 Unicode 值可能有三种组合：

- 提供的 **glyphname**，**uv=0**：这相当于没有 Unicode 值的编码文件；
- 提供 **uv**，但没有提供 **glyphname**：这相当于代码页文件；
- 提供 **glyphname** 和 **uv**：这相当于有 Unicode 值的编码文件。

可以在当前**对象**范围结束之前使用定义的编码。

范围 对象、文档、页面、图案、模板、路径、字体、字形

3.4 简单文本输出

注：在本节中，所有提供给函数的文本都必须匹配用 `PDF_load_font()` 和指定文本格式选择的编码。由于 *Acrobat* 中的限制，文本字符串的长度不能超过 32 KB。

表 3.7 和表 3.8 列出了本节的相关参数及相应的值（参见第 11 页上的第 2.1 节“参数处理”）。

表 3.7 `PDF_get/set_parameter()` 的与文本相关的关键字

关键字	说明
autospace	如果为 true ，且当前字体兼容 <i>Unicode</i> ， <i>PDFlib</i> 将在每个用 <i>show</i> 操作生成的文本输出后添加一个空格字符 (0x20)。这可能会对生成标签 <i>PDF</i> 很有用。请注意，在 <i>show</i> 操作后添加空格会改变当前文本位置。默认值： false 。范围：任意
charref	如果为 true ，将启用数字和字符实体引用与字形名称引用的替换。默认值： false
escape-sequence	若为 true ，将在内容字符串、超文本字符串和名称字符串中启用转义序列的替代。默认值： false
fakebold	如果为 true ，将通过压印三次模拟粗体。强烈建议使用粗体变体突出文本，此参数将创建效果次于真实粗体文本的文本输出，并可能禁止文本提取。默认值： false
glyphcheck	默认的字形检查策略（参见表 4.1）。默认值： replace 。范围：任意
glyphwarning	不常用，使用 <i>errorpolicy</i>
 Kerning	如果为 true ，将为已用 <i>kerning</i> 选项打开的字体启用字距微调；如果为 false ，则停用字距微调。默认值： true 。范围：任意
textformat	（仅用于不兼容 <i>Unicode</i> 的语言绑定）文本输出函数预计客户端提供字符串时使用的格式。其可选值为 <i>bytes</i> 、 <i>utf8</i> 、 <i>ebcdicutf8</i> （仅在 <i>iSeries</i> 和 <i>zSeries</i> 上）、 <i>utf16</i> 、 <i>utf16</i> 、 <i>utf16be</i> 、和 <i>auto</i> 。默认值： auto 。范围：任意
underline overline strikeout	当前下划线、上划线和删除线模式；显式更改和开始新页面之前，将一直保留这些模式。这些模式可以彼此独立设置，并在每页开头重设为 false 。默认值： false 。范围：页面、图案、模板、字形 true 下划线 / 上划线 / 删除线文本 false 不使用下划线 / 上划线 / 删除线文本

表 3.8 `PDF_get/set_value()` 的与文本相关的关键字

关键字	说明
charspacing	字符间距，即：在字符串中放置具体字符之后，当前点的位移。它以用户坐标系统的单位指定，并将在每页的开头和结尾重设为缺省值 0。要扩大字符间距，可在横排书写模式中使用正值，在直排书写模式中使用负值。范围：页面、图案、模板、字形、文档。
font¹	如果没有设置字体，则为已经用 <i>PDF_setfont()</i> 设置的当前字体的标识符，或 -1（在 <i>PHP</i> 中：0）。范围：页面、图案、模板、字形
fontsize¹	当前字体的大小，必须事先用 <i>PDF_setfont()</i> 设置。范围：页面、图案、模板、字形
horizscaling	缩放至给定百分比（必须不等于 0）的水平文本。文本缩放比例可按给定的比例收缩或扩展文本。在每页的开头和结尾，此参数将设为缺省值 100。文本缩放比例始终与水平坐标相对。范围：页面、图案、模板、字形、文档。
italicangle	文本的斜体（倾斜）角度的度数，范围在 -90° 和 90° 之间。如果只有常规字体，可使用负值模拟斜体文本，尤其是 <i>CIK</i> 字体。默认值：0（将在每页的开头和结尾重设此参数）。范围：页面、图案、模板、字形、文档
leading	行距，即相邻文本行的基线之间的距离。 <i>leading</i> 用于 <i>PDF_continue_text()</i> 。使用 <i>PDF_setfont()</i> 选择新字体时，它设为字体大小的值。设置 <i>leading</i> 等于字体大小将导致密集行距（ <i>leading</i> = 0 将导致叠印行）。但是，相邻行的字母上缘和字母下缘通常不会重叠。范围：页面、图案、模板、字形

表 3.8 PDF_get/set_value() 的与文本相关的关键字（续）

关键字	说明
textrendering	当前文本渲染模式。在每页的开头，它设为缺省值 o 。范围：页面、图案、模板、字形。 <div><div><div><div><div>0</div><div>填充文本</div></div><div><div>1</div><div>描边文本（轮廓）</div></div><div><div>2</div><div>填充和描边文本</div></div><div><div>3</div><div>不可见文本</div></div><div><div>4</div><div>填充文本并将其添加到剪贴路径</div></div><div><div>5</div><div>描边文本并将其添加到剪贴路径</div></div><div><div>6</div><div>填充并描边文本并将其添加到剪贴路径</div></div><div><div>7</div><div>将文本添加到剪贴路径</div></div></div></div></div>
textrise	文本抬升参数，该参数指定所需文本位置与缺省基线之间的距离。如文本抬升参数为正值，将向上移动文本。文本抬升始终与垂直坐标相对。这可能对上标和下标很有用。在每页开头，文本抬升设为缺省值 o 。范围：页面、图案、模板、字形
textx¹ texty¹	当前文本位置的 x 或 y 坐标。默认值： o 。范围：页面、图案、模板、字形
underline-position	下划线文本的描边线相对于基线的位置（以字体大小的分数表示）。可提供值 1000000 ，以设定将从字体规格或轮廓文件返回的字体特定值。默认值： 1000000
underline-width	下划线文本的绝对线宽。可提供值 o ，以设定将从字体规格或轮廓文件返回的字体特定值。默认值： o
wordspacing	字间距，即：在字符串中放置具体单字之后，当前点的位移。换言之，当前点将水平移至每个空格字符 (U+0020) 之后。间距值以文本间隔单位指定，并将在每页的开头和结尾重设为缺省值 o 。范围：页面、图案、模板、字形、文档

1. 仅用于 PDF_get_value()

C++ Java

void PDF_setfont(int font, double fontsize)

Perl PHP

PDF_setfont(resource p, int font, float fontsize)

C

void PDF_setfont(PDF *p, int font, double fontsize)

以指定的大小设置当前字体。

字体 PDF_load_font() 返回的字体句柄。

fontsize 字体的大小，以当前用户坐标系统的单位表示。字体大小不能为 **o**；如果为负值，将产生与当前转换矩阵相对的镜像文本。

详细说明 在调用任何简单文本输出函数之前，必须在每个页面上设置字体。不能在多个页面间保留字体设置。在每一页上，更改当前字体的次数不受任何限制。在所有文本格式化函数中（参见第 47 页上的第 4 章“格式化函数”）。可以使用 **font** 和 **fontsize** 选项指定字体。

范围 页面、图案、模板、字形

参数 此函数自动将 **leading** 参数设为 **fontsize**。

C++ Java `void set_text_pos(double x, double y)`
Perl PHP `PDF_set_text_pos(resource p, float x, float y)`
C `void PDF_set_text_pos(PDF *p, double x, double y)`

设置文本输出在页面上的位置。

x, y 要设置的当前文本位置。

详细说明 在每页开头，文本位置设为缺省值 (0, 0)。图形输出的当前点和当前文本位置将单独维护。

范围 页面、图案、模板、字形

参数 参见表 3.7 和表 3.8。

C++ Java `void show(String text)`
Perl PHP `PDF_show(resource p, string text)`
C `void PDF_show(PDF *p, const char *text)`
C `void PDF_show2(PDF *p, const char *text, int len)`

在当前文本位置，使用当前字体和大小打印文本。

文本 （内容字符串）要打印的文本。在 C 语言中，因为假定以 `null` 终结，所以使用 `PDF_show()` 时，`text` 不能包含 `null` 字符；对于可能包含 `null` 字符的字符串，应使用 `PDF_show2()`。

len （仅用于 `PDF_show2()`）UTF-16 字符串的 `text` 长度（以字节为单位）。如果 `len = 0`，则必须提供以 `null` 终结的字符串。

详细说明 必须已经使用 `PDF_setfont()` 设置了字体。当前文本位置移至打印文本的结尾。

范围 页面、图案、模板、字形

参数 参见表 3.7 和表 3.8。

绑定 `PDF_show2()` 只在 C 语言中使用；这是因为，在所有其他语言绑定中，可以使用 `PDF_show()` 提供任意字符串内容。

C++ Java `void xshow(String text, const double *xadvancelist)`
C `void PDF_xshow(PDF *p, const char *text, int len, const double *xadvancelist)`

使用具体的水平位置，以当前字体和大小打印文本。

文本 （内容字符串）要打印的文本。

len （仅用于 C 语言绑定）UTF-16 字符串的 `text` 长度（以字节为单位）。如果 `len = 0`，则必须提供以 `null` 终结的字符串。

xadvancelist 一组用于设定文本中的字形的 `x` 预置值。每个值都指定放置字形后的相对水平置换（使用用户坐标）。数组长度必须等于文本中的字形数量（不必与字节数 `len` 相等）。

详细说明 必须已经使用 `PDF_setfont()` 设置了字体。

范围 页面、图案、模板、字形

参数 参见表 3.7 和表 3.8。

绑定	仅用于 C 和 C++ 语言绑定。其他语言绑定可以使用 <code>PDF_fit_textline()</code> 中的 <code>xadvancelist</code> 选项实现相同的功能。
----	---

C++ Java	<code>void show_xy(String text, double x, double y)</code>
Perl PHP	<code>PDF_show_xy(resource p, string text, float x, float y)</code>
C	<code>void PDF_show_xy(PDF *p, const char *text, double x, double y)</code>
C	<code>void PDF_show_xy2(PDF *p, const char *text, int len, double x, double y)</code>

以当前字体打印文本。

文本 （内容字符串）要打印的文本。在 C 语言中，因为假定以 `null` 终结，所以使用 `PDF_show_xy()` 时，`text` 不能包含 `null` 字符；对于可能包含 `null` 字符的字符串，应使用 `PDF_show_xy2()`。

x, y 用户坐标系统中的位置，将在其上打印文本。

len （仅用于 `PDF_show_xy2()`）UTF-16 字符串的 `text` 长度（以字节为单位）。如果 `len = 0`，则必须提供以 `null` 终结的字符串。

详细说明 必须已经使用 `PDF_setfont()` 设置了字体。当前文本位置移至打印文本的结尾。

范围 页面、图案、模板、字形

参数 参见表 3.7 和表 3.8。

绑定 `PDF_show_xy2()` 只在 C 语言中使用；这是因为，在所有其他语言绑定中，可以使用 `PDF_show_xy()` 提供任意字符串内容。

C++ Java	<code>void continue_text(String text)</code>
Perl PHP	<code>PDF_continue_text(resource p, string text)</code>
C	<code>void PDF_continue_text(PDF *p, const char *text)</code>
C	<code>void PDF_continue_text2(PDF *p, const char *text, int len)</code>

在下一行打印文本。

文本 （内容字符串）要打印的文本。如果是空字符串，文本位置仍将移至下一行。在 C 语言中，因为假定以 `null` 终结，所以使用 `PDF_continue_text()` 时，`text` 不能包含 `null` 字符；对于可能包含 `null` 字符的字符串，应使用 `PDF_continue_text2()`。

len （仅用于 `PDF_continue_text2()`）UTF-16 字符串的 `text` 长度（以字节为单位）。如果 `len = 0`，必须像 `PDF_continue_text()` 中那样，提供以 `null` 终结的字符串。

详细说明 文本的位置（`x` 和 `y` 位置）以及各行间的间隔由 `leading` 参数及最近一次 `PDF_fit_textline()`、`PDF_show_xy()` 或 `PDF_set_text_pos()` 调用决定。当前点将移至打印文本的结尾处；此函数后续调用的 `x` 位置将不会更改。

范围 页面、图案、模板、字形；不应在直排书写模式下使用此函数。

参数 参见表 3.7 和表 3.8。

绑定 `PDF_continue_text2()` 只在 C 语言中使用；这是因为，在所有其他语言绑定中，可以使用 `PDF_continue_text()` 提供任意字符串内容。

C++ Java	<code>double stringwidth(String text, int font, double fontsize)</code>
Perl PHP	<code>float PDF_stringwidth(resource p, string text, int font, float fontsize)</code>
C	<code>double PDF_stringwidth(PDF *p, const char *text, int font, double fontsize)</code>
C	<code>double PDF_stringwidth2(PDF *p, const char *text, int len, int font, double fontsize)</code>

以任意字体计算 `text` 的宽度。

文本 （内容字符串）请求其宽度的文本。在 C 语言中，因为假定以 `null` 终结，所以使用 `PDF_stringwidth()` 时，`text` 不能包含 `null` 字符；对于可能包含 `null` 字符的字符串，应使用 `PDF_stringwidth2()`。

len （仅用于 `PDF_stringwidth2()`）UTF-16 字符串的 `text` 长度（以字节为单位）。如果 `len = 0`，则必须提供以 `null` 终结的字符串。

font `PDF_load_font()` 返回的字体句柄。

fontsize 字体的大小，以用户坐标系统的单位表示（参见 `PDF_setfont()`）。

返回值	使用某一字体的 <code>text</code> 的宽度；该字体用 <code>PDF_load_font()</code> 和提供的 <code>fontsize</code> 选择。返回的宽度值可以是负值（例如，已设置负值的水平缩放比例）。在垂直书写模式下，将返回最宽的字形的宽度（使用 <code>PDF_info_textline()</code> 确定文本的实际高度）。
详细说明	宽度的计算会考虑下列文本参数的当前值： <code>horizscaling</code> 、 <code>kerning</code> 、 <code>charspacing</code> 和 <code>wordspacing</code> 。
范围	字体、页面、图案、模板、路径、字形、文档
参数	参见表 3.7 和表 3.8。
绑定	<code>PDF_stringwidth2()</code> 仅用于 C 语言；这是因为，在所有其他语言绑定中，可以使用 <code>PDF_stringwidth()</code> 提供任意字符串内容。

3.5 Unicode 转换函数

这些函数可能对于 Unicode 字符串转换很有用。它们是专为在不识别 Unicode 的语言环境下工作的用户提供的。

C++	<code>string utf16_to_utf8(string utf16string)</code>
Perl PHP	<code>string PDF_utf16_to_utf8(resource p, string utf16string)</code>
C	<code>const char *PDF_utf16_to_utf8(PDF *p, const char *utf16string, int len, int *size)</code>

将字符串从 UTF-16 格式转换为 UTF-8。

utf16string 要转换的字符串。将解释字符串内的字节顺序标记 (BOM)。如果 BOM 缺失，则假定为平台的本身字节顺序。

len （仅用于 C 语言绑定）**utf16string** 的长度（以字节为单位）。

size （仅用于 C 语言绑定）指向内存位置的 C 语言指针，其中存储返回字符串的长度（以字节为单位）。如果指针为 NULL，将忽略指针。

返回值 转换后的 UTF-8 字符串。生成的 UTF-8 字符串将以 UTF-8 BOM (`\xEF\xBB\xBF`) 开头。在 EBCDIC 平台上，包括 BOM 的转换结果将最终转换为 EBCDIC。在下次调用任何 PDFlib 函数之前或者在引发异常之前，返回的字符串将一直有效。如果此后仍需要该字符串，则客户端必须复制该字符串。转换的字符串使用的内存将由 PDFlib 管理。

范围 任意

绑定 此函数不能用于支持 Unicode 的语言绑定。

C++	<code>string utf8_to_utf16(string utf8string, string ordering)</code>
Perl PHP	<code>string PDF_utf8_to_utf16(resource p, string utf8string, string ordering)</code>
C	<code>const char *PDF_utf8_to_utf16(PDF *p, const char *utf8string, const char *ordering, int *size)</code>

将字符串从 UTF-8 格式转换为 UTF-16。

utf8string 要转换的字符串，其中必须包含有效的 UTF-8 序列（在 EBCDIC 平台上，必须用 EBCDIC 编码）。如果存在字节顺序标记 (BOM)，该标记将被删除。

ordering 指定结果字符串的字节顺序。

- ▶ **utf16** 或空字符串：转换后的字符串将不包含任何 BOM，并将按平台的本身字节顺序存储。
- ▶ **utf16le**：转换后的字符串将用 **little endian** 格式进行格式化，并将添加 **little-endian** BOM (`\xFF\xFE`) 前缀。
- ▶ **utf16be**：转换后的字符串将用 **big endian** 格式进行格式化，并将添加 **big endian** BOM (`\xFE\xFF`) 前缀。

size （仅用于 C 语言绑定）指向内存位置的 C 语言指针，其中存储返回字符串的长度（以字节为单位）。

返回值	转换后的 UTF-16 字符串。在下次调用任何 PDFlib 函数之前或者在引发异常之前，返回的字符串将一直有效。如果此后仍需要该字符串，则客户端必须复制该字符串。转换的字符串使用的内存将由 PDFlib 管理。
范围	任意
绑定	此函数不能用于支持 Unicode 的语言绑定。
<hr/>	
C++	<i>string utf32_to_utf16(string utf32string, string ordering)</i>
Perl PHP	<i>string PDF_utf32_to_utf16(resource p, string utf32string, string ordering)</i>
C	<i>const char *PDF_utf32_to_utf16(PDF *p, const char *utf32string, int len, const char *ordering, int *size)</i>
<hr/>	
将字符串从 UTF-32 格式转换为 UTF-16。	
utf32string 要转换的字符串，其中必须包含有效的 UTF-32 序列（在 EBCDIC 平台上，必须用 EBCDIC 编码）。如果存在字节顺序标记 (BOM)，该标记将被解释。	
len （仅用于 C 语言绑定） <i>utf32string</i> 的长度（以字节为单位）。	
ordering 指定结果字符串的字节顺序。	
<ul style="list-style-type: none">▶ utf16 或空字符串: 转换后的字符串将不包含任何 BOM，并将按平台的本身字节顺序存储。▶ utf16le: 转换后的字符串将用 little endian 格式进行格式化，并将添加 little endian BOM (\xFF\xFE) 前缀。▶ utf16be: 转换后的字符串将用 big endian 格式进行格式化，并将添加 big endian BOM (\xFE\xFF) 前缀。	
size （仅用于 C 语言绑定）指向内存位置的 C 语言指针，其中存储返回字符串的长度（以字节为单位）。	
返回值	转换后的 UTF-16 字符串。在调用除 <i>PDF_utf16_to_utf8()</i> 、 <i>PDF_utf8_to_utf16()</i> 、和 <i>PDF_utf32_to_utf16()</i> 之外的其他 PDFlib 函数或引发异常之前，返回的字符串将一直有效。如果此后仍需要该字符串，则客户端必须复制该字符串。转换的字符串使用的内存将由 PDFlib 管理。
范围	任意
绑定	此函数不能用于支持 Unicode 的语言绑定。

4 格式化函数

4.1 使用文本流的单行文本

C++ Java	<code>void fit_textline(String text, double x, double y, String optlist)</code>
Perl PHP	<code>PDF_fit_textline(resource p, string text, float x, float y, string optlist)</code>
C	<code>void PDF_fit_textline(PDF*p, const char *text, int len, double x, double y, const char *optlist)</code>

根据各种选项，在位置 (x,y) 上放置单行文本。

文本 （内容字符串）要打印的文本。

len （仅用于 C 语言绑定）UTF-16 字符串的 text 长度（以字节为单位）。如果 len = 0，则必须提供以 null 终结的字符串。

x, y 用户坐标系统中的参考点坐标；将根据各种选项，在此坐标上放置文本。

optlist 选项列表，根据表 4.1 指定选项。可以使用下列选项：

- ▶ 字体相关的选项：encoding、字体、fontname、fontsize
fontname 和 encoding （对应于 PDF_load_font() 中的同名参数）选项都可以用来选择字体。此外，font 选项也可用于提供之前使用的 PDF_load_font() 调用创建的字体句柄。如果指定了 font，将忽略 fontname 和 encoding 选项。要求使用 fontsize 选项。
- ▶ PDF_load_font() 的所有选项（参见表 3.4）。仅当同时提供了 fontname 和 encoding 选项，以下这些选项才可用（font 选项除外）：
ascender、autocidfont、auto-subsetting、capheight、descender、embedding、fontstyle、keepnative、kerning、linegap、metadata、monospace、replacementchar、subsetlimit、subsetminsize、子集化、unicodemap、vertical、xheight
- ▶ 格式化：alignchar、boxsize、fitmethod、leader、margin、orientate、position、rotate、stamp、xadvancelist
- ▶ 外观：charref、charspacing、dasharray、escape-sequence、fakebold、fillcolor、glyphcheck、horizscaling、italicangle、kerning、匹配框、overline、showborder、shrinklimit、strikeout、strokecolor、strokewidth、textformat、textrendering、textrise、underline、underline-position、underline-width、wordspacing

详细说明 除非用选项显式覆盖，否则将使用当前文本和图形状态参数控制文本输出的外观。另一方面，此函数不会修改当前文本和图形状态（尤其是，不会影响当前字体）。但是，textx/texty 参数将调整到指向生成文本输出的结尾处。

范围 页面、图案、模板、字形；不应在直排书写模式下使用此函数。

参数 参见表 3.7 和表 3.8。

C++ Java	<code>double info_textline(String text, String keyword, String optlist)</code>
Perl PHP	<code>float PDF_info_textline(resource p, string text, string keyword, string optlist)</code>
C	<code>double PDF_info_textline(PDF*p, const char *text, int len, const char *keyword, const char *optlist)</code>

执行文本行格式化和查询规格结果。

文本 （内容字符串）文本行的内容。

表 4.1 `PDF_fit_textline()` 和 `PDF_info_textline()` 的选项

关键字	说明
alignchar	(<i>Unichar</i> 或关键字) 若在文本中找不到指定字符, 其左下角将在参考点处对齐。对于 <code>orientate=north</code> 或 <code>south</code> 的水平文本, <code>position</code> 选项中提供的第一个值将定义位置。对于 <code>orientate=west</code> 或 <code>east</code> 的水平文本, <code>position</code> 选项中提供的第二个值将定义位置。如果在文本中不存在指定的对齐字符, 将忽略此选项。值 <code>o</code> 和关键字 <code>none</code> 将禁用对齐字符。尽管因 <code>alignchar</code> 的强制定位而不能将文本放置在限定框中, 仍将应用指定的 <code>fitmethod</code> 。默认值: <code>none</code>
boxsize	(浮点值列表) 指定框宽度和高度的两个值, 将相对于这两个值放置并可能缩放文本。框的左下角即为参考点 (x, y) 。放置文本并使其填充此框的操作由 <code>position</code> 和 <code>fitmethod</code> 选项控制。如果 <code>width=0</code> , 则只考虑高度; 如果 <code>height=0</code> , 则只考虑宽度。在这些情况下, 将分别相对于 (x, y) 到 $(x, y+height)$ 的垂直直线或 (x, y) 到 $(x+width, y)$ 的水平直线放置文本。默认值: <code>{0 0}</code>
charref	(布尔值) 如果为 <code>true</code> , 将启用数字和字符实体引用与字形名称引用的替换。 默认值: 全局 <code>charref</code> 参数
charspacing	(浮点值或百分比) 字符间距 (参见表 3.8)。其值为字体大小的百分比。 <code>PDF_add/create_textflow()</code> 的默认值 ¹ : <code>o</code>
dasharray	(两个浮点值的列表) 描边 (轮廓) 文本和修饰的虚线和间隙长度。 <code>PDF_add/create_textflow()</code> 的默认值 ¹ : <code>{0 0}</code> (即实线)
encoding	(字符串; 必须与 <code>fontname</code> 选项一起使用; 如果指定了 <code>font</code> , 将被忽略) 编码的名称
errorpolicy	(关键字) 控制出现错误时的行为 (参见第 28 页上的第 2.6 节 “异常处理”)
escape-sequence	(布尔值) 若为 <code>true</code> , 将在内容字符串、超文本字符串和名称字符串中启用转义序列的替代。 默认值: 全局 <code>escapesequences</code> 参数
fakebold	如果为 <code>true</code> , 将通过压印三次模拟粗体。强烈建议使用粗体变体突出文本, 此选项将创建效果次于真实粗体文本的文本输出, 并可能禁止文本提取。默认值 ¹ : <code>false</code>
fillcolor	(颜色) 填充文本的颜色。 <code>PDF_fit_textline()</code> 的默认值: 当前图形状态的对应参数。 <code>PDF_add/create_textflow()</code> 的默认值: <code>{gray 0}</code>
fitmethod	(关键字) 指定使文本填充指定框的方法。如果尚未有指定框, 将忽略此选项。默认值: <code>nofit</code> 。 nofit 只确定文本位置, 不进行任何缩放或剪裁处理。 clip 确定文本位置, 并在框的边缘剪裁文本。 meet 根据 <code>position</code> 选项确定文本位置, 并在保持长宽比的同时进行缩放以使其完全装入框内。通常, 至少有文本的两个边缘与框的对应边缘对齐。 auto 这种方法会自动尝试将文本完全置于框内。具体讲: 如果文本可以填充于框内, 则相当于 <code>nofit</code> 。否则, 将计算缩放因子, 以使文本填入框内。如果此因子大于 <code>shrinklimit</code> 选项, 将扭曲文本以填充框; 否则, 将应用 <code>meet</code> 方法。 slice 根据 <code>position</code> 选项确定文本位置, 并进行缩放以使其覆盖整个框, 可在保持长宽比的同时确保至少文本的一个尺寸可完全包含在框内。通常, 文本的某些其他尺寸会延伸到框外, 因此, 将被剪裁。 entire 根据 <code>position</code> 选项确定文本的位置, 并进行缩放以使其覆盖整个框。通常, 这种方法会扭曲文本。 <code>scale</code> 选项将被忽略。
字体	(字体句柄) <code>PDF_load_font()</code> 返回的字体句柄。默认值: 当前字体
fontname	(名称字符串; 必须与 <code>encoding</code> 选项一起使用; 如果指定了 <code>font</code> , 将被忽略) 字体的名称
fontsize	(浮点型、百分比或选项列表; 如提供了 <code>font</code> 选项, 则要求此选项) 字体的大小, 以当前用户坐标系统的单位表示。在 <code>PDF_fit_textline()</code> 百分比与框的宽度 (对于 <code>orientate=north</code> 和 <code>south</code>) 或高度 (对于 <code>orientate=east</code> 和 <code>west</code>) 有关。文本流: 百分比与先前文本的大小有关。默认值: 当前字体大小。 如果提供了选项列表, 则它必须包含关键字和数字。关键字描述所需字体规格, 数字包含所需的大小值: ascender 此数字将被解释为上伸高度。 bodyheight 此数字将被解释为基线之间的最小距离, 即如果将此值用作行距, 则相邻行的下伸和上伸部分将恰好相接触。如果未提供关键字, 这将是默认行为。 capheight 此数字将被解释为大写字母的高度。 xheight 此数字将被解释为小写字母的高度。

表 4.1 `PDF_fit_textline()` 和 `PDF_info_textline()` 的选项（续）

关键字	说明
glyphcheck	（关键字）指定字形检查策略：文本中包含不能被映射到选定字体中的字形时将发生的情况。 默认值 ¹ ： replace 。 none 不检查 error 字形不可用时将引发异常。可以使用 <code>PDF_get_errmsg()</code> 检索详细的错误消息。 replace <i>PDFlib</i> 将尝试使用适当的替换字形替换不可用的字形；将分解连字。如果没有适当的替换字形，将使用 <code>replacementchar</code> 替换该字形。
glyphwarning	不常用，使用 <code>errorpolicy</code>
horizscaling	（浮点值或百分比；不能为 0）水平文本缩放（参见表 3.8）。 <i>PDF_add/create_textflow()</i> 的默认值 ¹ ： 100
italicangle	（浮点型）指定文本的斜体（倾斜）角度的度数。 <i>PDF_add/create_textflow()</i> 的默认值 ¹ ： 0
kerning	（布尔值）字距调整行为（参见表 3.7）。默认值：全局 <code>kerning</code> 参数。
leader	（关键字；若未指定 <code>boxsize</code> ，将被忽略）指定将在文本框的边框和文本之间重复插入的填充文本（默认值：无前导字符）： alignment （一个或两个关键字）第一个关键字指定限定框的左边框与文本行之前的前导字符的对齐方式；第二个关键字指定文本行与限定框的右边框之间的前导字符的对齐方式。如果仅指定一个关键字，它将被用作文本行与限定框右边框之间的前导字符。支持的关键字（默认值： <code>{none grid}</code> ）： center 前导字符在文本行和限定框的边框之间分散对齐。 grid <i>PDFlib</i> 将前导文本的位置与文本行的左侧或右侧下一个前导文本半宽度的整数倍位置对齐。这可能会导致文本行与前导文本之间产生最大跨 50% 前导文本宽度的间距。 justify 通过适当的 <code>charspacing</code> ，将前导字符在文本行和限定框的边框之间分散对齐。 left 前导字符起始于限定框的左边框或文本行的末端。 none 无前导字符 right 前导字符终止于文本行的开端或限定框的右边框。 encoding （字符串；必须与 <code>fontname</code> 一起使用；如果指定了 <code>font</code> ，将被忽略）编码的名称 fillcolor （颜色）前导字符的颜色。默认值：文本行的颜色 font （字体句柄）用于前导字符的字体句柄。默认值：文本行的字体 fontname （名称字符串；必须与 <code>encoding</code> 一起使用；如果指定了 <code>font</code> ，将被忽略）用于前导字符的字体的名称 fontsize （浮点值或选项列表）前导字符的大小。默认值：文本行的字体大小 text （内容字符串）用于前导字符的文本。默认值： <code>U+002E'.'</code> （句点） yposition （浮点值或关键字）根据基线或通过关键字 <code>fontsize</code> 、 <code>ascender</code> 、 <code>xheight</code> 、 <code>baseline</code> 、 <code>descender</code> 、 <code>textrise</code> 之一指定前导字符的垂直位置。默认值： <code>baseline</code> 另外，可以提供 <i>PDF_load_font()</i> 中的所有选项。
locallink	不常用；使用匹配框功能在文本中创建链接（参见第 70 页上的第 4.4 节“匹配框”）
margin	（浮点值列表）一个或两个浮点值，描述文本框的额外水平和垂直扩展。默认值： 0
匹配框	（选项列表）根据表 4.13，包含匹配框详细信息的选项列表
orientate	（关键字）指定放置文本时所需的文本方向。默认值： north 。 north 向上 east 指向右侧 south 向下 west 指向左侧

表 4.1 PDF_fit_textline() 和 PDF_info_textline() 的选项（续）

关键字	说明
overline	（布尔值）上划线模式（参见表 3.7）。PDF_add/create_textflow() 的默认值 ¹ : false
position	（浮点值或关键字列表）对齐控制：一个或两个值，指定文本定界框内的参考点位置 (x,y)；{0 0} 表示位于文本框左下角，{100 100} 表示位于右上角。如果已经指定了 boxsize 选项，则 position 选项还可指定目标框的位置。这些值表示为文本宽度和高度的百分比。如果两个百分比相等，指定一个浮点值即可。 关键字 left、center、right (x 方向) 或者 bottom、center、top (y 方向) 可用来替代 0、50 和 100。如果只指定了一个关键字，将会添加另一方向对应的关键字。默认值：{left bottom}。示例： {0 50} 或 {left center} 左对齐文本 {50 50} 或 {center} 令文本居中对齐 {100 50} or {right center} 令文本右对齐
rotate	（浮点型）旋转坐标系，将中心用作参考点并将指定的值用作以度数表示的旋转角度。这将导致框和文本被旋转。一旦文本已放置，旋转将重置。默认值：0
showborder	（布尔值）如果为 true，将对限定框的边框做描边处理（使用当前图形状态）。如果创建了图章，也将对图章的边框做描边处理。这可能对开发和调试很有用。默认值：false
shrinklimit	（浮点值或百分比）收缩因子的下限，将用来调整 fitmethod=auto 的文本。默认值：0.75
stamp	（关键字；若未指定 boxsize，将被忽略）此选项可以用于在 boxsize 选项指定的框中创建对角图章。包含此图章的文本将尽可能的大。在框中放置图章文本时，将遵守选项 position、fitmethod 和 orientate（仅 north 和 south）。默认值：none。 llzur 图章将沿对角线从左下角延伸到右上角。 ulzlr 图章将沿对角线从左上角延伸到右下角。 none 不创建图章。
strikeout	（布尔值）删除线模式（参见表 3.7）。PDF_add/create_textflow() 的默认值 ¹ : false
strokecolor	（颜色）文本描边色。PDF_add/create_textflow() 的默认值 ¹ : {gray 0}
strokewidth	（浮点值、百分比或关键字）描边文本的线宽度（绝对值或相对于 fontsize）。关键字 auto 指定字体的 underlinewidth 值。默认值：auto
textformat	（关键字；仅用于不兼容 Unicode 的语言绑定）用来解释提供的文本的格式。默认值：全局 textformat 参数。
textrendering	（整型）文本渲染模式（参见表 3.8）。PDF_add/create_textflow() 的默认值 ¹ : 0
textrise	（浮点值或百分比）文本提升（参见表 3.8）。其值为字体大小的百分比。 PDF_add/create_textflow() 的默认值 ¹ : 0
underline	（布尔值）下划线模式（参见表 3.7）。PDF_add/create_textflow() 的默认值 ¹ : false
underline-position	（浮点型、百分比或关键字）下划线文本的描边线相对于基线的位置（绝对值或相对于 fontsize 的值；典型值为 -10%）。关键字 auto 指定将从字体规格或轮廓文件返回的字体特定值。默认值：auto
underline-width	（浮点型、百分比或关键字）下划线文本的线宽（绝对值或相对于 fontsize 的值；典型值为 5%）。关键字 auto 或值 0 指定将从字体规格或轮廓文件返回的字体特定值。默认值：auto
weblink	不常用，使用 PDF_fit_textline() 和 PDF_create_annotation() 中的 matchbox 选项在文本中创建链接（参见第 70 页上的第 4.4 节“匹配框”）。
wordspacing	（浮点值或百分比）字间距（参见表 3.8）。其值为字体大小的百分比。PDF_add/create_textflow() 的默认值 ¹ : 0
xadvancelist	（浮点值列表）使用用户坐标，指定文本中所有字形的预置宽度。列表的长度必须小于或等于文本中的字形数。将使用 xadvance 值替代标准字形宽度。字距微调 and 字符间距等其他效果不受影响。

1. PDF_fit_textline() 的默认值：当前图形状态的对应参数

len （仅用于 C 语言绑定）以字节为单位的文本长度，以 null 终结的字符串的长度为 0。

keyword 根据表 4.2 指定请求的信息的关键字。

optlist 选项列表，根据表 4.1 指定文本行选项。将忽略与文本的大小计算无关的选项，而不返回任何信息。

返回值 *keyword* 请求的、某些文本规格值的值。

详细说明 此函数将根据提供的选项执行放置文本所需要的所有计算，但不在页面上实际创建任何输出。假定文本参考位置为 {0 0}。

范围 任意对象除外。

表 4.2 PDF_info_textline() 的关键字

keyword	说明
angle	基线的旋转角度，即文本的旋转角度
ascender capheight descender	对应的排版值，以用户坐标为单位
endx, endy	文本结束位置的 <i>x/y</i> 坐标，以用户坐标系统为单位
height	根据匹配框的 <i>boxheight</i> 规范的文本字符串的高度
perpendiculardir	垂直 <i>writingdir</i> 的单位矢量：对于标准水平文本，将为 (0, 1)，对于直排文本，将为 (1, 0)
scalex, scaley	水平书写和垂直缩放因子。如果这些关键字的值不等于 1，将缩放文本以填入文本框。
startx, starty	文本起始位置的 <i>x/y</i> 坐标，以用户坐标系统为单位
unmappedglyphs	文本中无法映射的字形的数目
width	文本字符串的宽度（水平书写模式）或字形的最大宽度（垂直书写模式）
writingdirx writingdiry	书写方向（内嵌文本排放的方向）的 <i>x/y</i> 坐标，即从 (startx, starty) 到 (endx, endy) 的单位适量。对于标准水平文本，将为 (1, 0)，对于直排文本，将为 (0, -1)
xheight	对应的排版值，以用户坐标为单位

4.2 使用文本流的多行文本

C++ Java

int add_textflow(int textflow, String text, String optlist)

Perl PHP

int PDF_add_textflow(resource p, int textflow, string text, string optlist)

C

int PDF_add_textflow(PDF *p, int textflow, const char *text, int len, const char *optlist)

创建一个 `Textflow` 对象，可向现有 `Textflow` 添加文本和显式选项。

textflow 之前调用 `PDF_create_textflow()` 或 `PDF_add_textflow()` 返回的 `Textflow` 句柄或 -1（PHP 中：o）以创建新的 `Textflow`。

文本 （内容字符串）文本流的内容。文本中不能包含任何内嵌选项。

len （仅用于 C 语言绑定）以字节为单位的文本长度，以 `null` 终结的字符串的长度为 o。

optlist 选项列表，根据表 4.1 和表 4.3 指定文本流选项。可以使用下列选项：

- 常规： *errorpolicy*
- 字体相关的选项： *encoding*、字体、 *fontname*、 *fontsize*
fontname 和 *encoding* （对应于 `PDF_load_font()` 中的同名参数）选项都可以用来选择字体。此外， *font* 选项也可用于提供之前使用的 `PDF_load_font()` 调用创建的字体句柄。如果指定了 *font*，将忽略 *fontname* 和 *encoding* 选项。要求使用 *fontsize* 选项。如果将 *fontsize* 指定为百分比，它将被解释成先前的 *fontsize* 的百分比；初始值为 1。
- `PDF_load_font()` 的所有选项（参见表 3.4）。仅当同时提供了 *fontname* 和 *encoding* 选项，这些选项才可用（*font* 选项除外）：
ascender、 *autocidfont*、 *auto-subsetting*、 *capheight*、 *descender*、 *embedding*、 *fontstyle*、 *keepnative*、 *kerling*、 *linegap*、 *metadata*、 *monospace*、 *replace-mentchar*、 *subsetlimit*、 *subsetminsize*、子集化、 *unicodemap*、 *vertical*、 *xheight*
- `PDF_fit_textline()` 的下列外观选项（参见表 4.1）：
charref、 *charspacing*、 *dasharray*、 *escape-sequence*、 *fakebold*、 *fillcolor*、字体、 *fontsize*、 *glyphcheck*、 *horizscaling*、 *italicangle*、 *kerling*、匹配框、 *overline*、 *strikeout*、 *strokecolor*、 *strokewidth*、 *textformat*、 *textrendering*、 *textrise*、 *underline*、 *underline-position*、 *underline-width*、 *wordspacing*
- 文本文义： *charclass*、 *charmapping*、 *hyphenchar*、 *tabalignchar*
- 文本格式： *alignment*、 *avoid-emptybegin*、 *fixedleading*、 *hortabsize*、 *hortabmethod*、 *lastalignment*、 *leader*、 *leading*、 *leftindent*、 *minlinecount*、 *parindent*、 *rightindent*、 *ruler*、 *tabalignment*
- 控制换行算法的选项： *adjustmethod*、 *avoidbreak*、 *maxspacing*、 *minspacing*、 *nofitlimit*、 *shrinklimit*、 *spreadlimit*
- 用作命令的选项： *comment*、 *mark*、 *nextline*、 *nextparagraph*、 *resetfont*、 *return*、 *space*

返回值 文本流句柄，可在 `PDF_add_textflow()`、 `PDF_fit_textflow()`、 `PDF_info_textflow()`、和 `PDF_delete_textflow()` 调用中使用。在关闭文档范围结束和用该句柄调用 `PDF_delete_textflow()` 之前，句柄将一直有效。

如果 *textflow* 参数为 -1，将创建新的 `Textflow`，并将返回对应的句柄。否则将返回 *textflow* 参数中提供的句柄。默认情况下，如果出现错误，函数将返回 -1（在 PHP 中：o）。不过，可以使用 *errorpolicy* 参数或选项更改此行为。如果出现错误，在后续函数调用中不能再使用 *textflow* 参数中提供的句柄（如果值不等于 -1， `PDF_delete_textflow()` 中除外）。

详细说明 此函数处理提供的文本并据此创建内部数据结构。它决定稍后将由文本格式器使用的文本部分（例如单词），处理内嵌选项列表，在可能的情况下将文本转换为 Unicode 字符串，确定潜在的换行符并基于字体和文本选项计算文本部分的宽度。

与期望在一次调用返回所有文本内容和选项的 `PDF_create_textflow()` 相反，此函数可用于在单独的调用中为文本流提供文本内容和选项。它将提供的 `text` 和 `optlist` 添加到新的或现有的 `Textflow` 中。在处理 `text` 之前，将计算 `optlist` 中指定的选项。`text` 和 `optlist` 都可以为空。

如果 `textflow=-1`，则此函数几乎等同于 `PDF_create_textflow()`。然而，此函数与 `PDF_create_textflow()` 不同，它不搜索 `text` 中的内嵌选项。因此不必重定义内嵌选项列表的起始字符或使用内嵌选项指定文本的长度（甚至非 Unicode 文本和 UTF-16 文本也不必）。

此函数不会在生成的 PDF 文档中创建任何输出，只准备文本。使用 `PDF_fit_textflow()`，用预处理的文本流句柄创建输出。

缺省情况下，将用 Unicode 兼容字体中的 U+000B (VT)、U+2028 (LS)、U+000A (LF)、U+000D (CR)、CRLF、U+0085 (NEL)、U+2029 (PS) 和 U+000C (FF) 字符强制生成新行。除 VT 和 LS 之外，所有这些字符都可强制生成新段落（这表示，`parindent` 选项将有效）。FF 立即停止使文本填充限定框的操作（将退出 `PDF_fit_textflow()` 函数，并返回字符串 `_nextpage`）。

水平制表符 (HT) 设置后续文本的新起点。具体由 `hortabmethod` 和 `hortabsize` 选项控制。

如果自动连字符后是一个换行符，将用 `hyphenchar` 选项中指定的字符替换自动连字符 (SHY)。

不支持直排书写模式。

范围 任意，对象除外

表 4.3 用于 `PDF_create_textflow()`、`PDF_add_textflow()` 的选项和 `PDF_create_textflow()` 中的内嵌选项

选项	说明
<code>adjustmethod</code>	（关键字）当使用大于等于 <code>minspacing</code> 并小于等于 <code>maxspacing</code> 的单词距离选项进行增加或缩减单词间距离的调整后，仍有部分文本不能排入同一文本行时，所采用的方法。默认值： <code>auto</code> 。
<code>auto</code>	按顺序使用以下方法： <code>shrink</code> 、 <code>spread</code> 、 <code>nofit</code> 、 <code>split</code> 。
<code>clip</code>	类似于 <code>nofit</code> ，除了将剪切定界框（考虑 <code>rightindent</code> 选项）右边缘的长的部分以外。
<code>nofit</code>	末尾单词将移动到下一行，以使剩余（短）行将不会少于 <code>nofitlimit</code> 选项中指定的百分比。即使对齐的段落也可能看起来有点不规则。
<code>shrink</code>	若单词不能完全排入当前行，将缩减文本直至 <code>shrinklimit</code> 。若仍不能满足要求，则将使用 <code>nofit</code> 方法。
<code>split</code>	末尾单词将不会移动到下一行，而是被强制在文本框内的最后一个字母断字。若字体为文本，而非符号或 <code>hyphenchar=none</code> ，则插入连字符。
<code>spread</code>	末尾字将移动到下一行，并将根据 <code>spreadlimit</code> 通过增加词中字符之间的距离对齐剩余（短）行。若仍无法实现对齐，则将使用 <code>nofit</code> 方法。
<code>alignment</code>	（关键字）指定文本行在段落中的对齐方式。默认值： <code>left</code> 。
<code>left</code>	起于 <code>leftindent+parindent</code> （对于段落的第一行）和 <code>leftindent</code> （所有其他行），向左对齐
<code>center</code>	以 <code>leftindent</code> 和 <code>rightindent</code> 之间的中心对齐
<code>right</code>	向右对齐，止于 <code>rightindent</code>
<code>justify</code>	左右两边对齐
<code>avoidbreak</code>¹	（布尔值）若为 <code>true</code> ，则避免任何换行直到 <code>avoidbreak</code> 重置为 <code>false</code> 。换言之，将按非 Unicode 兼容字体处理文本。拆分过程不受此选项影响。默认值： <code>false</code>
<code>avoid-emptybegin</code>	（布尔值）如果为 <code>true</code> ，将删除限定框开头的空行。默认值： <code>false</code>

表 4.3 用于 `PDF_create_textflow()`、`PDF_add_textflow()` 的选项和 `PDF_create_textflow()` 中的内嵌选项（续）

选项	说明
charclass¹	（成对列表，其中，每对的第一元素是一个关键字，第二个元素是一个 <i>unichar</i> 或一组 <i>unichar</i> 的列表）指定的 <i>unichar</i> 将按指定的关键字分类，以确定这些字符的换行行为： letter 行为如同一个字母（例如 <i>B</i> ） punct 行为如同一个标点符号（例如 <i>+ / ; :</i> ） open 行为如同一个左括号（例如 <i>[</i> ） close 行为如同一个右括号（例如 <i>]</i> ） default 将所有字符类重置为 <i>PDFlib</i> 的内置默认值 示例： <code>charclass={ close » open « letter={ / := } punct & }</code>
charmapping¹	（成对列表，每对包含两个单字符或一个单字符和一个单字符和整型的列表）使用另一字符的一个或多个实例替换各字符。选项列表包含一对或多对单字符。每对中的第一个字符将由第二个字符替代。每对中的第二个元素可以是包含一个单字符和一个计数的选项列表，而非一一对应： count > 0 此计数指示字符替代将重复的次数。 count < 0 多个字符示例的序列将缩减到指定数字的绝对值。 count = 0 将删除字符。 示例： <code>charmapping={ hortab space CRLF space LF space CR space }</code> <code>charmapping={ shy {shy 0} }</code> <code>charmapping={ hortab {space 4} }</code>
comment	（字符串）将被忽略的任意文本，对注释选项列表或宏很有用
encoding	（字符串；必须与 fontname 选项一起使用；如果指定了 font ，将被忽略）编码的名称
errorpolicy	（关键字）控制出现错误时的行为（参见第 28 页上的第 2.6 节“异常处理”）
fixedleading	（布尔值）如果值为 true ，每行的第一个行距将作为该行的 <i>leading</i> 值。否则，每行的最大行距将作为该行的 <i>leading</i> 值。默认值： false
fontname	（名称字符串；必须与 encoding 选项一起使用；如果指定了 font ，将被忽略）字体的名称；切记在包含空格字符的字体名称两边加上 { 和 }
hortabsize¹	（浮点值或百分比）水平制表键的宽度 ² 。具体解释将取决于 hortabmethod 选项。默认值： 7.5%
hortabmethod¹	（关键字）文本中的水平制表键的处理。若计算的位置位于当前文本位置的左侧，则将忽略制表键。默认值： relative 。 relative 位置将按 hortabsize 中指定的数量前进。 typewriter 位置将移到下一个 hortabsize 的倍数的位置。 ruler 位置将前进到 ruler 选项中的第 <i>n</i> 个制表键值，此处 <i>n</i> 为目前在行中找到的选项卡数量。若 <i>n</i> 大于制表键的数量，则将使用相关方法。
hyphenchar¹	（ <i>Unichar</i> 或关键字）在换行符处替代自动连字符的字符。值 o 和关键字 none 将完全禁用连字符。默认值：如果自动连字符 (<i>U+00AD</i>) 存在于字体中，缺省值为 <i>U+00AD</i> ；否则，为 <i>U+002D</i> （连字符减号）。
lastalignment	（关键字）排版段落中的最后一行。支持所有 alignment 选项的关键字，外加以下（默认值： auto ）： auto 若 alignment 选项值不是 justify ，使用该方式排版。在后者情况下，将使用 left 方式。
leader	（选项列表）指定将重复插入的填充文本（例如圆点前导字符）。将插入前导字符直至下一个制表键，或者若制表键不可用，直至行尾。前导字符从不跨多行（默认值：无前导字符）： alignment （关键字）前导字符对齐方式（默认值： grid ）： center 前导字符在上一文本片段（或者若无文本，则为行首）和制表键（或者若无制表键，则为行尾）之间居中对齐。 grid <i>PDFlib</i> 将第一个前导字符的位置与下一个前导文本半宽度的整数倍位置对齐。这可能会导致文本与前导字符之间产生最大 50% 前导字符宽度的间距。 justify 前导字符在上一文本片段（或者若无文本，则为行首）和制表键（或者若无制表键，则为行尾）之间通过应用适当的 charspacing 分散对齐。 left 前导字符紧接上一文本片段（或者若无文本，则为行首）。 right 前导字符终止于紧接制表键之前的位置（或者若无制表键，则为行尾）。 同时还支持 <i>PDF_fit_textline()</i> 中 leader 选项的下列子选项（参见表 4.1）： encoding 、 fillcolor 、 font 、 fontname 、 fontsize 、 text 、 yposition 。

表 4.3 用于 `PDF_create_textflow()`、`PDF_add_textflow()` 的选项和 `PDF_create_textflow()` 中的内嵌选项（续）

选项	说明
leading	（浮点值或百分比）相邻文本基线间的距离 ³ 。实际值将按以下方式确定：如果行的开头有选项列表，将根据最后一个相关选项（ <code>font</code> 、 <code>fontsize</code> 、 <code>leading</code> 等）确定行距。如果同一行内还有其他选项列表，只有当 <code>fixedleading=false</code> 时，才考虑其任何与行距相关的选项。如果行内没有选项列表，将考虑前一个行距值。默认值： <code>100%</code>
leftindent	（浮点值或百分比）文本行的左缩进 ² 。若在行内指定 <code>leftindent</code> ，并且确定的位置在当前文本位置的左侧，则将为此行忽略此选项。默认值： <code>0</code>
mark	（整型）作为标记，在内部存储提供的值。稍后，可使用 <code>PDF_info_textflow()</code> 调用最近一次存储的标记。这可能会对确定文本的哪些部分已置于页面很有用。
matchbox	（选项列表）根据表 4.13，包含匹配框详细信息的选项列表
maxspacing¹ minspacing¹	（浮点值或百分比）单词间最大或最小的距离（以用户坐标或间隔字符的宽度的百分比为单位）。计算的单词间距受提供的值的限制（但是仍将添加 <code>wordspacing</code> 选项）。默认值： <code>minspacing=50%</code> ， <code>maxspacing=500%</code>
minlinecount	（整型）限定框中最后一个段落的最小行数。如果行数小于该值，将置于下一限定框。值 <code>2</code> 可用来防止在限定框结尾处出现单行（» 孤行 «）。默认值： <code>1</code>
nextline nextparagraph	（布尔值）强制生成新行或新段落；即使使用不兼容 <code>Unicode</code> 的字体，也是如此。
nofitlimit	（浮点值或百分比）使用 <code>nofit</code> 方法 ² 的行的长度下限。默认值： <code>75%</code>
parindent	（浮点值或百分比）段落第一行的左缩排 ² 。该数值将追加到 <code>leftindent</code> 上。在一行中指定这一选项，其行为就如制表键。默认值： <code>0</code>
resetfont	（布尔值）将 <code>font</code> 和 <code>fontsize</code> 选项重设为它们以前的值。这可能会对插入操作之后重设字体很有用，例如斜体文本。 <code>font</code> 选项优先于此选项。只有在第二次设置任何字体相关参数后，才能使用此命令；否则，此命令将被忽略。
return	（字符串；不必以下划线字符_开头）使用提供的字符串作为返回值退出 <code>PDF_create_textflow()</code> 或 <code>PDF_add_textflow()</code> 。
rightindent	（浮点值或百分比）所有文本行的右缩排 ² 。默认值： <code>0</code>
ruler¹	（浮点值或百分比列表）当 <code>hortabmethod=ruler</code> 时，制表键绝对值列表 ² 。这个列表可包含多至 <code>32</code> 个非负数的值，以上升秩序排列。默认值： <code>hortabsize</code> 的整数倍数
shrinklimit	（百分比）对于 <code>adjustmethod=shrink</code> 方式，可缩减文本的下限；计算的收缩因素受限于提供的值，但是将乘以 <code>horizscaling</code> 选项。默认值： <code>85%</code>
space	（浮点或百分比）文本位置将向前移动提供的值 ³ 。这同样适用于不兼容 <code>Unicode</code> 的字体。
spreadlimit¹	（浮点或百分比） <code>spread</code> 方法中，两个字符之间的距离上限 ³ ；计算的值将与 <code>charspacing</code> 选项的值相加。默认值： <code>0</code>
tabalignchar¹	（ <code>Unichar</code> ）将在此对齐小数制表符的字符。默认值： <code>U+002E''</code>
tabalignment¹	（关键字列表）制表键对齐方式。列表中的每个项都定义 <code>ruler</code> 选项中的相应的项的对齐方式。 默认值： <code>left</code> 。 center 制表键内文本向中对齐。 decimal 第一个 <code>tabalignchar</code> 示例将在制表键位置左对齐。如果找不到 <code>tabalignchar</code> ，则将采用向右对齐。 left 制表键内文本向左对齐。 right 制表键内文本向右对齐。
textwarning	不常用，使用 <code>errorpolicy</code>

1. 此选项不影响使用不兼容 `Unicode` 的字体的文本

2. 使用用户坐标，或限定框宽度的百分比

3. 使用用户坐标，或字体大小的百分比

文本流的宏选项 文本流的选项列表（无论是在 `PDF_create_textflow()` 或 `PDF_add_textflow()` 的 `optlist` 参数中，或是提供给 `PDF_create_textflow()` 的文本中的内嵌行）可以包含宏定义和宏调用；具体规则，参见表 4.4。宏可用于集中定义多次使用的选项值，如字体名称、缩进量等。解析选项列表之前，将用宏定义中提供的对应选项列表的内容替代其中包含的每个宏。然后，会解析由此生成的选项列表。以下是两个宏的宏定义示例：

```
<macro {
  comment { The following macros are used as paragraph styles }
  H1 {fontname=Helvetica-Bold encoding=winansi fontsize=14 }
  body {fontname=Helvetica encoding=winansi fontsize=12 }
}>
```

这些宏在选项列表中的使用方式如下：

```
<&H1>Chapter 1
<&body>This chapter talks about...
```

宏的定义和使用应遵守下列规则：

- 宏的嵌套可达到任意深度（宏定义可以包含对其他宏的调用）
- 宏不能在定义他们的选项列表中使用。在 `PDF_create_textflow()` 中，可以在定义宏的内嵌选项列表结束之后，立即开始使用该宏的新内嵌选项列表。在使用 `PDF_add_textflow()` 时，需要用函数调用定义宏，另一个使用宏（原因是 `PDF_add_textflow()` 一次只接受一个选项列表）。
- 宏名称不区分大小写。
- 未定义的宏将导致异常。
- 可随时重新定义宏。

表 4.4 `PDF_add_textflow()`、`PDF_create_textflow()` 和 `fit_textflow()` 的选项列表宏定义和宏调用

选项	说明
comment	（字符串）将被忽略的任意文本，对宏注释很有用
macro	（成对列表）每一对都分别描述一个宏的名称和定义；具体如下： name （字符串）稍后可用于宏调用的宏名称。稍后可重新定义已经定义的宏。将忽略特殊名称注释。 suboptlist 调用宏时，将实际替代宏名称的选项列表。将忽略前导空白和尾随空白。
&name	将扩展具有指定名称的宏，并用宏的内容替代宏名称（包括 & 字符），即：已经为宏定义的子选项列表（两端没有大括号）。宏名称以空白、{、}、= 或 & 终结。因此，不能将这些字符用作宏名称的一部分。 将扩展嵌套的宏，且没有任何嵌套限制。同时，还将扩展字符串选项中的宏。宏替代必须生成有效的选项列表。

C++ Java	<code>int create_textflow(String text, String optlist)</code>
Perl PHP	<code>int PDF_create_textflow(resource p, string text, string optlist)</code>
C	<code>int PDF_create_textflow(PDF *p, const char *text, int len, const char *optlist)</code>

从文本内容、嵌入选项和显式选项创建一个 `Textflow` 对象。

文本 （内容字符串）文本流的内容。它可能包含表 4.3 和表 4.5 中列出的各种编码、宏（参见第 56 页上的“文本流的宏选项”）以及内嵌选项列表所产生的文本（另请参见第 58 页上的“文本流的内嵌选项列表”）。不过，如果 `text` 为空字符串，将返回有效的文本流句柄。

len （仅用于 C 语言绑定）以字节为单位的文本长度，以 `null` 终结的字符串的长度为 0。

optlist 指定 `Textflow` 选项的选项列表。将在 `text` 中的内嵌选项列表之前计算 `optlist` 参数中指定的选项，以使内嵌选项优先于 `optlist` 参数中提供的选项。可以使用下列选项：

- ▶ `PDF_add_textflow()` 的所有选项 （参见 `PDF_add_textflow()` 的选项列表和表 4.3）
- ▶ 控制嵌入选项列表的处理的选项，参见表 4.5：
`begoptlistchar`、`endoptlistchar`、`fixedtextformat`、`textlen`

返回值 文本流句柄，可在 `PDF_add_textflow()`、`PDF_fit_textflow()`、`PDF_info_textflow()`、和 `PDF_delete_textflow()` 调用中使用。在关闭文档范围结束和用该句柄调用 `PDF_delete_textflow()` 之前，句柄将一直有效。默认情况下，如果出现错误，此函数返回 -1（在 PHP 中：0）。不过，可以使用 `errorpolicy` 参数或选项更改此行为。

详细说明 此函数接受将为文本流准备的选项和文本。与 `PDF_add_textflow()` 不同，文本中可以包含任何内嵌选项。可应用 `optlist` 参数中的 `textlen` 选项，对部分或全部文本停用内嵌选项列表搜索（第 58 页上的“文本流的内嵌选项列表”）。

此函数不会在生成的 PDF 文档中创建任何输出，只根据提供的选项准备文本。使用 `PDF_fit_textflow()`，用得到的文本流句柄创建输出。

有关特殊字符、断行等的详细信息，请参见 `PDF_add_textflow()` 的 *Details* 一节。

范围 任意，对象除外

表 4.5 用于 `PDF_create_textflow()` 中内嵌选项列表处理的附加选项

选项	说明
<code>begoptlistchar</code>	（单字符或关键字）内嵌选项列表开头使用的字符。如果文本本身出现了缺省字符，则替换该字符会很有用（参见第 58 页上的“文本流的内嵌选项列表”）。如果未指定 <code>textlen</code> ，将按照与前面的文本相同的文本格式和编码对文本中的 <code>begoptlistchar</code> 字符进行编码。这表示必须选择 <code>begoptlistchar</code> 的 <i>Unicode</i> 值，以使其包含在前面的文本的编码中。可使用关键字 <code>none</code> 完全停用搜索选项列表的操作。默认值： <code>U+003C (<)</code>
<code>endoptlistchar</code>	（单字符；不允许使用 <code>U+007D “}”</code> ）终止内嵌选项列表的字符。默认值： <code>U+003F (>)</code>
<code>fixedtextformat</code>	（布尔型；在识别 <i>Unicode</i> 的语言绑定中将被忽略；此选项中内嵌选项列表中无意义，只能在 <code>optlist</code> 参数中使用）如果为 <code>true</code> ，所有的文本片段和内嵌选项列表将使用相同的 <code>textformat</code> （必须为下列值之一： <code>utf8</code> 、 <code>utf16</code> 、 <code>utf16be</code> 或 <code>utf16le</code> ）。如果文本和内嵌选项来源相同，这可能很有用。 如果为 <code>false</code> ，无论实际文本中使用何种格式，包括分隔符的内嵌选项列表必须用 <code>textformat=bytes</code> 编码。例如，这允许使用 <i>UTF-16</i> 文本与 <i>ASCII</i> 编码的内嵌选项列表的组合（文本可以来自一个 <i>Unicode</i> 数据库，而内嵌选项被构造成应用程序内的 <i>ASCII</i> 文本）。默认值： <code>false</code>
<code>textlen</code>	（整型或关键字；使用不兼容 <i>Unicode</i> 的字体的文本或者使用不识别 <i>Unicode</i> 语言且 <code>fixedtextformat=false</code> 、 <code>textformat=utf16xx</code> 的文本片段要求此选项）字节数或（在识别 <i>Unicode</i> 的语言中）下一内嵌选项列表之前的字符数（参见第 58 页上的“文本流的内嵌选项列表”）。字符数在解析字符引用之前计算，例如 <code><textlen=8>&#x2460;<...></code> 。关键字 <code>all</code> 指定全部剩余文本。默认值：将搜索文本中的下一个 <code>begoptlistchar</code> 。

文本流的内嵌选项列表 `PDF_create_textflow()`（而非 `PDF_add_textflow()`）的 `text` 参数中提供的内容可以包括任意数量的选项列表（内嵌选项），根据表 4.3 指定文本流选项。另外，也可以在 `PDF_create_textflow()` 和 `PDF_add_textflow()` 的 `optlist` 参数中提供所有这些选项。在一个选项列表中，同一选项可指定多次；这时，将只考虑最后一次指定的选项。

必须使用 `begoptlistchar` 和 `endoptlistchar` 选项中指定的字符围绕内嵌选项列表（默认值：`<` 和 `>`）。很显然，如果实际文本中必须使用用来开始内嵌选项列表的字符，可能会引发冲突。有几种方法可以解决这个冲突；具体取决于文本是否包含任何内嵌选项列表。切记 `PDF_add_textflow()` 完全分隔文本和选项，因此这里不会发生冲突。

如果文本不包含任何内嵌选项列表，可以通过下列一种方法完全停用搜索内嵌选项列表的操作：

- ▶ 在 `PDF_create_textflow()` 的 `optlist` 参数中，设置 `begoptlistchar=none`。
- ▶ 将 `PDF_create_textflow()` 的 `optlist` 参数中的 `textlen` 选项设为整个文本的长度。

如果文本确实包含内嵌选项列表，可通过下列一种方法避免文本内容与开始内嵌选项列表的 `begoptlistchar` 之间的冲突：

- ▶ 用对应的数字或字符实体引用（`<` 或 `<`）替换出现的所有 `<` 字符，或用文字 `<` 字符开始内嵌选项列表：

`A<B<fontname=Helvetica encoding=winansi>`

- ▶ 将 `PDF_create_textflow()` 的 `optlist` 参数中的 `begoptlistchar` 选项设为文本中没有使用的字符（例如 `$`），并使用此字符开始内嵌选项列表：

`<begoptlistchar=$>A<B$fontname=Helvetica encoding=winansi>`

- ▶ 在前一个内嵌选项列表中，使用 `textlen` 选项指定下一文本片段的长度（直至下一内嵌选项列表开始为止）：

`<textlen=3>A<B<fontname=Helvetica encoding=winansi>`

注：如果直接在前一个选项列表之后提供了另一个内嵌选项列表，将假定他们之间存在长度为 0 的文本片段。在第一个选项列表中提供 `textlen` 选项时，这一点很重要。

C++ Java	<i>String fit_textflow(int textflow, double llx, double lly, double urx, double ury, String optlist)</i>
Perl PHP	<i>string PDF_fit_textflow(resource p, int textflow, float llx, float lly, float urx, float ury, string optlist)</i>
C	<i>const char *PDF_fit_textflow(PDF *p, int textflow, double llx, double lly, double urx, double ury, const char *optlist)</i>

将文本流的下一部分格式化成 一个矩形区域。

textflow *PDF_create_textflow()* 或 *PDF_add_textflow()* 调用返回的文本流句柄。

llx, lly, urx, ury 用户坐标系统中，目标矩形（限定框）左下角和右上角的 *x* 和 *y* 坐标。也可以按相反的顺序指定左下角和右上角坐标。可以使用 *wrap* 选项填充矩形以外的形状。

optlist 选项列表，根据表 4.6 指定处理选项。可以使用下列选项：

blind、*featherlimit*、*firstlinedist*、*fitmethod*、*fontscale*、*keep*、*lastlinedist*、*linespread-limit*、*maxlines*、*minfontsize*、*orientate*、*rewind*、*rotate*、*showborder*、*showtabs*、*verticalalign*、*wrap*

返回值 指定函数返回的原因的字符串。

- ▶ *_stop*: 文本流内的所有文本均已处理完毕。
- ▶ *_nextpage*: 等候下一页（由于换页符 U+000C 导致）。处理剩余文本要求再次调用 *PDF_fit_textflow()*。
- ▶ *_boxfull*: *fitbox* 中的空间已全部用完，或框中已设置了最大行数（如通过 *maxlines* 选项指定的行数），或者已经指定了 *fitmethod=auto* 和 *minfontsize*，但文本不能完全填入文本框。处理剩余文本要求再次调用 *PDF_fit_textflow()*。
- ▶ *_boxempty*: 处理之后文本框中不包含任何文本。如果 *fitbox* 太小而不能容纳任何文本，可能会发生这种情况。为了避免无限循环，不应再对同一限定框发出 *PDF_fit_textflow()* 调用。
- ▶ 任何其他字符串：在内嵌选项列表中，提供给 *return* 命令的字符串。

如果同时由多种原因导致返回，将报告列表中的第一项（由上至下）。在下次调用此函数前，返回的字符串将一直有效。

详细说明 当前文本和图形状态不影响用此函数创建的文本输出（这不同于 *PDF_fit_textline()*）。可在 *PDF_create_textflow()* 或 *PDF_add_textflow()* 中使用 *fillcolor*、*strokecolor* 和其他外观选项（参见表 4.1）控制文本的外观。从此函数返回后，文本状态不会发生更改。但是，*textx/texty* 参数将调整到生成的文本输出的末尾（除非 *blind* 选项已设为 *true*）。

范围 页面、图案、模板、字形

表 4.6 PDF_fit_textflow() 的选项

选项	说明
blind	(布尔值) 如果值为 true ，不生成输出，但将执行所有计算，并可使用 PDF_info_textflow() 检查格式化结果。默认值: false
firstlinedist	(浮点型、百分比或关键字) 定义框顶部与第一行文本的基线之间的距离，以用户坐标或相关字体大小的百分比为单位 (用该行的第一个字体大小为单位，如果 fixedleading=true 并且该字体大小是该行中最大的)，或者用关键字。 leading 第一行的行距值；常用可触及定义框顶部的典型字符如 Å 为基准。 ascender 第一行的字体上伸部分；常用可触及定义框顶部的带较大的上伸部分的典型字符如 d 和 h 为基准。 capheight 为首行确定的大写高度值；常用可触及定义框顶部的典型的大写字母字符如 H 为基准。 xheight 第一行的行距值；常用可触及定义框顶部的典型字符如 x 为基准。 若 fixedleading=false 第一行里 leading 、 ascender 、 xheight 或 capheight 中的最大值将被使用。
fitmethod	(关键字) 指定使文本填充限定框的方法。默认值: clip auto PDF_fit_textflow() 将在隐蔽模式下，使用缩小的字体大小及其他字体相关选项 (参见 fontscale)，直至将文本完全排入限定框 (但另请参见选项 minfontsize) clip 文本将在限定框底部截断。 nofit 文本可能超出限定框底线。
fontscale	(浮点值或百分比) 将用提供的缩放因子或百分比乘以 fontsize 的值以及 leading 、 minspacing 、 maxspacing 、 spreadlimit 和 space 的绝对值 (而非百分比)。默认值: 如果 rewind=0 ，缺省值为 1 ；否则，将为与相应 PDF_fit_textflow() 调用提供的值。
lastlinedist	(浮点型、百分比或关键字；若 fitmethod=nofit ，将忽略) 最后的文本行基线与定义框底部之间的最小距离，以用户坐标或相关字体大小的百分比为单位 (用该行的第一个字体大小为单位，如果 fixedleading= true 并且该字体大小是该行中最大的)，或者使用关键字。默认值: 0 ，即：限定框的底部将用作基线，典型的字母下缘将延伸到限定框下方)。可以使用下列关键字： descender 最后一行的字体下伸值；常用可触及定义框底部的带下伸部分的典型字符如 g 和 j 为基准。 如果 fixedleading=false ，最后一行 descender 中的最大值将被使用。
linespread-limit	(浮点值或百分比；只有当 verticalalign=justify 时有效) 垂直排版行距调整最大值，以用户坐标或相关行距的百分比为单位。默认值: 200%
maxlines	(整型或关键字) 该属性值可以是整数代表定义框可容纳文本行的最大值；亦可是关键字 auto ，代表定义框所含文本行数没有限制。若已放置最大行数，则 PDF_fit_textflow() 将返回 _boxfull 。 默认值: auto
minfontsize	(浮点值或百分比) 缩小文本以填入定义框时允许使用的最小字体大小，专门用于 fitmethod=auto 的情况，使用用户坐标或定义框高度的百分比指定限制值。如果到了极限而文本仍不能填充，将返回字符串 _boxfull 。默认值: 0.1%
orientate	(关键字) 指定放置文本时所需的文本方向。默认值: north 。 north 向上 east 指向右侧 south 向下 west 指向左侧
rewind	(整型: -2 、 -1 、 0 或 1) 提供文本流的状态重设为使用同一文本流句柄的另一 PDF_fit_textflow() 调用之前的状态。默认值: 0 。 1 退回第一次调用 PDF_fit_textflow() 之前的状态。 0 不重设文本流。 -1 退回上一次调用 PDF_fit_textflow() 之前的状态。 -2 退回倒数第二次调用 PDF_fit_textflow() 之前的状态。
rotate	(浮点型) 旋转坐标系，以定义框的左下角为中心并将指定的值作为旋转角 (以度为单位)。这将导致定义框和文本被旋转。一旦文本已放置，旋转将重置。默认值: 0

表 4.6 PDF_fit_textflow() 的选项 （续）

选项	说明
showborder	（布尔值）如果为 true ，将对限定框的边框做描边处理 （使用当前图形状态）。这可能对开发和调试很有用。默认值: false
showtabs	（关键字）将使用垂直线可视化显示制表位以辅助调试。将根据图形状态 （在调用 <i>PDF_fit_textflow()</i> 之前为活动）划线 （默认值: none): none 不划线 fitbox 将在限定框全高之上划线 validarea 只在有效的垂直区域划线
verticalalign	（关键字）限定框中文本使用的垂直对齐方式；将根据需要，考虑 firstlinedist 和 lastlinedist 选项 （缺省值: top): top 排版文本将从第一行开始，由上至下。若文本不能填满定界框，会在文本下端留出空白。 center 文本将以定界框的垂直中心对齐。若文本不能填满定界框，会在文本上、下端留出空白。 bottom 排版文本将从最后一行开始，由下至上。若文本不能填满定界框，会在文本上端留出空白。 justify 文本将对齐定界框的顶部及底部。为了获得此效果，行距可增加至 <i>linespreadlimit</i> 定义的最大限。当 firstlinedist=leading 时，第一行的高度将会被增加。
wrap	（选项列表，参见表 4.7）文本将环绕使用表 4.7 中列出的子选项指定的区域。这可用于将图形放入文本流内并使文本围绕图形，或用文本填充任意形状。将按照奇偶定则从 <i>fitbox</i> 的边框开始填充 <i>fitbox</i> 。 默认情况下，指定区域内不包含任何文本 （重叠区域除外），即文本环绕形状。这可用于将图形置于形状内。 使用 addfitbox 选项可获得相反的效果：文本将填充指定区域，定界框和形状之间的外部区域保持空白。这样便可使用文本填充任意形状 （而不仅仅填充矩形）。 将使用用户坐标系解释绝对值；使用定界框坐标系解释百分比，即定界框的左下角为 (0,0)，右上角为 (100,100)。最多有 32 个百分比值可供使用。示例： 排除定界框的右上部分: wrap={ boxes={{50% 50% 100% 100%}} } 填充三角形: wrap={ addfitbox polygons={{50% 80% 30% 40% 70% 40% 50% 80%}} } 使用名为 image1 的匹配框排除图像区域: wrap={ usematchboxes={{ image1 }}

表 4.7 PDF_fit_textflow() 的 wrap 选项的子选项

选项	说明
addfitbox	（布尔型）向环绕区域添加定界框。因此，将使用文本填充通过其他环绕选项指定的形状，而不是使用文本环绕形状。默认值: false
boxes	（矩形列表）一个或多个矩形。
lineheight	（两个元素组成的列表，每个元素都是一个正浮点值或关键字）定义文本行的垂直范围，用于计算与环绕元素相交的区域。两个关键字 / 浮点值指定文本基线以上和以下的范围。支持的关键字： none （无范围）、 xheight 、 descender 、 capheight 、 ascender 、 fontsize 、 leading 、 textrise 默认值: { ascender descender }
usematch-boxes	（字符串列表的列表）每个列表中的第一个元素是一个指定匹配框的名称字符串。第二个元素是一个指定所需矩形数目的整数，或者是关键字 all ，指定参考所选择的匹配框的所有矩形。如果第二个元素缺失，则默认为 all 。使用每个矩形的定界框作为文本环绕的形状。
offset	（浮点值或百分比）文本与环绕区域周线之间的水平距离，以用户坐标或定界框宽度的百分比为单位。这可用于以水平方式扩展环绕区域。默认值: 0
polygons	（折线列表）一条或多条折线 （不一定闭合）。文本流将 ascender 和 descender 用作文本行的垂直范围。

C++	Java	<code>double info_textflow(int textflow, String keyword)</code>
Perl	PHP	<code>float PDF_info_textflow(resource p, int textflow, string keyword)</code>
C		<code>double PDF_info_textflow(PDF *p, int textflow, const char *keyword)</code>

`PDF_fit_textflow()` 之后查询文本流的当前状态。

textflow 使用 `textflowhandle` 选项调用 `PDF_add/create_textflow()` 或 `PDF_fill_textblock()` 返回的文本流句柄。

keyword 根据表 4.8 指定请求的信息的关键词。

返回值 *keyword* 请求的、某些文本流参数的值。即使在隐蔽模式下，此参数也可返回正确的几何信息（不同于 *textx/texty* 参数）。

范围 任意，对象除外

表 4.8 `PDF_info_textflow()` 的关键词

keyword	说明
<i>boxlinecount</i>	最后一个限定框中的行数
<i>firstparalinecount</i>	限定框中第一个段落的行数
<i>firstlinedist</i>	第一文本基线和上面的假想基线（如果 <code>fitmethod=top</code> ，将为限定框的上边框）之间的距离
<i>lastmark</i>	最后一个限定框内已处理文本流部分中找到的最后一个标记的编号（可用 <code>mark</code> 选项设置标记）
<i>lastlinedist</i>	最后一条文本基线和下面的假想基线（假定行距保持不变；如果 <code>fitmethod=bottom</code> ，将为限定框的下边框）之间的距离
<i>leading</i>	<code>leading</code> 选项的当前值，由文本流中的文本和选项确定。
<i>lastparalinecount</i>	限定框中最后一个段落的行数。
<i>leftlinex¹, leftliney¹</i>	分别表示最近填充的限定框内、从最左边开始的行的 <i>x</i> 和 <i>y</i> 坐标，以当前的用户坐标为单位。
<i>maxlinelength</i>	表示最近填充的限定框中、最长文本行的长度。
<i>maxliney¹</i>	表示最近填充的限定框中、最长文本行的基线 <i>y</i> 坐标，以当前用户坐标为单位。
<i>minlinelength</i>	表示最近填充的限定框中、最短文本行的长度。
<i>minliney¹</i>	表示最近填充的限定框中、最短文本行的基线 <i>y</i> 坐标，以当前用户坐标为单位。
<i>remainchars</i>	（不常用）尚未处理的字符数。此计数不包括内嵌选项列表和字符引用中的字符数。由于各种文本替换过程（例如 <i>CR/NL</i> 组合），此值可能不可靠。
<i>returnreason</i>	表示最近直接或间接调用 <code>PDF_fit_textflow()</code> 返回的原因的字符串索引（参见表 2.1）。该字符串是 <code>PDF_fit_textflow()</code> 的返回字符串之一。查询由 <code>PDF_fill_textblock()</code> 在内部发出的间接文本流调用的结果时，这可能很有用。
<i>rightlinex¹, rightliney¹</i>	分别表示最近填充的限定框内、在最右边结束的行的 <i>x</i> 和 <i>y</i> 坐标，以当前的用户坐标为单位。
<i>split</i>	指定在最后一个限定框内是否发生了文本拆分： <div> <div>0</div> <div>不需拆分任何文本行。</div> </div> <div> <div>1</div> <div>至少需要拆分一个文本行。</div> </div>
<i>textendx, textendy</i>	表示最近填充的限定框之后、当前文本位置的 <i>x</i> 或 <i>y</i> 坐标，以当前用户坐标为单位
<i>textheight</i>	整个文本的界定框的高度（考虑到 <code>firstlinedist</code> 和 <code>lastlinedist</code> ），以当前用户坐标为单位。
<i>textwidth</i>	整个文本的界定框的宽度，以当前用户坐标为单位
<i>used</i>	到目前为止，已放置的文本的百分比 (0...100)
<i>x1, y1, ..., x4, y4</i>	整个文本的界定框的坐标（考虑到 <code>firstlinedist</code> 和 <code>lastlinedist</code> ），以当前用户坐标为单位。

1. 如果 `rotate` 不等于 0，此值指旋转系统。

C++ Java	<code>void delete_textflow(int textflow)</code>
Perl PHP	<code>PDF_delete_textflow(resource p, int textflow)</code>
C	<code>void PDF_delete_textflow(PDF *p, int textflow)</code>

删除文本流及其所有相关数据结构。

textflow `PDF_create_textflow()` 或 `PDF_add_textflow()` 调用返回的文本流句柄。

详细说明 尚未使用此函数删除的文本流将在关闭文档范围结束之后自动删除。不过，如果生成许多文本流而不调用 `PDF_delete_textflow()`，可能会显著减慢应用程序。

范围 任意

4.3 表格式化

C++ Java

int add_table_cell(int table, int column, int row, string text, string optlist)

Perl PHP

int PDF_add_table_cell(resource p, int table, int column, int row, string text, string optlist)

C

int PDF_add_table_cell(PDF *p,
int table, int column, int row, const char *text, int len, const char *optlist)

在新表或现有表中添加一个单元格。

表 再次调用 `PDF_add_table_cell()` 返回的有效表句柄，或者对于第一次调用为 `-1`（在 PHP 中： `o`）。必须尚未在对 `PDF_fit_table()` 的调用中使用该表句柄，即，所有的表内容必须在将表放置到页面上之前定义。

column, row 包含单元格的列号和行号。如果单元格跨多列和 / 或多行，必须提供最左列的列号和最顶行的行号。第一列 / 行的号码为 `1`。

文本 （内容字符串）用于填充单元格的文本。如果 `text` 不为空，将使用该文本通过 `PDF_fit_textline()` 填充单元格。

len （仅用于 C 语言绑定）UTF-16 字符串的 `text` 长度（以字节为单位）。如果 `len = o`，则必须提供以 `null` 终结的字符串。

- optlist** 选项列表，根据表 4.9 指定详细的表单元格格式化信息。可以使用下列选项：
- ▶ 常规： `errorpolicy`
 - ▶ 列和行定义： `colwidth`、 `colscale-group`、 `minrow-height`、 `return`、 `rowheight`、 `rowjoin-group`、 `rowscale-group`
 - ▶ 单元格定义： `checkword-splitting`、 `colspan`、 `margin`、 `marginleft`、 `marginbottom`、 `marginright`、 `marginintop`、 `matchbox`、 `rowspan`
 - ▶ 单元格内容： `fittextline`、 `textflow`、 `fittextflow`、 `image`、 `fitimage`、 `pdipage`、 `fitpdipage`

返回值	表句柄，可用于后续的表相关调用。必须检查返回值是否为 <code>-1</code> （在 PHP 中：为 <code>o</code> ）；这表示出现错误。如果出现错误，将只忽略最后一个单元格定义；不在表中添加任何内容，但表句柄仍然有效。不能在多个 PDF 输出文档之间使用返回的表句柄。
详细说明	<p>可以使用图像、导入的 PDF 页、文本流或文本行填充表单元格。可以为一个特定的单元格或一个函数调用指定多个内容类型。</p> <p>必须均匀地提供列宽度，以用户坐标或百分比为单位。PDFlib 将基于表的第一个限定框的宽度计算未指定的列宽度。所有未指定宽度的列将具有相同的宽度，以便表能够跨限定框的整个宽度。例外：如果一列中包含一个或多个文本行（而非文本流），PDFlib 将用列中最大的文本宽度作为该列的宽度。</p> <p>对于直排文本，将使用最宽的字符的宽度作为列宽度。对于朝西或朝东的文本，将使用两倍文本高度。文本高度是文本行匹配框的高度，默认的匹配框从基线延伸到大写高度。</p> <p>行高将以与列宽度类似的方式计算。如果一行中只包含文本流单元格，则行高的默认计算方式可能会产生不期望的结果，原因是高度可能会过大。这可以通过使用 <code>rowheight</code> 选项指定最小行高度来避免（PDFlib 始终将按 <code>Textflow</code> 调整行高）。对于朝西或朝东、只包含文本流的列，情况与此类似。</p>
范围	任意，对象除外

表 4.9 PDF_add_table_cell() 的选项

关键字	说明
checkword-splitting	(布尔值) 如果为 true，表格式器将检查文本流单元格是否需要至少一次强制断字。如果需要，将增加单元格宽度以尝试避免断字。强烈建议在 <i>PDF_add/create_textflow()</i> 中设置 <i>adjustmethod=nofit</i> 。在 <i>adjustmethod=auto</i> 的默认情况下，文本流可能会以不希望出现的方式收缩 (受 <i>shrinklimit</i> 选项控制)。由于在 <i>adjustmethod=auto</i> 的情况下，如果 <i>adjustmethod=auto</i> ，表格式器必须在无任何断字的情况下反复确定文本流的状态，可能会大大延长运行时间。 默认值: true
colscale-group¹	(字符串) 列组的名称，将向该列组添加列。如果必须放大其中一列以完全填入长文本，组内的所有列都将均匀缩放。如果一个单元格跨多个列，受影响的列将自动形成一个缩放组。
colspan	(整型) 单元格所跨的列数。默认值: 1
colwidth¹	(浮点值或百分比) 在 <i>column</i> 参数中指定的列宽度，以用户坐标 ² 或表的第一个限定框的宽度的百分比为单位 (参见 <i>PDF_fit_table()</i>)。如果使用百分比，要求使用此选项，且所有列宽度都必须以百分比指定。
errorpolicy	(关键字) 控制出现错误时的行为 (参见第 28 页上的第 2.6 节 “异常处理”)
fitimage	(选项列表; 仅与图像有关) <i>PDF_fit_image()</i> 的选项列表。将应用此选项列表在单元格内放置提供的图像。将使用内单元格框的左下角作为参考点。 默认值: <i>boxsize={<width> <height>}</i> <i>fitmethod=meet position=center</i> , 其中 <i><width></i> 和 <i><height></i> 是内单元格框的计算宽度和高度。计算的此选项列表将被附加到用户指定的选项列表前面。将自动计算框大小，并忽略所提供的选项列表中的任何 <i>boxsize</i> 选项。
fitdipage	(选项列表; 仅与 <i>PDI</i> 页有关) <i>PDF_fit_pdi_page()</i> 的选项列表。将应用此选项列表在单元格内放置提供的页面。将使用内单元格框的左下角作为参考点。 默认值: <i>boxsize={<width> <height>}</i> <i>fitmethod=meet position=center</i> , 其中 <i><width></i> 和 <i><height></i> 是内单元格框的计算宽度和高度。计算的此选项列表将被附加到用户指定的选项列表前面。框大小将自动计算，并将在所提供的选项列表中被忽略。
fittextflow	(选项列表; 仅与文本流有关) <i>PDF_fit_textflow()</i> 的选项列表。将应用此选项列表在单元格内放置提供的文本流。内单元格框将被用作限定框。 默认值: <i>verticalalign=center lastlinedist=descender</i> 。此选项列表将被附加到用户指定的选项列表前面。
fittextline	(选项列表; 仅与文本行有关) <i>PDF_fit_textline()</i> 的选项列表。将应用此选项列表将提供的文本填入单元格。将使用内单元格框的左下角作为参考点。尚未指定的选项将由相应的默认值替代; 不考虑当前文本状态。 默认值: <i>boxsize={<width> <height>}</i> <i>fitmethod=nofit position=center</i> , 其中 <i><width></i> 和 <i><height></i> 是内单元格框的计算宽度和高度。计算的此选项列表将被附加到提供的选项列表前面。将自动计算框大小，并忽略所提供的选项列表中的任何 <i>boxsize</i> 选项。
image	(图像句柄) 与将放置在内单元格框中的句柄关联的图像。
margin marginleft marginbottom marginright marginintop	(浮点值或百分比) 左 / 底 / 右 / 顶单元格边距，以用户坐标 (必须大于等于 0) 或单元格宽度或高度的百分比 (必须小于 100%) 为单位。指定的边距定义用作填入单元格内容的限定框的内单元格框。 <i>margin</i> 的默认值: 0; 其他所有选项的默认值: <i>margin</i>
matchbox	(选项列表) 根据表 4.13, 包含匹配框详细信息的选项列表。
minrow-height¹	(浮点值或百分比) 如果无法完全将一行放置在一个表实例中，此选项指定是否能拆分行以及可以拆分成多小的片段。可以使用用户坐标或行高度的百分比指定最小片段大小。默认值: 100%，即不拆分
pdipage	(页面句柄) 与将放置在内单元格框中的句柄关联的导入 <i>PDF</i> 页面。默认值: none
return¹	(字符串) 放置指定行后， <i>PDF_fit_table()</i> 将停止并返回指定的字符串。字符串不能用下划线 “_” 字符开头。如果指定行属于连接组，则它必须是该组的最后一行，否则将出现错误。
rowheight¹	(浮点值或百分比) 在 <i>row</i> 参数中指定的行高度，以用户坐标 ² 或表的第一个限定框的高度的百分比为单位 (参见 <i>PDF_fit_table()</i>)。如果使用百分比，要求使用此选项，且所有行高度都必须以百分比指定。

表 4.9 PDF_add_table_cell() 的选项（续）

关键字	说明
rowscale-group¹	（字符串）行组的名称，将向该行组添加行。如果必须放大其中一行以完全填入长文本，组内的所有行都将均匀缩放。如果一个单元格跨多个行，受影响的行将自动形成一个缩放组。
rowjoin-group¹	（字符串）行组的名称，将向该行组添加行。组内的所有行都将在表实例中保持在一起。组内的行必须连续编号。如果一个单元格跨多个行，受影响的行将不会自动形成一个连接组。
rowspan	（整型）单元格所跨的行数。默认值：1
textflow	（文本流句柄）与将放置在内单元格框中的句柄关联的文本流。文本流只能在一个表中使用一次，并且不能在该表之外使用。默认值：none

1. 此选项的最后一个说明起决定作用，前面对同一行或列所作的说明将被忽略。
2. 更准确地说，当调用 PDF_fit_table() 以放置第一个表实例时有效的坐标系统。

```
C++ Java String fit_table(int table, double llx, double lly, double urx, double ury, String optlist)
Perl PHP string PDF_fit_table(resource p, int table, float llx, float lly, float urx, float ury, string optlist)
C const char *PDF_fit_table(PDF *p,
                             int table, double llx, double lly, double urx, double ury, const char *optlist)
```

将表完全或部分地放在页面上。

表 通过调用 PDF_add_table_cell() 返回的有效表句柄。

llx, lly, urx, ury 用户坐标系统中，表实例的目标矩形（限定框）左下角和右上角的坐标。也可以按相反的顺序指定左下角和右上角坐标。

optlist 选项列表，根据表 4.10 指定详细的填充信息。可以使用下列选项：

- 常规选项：*blind*、*errorpolicy*、*horshrinklimit*、*rewind*、*vertshrink-limit*
- 表内容：*填充*、*header*、*footer*、*描边*
- 用于开发和调试的可视化辅助：*debugshow*、*showborder*、*showcells*、*showgrid*

返回值 指定函数返回的原因的字符串。

- *_stop*: 表内的所有行均已处理完毕。
- *_boxfull*: 还有行需要放置，但表的限定框内已经没有足够的空间；需要再次调用 PDF_fit_table() 以处理剩余的行。
- *_error*: 出现错误；调用 PDF_get_errmsg() 可获取有关此问题的详细信息。
- 任何其他字符串：在调用 PDF_add_table_cell() 时提供给 return 选项的字符串。

详细说明 将表放置在页面上。必须已经使用先前对 PDF_add_table_cell() 的调用填充了表单元格。如果整个表不能完全填入页面，将放置一个表实例；可以根据返回值使用对此函数的后续调用放置一个或多个实例。将以下列顺序放置表单元格的内容：

- 阴影：将按下列顺序填充使用 *fill* 选项指定的区域：*table*、*colother*、*colodd*、*coleven*、*col#*、*collast*、*rowother*、*rowodd*、*roweven*、*row#*、*rowlast*、*header*、*footer*。
- 匹配框阴影：由 *matchbox* 所定义的单个单元格区域。
- 内容：将以下列顺序放置指定的单元格内容：图像、导入的 PDF 页、文本流、文本行。
- 匹配框划线：由 *matchbox* 所定义的单个单元格区域。
- 划线：将按下列顺序使用 *linecap=projecting* 和 *linejoin=miter* 对使用 *stroke* 选项指定的行进行描边：*other*、*horother*、*hor#*、*horlast*、*vertother*、*vert#*、*vertlast*、*frame*。跨多行或多列的单元格将不受描边的影响。类似地，将不会围绕带指定边框修饰的 *matchbox* 的单元格描边（除非该匹配框使用内单元格框）。如果指定了 *frame*，将禁用表边线 *verto*、*horo*、*vertN* 和 *horN*。
- 指定的匹配框：可以在表函数之外使用其他元素（例如注释、表域、图像等）填充这些匹配框。

表 4.10 *PDF_fit_table()* 的选项

关键字	说明
blind	(布尔值) 如果为 true , 将执行所有计算, 但不创建输出。默认值: false
debugshow	(布尔型) 如果为 true , 则会禁止表出现太高、太宽或单元格太小等所有错误, 并将错误记录下来。得到的表实例虽然已损坏, 但它仍可辅助调试。默认值: false
errorpolicy	(关键字) 控制出现错误时的行为 (参见第 28 页上的第 2.6 节 “异常处理”)
填充	(选项列表的列表) 此选项可用来使用颜色填充行或列 (matchbox 选项可用来使用颜色填充单个单元格, 参见第 70 页上的第 4.4 节 “匹配框”)。支持下列子选项: area (关键字) 要填充的表区域: col# 表中的列编号 # collast 最后一列 coleven 所有偶数列 (参见 <i>PDF_add_table_cell()</i> 中的 col) colodd 所有奇数列 colother 所有其他列 row# 表中的行编号 # rowlast 表实例中的最后一个正文行 roweven 所有偶数行 (参见 <i>PDF_add_table_cell()</i> 中的 row) rowodd 所有奇数行 header 页眉组中的所有行 footer 页脚组中的所有行 rowother 所有其他正文行 table 整个表区域 (即表中的所有行) fillcolor (颜色; 必需的) 填充区域的颜色 示例: 使用红色填充表中的所有行: <code>fill = { {area=table fillcolor={rgb 1 0 0}} }</code> 使用绿色填充奇数行, 使用红色填充偶数行: <code>fill = { {area=rowodd fillcolor={rgb 0 1 0}} {area=roweven fillcolor={rgb 1 0 0}} }</code>
footer	(整型) 表定义中最后的 (页脚) 行 (将在表实例的底部重复出现) 的行数。 默认值: 0 (无页脚行)
header	(整型) 表定义中初始的 (页眉) 行 (将在表实例的顶部重复出现) 的行数。 默认值: 0 (无页眉行)
horshrinklimit	(浮点值或百分比) 将在收缩表以完全排入表的限定框时使用的水平收缩因子的下限 (如果提供了百分比), 或表宽度与限定框宽度的绝对差 (如果提供了浮点值)。默认值: 50%
rewind	(整型: -1 、 0 、或 1) 表的状态重设为另一 <i>PDF_fit_table()</i> 调用之前的状态。目前, 支持以下值。 默认值: 0 1 退回第一次调用 <i>PDF_fit_table()</i> 之前的状态。 0 不重设表。 -1 退回上一次调用 <i>PDF_fit_table()</i> 之前的状态。
showborder	(布尔值) 如果为 true , 将对表的外边框做描边处理 (使用当前图形状态)。默认值: false

表 4.10 PDF_fit_table() 的选项（续）

关键字	说明
showcells	（布尔值）如果为 true，将对每个内单元格框的边框做描边处理（使用当前图形状态）。默认值：false
showgrid	（布尔值）如果为 true，将对所有列和行的垂直和水平边界做描边处理。默认值：false
描边	（选项列表的列表）此选项可用在单元格边框处创建描边。支持下列子选项： line（关键字）将做描边处理的表线： vert# 列编号 # 的右边框处的垂直线；vert0 为表的左边框 vertfirst 第一条垂直线（等同于 vert0） vertlast 最后一条垂直线 vertother 所有其他垂直线 hor# 表中行号 # 处的底部水平线；row0 为顶边框 horfirst 表实例中的第一条水平线 horother 所有其他水平线 horlast 表实例中的最后一条水平线 frame 表的外边框 other 所有未指定的线 linewidth（浮点型）线宽，0 表示没有线。默认值：1 strokecolor（颜色）线的颜色。默认值：black dasharray（浮点值列表）0-8 个值，定义虚线样式（参见 PDF_setdashpattern()）。默认值：{}（空列表，即描边线） dashphase（浮点型）虚线样式之间的距离（参见 PDF_setdashpattern()）。默认值：0 示例： 使用黑色和线宽 1 对所有线做描边处理：stroke = {line=other} 使用线宽 0.5 对外边框线做描边处理：stroke = { {line=frame linewidth=0.5} } 使用线宽 0.5 对外边框线做描边处理，对所有其他线使用线宽 0.1 做描边处理： stroke = { {line=frame linewidth=0.5} {line=other linewidth=0.1} } vertshrink-limit（浮点值或百分比）将在收缩表以完全排入表的限定框时使用的垂直收缩因子的下限（如果提供了百分比），或表实例的高度与限定框高度的绝对差（如果提供了浮点值）。默认值：90%

C++ Java double info_table(int table, String keyword)

Perl PHP float PDF_info_table(resource p, int table, string keyword)

C double PDF_info_table(PDF *p, int table, const char *keyword)

检索与最近放置的表实例相关的表信息。

表 通过调用 PDF_add_table_cell() 返回的有效表句柄。由于只有将表实例放置到页面上之后，返回值才有意义，所以必须已经在至少一次 PDF_fit_table() 调用中使用了表句柄。

keyword 根据表 4.11 指定请求的信息的关键字。

返回值 keyword 请求的、某些表参数的值。即使在隐蔽模式下，此函数也可返回正确的几何信息。

范围 任意，对象除外

表 4.11 PDF_info_table() 的关键字

keyword	说明
firstbodyrow	在最近放置的表实例中第一个正文行的编号
height	表实例的高度
horboxgap	表实例宽度与限定框宽度之差。如果必须收缩表，该值将指定与限定框宽度的偏差（即一个负值）。
horshrinking	水平收缩因子，以计算表宽度的百分比表示。如果必须水平收缩表，该值将指定收缩百分比，否则将为 100。
lastbodyrow	在最近放置的表实例中最后一个正文行的编号

表 4.11 `PDF_info_table()` 的关键字 （续）

keyword	说明
returnreason	返回原因的字符串索引
rowcount	最近放置的表实例中的行数 （包括页眉和页脚）
rowsplit	如果必须拆分最后一行，为 <code>1</code> ；否则为 <code>0</code>
verboxgap	表实例高度与限定框高度之差。如果必须收缩表，该值将指定与限定框高度的偏差 （即一个负值）。
vert-shrinking	垂直收缩因子，以计算表高度的百分比表示。如果必须垂直收缩表，该值将指定收缩百分比，否则将为 <code>100</code> 。
width	表实例的宽度
x1, y1, ..., x4, y4	表实例各角的坐标，以用户坐标为单位，按逆时针方向从左下角开始
xvertline#	带编号 <code>#</code> 的垂直线的 <code>x</code> 坐标。 <code>xvertline0</code> 为左表边框。
yhorline#	带编号 <code>#</code> 的水平线的 <code>x</code> 坐标。 <code>yhorline0</code> 为顶表边框。

C++ Java

`void delete_table(int table, String optlist)`

Perl PHP

`PDF_delete_table(resource p, int table, string optlist)`

C

`void PDF_delete_table(PDF *p, int table, const char *optlist)`

删除表及其所有相关数据结构。

表 通过调用 `PDF_add_table_cell()` 返回的有效表句柄。

optlist 选项列表，根据表 4.12 指定清理选项。

详细说明 尚未使用此函数删除的表将在关闭文档范围结束之后自动删除。

范围 任意

表 4.12 `PDF_delete_table()` 的选项

关键字	说明
keephandles	（布尔值）如果为 <code>false</code> ，将自动删除提供给 <code>PDF_add_table_cell()</code> 的 <code>textflow</code> 、 <code>image</code> 和 <code>pdipage</code> 选项的所有句柄。默认值： <code>false</code>

4.4 匹配框

匹配框不是用专用函数定义的，而是在创建实际元素的函数调用中使用 `matchbox` 选项调用的。

- ▶ 文本行: `textflow` 属性设置为 `false` 的 `PDF_fit_textline()`、`PDF_fill_textblock()`
- ▶ 文本流片段: `textflow` 属性设置为 `true` 的 `PDF_create_textflow()`、`PDF_add_textflow()`、`PDF_fill_textblock()`
- ▶ 导入的 PDF 页: `PDF_fit_pdi_page()`、`PDF_fill_pdf_block()`
- ▶ 图像和模板: `PDF_fit_image()`、`PDF_fill_image_block()`
- ▶ 表单元格: `PDF_add_table_cell()`

使用这些函数的 `matchbox` 选项定义匹配框。它需要支持表 4.13 中所列的子选项的选项列表。与匹配框对应的矩形的详细信息可通过 `PDF_info_matchbox()` 查询。

表 4.13 各种函数的 `matchbox` 选项的子选项

选项	说明
fillcolor	(颜色) 填充矩形的颜色。默认值: <code>none</code>
strokecolor	(颜色) 矩形边框的描边色。默认值: <code>black</code>
borderwidth	(浮点型) 矩形边框的线宽。默认值: <code>0</code>
boxheight	(两个关键字的列表; 仅用于文本行和文本流) 定义文本框的垂直范围。可用两个关键字分别指定基线以上和以下的范围: <code>none</code> (无范围), <code>xheight</code> 、 <code>descender</code> 、 <code>capheight</code> 、 <code>ascender</code> 、 <code>fontsize</code> 、 <code>leading</code> 文本流: 将使用与匹配框开头处的文本对应的值。默认值: <code>{capheight none}</code>
boxwidth	(浮点型; 仅用于文本流) 匹配框的宽度。如果提供此选项, 将在 <code>matchbox</code> 选项和下一个文本片段或 <code>matchbox end</code> 说明之间插入指定宽度的水平空间。这可用于保留空间, 以便在文本流中插入图像、模板或 PDF 页。默认值: <code>0</code>
clipping	(4 个浮点值或 4 个百分比的列表; 仅用于图像和导入的 PDF 页; 如果指定了 <code>innerbox</code> 选项, 将被忽略) 图像或页面中矩形的左下角和右上角的坐标, 指定应显示哪一部分。对于图像, 可使用像素点或宽度 / 高度的百分比指定剪切矩形。对于 PDF 页, 可使用默认单位或页面的裁剪框宽度 / 高度的百分比指定剪切矩形。默认值: <code>{0% 0% 100% 100%}</code>
dasharray	(浮点值列表) 0-8 个值的列表, 使用用户坐标定义矩形边框的虚线样式 (参见 <code>PDF_setdashpattern()</code>)。默认值: <code>{}</code>
dashphase	(浮点型) 虚线样式内矩形边框的虚线开始的距离 (参见 <code>PDF_setdashpattern()</code>)。默认值: <code>0</code>
drawleft drawbottom drawright drawtop	(布尔值) 如果为 <code>true</code> , 将画出矩形的对应边框。默认值: <code>true</code>
end	(布尔型; 仅用于文本流) 指定匹配框的结尾。如果为 <code>true</code> , 将忽略所有其他 <code>matchbox</code> 选项。文本流中的匹配框不能嵌套。文本流匹配框的宽度通过选项 <code>boxwidth</code> (如果已指定) 和 <code>matchbox</code> 与 <code>matchbox= end</code> 选项包括的文本的范围来定义。如果未指定 <code>end</code> 选项, 匹配框将在文本流中最后一个字符之后结束。
innerbox	(布尔型; 仅用于表单元格和 <code>TIFF</code> 和 <code>JPEG</code> 图像) 表单元格: 如果为 <code>true</code> , 将按为单元格定义的边距缩小单元格框; 否则将使用整个单元格。 <code>TIFF</code> 和 <code>JPEG</code> 图像: 如果图像包含剪切路径, 则使用剪切路径的定界框, 而非整个图像。 默认值: <code>false</code>
linecap	(整型或关键字) 路径结尾处的形状 (参见 <code>PDF_setlinecap()</code>); 必须为 <code>0</code> 、 <code>1</code> 或 <code>2</code> , 或对应的关键字 <code>butt</code> 、 <code>round</code> 或 <code>projecting</code> 之一。默认值: <code>butt</code>
linejoin	(整型或关键字) 路径各角处的形状 (参见 <code>PDF_setlinejoin()</code>); 必须为 <code>0</code> 、 <code>1</code> 或 <code>2</code> , 或对应的关键字 <code>miter</code> 、 <code>round</code> 或 <code>bevel</code> 之一。默认值: <code>miter</code>
margin	(浮点值或百分比) 匹配框矩形的附加边距, 以用户坐标 (必须大于等于 <code>0</code>) 或矩形宽度或高度的百分比 (必须小于 <code>100%</code>) 为单位。默认值: <code>0</code>

表 4.13 各种函数的 `matchbox` 选项的子选项（续）

选项	说明
<code>name</code>	（名称字符串）可用于 <code>PDF_info_matchbox()</code> 的匹配框名称。如果名称已经被指定给一个匹配框，将为此名称另外创建一个矩形。可以在当前页结束之前使用匹配框名称。默认值：无名称
<code>offsetleft</code> <code>offsetbottom</code> <code>offsetright</code> <code>offsettop</code>	（浮点值或百分比）计算而得的初始矩形和期望框左 / 右 / 底 / 顶边缘的用户定义的偏移。使用用户坐标或矩形宽度（对于 <code>offsetleft/offsetright</code> ）或高度（对于 <code>offsetbottom/offsettop</code> ）的百分比指定值。允许使用负值扩展匹配框。 <code>offsetleft/offsetbottom</code> 的默认值： <code>margin</code> ； <code>offsetright/offsettop</code> 的默认值： <code>-margin</code>
<code>openrect</code>	（布尔型；仅用于文本流和表单元格）文本流：如果匹配框矩形被拆分到下一行，将不画第一个矩形的右边框和第二个矩形的左边框。表单元格：如果表行被拆分到下一个表实例，将不画第一部分的底边框和第二部分的顶边框。默认值： <code>false</code>
<code>create-wrapbox</code>	（布尔型；仅用于文本流）包含匹配框的矩形在计算之后将作为环绕框插入文本流。包含匹配框的行之后的行将环绕矩形。默认值： <code>false</code>

```
C++ Java double info_matchbox(String boxname, int num, String keyword)
Perl PHP float PDF_info_matchbox(resource p, string boxname, int num, string keyword)
C double PDF_info_matchbox(PDF *p,const char *boxname, int len, int num, const char *keyword)
```

查询有关当前页上的匹配框的信息。

boxname （名称字符串）匹配框的名称。定义匹配框时，必须已经使用 `matchbox` 选项的 `name` 子选项定义了名称。

len （仅用于 C 语言绑定）UTF-16 字符串的 `name` 长度（以字节为单位）。如果 `len = 0`，则必须提供以 `null` 终结的字符串。

num 请求的匹配框矩形的编号（第一个为 1）。

keyword 根据表 4.14 指定请求的信息的关键字。

返回值 `keyword` 请求的、某些匹配框参数的值。如果在当前页面上不存在指定名称或编号的匹配框，所有关键字将返回值 0。

范围 页面、图案、模板、字形、路径、字体

表 4.14 `PDF_info_matchbox()` 的关键字

keyword	说明
<code>count</code>	（将忽略 <code>num</code> 参数）矩形的数目
<code>exists</code>	如果矩形存在，为 1；否则为 0
<code>height</code>	矩形的高度，以用户坐标为单位
<code>width</code>	矩形的宽度，以用户坐标为单位
<code>x1, y1, ..., x4, y4</code>	矩形第 <i>i</i> 个角的位置 (<i>i</i> =1, 2, 3, 4)，以用户坐标为单位 在对应填充元素（图像、文本等）的坐标系中， <code>x1, y1</code> 对应于左上角， <code>x2, y2</code> 对应于左下角， <code>x3, y3</code> 对应于右下角， <code>x4, y4</code> 对应于右上角



5 图形函数

5.1 图形状态

在页面开头，所有图形状态参数都将还原为缺省值。相关函数说明中记录了这些缺省值。
第 31 页上的第 3 章“文本函数”中列出了与文本状态相关的函数。

注：路径范围内不能使用任何图形状态函数。

C++ Java	<code>void setdash(double b, double w)</code>
Perl PHP	<code>PDF_setdash(resource p, float b, float w)</code>
C	<code>void PDF_setdash(PDF *p, double b, double w)</code>
设置当前虚线样式。	
<i>b, w</i> 交替显示的黑白单位的数量。 <i>b</i> 和 <i>w</i> 必须为正值数值。	
详细说明	要生成实线，应设置 <i>b=w=0</i> 。在每页的开头， <i>dash</i> 参数将设为实线。
范围	页面、图案、模板、字形

C++ Java	<code>void setdashpattern(String optlist)</code>
Perl PHP	<code>PDF_setdashpattern(resource p, string optlist)</code>
C	<code>void PDF_setdashpattern(PDF *p, const char *optlist)</code>
设置选项列表定义的虚线样式。	
<i>optlist</i> 根据表 5.1 提供的选项列表。空列表将生成实线。可以使用下列选项： <i>dasharray</i> 、 <i>dashphase</i>	

表 5.1 `PDF_setdashpattern()` 的选项

选项	描述
<i>dasharray</i>	(浮点值列表) 一个由 2 至 8 个交替值组成的列表，指示虚线的长度和描边路径的间隔（使用用户坐标系表示）。数组值必须大于 0。这些值将循环使用，直至整个描边路径完成为止。
<i>dashphase</i>	(浮点型) 虚线样式之间的距离。默认值：0

详细说明	在每页的开头， <i>dash</i> 参数将设为实线。
范围	页面、图案、模板、字形

C++ Java	<code>void setflat(double flatness)</code>
Perl PHP	<code>PDF_setflat(resource p, float flatness)</code>
C	<code>void PDF_setflat(PDF *p, double flatness)</code>
设置 <i>flatness</i> 容差。	
<i>flatness</i> 一个正值，描述路径之间的最大距离（以设备像素为单位）和根据直线段计算的近似值。	

详细说明 在每页开头，*flatness* 容差设为缺省值 1。

范围 页面、图案、模板、字形

C++ Java

void setlinejoin(int linejoin)

Perl PHP

PDF_setlinejoin(resource p, int linejoin)

C

void PDF_setlinejoin(PDF *p, int linejoin)

设置 *linejoin* 样式。
linejoin 指定描边路径拐角的形状，参见表 5.2。

详细说明 在每页开头，*linejoin* 样式设为缺省值 0。

范围 页面、图案、模板、字形

表 5.2 linejoin 样式的值

值	描述（摘自《PDF Reference》）	示例
0	斜角连接：两个线段的描边外缘继续延伸，直至相接为止。当斜角外延超出斜角极限时，将由斜削连接取代。	
1	圆角连接：以在线段相接的点为圆心，绘制直径等于线宽的圆弧，并填充圆弧，生成圆角。	
2	斜面连接：绘制两个具有平头端点的路径线段（参见有关 linecap 参数的讨论），用三角形填充生成的线段端点以外的切口。	

C++ Java

void setlinecap(int linecap)

Perl PHP

PDF_setlinecap(resource p, int linecap)

C

void PDF_setlinecap(PDF *p, int linecap)

设置 *linecap* 参数。
linecap 对于描边，控制路径端点的形状，参见表 5.3。

详细说明 在每页开头，*linecap* 参数设为缺省值 0。

范围 页面、图案、模板、字形

表 5.3 linecap 参数的值

值	描述（摘自《PDF Reference》）	示例
0	平头端点：描边在路径端点处切成方形。	
1	圆头端点：以端点为圆心，绘制直径等于线宽的半圆弧，并填充圆弧。	
2	方形突出终端点：描边延伸出线的端点（距离为线宽的一半），并以方形结束。	

C++ Java void setmiterlimit(double miter)

Perl PHP PDF_setmiterlimit(resource p, float miter)

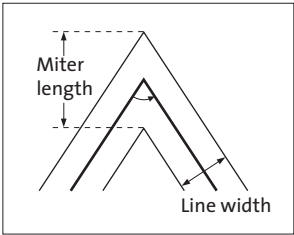
C void PDF_setmiterlimit(PDF *p, double miter)

设置斜接限制。

miter 一个大于或等于 1 的值，控制斜接连接生成的尖端。

详细说明

如果 *linejoin* 样式设为 *o*（斜角连接），以小角度连接的两个线段将生成很尖锐的尖端。斜接长度和线宽的比例超出斜接限制时，将用垂直端点替代尖端（即，斜接连接将改为斜面连接）。在每页开头，斜接限制设为缺省值 *10*。这大约相当于 *11.5* 度角。



范围 页面、图案、模板、字形

C++ Java void setlinewidth(double width)

Perl PHP PDF_setlinewidth(resource p, float width)

C void PDF_setlinewidth(PDF *p, double width)

设置当前线宽。

width 以当前用户坐标系统的单位设置的线宽。

详细说明 在每页开头，*width* 参数设为缺省值 *1*。

范围 页面、图案、模板、字形

C++ Java void initgraphics()

Perl PHP PDF_initgraphics(resource p)

C void PDF_initgraphics(PDF *p)

将所有颜色和图形状态参数重设为他们的缺省值。

详细说明

填充色和描边色、线宽、线条端点样式、样条连接样式、斜接限制、虚线样式和容差平滑度设置以及坐标系统（但不包括文本状态参数）都重设为他们各自的缺省值。不影响当前剪切路径。

在程序流不允许方便地使用 *PDF_save()/PDF_restore()* 时，此函数可能会很有用。

范围 页面、图案、模板、字形

5.2 保存和恢复图形状态

C++ Java	<code>void save()</code>
Perl PHP	<code>PDF_save(resource p)</code>
C	<code>void PDF_save(PDF *p)</code>

将当前图形状态保存到一个堆栈。

详细说明 图形状态包含控制各种类型的图形对象的参数。*PDF* 不要求保存图形状态；只有应用程序希望稍后返回某些特定图形状态（例如，自定义坐标系统），而又不想再次显式设置所有相关参数时，才需要保存图形状态。下列项目受 *save/restore* 控制：

- ▶ 已经使用相应函数设置的图形参数：剪切路径、坐标系统、当前点、平滑度 (*flatness*) 容差、线条端点样式、虚线样式、线条连接样式 (*line join style*)、线宽、斜接限制；
- ▶ 颜色参数：填充色和描边色；
- ▶ 已经使用 *PDF_set_gstate()* 中的显式图形状态设置的图形参数（参见第 79 页上的第 5.4 节“显式图形状态”）；
- ▶ 文本位置及下列与文本相关的参数：*charspacing*、*wordspacing*、*horizscaling*、*italicangle*、*leading*、*font*、*fontsize*、*textrendering*、*textrise*。

PDF_save() 和 *PDF_restore()* 可以嵌套成对。尽管 *PDF* 规范没有限制 *save/restore* 对的嵌套深度，但应用程序必须保持嵌套深度小于 26，以避免由 *PDF* 浏览器所生成的带有限制的 *PostScript* 输出导致的打印问题，并允许 *PDFlib* 内部要求的额外 *save* 级别。

范围 页面、图案、模板、字形：必须与对应的 *PDF_restore()* 调用成对使用。在每个页面、图案、模板和字形描述中，*PDF_save()* 和 *PDF_restore()* 调用必须成对出现。

参数 大多数与文本相关的参数受 *save/restore* 的影响；参见以上列表。下列参数不受 *save/restore* 控制：*fillrule*、*kertering*、*underline*、*overline*、*strikeout*。

C++ Java	<code>void restore()</code>
Perl PHP	<code>PDF_restore(resource p)</code>
C	<code>void PDF_restore(PDF *p)</code>

从堆栈中恢复最近一次保存的图形状态。

详细说明 必须是已经在同一页面、图案或模板上保存了的相应的图形状态。

范围 页面、图案、模板、字形：必须与对应的 *PDF_save()* 调用成对使用。在每个页面、图案、模板和字形描述中，*PDF_save()* 和 *PDF_restore()* 调用必须成对出现。

5.3 坐标系统转换

所有转换函数（`PDF_translate()`、`PDF_scale()`、`PDF_rotate()`、`PDF_skew()`、`PDF_concat()`、`PDF_setmatrix()` 和 `PDF_initgraphics()`）都可更改绘制后续对象使用的坐标系统。他们不会对页面上的现有对象产生任何影响。

C++ Java

void translate(double tx, double ty)

Perl PHP

PDF_translate(resource p, float tx, float ty)

C

void PDF_translate(PDF *p, double tx, double ty)

转换坐标系的原点。

tx, ty 坐标系统的新原点为旧坐标系统中的 (tx, ty) 点。

范围

页面、图案、模板、字形

C++ Java

void scale(double sx, double sy)

Perl PHP

PDF_scale(resource p, float sx, float sy)

C

void PDF_scale(PDF *p, double sx, double sy)

缩放坐标系统。

sx, sy x 和 y 方向上的缩放因子。

详细说明 此函数可按 **sx** 和 **sy** 值缩放坐标系统。通过使用负值缩放因子，还可使用此函数实现对称（镜像）效果。在新坐标系统中，x 方向上的一个单位等于旧坐标系统中 x 方向上的 **sx** 个单位，y 坐标同样如此。

范围

页面、图案、模板、字形

C++ Java

void rotate(double phi)

Perl PHP

PDF_rotate(resource p, float phi)

C

void PDF_rotate(PDF *p, double phi)

旋转坐标系统。

phi 旋转角度的度数。

详细说明 自当前坐标系统的正 x 轴起，按逆时针方向测量角度。将旧坐标轴旋转 **phi** 指示的度数，计算新坐标轴。

范围

页面、图案、模板、字形

C++ Java

void skew(double alpha, double beta)

Perl PHP

PDF_skew(resource p, float alpha, float beta)

C

void PDF_skew(PDF *p, double alpha, double beta)

歪斜坐标系统。

alpha, beta x 和 y 方向上的歪斜角度数。

详细说明 歪斜（或剪切）会在 x 和 y 方向上按指定的角度扭曲坐标系统。自当前坐标系统的正 x 轴起，按逆时针方向测量 $alpha$ ；自正 y 坐标起，按顺时针方向测量 $beta$ 。两个角都必须在 $-360^\circ < alpha, beta < 360^\circ$ 范围内，且必须不同于 -270° 、 -90° 、 90° 、和 270° 。

范围 页面、图案、模板、字形

C++ Java

void concat(double a, double b, double c, double d, double e, double f)

Perl PHP

PDF_concat(resource p, float a, float b, float c, float d, float e, float f)

C

void PDF_concat(PDF *p, double a, double b, double c, double d, double e, double f)

对当前坐标系统应用转换矩阵。

a, b, c, d, e, f 转换矩阵的元素。这六个值以 PostScript 和 PDF 中相同的方式构成坐标系统（参见相关参考）。为了避免转换退化， $a*d$ 不能等于 $b*c$ 。

详细说明 此函数对当前坐标系统应用矩阵。它允许大多数常规的转换形式。使用此函数需对转换矩阵很熟悉，否则建议另用 `PDF_translate()`、`PDF_scale()`、`PDF_rotate()` 和 `PDF_skew()` 函数。在每页的开头将坐标系统重设为默认坐标系统（即当前转换矩阵为单位矩阵 $[1, 0, 0, 1, 0, 0]$ ）。

范围 页面、图案、模板、字形

C++ Java

void setmatrix(double a, double b, double c, double d, double e, double f)

Perl PHP

PDF_setmatrix(resource p, float a, float b, float c, float d, float e, float f)

C

void PDF_setmatrix(PDF *p, double a, double b, double c, double d, double e, double f)

显示设置当前转换矩阵。

a, b, c, d, e, f 参见 `PDF_concat()`。

详细说明 此函数类似于 `PDF_concat()`。但是，它会废弃当前转换矩阵，并用新的矩阵完全替代旧矩阵。

范围 页面、图案、模板、字形

5.4 显式图形状态

C++ Java	<code>int create_gstate(String optlist)</code>
Perl PHP	<code>int PDF_create_gstate(resource p, string optlist)</code>
C	<code>int PDF_create_gstate(PDF *p, const char *optlist)</code>

根据各种选项创建图形状态对象。

optlist 选项列表，根据表 5.4 包含图形状态的选项。可以使用下列选项：
alphaishape、*blendmode*、*flatness*、*linecap*、*linejoin*、*linewidth*、*miterlimit*、*opacityfill*、*opacitystroke*、*overprintfill*、*overprintmode*、*overprintstroke*、*renderingintent*、*smoothness*、*strokeadjust*、*textknockout*

返回值	图形状态句柄；在关闭的文档范围中，可在后续 <code>PDF_set_gstate()</code> 调用中使用此句柄。
详细说明	此选项列表可以包含任何数量的图形状态参数。并非所有参数都可用于所有 PDF 版本。表中列出了要求的最低 PDF 版本。

范围 文档、页面、图案、模板、字形

表 5.4 `PDF_create_gstate()` 的选项

关键字	说明和可能的值
<i>alphaishape</i>	（布尔型；PDF 1.4）此值为真 (<i>true</i>) 则 <i>Alpha</i> 的数据描述形状，若为假 (<i>false</i>) 则描述不透明度。默认值：false
<i>blendmode</i>	（关键字列表；PDF 1.4；如果在 PDF/A 模式下使用，必须取值 Normal）混合模式的名称。可指定多个混合模式。可能的值：Color、ColorDodge、ColorBurn、Darken、Difference、Exclusion、HardLight、Hue、Lighten、Luminosity、Multiply、None、Normal、Overlay、Saturation、Screen、SoftLight。默认值：None
<i>flatness</i>	（浮点型）路径与其近似值之间的最大距离（参见 <code>PDF_setflat()</code> ），必须 > 0 。
<i>linecap</i>	（整型或关键字）路径结尾处的形状（参见 <code>PDF_setlinecap()</code> ）；必须为 0、1 或 2，或对应的关键字 butt、round 或 projecting 之一。
<i>linejoin</i>	（整型或关键字）路径各角处的形状（参见 <code>PDF_setlinejoin()</code> ）；必须为 0、1 或 2，或对应的关键字 miter、round 或 bevel 之一。
<i>linewidth</i>	（浮点型）线宽（参见 <code>PDF_setlinewidth()</code> ）；必须 > 0 。
<i>miterlimit</i>	（浮点型）控制斜接连接生成的尖角，必须 ≥ 1 （参见 <code>PDF_setmiterlimit()</code> ）。
<i>opacityfill</i>	（浮点型；PDF 1.4；如果在 PDF/A 模式下使用，必须取值 1）填充操作的常数 <i>alpha</i> ；必须 ≥ 0 和 ≤ 1 。
<i>opacitystroke</i>	（浮点型；PDF 1.4；如果在 PDF/A 模式下使用，必须取值 1）描边操作的常数 <i>alpha</i> ；必须 ≥ 0 和 ≤ 1 。
<i>overprintfill</i>	（布尔值）压印除描边之外的其他操作。默认值：false
<i>overprintmode</i>	（整型）压印模式。0 表示每个颜色组件均替代以前放置的标记；模式 1（在 Acrobat 中，称“压印缺省值是非零压印”）表示，颜色组件值为 0 时将不对相应的组件做任何更改。默认值：0
<i>overprintstroke</i>	（布尔值）描边操作的压印。默认值：false
<i>renderingintent</i>	（关键字）色域压缩使用的颜色渲染方法；可能的值：Auto、AbsoluteColorimetric、RelativeColorimetric、Saturation、Perceptual
<i>smoothness</i>	（浮点型）着色线性插值的最大误差，必须 ≥ 0 并且 ≤ 1 。
<i>strokeadjust</i>	（布尔值）是否应用自动描边调整。默认值：false
<i>textknockout</i>	（布尔型；PDF 1.4）对于复合操作，文本对象中的字形将视为单独对象 (<i>false</i>) 或单一对象 (<i>true</i>)。默认值：true

C++ Java

void set_gstate(int gstate)

Perl PHP

PDF_set_gstate(resource p, int gstate)

C

void PDF_set_gstate(PDF *p, int gstate)

启动图形状态对象。

gstate 由 `PDF_create_gstate()` 生成的的图形状态对象的句柄。

详细说明 将设置图形状态对象中包含的所有选项。多次调用此函数时，会累积图形状态选项。未在图
形状态对象中明确设置的选项将保留各自的当前值。在页面开头，所有图形状态选项都将重
设为缺省值。

范围 页面、图案、模板、字形

5.5 路径构建

表 5.5 列出了本节的相关值关键字名称。

表 5.5 PDF_get_value() 的关键字（参见第 11 页上的第 2.1 节“参数处理”）

关键字	说明
currentx ¹ currenty ¹	分别指示当前点的 <i>x</i> 或 <i>y</i> 坐标（以当前坐标系统的单位表示）。范围：页面、图案、模板、路径
ctm_a ¹ ctm_b ¹ ctm_c ¹ ctm_d ¹ ctm_e ¹ ctm_f ¹	矢量图形的当前转换矩阵 (CTM) 的要素。范围：页面、图案、模板、路径

1. 不能用于 PDF_set_value(), 因为可以使用对应的 API 函数来设置这些值

注：使用本节介绍的函数之后，一定要调用第 84 页上的第 5.6 节“路径绘制和剪切”中的一个函数；否则，构建的路径不会有任何效果，而且后续操作也可能会引发异常。

C++ Java	void moveto(double x, double y)
Perl PHP	PDF_moveto(resource p, float x, float y)
C	void PDF_moveto(PDF *p, double x, double y)
	设置图形输出的当前点。
	<i>x, y</i> 新当前点的坐标。
详细说明	在每页开头，当前点设为 <i>undefined</i> 的缺省值。图形的当前点和当前文本位置将分别维护。
范围	页面、图案、模板、路径、字形；此函数是路径范围的开头。
参数	currentx, currenty

C++ Java	void lineto(double x, double y)
Perl PHP	PDF_lineto(resource p, float x, float y)
C	void PDF_lineto(PDF *p, double x, double y)
	从当前点至另一点，绘制一条直线。
	<i>x, y</i> 直线第二个端点的坐标。
详细说明	此函数将在当前路径上添加一条自当前点至 (<i>x, y</i>) 的直线。使用此函数之前，必须先设置当前点。点 (<i>x, y</i>) 将成为新的当前点。 绘制的直线将围绕 » 理想 « 直线居中放置，即：在连接两个端点的直线两侧分别绘制线宽的一半（根据 <i>linewidth</i> 参数的值确定）。端点处的行为由 <i>linecap</i> 参数的值设置。
范围	路径
参数	currentx, currenty

C++ Java	<code>void curveto(double x1, double y1, double x2, double y2, double x3, double y3)</code>
Perl PHP	<code>PDF_curveto(resource p, float x1, float y1, float x2, float y2, float x3, float y3)</code>
C	<code>void PDF_curveto(PDF *p, double x1, double y1, double x2, double y2, double x3, double y3)</code>

使用另外三个控制点，自当前点起，绘制 Bézier 曲线。

`x1, y1, x2, y2, x3, y3` 三个控制点的坐标。

详细说明	将 $(x1, y1)$ 和 $(x2, y2)$ 用作控制点，在当前路径上添加一条从当前点至 $(x3, y3)$ 的 Bézier 曲线。使用此函数之前，必须先设置当前点。该曲线的端点 $(x3, y3)$ 将成为新的当前点。
范围	路径
参数	<code>currentx, currenty</code>

C++ Java	<code>void circle(double x, double y, double r)</code>
Perl PHP	<code>PDF_circle(resource p, float x, float y, float r)</code>
C	<code>void PDF_circle(PDF *p, double x, double y, double r)</code>

绘制一个圆形。

`x, y` 圆形原点的坐标。

`r` 圆形的半径。

详细说明	此函数作为完全子路径，在当前路径上添加一个圆形。点 $(x + r, y)$ 将成为新的当前点。生成的形状将为使用用户坐标表示的圆形。如果坐标系统在 x 和 y 方向上的缩放比例不同，生成的曲线将为椭圆形。
范围	页面、图案、模板、路径、字形；此函数是路径范围的开头。
参数	<code>currentx, currenty</code>

C++ Java	<code>void arc(double x, double y, double r, double alpha, double beta)</code>
Perl PHP	<code>PDF_arc(resource p, float x, float y, float r, float alpha, float beta)</code>
C	<code>void PDF_arc(PDF *p, double x, double y, double r, double alpha, double beta)</code>

绘制一个逆时针圆弧线段。

`x, y` 圆弧线段中心点的坐标。

`r` 圆弧线段的半径。 r 必须为非负值。

`alpha, beta` 圆弧线段开头和结尾角度的度数。

详细说明	此函数将在当前路径上，按逆时针方向添加一条圆弧线段，自 $alpha$ 角度延伸至 $beta$ 角度。对于 <code>PDF_arc()</code> 和 <code>PDF_arcn()</code> ，都自当前坐标系统的正 x 轴起，按逆时针方向测量角度。如果存在当前点，将自当前点起至圆弧的起始点止，绘制另一条直线。该圆弧的端点将成为新的当前点。 圆弧线段将为使用用户坐标表示的圆形。如果坐标系统在 x 和 y 方向上的缩放比例不同，生成的曲线将为椭圆形。
范围	页面、图案、模板、路径、字形；此函数是路径范围的开头。
参数	<code>currentx, currenty</code>

C++ Java	<i>void arcn(double x, double y, double r, double alpha, double beta)</i>
Perl PHP	<i>PDF_arcn(resource p, float x, float y, float r, float alpha, float beta)</i>
C	<i>void PDF_arcn(PDF *p, double x, double y, double r, double alpha, double beta)</i>

绘制一个顺时针圆弧线段。

详细说明

除绘制方向外，此函数的行为与 *PDF_arc()* 完全相同。尤其是，仍将自正 *x* 轴起，按*逆时针*方向测量角度。

C++ Java	<i>void rect(double x, double y, double width, double height)</i>
Perl PHP	<i>PDF_rect(resource p, float x, float y, float width, float height)</i>
C	<i>void PDF_rect(PDF *p, double x, double y, double width, double height)</i>

绘制一个矩形。

x, y

矩形左下角的坐标。

width, height

矩形的大小。

详细说明

此函数作为完全子路径，在当前路径上添加一个矩形。使用此函数前，不要求设置当前点。点 *(x, y)* 将成为新的当前点。绘制的直线段将围绕 » 理想 « 直线居中放置，即：在连接各个端点的直线两侧分别绘制线宽的一半（根据 *linewidth* 参数的值确定）。

范围

页面、图案、模板、路径、字形；此函数是路径范围的开头。

参数

currentx, currenty

C++ Java	<i>void closepath()</i>
Perl PHP	<i>PDF_closepath(resource p)</i>
C	<i>void PDF_closepath(PDF *p)</i>

关闭当前路径。

详细说明

此函数将关闭当前子路径，即：添加一条自当前点起、至子路径起始点止的直线。

范围

路径

参数

currentx, currenty

5.6 路径绘制和剪切

表 5.6 列出了本节中的相关参数关键字名称（参见第 11 页上的第 2.1 节“参数处理”）。

表 5.6 PDF_get/set_parameter() 中与路径有关的关键字

关键字	说明
<i>fillrule</i>	将当前填充规则设为 <i>winding</i> 或 <i>evenodd</i> 。PDF 查看程序将使用此填充规则确定形状的内部，以进行填充或剪切。因为对于简单形状而言，这两种算法将产生相同的结果，所以大多数应用程序都无需更改填充规则。在每页开头，填充规则将重设为缺省的 <i>winding</i> 。范围：页面、图案、模板、字形。

注：本节内的大多数函数都将清除路径，而不定义当前点。因此，调用其中一个函数之后，后续的绘制操作都必须明确设置当前点（例如，使用 *PDF_moveto()*）。

C++ Java	<i>void stroke()</i>
Perl PHP	<i>PDF_stroke(resource p)</i>
C	<i>void PDF_stroke(PDF *p)</i>

使用当前线宽和当前描边色描边路径，并清除路径。

范围 路径：此函数终结路径范围。

C++ Java	<i>void closepath_stroke()</i>
Perl PHP	<i>PDF_closepath_stroke(resource p)</i>
C	<i>void PDF_closepath_stroke(PDF *p)</i>

关闭路径，并描边。

详细说明 此函数关闭当前子路径（添加一条自当前点起、至路径起点止的直线段）并使用当前线宽和当前描边色描边整个当前路径。

范围 路径：此函数终结路径范围。

C++ Java	<i>void fill()</i>
Perl PHP	<i>PDF_fill(resource p)</i>
C	<i>void PDF_fill(PDF *p)</i>

使用当前填充色填充路径的内部。

详细说明 此函数使用当前填充色填充当前路径的内部。路径的内部可由其中一种算法确定（参见 *fillrule* 参数）。填充之前，会自动关闭仍开放的路径。

范围 路径：此函数终结路径范围。

参数 *fillrule*

C++ Java	<code>void fill_stroke()</code>
Perl PHP	<code>PDF_fill_stroke(resource p)</code>
C	<code>void PDF_fill_stroke(PDF *p)</code>

使用当前填充色和描边色，填充并描边路径。

范围 路径；此函数终结路径范围。

参数 *fillrule*

C++ Java	<code>void closepath_fill_stroke()</code>
Perl PHP	<code>PDF_closepath_fill_stroke(resource p)</code>
C	<code>void PDF_closepath_fill_stroke(PDF *p)</code>

关闭路径，填充并描边路径。

详细说明 此函数关闭当前子路径（添加一条自当前点起、至路径起点止的直线段），然后填充并描边整个当前路径。

范围 路径；此函数终结路径范围。

参数 *fillrule*

C++ Java	<code>void clip()</code>
Perl PHP	<code>PDF_clip(resource p)</code>
C	<code>void PDF_clip(PDF *p)</code>

将当前路径用作剪切路径并终结路径。

详细说明 此函数将当前路径与当前剪贴路径的交叉部分用作后续操作的剪切路径。在每页开头，剪切路径设为页面大小的缺省值。剪切路径受 *PDF_save()/PDF_restore()* 影响。只能通过 *PDF_save()/PDF_restore()* 的方式放大剪切路径。

范围 路径；此函数终结路径范围。

C++ Java	<code>void endpath()</code>
Perl PHP	<code>PDF_endpath(resource p)</code>
C	<code>void PDF_endpath(PDF *p)</code>

结束当前路径，既不填充也不描边。

详细说明 此函数不会在页面上显式任何效果。它会在页面上生成不可见的路径。

范围 路径；此函数终结路径范围。

5.7 图层

C++ Java

int define_layer(String name, String optlist)

Perl PHP

int PDF_define_layer(resource p, string name, string optlist)

C

int PDF_define_layer(PDF *p, const char *name, int len, const char *optlist)

创建新的图层定义（要求 PDF 1.5）。

name （超文本字符串）图层的名称。

len （仅用于 C 语言绑定）UTF-16 字符串的 **name** 长度（以字节为单位）。如果 **len = 0**，则必须提供以 null 终结的字符串。

optlist 选项列表，根据表 5.7 指定图层设置。可以使用下列选项：
creatorinfo、*defaultstate*、*hypertext-encoding*、*hypertext-format*、*initial-exportstate*、*initial-printstate*、*initial-viewstate*、*intent*、*language*、*onpanel*、*pageelement*、*printsubtype*、*zoom*

- 返回值
- 图层句柄，可用于 *PDF_begin_layer()* 和 *PDF_set_layer_dependency()* 调用，直至关闭文档范围结束为止。
- 详细说明
- 如果定义了图层，但未在文档中使用，PDFlib 将发出警告。
- 范围
- 文档、页面

表 5.7 *PDF_define_layer()* 的选项

选项	说明
creatorinfo	（选项列表）描述内容和创建应用程序的选项列表。如果使用此选项，要求同时提供以下两个条目： creator （超文本字符串）创建图层的应用程序名称 subtype （字符串）内容的类型。建议使用 Artwork 和 Technical 值。
defaultstate	（布尔值）默认值：true
hypertext-encoding	（关键字）指定 name 参数的编码方式和 creator 选项。空字符串等效于 unicode 。默认值：全局 hypertextencoding 参数
hypertext-format	（关键字）设置 name 参数的格式。其可选值为 bytes 、 utf8 、 utf16 、 utf16le 、 utf16be 和 auto 。默认值： hypertextformat 参数的值
initial-exportstate	（布尔值）指定图层的建议导出状态。如果为 true ， Acrobat 将在转换 / 导出到旧 PDF 版本或其他文档格式时包括图层。默认值：true
initial-printstate	（布尔值）图层的建议打印状态。如果为 true ， Acrobat 将在打印文档时包括图层。默认值：true
initial-viewstate	（布尔值）图层的建议查看状态。如为 true ， Acrobat 将在打开文档时显示图层。默认值：true
intent	（关键字）图形的预期用途： 查看或设计 。默认值： View
language	（选项列表）指定图层的语言： lang （字符串；必需的）在表 2.3 中描述的 lang 选项的语言和可能的区域设置。 preferred （布尔值）如果为 true ，且图层语言只部分匹配系统语言，将使用此图层。默认值：false

表 5.7 PDF_define_layer() 的选项 （续）

选项	说明
onpanel	（布尔值）如果为 false ，将不在 Acrobat 的图层面板显示图层名称，因此用户不能操作图层。 默认值： true
pageelement	（关键字）指定图层包含分页伪像： HF （页眉 / 页脚）、 FG （前景图像或图形）、 BG （背景图像或图形）或 L （徽标）之一。
printsubtype	（选项列表）指定图层是否用于打印： subtype （关键字） Trapping 、 PrintersMarks 或 Watermark 之一，指定图层内容的类型。 printstate （布尔值）如果为 true ， Acrobat 将在打印时启动图层内容。
zoom	（浮点型或百分比列表）一个或两个值，根据缩放因子指定图层的可见性（ 1.0 表示缩放因子为 100% ）。如果只提供一个值，将用作图层显示时使用的最大缩放因子；如果提供两个值，则指定最大和最小缩放因子。可使用关键字 maxzoom ，指定可能的最大缩放因子。

C++ Java

void set_layer_dependency(String type, String optlist)

Perl PHP

PDF_set_layer_dependency(resource p, string type, string optlist)

C

void PDF_set_layer_dependency(PDF *p, const char *type, const char *optlist)

定义图层间的层次、分组和锁定条件（要求 PDF 1.5）。

类型 依存关系的类型，必须为下列值之一：

- ▶ **GroupAllOn**: 如果 **group** 选项中指定的所有图层都可以显示，将显示 **depend** 选项中指定的图层。此类型的特定选项：**depend**、**group**
- ▶ **GroupAnyOn**: 如果 **group** 选项中指定的任一图层可以显示，将显示 **depend** 选项中指定的图层。此类型的特定选项：**depend**、**group**
- ▶ **GroupAllOff**: 如果 **group** 选项中指定的所有图层都不显示，将显示 **depend** 选项中指定的图层。此类型的特定选项：**depend**、**group**
- ▶ **GroupAnyOff**: 如果 **group** 选项中指定的任一图层不显示，将显示 **depend** 选项中指定的图层。此类型的特定选项：**depend**、**group**
- ▶ **Lock**: (PDF 1.6) **group** 选项中指定的图层将被锁定，即无法在 **Acrobat** 中交互更改它们的状态。此类型的特定选项：**group**
- ▶ **Parent**: 指定 **parent** 选项中指定的图层与 **children** 选项中指定的图层之间的层次关系。一个图层只能属于一个父级图层。此类型的特定选项：**children**、**parent**
- ▶ **Radiobtn**: 指定 **group** 选项中指定的各个图层间的单选按钮关系。这表示，一次最多只能显示该组中的一个图层；这对多语言图层很有用。此类型的特定选项：**group**
- ▶ **Title**: **parent** 选项中指定的图层句柄不能直接控制任何页面内容，但可以用作 **children** 选项中指定的图层的父级图层节点。此类型的特定选项：**children**、**parent**

optlist 选项列表，根据表 5.8 指定图层依存关系。

范围 文档、页面

C++ Java

void begin_layer(int layer)

Perl PHP

PDF_begin_layer(resource p, int layer)

C

void PDF_begin_layer(PDF *p, int layer)

开始一个图层，用于后续的页面输出（要求 PDF 1.5）。

layer 图层的句柄，必须用 **PDF_define_layer()** 返回。

详细说明 在此调用之后、但在任何后续 **PDF_begin_layer()** 或 **PDF_end_layer()** 调用之前置于页面上的所有内容都将构成指定图层的一部分。内容的可见性取决于图层的设置。

表 5.8 `PDF_set_layer_dependency()` 的选项

选项	说明
<i>children</i>	（图层句柄的列表；仅用于 <code>type=Parent</code> 和 <code>Title</code> 的情况）一个或多个图层句柄，指定从属于提供的父级图层的图层。
<i>depend</i>	（图层句柄；仅用于 <code>type=GroupAllOn</code> 、 <code>GroupAnyOn</code> 、 <code>GroupAllOff</code> 和 <code>GroupAnyOff</code> 的情况）由 <code>group</code> 选项中指定的图层控制的图层。
<i>group</i>	（图层句柄；仅用于 <code>type=GroupAllOn</code> 、 <code>GroupAnyOn</code> 、 <code>GroupAllOff</code> 、 <code>GroupAnyOff</code> 、 <code>Lock</code> 和 <code>Radiobtn</code> 的情况）构成该组的一个或多个图层句柄。对于 <code>type=Lock</code> 的情况，将锁定组内的所有图层。
<i>parent</i>	（图层句柄；仅用于 <code>type=Parent</code> 和 <code>Title</code> 的情况） <code>children</code> 选项中指定的图层的父级图层。

此函数将启动指定图层，并停用任何当前现用图层。

注释、图像、模板和表单域的图层也可以使用相应函数的 *layer* 选项来控制。在这些情况下，不需要调用 `PDF_define_layer()`。

范围 页面

C++	Java	<code>void end_layer()</code>
Perl	PHP	<code>PDF_end_layer(resource p)</code>
C		<code>void PDF_end_layer(PDF *p)</code>

停用所有活动图层（要求 PDF 1.5）。

详细说明 此调用之后置于页面上的内容不属于任何图层。所有图层都必须在页面结尾处结束。
只需调用一次 `PDF_begin_layer()` 即可从图层 A 切换到图层 B；不需要显示调用 `PDF_end_layer()` 来关闭图层 A。只需使用 `PDF_end_layer()` 创建无条件内容（始终可见）并在页面的结尾关闭所有图层。

范围 页面

6 颜色函数

6.1 设置颜色和色彩空间

表 6.1 列出了本节的相关参数关键字名称（参见第 11 页上的第 2.1 节“参数处理”）。

表 6.1 PDF_get/set_parameter() 的色彩相关关键字

关键字	说明
preserveold-pantonenames	如果为 false，旧版 <i>Pantone</i> 专色名称将转换为对应的新颜色名称；否则，将保留旧名称。 默认值: false。范围: 任意
spotcolorlookup	如果为 false， <i>PDFlib</i> 将不使用它的内部专色名称数据库。可以使用此参数提供已知专色的自定义，这将为匹配其他应用程序使用的定义提供了变通方式。使用此功能时要小心，建议不使用此功能。默认值: true。范围: 任意

色彩空间 *PDFlib* 客户端可以指定用于填充和描边路径内部和文本字符的颜色。可以使用几个色彩空间指定颜色（每个列表项目都以 *PDF_setcolor()* 和色彩选项的相应的色彩空间关键字开头）：

- ▶ *gray*: 介于 0（表示黑色）和 1（表示白色）之间的灰度值；
- ▶ *rgb*: RGB 三原色组合，即用于指定红色 (red)、绿色 (green) 和蓝色 (blue) 的百分比的介于 0 和 1 之间的三个值；其中，(0, 0, 0) 表示黑色，(1, 1, 1) 表示白色。根据 *PDFlib* 的要求，介于 0 到 255 之间的常用 RGB 颜色值必须除以 255，计算出 0 到 1 之间的值。
- ▶ *cmymk*: 介于 0（表示无颜色）和 1（表示完全色）之间的四个 CMYK 值，分别表示青色 (cyan)、洋红色 (magenta)、黄色 (yellow) 和黑色 (black) 值；其中，(0, 0, 0, 0) 表示白色，(0, 0, 0, 1) 表示黑色。请注意，这不同于 RGB 规范。
- ▶ *iccbasedgra/rgb/cmyk*: 在 ICC 色彩特征描述文件的帮助下，指定基于 ICC 的颜色。
- ▶ *spot*: 专色（分色色彩空间）：一个预定义或任意命名的自定义颜色表示以上色彩空间的某一颜色；通常用于为打算在使用一种或多种自定义颜色的胶印机上印刷文档作准备。色调值（百分比）范围是从 0（表示无颜色）到 1（表示专色最大亮度）。
- ▶ *lab*: CIE L*a*b* 色彩空间内独立于设备的颜色由一个在 0-100 范围内的亮度值和两个在 -127 到 128 范围内的颜色值指定。
- ▶ *pattern*: 使用任意文本、矢量或图像图形组成的对象平铺图案。
- ▶ 着色（平滑混合）提供基于另一色彩空间的两种颜色之间的逐渐过渡。可以使用 *PDF_shading()* 创建着色。
- ▶ 索引色彩空间本身不是真正的色彩空间，而是其他色彩空间的有效代码。当导入索引（基于调色板）图像时，将自动生成索引色彩空间，但不能直接指定。

描边和填充操作的默认颜色是黑色。

选项列表中的色彩规范 有关选项列表中色彩数据类型的说明和示例，请参见第 5 页上的第 1.2 节“选项列表”。

C++ Java	<code>void setcolor(String fstyle, String colorspace, double c1, double c2, double c3, double c4)</code>
Perl PHP	<code>PDF_setcolor(resource p, string fstyle, string colorspace, float c1, float c2, float c3, float c4)</code>
C	<code>void PDF_setcolor(PDF *p, const char *fstyle, const char *colorspace, double c1, double c2, double c3, double c4)</code>

设置当前色彩空间和颜色。

fstyle *fill*、*stroke* 或 *fillstroke* 之一，指定是否将颜色设为填充、描边，或同时设为填充和描边。

colorspace *gray*、*rgb*、*cmymk*、*spot*、*pattern*、*iccbasedgray*、*iccbasedrgb*、*iccbasedcmymk* 或 *lab* 之一，指定色彩空间。

PDF/X-1a: 不允许 *colorspace=rgb*、*iccbasedgray/rgb/cmymk* 和 *lab*。

PDF/X-3: 除非 PDF/X 输出方法为灰度或 CMYK 设备，否则 *colorspace=gray* 要求事先设置 *PDF_begin_page_ext()* 中的 *defaultgray* 选项。除非 PDF/X 输出方法为 RGB 设备，否则 *colorspace=rgb* 要求事先设置 *PDF_begin_page_ext()* 中的 *defaultrgb* 选项。除非 PDF/X 输出方法为 CMYK 设备，否则 *colorspace=cmymk* 要求事先设置 *PDF_begin_page_ext()* 中的 *defaultcmymk* 选项。使用 *iccbasedgray/rgb/cmymk* 和 *lab* 颜色要求在输出方法中提供 ICC 色彩特征描述文件（在这种情况下，标准名称是不够的）。

PDF/A: *colorspace=gray* 要求已指定输出方法（任意类型）。*colorspace=rgb* 或 *cmymk* 分别要求已指定 RGB 或 CMYK 输出方法。

c1, c2, c3, c4 所选色彩空间的色值组件。这些值的解释取决于 *colorspace* 参数：

- ▶ *gray*: *c1* 指定灰色值；
- ▶ *rgb*: *c1*、*c2*、*c3* 分别指定红色、绿色和蓝色值。
- ▶ *cmymk*: *c1*、*c2*、*c3*、*c4* 分别指定青色、洋红色、黄色和黑色值；
- ▶ *iccbasedgray*: *c1* 指定灰色值；
- ▶ *iccbasedrgb*: *c1*、*c2*、*c3* 分别指定红色、绿色和蓝色值；
- ▶ *iccbasedcmymk*: *c1*、*c2*、*c3*、*c4* 分别指定青色、洋红色、黄色和黑色值；
- ▶ *spot*: *c1* 指定 *PDF_makespotcolor()* 返回的专色句柄，而 *c2* 指定介于 0 和 1 之间的色调值；
- ▶ *lab*: *c1*、*c2* 和 *c3* 分别指定 CIE L*a*b* 色彩空间中的色值，用 D50 标准光源解释。*c1* 指定介于 0 到 100 之间的 L*（亮度），而 *c2*、*c3* 指定介于 -127 到 128 之间的 a*、b*（色度）值。
- ▶ *pattern*: *c1* 指定 *PDF_begin_pattern()* 或 *PDF_shading_pattern()* 返回的图案句柄。

详细说明 *gray*、*rgb* 和 *cmymk* 色彩空间的所有颜色值以及 *spot* 色彩空间的 *tint* 值都必须为 0 到 1 之间的数值，包括 0 和 1。不使用的参数应设为 0。

在每页开头，*gray*、*rgb* 和 *cmymk* 色彩空间的填充和描边颜色值设为黑色的缺省值。专色和图案颜色没有缺省值。

如果使用 *iccbasedgray/rgb/cmyk* 色彩空间，则在使用 *setcolor:iccprofilegray/rgb/cmyk* 参数之前，必须已设置好合适的 ICC 色彩特征描述文件（参见表 6.3）。

范围 页面、图案（仅限于图案的绘画类型为 1 的情况）模板、字形（仅限于 Type 3 字体的 *colorized* 选项为 *true* 的情况）、文档；图案颜色不能在自己的定义中使用。在文档范围中设置颜色可能对使用 *PDF_makespotcolor()* 定义专色很有用。

参数 *setcolor:iccprofilegray/rgb/cmyk*

C++ Java *int makespotcolor(String spotname)*

Perl PHP *int PDF_makespotcolor(resource p, String spotname)*

C *int PDF_makespotcolor(PDF *p, const char *spotname, int reserved)*

查找内置专色名称，或通过当前填充色创建命名专色。

spotname 内置专色的名称，或要定义的专色的任意名称。此名称最长不得超过 126 字节。专色名称只支持单字节（8 位）字符，不支持 Unicode 或嵌入的 null 字符。PDF/X-1a 模式不支持 PANTONE® 颜色。

reserved （仅用于 C 语言绑定）保留，必须为 0。

返回值 颜色句柄，可用于整个文档中的后续 *PDF_setcolor()* 调用。可在所有页面之间重复使用专色句柄，但不能跨越不同文档。在一个文档中，专色的数量不受任何限制。

详细说明 如果 *spotname* 在内部颜色表中是已知的，并且 *spotcolorlookup* 参数为 *true*（这是缺省值），将使用提供的专色名称。否则，将使用当前填充色的（CMYK 或其他）颜色值定义新专色的外观。这些替代值只用于屏幕预览和低端打印。提供的专色名称将用来生成分色，而不使用替代值。

如果已在前一个 *PDF_makespotcolor()* 调用中使用了 *spotname*，返回值将与前一次调用中的值相同，不会影响当前颜色。

可使用特殊专色名称 *All* 将颜色应用到所有分色，这对绘制注册商标很有用。专色名称 *None* 将在任何分色上生成不可见的输出。

范围 页面、图案、模板、字形（仅限于 Type 3 字体的 *colorized* 选项为 *true* 的情况）文档；如果定义自定义颜色，当前填充色不能是专色或图案。

参数 *spotcolorlookup*、*preserveoldpantonenames*

6.2 ICC 色彩特征描述文件

C++ Java

int load_iccprofile(String profilename, String optlist)

Perl PHP

int PDF_load_iccprofile(resource p, string profilename, string optlist)

C

int PDF_load_iccprofile(PDF *p, const char *profilename, int len, const char *optlist)

搜索 ICC 色彩特征描述文件 (ICC Profile)，并准备稍后使用。

profilename （名称字符串）*ICCProfile* 资源的名称、基于磁盘的文件名称或虚拟文件名称，或表 6.5 中列出的 PDF/X 的标准输出条件名称之一。只有 *usage* 选项设置为 *outputintent*，才允许使用 PDF/X 的标准输出条件名称。可以使用 *StandardOutputIntent* 资源定义其他标准输出方法。

len （仅用于 C 语言绑定）UTF-16 字符串的 *profilename* 长度（以字节为单位）。如果 *len = 0*，则必须提供以 *null* 终结的字符串。

optlist 选项列表，根据表 6.2 描述描述文件处理的各个方面。可以使用下列选项：*description*、*embedprofile*、*errorpolicy*、*metadata*、*usage*

表 6.2 PDF_load_iccprofile() 的选项

关键字	说明和可能的值
description	（字符串）仅当 <i>usage=outputintent</i> ，才使用此选项。它包含对 <i>ICC</i> 色彩特征描述文件的可读描述；此描述文件将与 <i>PDF/X</i> 输出方法一起使用。默认值：如果 <i>profilename</i> 指的是标准输出方法，将从内部列表中提取此描述；否则，将不做任何描述。
embedprofile	（布尔值）仅当 <i>usage=outputintent</i> ，才使用此选项。即使已提供 <i>PDF/X</i> 的标准输出方法作为 <i>profilename</i> ，也要强制使用嵌入的 <i>ICC</i> 色彩特征描述文件。默认值：false
errorpolicy	（关键字）控制出现错误时的行为（参见第 28 页上的第 2.6 节“异常处理”）
metadata	（选项列表；PDF1.4）提供描述文件的元数据（参见第 144 页上的第 12.2 节“XMP 元数据”）
usage	（关键字）描述 <i>ICC</i> 色彩特征描述文件的预期用途（缺省值：iccbased） iccbased <i>ICC</i> 色彩特征描述文件将用来定义文本或图形的基于 <i>ICC</i> 的色彩空间，或应用到图像。 outputintent <i>ICC</i> 色彩特征描述文件将用于定义输出描述（主要用于 <i>PDF/X</i> 或 <i>PDF/A</i> ）。

返回值 描述文件句柄，可在后续 *PDF_load_image()* 调用中使用，或用来设置描述文件相关的参数。必须检查返回值是否为 -1（在 PHP 中为：0）；这表示出现错误。不能在多个 PDF 文档之间使用返回的描述文件句柄。而且，如果 *usage* 选项为 *outputintent*，则不能将返回句柄应用到图像。在一个文档中，*ICC* 色彩特征描述文件的数量不受任何限制。如果函数调用失败，则可以使用 *PDF_get_errmsg()* 请求返回失败的原因。

详细说明 如果 *usage* 选项为 *iccbased*，将根据描述文件搜索策略搜索指定的描述文件。如果找到描述文件，将检查它是否适用（例如，颜色组件的数量）。*sRGB* 描述文件始终都可在内部使用，但不能进行配置。

如果 *usage* 为 *outputintent*，将首先在内部标准输出描述列表中搜索指定的描述文件。如果搜索不成功，将在用户描述的输出描述列表中搜索该名称。如果已找到提供的名称，并根据内置或用户配置列表确认为标准输出描述，将不再搜索 ICC 色彩特征描述文件；而且，*description* 选项提供的名称将嵌入到 PDF 输出中，用作 PDF/X 输出描述。如果发现此名称不是标准输出描述标识符，则将其用作描述文件名称，并将对应的 ICC 色彩特征描述文件作为输出描述嵌入 PDF。

PDF/X: 必须使用此函数或使用 *PDF_process_pdi()* 复制导入文档的输出描述来设置输出描述。

PDF/A: 可以使用此函数或使用 *PDF_process_pdi()* 复制导入文档的输出描述来设置输出描述。但是，只要文档中使用了设备无关的颜色，就不要求任何输出描述。ICC 色彩特征描述文件必须符合 ICC 规范 ICC.1:1998-09 及其附录 ICC.1A:1999-04（内部描述文件版本为 2.2.0）。

范围 文档；应在 *PDF_begin_document()* 之后立即设置输出方法。如果 *usage=iccbased*，同时允许下列范围：页面、图案、模板、字形。

参数 请参见表 6.3 和表 6.4

表 6.3 *PDF_get/set_parameter()* 的关键字（参见第 11 页上的第 2.1 节“参数处理”）

关键字	说明
<i>ICCPProfile</i> <i>StandardOutputIntent</i>	如 <i>UPR</i> 文件中相关类别所示的对应资源文件行。可由多个调用向内部列表添加新条目（另请参见表 2.1 中的 <i>resourcefile</i> 。范围：任意
<i>iccwarning</i>	不常用，使用 <i>errorpolicy</i>

表 6.4 *PDF_get/set_value()* 的关键字（参见第 11 页上的第 2.1 节“参数处理”）

关键字	说明
<i>iccomponents</i>	<i>modifier</i> 中提供的句柄引用的 ICC 色彩特征描述文件内的颜色组件数量。
<i>setcolor:icc-profilegray</i>	ICC 色彩特征描述文件；它指定用于 <i>PDF_setcolor()</i> 的灰色色彩空间。范围：文档、页面、图案、模板、字形
<i>setcolor:icc-profilergb</i>	ICC 色彩特征描述文件；它指定用于 <i>PDF_setcolor()</i> 的 RGB 色彩空间。范围：文档、页面、图案、模板、字形
<i>setcolor:icc-profilecmyk</i>	ICC 色彩特征描述文件；它指定用于 <i>PDF_setcolor()</i> 的 CMYK 色彩空间。范围：文档、页面、图案、模板、字形

表 6.5 PDF/X 的标准输出方法（有关详细信息，请参见 www.color.org）

名称	印刷过程
CGATS TR 001	美国 SWOP（出版物）印刷 ANSI CGATS.6.
FOGRA27	ISO/DIS 12647-2:2004、符合 ISO 12647-2 的商业和专业胶版印刷、阳图版、1 或 2 型纸（光泽或无光涂布胶版纸、115 g/m ² ）、网屏频率 60/cm。
FOGRA28	ISO/DIS 12647-2:2004、符合 ISO 12647-2 的商业和专业胶版印刷、阳图版、3 型纸（涂布卷筒纸、60 g/m ² ）、网屏频率 60/cm。
FOGRA29	ISO/DIS 12647-2:2004、符合 ISO 12647-2 的商业和专业胶版印刷、阳图版、4 型纸（无涂布白色胶版纸、120 g/m ² ）、网屏频率 60/cm。
FOGRA30	ISO/DIS 12647-2:2004、符合 ISO 12647-2 的商业和专业胶版印刷、阳图版、5 型纸（无涂布稍黄胶版纸、115 g/m ² ）、网屏频率 60/cm。
FOGRA31	ISO/DIS 12647-2:2003、符合 ISO 12647-2 的连续表格印刷、阳图版、2 型纸（无光涂布胶版纸、115 g/m ² ）、网屏频率 60/cm。
FOGRA32	ISO/DIS 12647-2:2004、符合 ISO 12647-2 的连续表格印刷、阳图版、4 型纸（白色无涂布胶版纸、80 g/m ² ）网屏频率 60/cm。
FOGRA33	ISO/DIS 12647-2:2004、符合 ISO 12647-2 的连续表格印刷、阳图版、2 型纸（无光涂布胶版纸、115 g/m ² ）、网屏频率 54/cm。
FOGRA34	ISO/DIS 12647-2:2004、符合 ISO 12647-2 的连续表格印刷、阳图版、4 型纸（白色无涂布胶版纸、120 g/m ² ）、网屏频率 60/cm。
FOGRA35	ISO/DIS 12647-2:2004、符合 ISO 12647-2 的连续表格印刷、阴图版、2 型纸（无光涂布胶版纸、115 g/m ² ）、网屏频率 54/cm。
FOGRA36	ISO/DIS 12647-2:2004、符合 ISO 12647-2 的连续表格印刷、阴图版、4 型纸（白色无涂布胶版纸、120 g/m ² ）、网屏频率 54/cm。
FOGRA37	ISO/DIS 12647-2:2004、符合 ISO 12647-2 的连续表格印刷、阴图版、2 型纸（无光涂布胶版纸、115 g/m ² ）网屏频率 60/cm。
FOGRA38	ISO/DIS 12647-2:2004、符合 ISO 12647-2 的连续表格印刷、阴图版、4 型纸（白色无涂布胶版纸、120 g/m ² ）、网屏频率 60/cm。
IFRA26	ISO/DIS 12647-3:2004、冷凝胶版印刷、接触曝光的阴图版或计算机直接制版（阶调值提高 26%）新闻印刷、网屏线数 40/cm。
IFRA30	ISO/DIS 12647-3:2004、冷凝胶版印刷、接触曝光的阴图版或计算机直接制版（阶调值提高 30%）新闻印刷、网屏线数 40/cm（主要适用于 USA）。
JC200103	Japan Color 2001 涂布纸：ISO 12647-2:2004、单张纸胶印、阳图版、3 型纸（涂布纸、105 gsm）、网屏频率 69/cm。
JC200104	Japan Color 2001 无涂布纸：ISO 12647-2:2004、单张纸胶印、阳图版、4 型纸（无涂布、105 gsm）、网屏频率 69/cm。
JCN2002	用于新闻印刷的 Japan Color 2002：ISO/DIS 12647-3:2004、冷凝胶版印刷、阴图版、新闻印刷、网屏频率 39/cm。
JCW2003	用于卷筒纸胶印的 Japan Color 2003：ISO 12647-2:2004、热定型卷筒纸胶版印刷、阳图版、3 型纸（涂布纸、70 gsm）、网屏频率 69/cm。
下面是一些不常用的标准输出方法，即使仍然可用，也不应使用它们：	
FOGRA11-FOGRA26、	
IFRA 22、IFRA 28、	
OF COM PO P1 F6o、OF COM PO P2 F6o、OF COM PO-P3 F6o、OF COM PO P4 F6o、OF COM NE P1 F6o、OF COM NE P2 F6o、OF COM NE P3 F6o、OF COM NE P4 F6o、SC GC2 CO F3o、Ifra NP_4olcm_neg+CTP_05.00	

6.3 图案和着色

C++ Java	<code>int begin_pattern(double width, double height, double xstep, double ystep, int painttype)</code>
Perl PHP	<code>int PDF_begin_pattern(resource p, float width, float height, float xstep, float ystep, int painttype)</code>
C	<code>int PDF_begin_pattern(PDF *p, double width, double height, double xstep, double ystep, int painttype)</code>

开始图案定义。

width, height 图案定界框的尺寸（以点数为单位）。

xstep, ystep 重复放置图案以描边或填充某个对象时的偏移量。大多数应用程序会将他们分别设为图案的 **width** 和 **height** 值。

painttype 如果 **painttype** 为 1，图案必须包含自己的颜色规范；将在使用图案时应用此颜色图案；如果 **painttype** 为 2，图案不能包含任何颜色规范，使用该图案进行填充或描边时将应用当前填充色或描边色。

返回值	图案句柄；在关闭的文档范围内，可在后续 <code>PDF_setcolor()</code> 调用中使用此句柄。
详细说明	此函数将所有文本、图形和颜色状态参数重置为其缺省值，并会根据全局 <code>topdown</code> 参数建立坐标系。图案定义过程中，不能使用超文本函数和打开图像的函数，但可以使用所有文本、图形和颜色函数（正在定义过程中的图案除外）。
范围	文档、页面；此函数是图案范围的开头，必须与对应的 <code>PDF_end_pattern()</code> 调用成对使用。
参数	<code>topdown</code>

C++ Java	<code>void end_pattern()</code>
Perl PHP	<code>PDF_end_pattern(resource p)</code>
C	<code>void PDF_end_pattern(PDF *p)</code>

完成图案定义。

范围	图案；此函数结束图案范围，必须与对应的 <code>PDF_begin_pattern()</code> 调用成对使用。
----	--

C++ Java	<code>int shading_pattern(int shading, String optlist)</code>
Perl PHP	<code>int PDF_shading_pattern(resource p, int shading, string optlist)</code>
C	<code>int PDF_shading_pattern(PDF *p, int shading, const char *optlist)</code>

使用着色对象定义着色图案（要求 PDF 1.4）。

shading `PDF_shading()` 返回的着色句柄。

optlist 选项列表，根据表 6.6 描述着色图案的各个方面。可以使用下列选项：`gstate`

返回值	图案句柄；在关闭的文档范围内，可在后续 <code>PDF_setcolor()</code> 调用中使用此句柄。
详细说明	此函数可用来使用着色填充任意对象。为此，必须使用 <code>PDF_shading()</code> 获取一个着色句柄，然后使用 <code>PDF_shading_pattern()</code> ，基于此着色定义图案。最后，可将图案句柄应用到 <code>PDF_setcolor()</code> ，将当前颜色设为着色图案。

范围 文档、页面、字体

表 6.6 PDF_shading_pattern() 的选项

关键字	说明和可能的值
<i>gstate</i>	(句柄) 图形状态句柄

C++ Java

void shfill(int shading)

Perl PHP

PDF_shfill(resource p, int shading)

C

void PDF_shfill(PDF *p, int shading)

基于着色对象，使用着色填充某个区域（要求 PDF 1.4）。

shading PDF_shading() 返回的着色句柄。

详细说明 通过此函数，可在不涉及 PDF_shading_pattern() 和 PDF_setcolor() 的情况下使用着色。但是，这只适用于简单的形状；其中，待填充对象的几何形状与着色本身完全相同。由于当前剪切区域将被着色（根据着色的 extend0 和 extend1 选项），此函数通常会与 PDF_clip() 结合使用。

范围 页面、图案（仅限于图案的绘画类型为 1 的情况）、模板、字形（仅限于 Type 3 字体的 colored 选项为 true 的情况）、文档

C++ Java

int shading(String shtype, double xo, double yo, double x1, double y1,
double c1, double c2, double c3, double c4, String optlist)

Perl PHP

int PDF_shading(resource p, string shtype, float xo, float yo, float x1, float y1,
float c1, float c2, float c3, float c4, string optlist)

C

int PDF_shading(PDF *p, const char *shtype, double xo, double yo, double x1, double y1,
double c1, double c2, double c3, double c4, const char *optlist)

通过当前填充色定义与另一颜色的混合效果（要求 PDF 1.4 或更高版本）。

shtype 着色的类型；对于线性着色，必须是 axial，或对于圆形着色必须是 radial。

xo, yo, x1, y1 对于轴向着色，(xo, yo) 和 (x1, y1) 是着色起点和端点的坐标。对于辐射着色，这些点可指定开始和结束圆形的圆心。

c1, c2, c3, c4 着色端点的颜色值；在当前填充色色彩空间中的解释方法与 PDF_setcolor() 中的颜色参数相同。如果当前填充色色彩空间为专色色彩空间，将忽略 c1，且 c2 将包含色调值。

optlist 选项列表，根据表 6.7 描述着色的各个方面。可以使用下列选项：antialias、extend0、extend1、N、ro、r1

返回值 着色句柄；在关闭的文档范围内，可在后续 PDF_shading_pattern() 和 PDF_shfill() 调用中使用此句柄。

详细说明 当前填充色将用作起始颜色，它不能基于图案。

范围 文档、页面、字体

表 6.7 PDF_shading() 的选项

关键字	说明和可能的值
<i>antialias</i>	(布尔值) 指定是否为着色启动消除锯齿功能。默认值: false
<i>extend0</i>	(布尔值) 指定是否将着色延伸出起点。默认值: false
<i>extend1</i>	(布尔值) 指定是否将着色延伸出结点。默认值: false
<i>N</i>	(浮点值) 颜色过渡函数的指数, 必须为 > 0 。缺省值: 1
<i>r0</i>	(浮点值; 仅用于辐射着色, 这里要求此选项) 起始圆形的半径
<i>r1</i>	(浮点值; 仅用于辐射着色, 这里要求此选项) 结束圆形的半径



7 图像和模板函数

表 7.1 和表 7.2 列出了本节的相关参数及相应的值关键字名称（参见第 11 页上的第 2.1 节“参数处理”）。

表 7.1 PDF_get/set_parameter() 的图像相关关键字

关键字	说明
honoriccprofile	读取图像内嵌入的 ICC 色彩特征描述文件，并将其应用到图像数据。默认值：true
imagewarning	不常用，使用 errorpolicy
renderingintent	图像的渲染方法。默认值：Auto Auto 不在 PDF 文件中指定任何渲染方法，而改为使用设备的默认渲染方法。 典型使用：未知情况 AbsoluteColorimetric 未对设备的白点（例如白纸）做出任何校正。色域外的颜色都将映射为设备的色域内最接近的值。典型使用：纯色的准确再现；不建议作其他使用。 RelativeColorimetric 颜色数据缩放到设备的色域内，原白点经微调后映射到另外一个白点。 典型使用：矢量图形 Saturation 当颜色值可能被移位时，将保留颜色的饱和度。典型使用：商业图形 Perceptual 通过修改色域内和色域外的颜色保留颜色关系，以提供令人满意的外观。 典型使用：扫描的图像

表 7.2 PDF_get/set_value() 的图像相关关键字

关键字	说明
imagewidth¹ imageheight¹	分别获取图像的宽度或高度，以像素为单位。modifier 是所选图像的整型句柄。范围：页面、图案、模板、字形、文档、路径
image:iccprofile¹	返回 modifier 中提供的图像句柄引用的图像内所嵌入的 ICC 色彩特征描述文件句柄。
orientation¹	获取图像的方向值。modifier 是所选图像的整型句柄。对于包含方向标记的 TIFF 图像，将返回此标记的值；在所有其他情况下，将返回 1；同时 PDFlib 也将相应调整图像数值以保持原貌。范围：页面、图案、模板、字形、文档、路径
resx¹ resy¹	分别获取图像的水平或垂直分辨率。modifier 是所选图像的整型句柄。范围：页面、图案、模板、字形、文档、路径。 如果值为正值，返回值为图像分辨率，单位是像素 / 英寸 (dpi)。如果返回值为负值，可使用 resx 和 resy 查找非方形像素的长宽比，但不具有任何绝对意义。如果返回值为零，则表示图像的分辨率未知。

1. 仅用于 PDF_get_value()

7.1 图像

C++ Java

int load_image(String imagetype, String filename, String optlist)

Perl PHP

int PDF_load_image(resource p, string imagetype, string filename, string optlist)

C

int PDF_load_image(PDF *p,
const char *imagetype, const char *filename, int len, const char *optlist)

根据各种选项打开基于磁盘的或虚拟的图像文件。

imagetype 字符串 *auto* 指示 PDFlib 自动检测图像文件类型（不能用于 CCITT 和原始图像）。用 *bmp*、*ccitt*、*gif*、*jpeg*、*jpeg2000*、*png*、*raw* 或 *tiff* 字符串明确指定图像格式可使性能略有提高。*jpeg2000* 类型要求 PDF 1.5 或更高版本，并且在 PDF/A 或 PDF/X 模式下不可用。类型 *ccitt* 不同于 TIFF，此类型包含 CCITT 压缩图像数据。

filename （名称字符串）通常指定要打开的图像文件的文件名。此名称必须为基于磁盘的文件或虚拟文件的文件名；PDFlib 不会从 URL 提取图像数据。

如果未能找到具有指定文件名的文件且 *imagetype=auto*，则 PDFlib 将尝试自动确定合适的文件名后缀；它会将下面列表中的所有后缀（包括小写和大写后缀）都附加到指定的 *filename*，然后尝试在 *searchpath* 指定的目录中查找具有该名称的文件：

.bmp、*.ccitt*、*.g3*、*.g4*、*.fax*、*.gif*、*.jpg*、*jpeg*、*.jpx*、*.jp2*、*.jpf*、*.jpm*、*.j2k*、*.png*、*.raw*、*.tif*、*.tiff*

len （仅用于 C 语言绑定）UTF-16 字符串的 *filename* 长度（以字节为单位）。如果 *len = 0*，则必须提供以 null 终结的字符串。

optlist 选项列表，根据表 7.3 指定图像相关属性。可以使用下列选项：

- ▶ 颜色相关的选项：*colorize*、*honor-iccprofile*、*iccprofile*、*invert*、*rendering-intent*
- ▶ Clipping 和 masking 选项：*clipping-pathname*、*honor-clippingpath*、*ignoremask*、*mask*、*masked*
- ▶ 有关使用图像的 PDF 特殊功能：*iconname*、*template*
- ▶ raw 和 CCITT 图像的选项：*bitreverse*、*bpc*、*components*、*height*、*K*、*width*
- ▶ 有关处理图像数据的选项：*ignore-orientation*、*inline*、*page*、*passthrough*
- ▶ 其他选项：*hypertext-encoding*、*errorpolicy*、*interpolate*、*layer*、*metadata*、*OPI-1.3*、*OPI-2.0*

返回值 图像句柄，可用于后续的图像相关调用。必须检查返回值是否为 -1（在 PHP 中：为 0）；这表示出现错误。不能在多个 PDF 文档之间使用返回的图像句柄。如果函数调用失败，则可以使用 *PDF_get_errmsg()* 请求返回失败的原因。

详细说明 此函数将打开并分析使用 *imagetype* 参数中确定的一种支持格式的光栅图形，并将相关图像数据复制至输出文档。此函数不会在输出上显示任何效果。要将导入图像实际置于生成的输出文档内，必须使用 *PDF_fit_image()*。不提倡在生成同一文档时多次打开同一图像；因为这样，实际的图像数据将向输出文档复制多次。

PDFlib 将根据 *filename* 打开图像文件，处理内容，并在返回之前关闭文件。尽管可以在文档中多次放置图像（参见 *PDF_fit_image()*），但实际的图像文件不会在此调用之后继续保持打开状态。

如果 *imagetype* = *raw* 或 *ccitt*，PDFlib 不能根据图像数据推算出 *width*、*height*、*components* 和 *bpc* 选项的值，故必须提供其相应值用户应负责提供与图像实际匹配的选项值。否则，可能会生成损坏的 PDF 输出，且 Acrobat 可能返回图像数据不足的错误信息。

如果 *imagetype* = *raw*，提供的图像数据的长度必须等于 $[width \times components \times bpc / 8] \times height$ （以字节为单位），括号内的项目将使用最接近的较大整数。图像样本将按照标准 PostScript/PDF 的顺序，即：由上而下、从左到右（假定没有坐标转换）。必须提供 16 位样本，首先提供最重要的字节（big-endian 或 »Mac« 字节顺序）。像素值的极性将在第 89 页上的第 6.1 节“设置颜色和色彩空间”讨论。如果 *bpc* 小于 8，则每个像素行都以字节边界开头，且颜色值必须从左到右封装在一个字节内。图像样本始终都采用交替顺序，即：首先提供第一个像素的所有颜色值，然后提供第二个像素的所有颜色值，依此类推。

PDF/X-1a: 不允许 RGB 图像。

PDF/X-3: 除非 PDF/X 输出方法为灰度或 CMYK 设备，否则灰度图像要求事先设置 *PDF_begin_page_ext()* 中的 *defaultgray* 选项。除非 PDF/X 输出方法为 RGB 设备，否则 RGB 图像要求事先设置 *PDF_begin_page_ext()* 中的 *defaultrgb* 选项。除非 PDF/X 输出方法为 CMYK 设备，否则 CMYK 图像要求事先设置 *PDF_begin_page_ext()* 中的 *defaultcmky* 选项。

PDF/A: 灰度图像要求已事先指定输出方法。RGB 或 CMYK 图像分别要求已事先指定 RGB 或 CMYK 输出方法。

范围 如果未提供 *inline* 选项，范围为文档、页面、字体；而且，此函数必须与对应的 *PDF_close_image()* 调用成对使用。在 *document* 或 *font* 范围内而不在 *page* 范围内加载图像可略微优化输出大小。
如果提供了 *inline* 选项，则范围为 *page*, *pattern*, *template*, *glyph*，并且禁止调用 *PDF_close_image()*。

参数 请参见表 7.1 和表 7.2

表 7.3 *PDF_load_image()* 的选项

关键字	说明
<i>bitreverse</i>	（布尔值；仅用于 <i>imagetype</i> = <i>ccitt</i> 的情况）如果为 <i>true</i> ，则对压缩数据中的所有字节执行位反转。 默认值: <i>false</i>
<i>bpc</i>	（整型；仅用于 <i>imagetype</i> = <i>raw</i> 的情况且是必需的）每个组件的位数，必须为 1、2、4 或 8。 在 PDF 1.5 中，也允许 <i>bpc</i> =16。
<i>clipping-pathname</i>	（字符串；仅用于 <i>imagetype</i> = <i>tiff</i> 和 <i>jpeg</i> 的情况；如果 <i>honorclippingpath</i> = <i>false</i> ，将忽略此选项）读取图像文件中具有指定名称的路径，并将其用作剪切路径。图像文件中必须存在指定的路径。 默认值: 图像文件中作为剪切路径提供的路径名称。
<i>colorize</i>	（专色句柄；在提供了 <i>iccprofile</i> 选项时忽略；不能用于 <i>imagetype</i> = <i>jpeg2000</i> 的情况）用专色句柄着色图像，必须事先使用 <i>PDF_makespotcolor()</i> 获取句柄。图像必须为黑白图像或灰度图像。
<i>components</i>	（整型；仅用于 <i>imagetype</i> = <i>raw</i> 的情况且是必需的）图像组件（通道）的数量，必须为 1、3 或 4。
<i>errorpolicy</i>	（关键字）控制出现错误时的行为（参见第 28 页上的第 2.6 节“异常处理”）
<i>height</i>	（整型；仅用于 <i>imagetype</i> = <i>raw</i> 和 <i>ccitt</i> 的情况，且是必需的）以像素为单位的图像高度。
<i>honor-iccprofile</i>	（布尔值；仅用于 <i>imagetype</i> = <i>jpeg</i> 、 <i>png</i> 和 <i>tiff</i> 的情况；在提供了 <i>colorize</i> 选项时将设置为 <i>false</i> ）读取嵌入的 ICC 色彩特征描述文件（如果有）并将其应用到图像。默认值: <i>honoriccprofile</i> 参数的值。

表 7.3 PDF_load_image() 的选项（续）

关键字	说明
honor-clippingpath	（布尔值；仅用于 imagetype=tiff 和 jpeg 的情况）读取图像文件中的剪切路径（如果有），并将其应用到图像。默认值：true
hypertext-encoding	（关键字）指定 iconname 选项的编码。空字符串相当于 unicode。默认值：全局 hypertextencoding 参数的值。
iccprofile	（ICC 句柄；仅用于 imagetype=jpeg、png 和 tiff 的情况）应用到图像的 ICC 色彩特征描述文件的句柄。默认值：如果图像中存在一个描述文件且 honoriccprofile=true，则为嵌入的描述文件。
iconname	（超文本字符串；如果 inline=true，将忽略此选项；如果 template=true，则是必需的）向图像附加一个名称，以便能够通过 JavaScript 引用，例如：将图像用作表单域的图标。
ignoremask	（布尔值）忽略图像中的透明度信息。默认值：false
ignore-orientation	（布尔值；仅用于 imagetype=tiff 的情况）忽略图像中的所有方向标记。这可能对弥补错误的方向信息很有用。默认值：false
imagewarning	不常用，使用 errorpolicy
inline	（布尔值；仅用于 imagetype=ccitt、jpeg 和 raw 的情况；建议仅在加载 Type 3 字体的位图字形时使用）如果为 true，图像将直接写入页面、图案、模板或字形描述的内容流中（建议仅用于 Type 3 字体中的字形）。
interpolate	（布尔值；对于 PDF/A，必须为 false）启用图像插值处理，以改善图像在屏幕和纸张上的显示效果。这对使用 Type 3 字体的字形描述的位图图像很有用。默认值：false
invert	（布尔值；除非 mask=true，否则不能用于 imagetype=jpeg2000 的情况）对图像进行反向处理（亮色和暗色互换）。这可用作一种变通方法，处理应用程序会有不同解释的图像。默认值：false
K	（整型；仅用于 imagetype=ccitt 的情况）用于压缩方案选择的 CCITT 参数。默认值：0 -1 G4 压缩 0 1 维 G3 压缩 (G3-1D) 1 混合 1 维和 2 维压缩 (G3, 2-D)
layer	（图层句柄；PDF 1.5）图像所属的图层。仅当相应图层可见时，图像才可见。请注意，图像将绑定到图层。如果需要将同一图像放置到属于不同图层的不同位置上，请在放置图像之前使用 PDF_begin_layer()。
mask	（布尔值；仅用于具有一个颜色组件的图像，包括索引颜色）。图像将用作蒙版。对于 1 位蒙版此选项是必需的，但对于像素位数大于 1 的蒙版此选项是可选的。但是，位数大于 1 的蒙版要求 PDF 1.4。默认值：false。蒙版有两种用途： ▶ 对另一图像蒙版：返回的图像句柄可用于后续调用，以打开另一图像，也可以提供给 masked 选项。 ▶ 放置一个着色透明图像：将图像中的 0 位像素视为透明，并使用当前填充色对 1 位像素着色。
masked	（图像句柄）作为蒙版应用到当前图像的图像句柄。图像句柄事先已通过 PDF_load_image() 调用返回，且尚未关闭。在 PDF 1.3 兼容模式下，蒙版句柄必须指向一个 1 位图像，并且必须已使用 mask 选项加载。在 PDF/A 和 PDF/X 模式下，此选项仅用于 1 位蒙版。
metadata	（选项列表；PDF 1.4）提供图像的元素数据（参见第 144 页上的第 12.2 节“XMP 元数据”）。
OPI-1.3	（选项列表；不适用于 PDF/A 和 PDF/X）选项列表，其中包含用作选项名称的 OPI 1.3 PostScript 注释；下列是必选条目： ALDImageFilename（字符串）、ALDImageDimensions（整数值列表）、ALDImageCropRect（具有整数值矩形）、ALDImagePosition（浮点值列表） 下列是可选条目： ALDImageID（字符串）、ALDObjectComments（字符串）、ALDImageCropFixed（矩形）、ALDImageResolution（浮点值列表）、ALDImageColorType（关键字：Process、Spot、Separation 之一；默认值：Spot）、ALDImageColorType（介于 0 到 1 之间的 4 个色彩值列表和一个色彩名称）、ALDImageTint（浮点值）、ALDImageOverprint（布尔值）、ALDImageType（整数值列表）、ALDImageGrayMap（整数值列表）、ALDImageTransparency（布尔值）、ALDImageAsciiTag（整数值 / 字符串对列表） 子选项 normalizefilename 控制文件名的处理：如果为 true，将按照 PDF 参考的规定进行规范化处理。如果为 false，将复制到输出，而不做任何修改。后者可能对使用某些无法正确处理规范化文件名的 OPI 服务器很有用。默认值：false

表 7.3 PDF_load_image() 的选项（续）

关键字	说明
OPI-2.0	（选项列表；不适用于 PDF/A 和 PDF/X）选项列表，其中包含用作选项名称的 OPI 2.0 PostScript 注释；下列是必选条目： ImageFilename（字符串） 下列条目应同时提供或同时省略： ImageCropRect（矩形）、ImageDimensions（浮点值列表） 下列是可选条目： MainImage（字符串）、TIFFASCIIITag（整数值 / 字符串对列表）、ImageOverprint（布尔值）、ImageInks（字符串 full_color、字符串 registration 或包含每个色料名称和色调的 monochrome 字符串和字符串 / 浮点值对的列表）、IncludedImageDimensions（整数列表）、IncludedImageQuality（具有 1、2 或 3 值的整数值） 同时，还支持 normalizefilename 选项（参见 OPI-1.3）。
page	（整数值；仅用于 imagedata=gif 或 tiff 的情况；如果用于其他格式，则必须为 1）根据给定数值，从多页图像文件中提取图像。第一个图像的编码为 1。缺省值：1
passthrough	（布尔值；仅用于 imagedata=tiff 或 jpeg 的情况）控制 TIFF 和 JPEG 图像数据的处理。
tiff	（默认值：true）如果为 true，压缩的 TIFF 图像数据将在可能的情况下直接传送到 PDF 输出。如果 TIFF 图像包含的数据受到破坏或者不完整，将此选项设为 false 会有所帮助。
jpeg	（默认值：false）如果为 false，PDFlib 将转换 JPEG 图像数据的代码以清理数据，使其与 Acrobat 相兼容。如果为 true，JPEG 图像数据将直接复制至 PDF 输出。对于 multiscan 和某些 CMYK JPEG 图像，将忽略此选项。将此选项设置为 true 可能会加快处理过程，但某些不常见的 JPEG 风格在 Acrobat 中将无法正确显示。
rendering-intent	（关键字）图像的渲染方法。有关可能的关键字列表及其含义，请参见表 7.1。默认值：全局 renderingintent 参数的值。
template	（布尔值）如果为 true，则生成嵌入在 Form XObject 中的 PDF Image XObject（在 PDFlib 中，称为模板），而不生成单纯的 Image XObject。这可用来创建只包含一个图像的表单域图标模板。使用 OPI-1.3 或 OPI-2.0 选项时，同样要求此选项，以便与某些 OPI 服务器相兼容。默认值：false。 范围：文档
width	（整数值；仅用于 imagedata=raw 和 ccitt 的情况；且是必需的）以像素为单位的图像宽度。

C++ Java void close_image(int image)

Perl PHP PDF_close_image(resource p, int image)

C void PDF_close_image(PDF *p, int image)

关闭图像。

image 由 PDF_load_image() 获取的有效图像句柄。

详细说明 此函数只会影响 PDFlib 的相关内部图像结构。如果已从文件中打开了图像，此调用不会影响实际图像文件；这是因为，已经在对应的 PDF_load_image() 调用结束时关闭了图像。由于此函数会断开 PDFlib 与图像的内部关联，所以用此函数关闭图像句柄后，将无法继续使用。

范围 文档、页面、字体；除非已经使用了 inline 选项，否则必须与对应的 PDF_load_image() 调用成对使用。

C++ Java	<code>void fit_image(int image, double x, double y, String optlist)</code>
Perl PHP	<code>PDF_fit_image(resource p, int image, float x, float y, string optlist)</code>
C	<code>void PDF_fit_image(PDF *p, int image, double x, double y, const char *optlist)</code>

根据各种选项，将图像或模板置于页面上的位置 (x,y)。

image 由 `PDF_load_image()` 或 `PDF_begin_template_ext()` 函数返回的有效图像或模板句柄。

x,y 用户坐标系统中的参考点坐标；将根据各种选项，在此坐标上放置图像或模板。

optlist 选项列表，根据表 7.4 指定详细的放置信息。可以使用下列选项：
`adjustpage`、`blind`、`boxsize`、`dpi`、`fitmethod`、`ignore-clippingpath`、`ignore-orientation`、`匹配框`、`orientate`、`position`、`rotate`、`scale`、`showborder`

详细说明 将相对于参考点 (x,y) 放置图像或模板（下文统称为对象）。缺省情况下，对象的左下角将置于参考点上。但是，`orientate`、`boxsize`、`position` 和 `fitmethod` 选项可修改此行为。缺省情况下，将根据其分辨率值缩放图像。可使用 `dpi`、`scale` 和 `fitmethod` 选项修改此行为。

范围 页面、图案、模板、字形（仅用于 Type 3 字体的 `colorized` 选项为 `true` 或图像为蒙版的情况）；只要尚未使用 `PDF_close_image()` 关闭图像句柄，便可在任意页面上多次调用此函数，次数不受任何限制。

表 7.4 `PDF_fit_image()` 和 `PDF_fit_pdi_page()` 的选项

关键字	说明
adjustpage	（布尔值； <code>blind=true</code> 时将忽略）将当前页面的尺寸调整为对象的尺寸，以使页面的右上角与对象的右上角加上 (x,y)x 和 y 参数的点重叠。将调整媒体框，且所有其他方框项都将重置为其默认值。 position 选项值为 0 时，应注意下列有用的情况： <code>x >= 0 且 y >= 0</code> 对象被白色边距环绕。边距的水平宽度为 y，垂直宽度为 x。 <code>x < 0 且 y < 0</code> 将从图像中裁切水平和垂直的条带。 此选项只在 <code>page</code> 范围内有效，且不能用于 <code>topdown</code> 参数设为 <code>true</code> 的情况。默认值：false
blind	（布尔值）如果为 <code>true</code> ，将完成所有定位和缩放计算，但不会将对象置于输出页面之上。这可用于处理页面上的块，而不实际使用页面的内容。默认值：false
boxsize	（浮点值列表）指定框宽度和高度的两个值，将相对于这两个值放置并可能缩放对象。框的左下角即为参考点 (x,y)。放置图像并使其填充此框的操作由 <code>position</code> 和 <code>fitmethod</code> 选项控制。如果 <code>width=0</code> ，则只考虑高度；如果 <code>height=0</code> ，则只考虑宽度。在这些情况下，将分别相对于 (x,y) 到 (x,y+height) 的垂直直线，或 (x,y) 到 (x+width,y) 的水平线放置对象。默认值：{0 0}
dpi	（浮点值列表）以像素 / 英寸为单位，用一或两个数值指定所需图像在水平和垂直方向的分辨率。如果只提供一个值，将同时用于两个方向。如果此值为 0，若图像内部的分辨率存在，将使用内部分辨率；否则，将使用 72 dpi 分辨率。除值 0 外，也可提供关键字 <code>internal</code> 。由此选项得到的缩放结果相对于当前用户坐标；如果已经过缩放，则生成的物理分辨率将不同于提供的值。 模板和 <code>PDF</code> 页面将忽略此选项，如果 <code>fitmethod</code> 选项已随 <code>auto</code> 、 <code>meet</code> 、 <code>slice</code> 或 <code>entire</code> 关键字一起提供，同样也会忽略此选项。默认值：internal

表 7.4 PDF_fit_image() 和 PDF_fit_pdi_page() 的选项（续）

关键字	说明
fitmethod	（关键字；除非提供了 boxsize ，否则将忽略此选项）指定使对象适合框的方法。默认值： nofit nofit 只确定对象位置，不进行任何缩放或剪裁处理。 clip 确定对象位置，并在框的边缘剪裁对象。 meet 根据 position 选项确定对象位置，并在保持长宽比的同时进行缩放以使其完全适合框。通常，至少有对象的两个边缘与框的对应边缘对齐。将忽略 dpi 和 scale 选项。 auto 相当于 meet 。 slice 根据 position 选项确定对象位置，并进行缩放以使其覆盖整个框，可在保持长宽比的同时确保至少对象的一个尺寸可完全包含在框内。通常，对象的某些其他尺寸会延伸到框外，因此将被剪裁。将忽略 dpi 和 scale 选项。 entire 根据 position 选项确定图像的位置，并进行缩放以使其覆盖整个框。通常，这种方法会扭曲对象。将忽略 dpi 和 scale 选项。
ignore-clippingpath	（布尔值；仅用于 TIFF 和 JPEG 图像）将忽略图像文件中存在的剪切路径。默认值： false ，即，将应用剪切路径。
ignore-orientation	（布尔值；仅用于 TIFF 图像）忽略图像中的所有方向标记。这可能会对弥补错误的方向信息很有用。默认值： PDF_load_image() 中 ignoreorientation 选项的值
匹配框	（选项列表）根据表 4.13，包含匹配框详细信息的选项列表。
orientate	（关键字）指定放置对象时对象的目标方向。默认值： north north 向上 east 指向右侧 south 向下 west 指向左侧
position	（浮点值或关键字列表）一个或两个值，指定参考点 (x, y) 在对象内的位置： {0 0} 表示位于对象的左下角， {100 100} 表示位于右上角。如果指定了 boxsize 对象，则 position 选项还可指定框的位置，即，框中的相应点将与参考点 (x, y) 对齐。这些值表示为对象宽度和高度的百分比。如果两个百分比相等，指定一个浮点值即可。示例： 0 或 {0 0} 左下角 {50 100} 顶部边缘中间 50 或 {50 50} 对象中央 关键字 left 、 center 、 right (x 方向) 或者 bottom 、 center 、 top (y 方向) 可用来替代 0 、 50 和 100 。如果只指定了一个关键字，将会添加另一方向对应的关键字。 默认值： {left bottom}
rotate	（浮点值）旋转坐标系，将中心用作参考点并将指定的值用作以度数表示的旋转角度。这将旋转框和对象。放置对象时，会重设旋转选项。默认值： 0
scale	（浮点值列表）按照指定的缩放因子（而非百分比），在水平和垂直方向上缩放对象。如果两个因子相等，指定一个浮点值即可。如果已为 fitmethod 选项提供了关键字 auto 、 meet 、 slice 或 entire 之一，则将忽略此选项。默认值： 1
showborder	（布尔值）如果为 true ，将对限定框的边框做描边处理（使用当前图形状态）。这可能会对开发和调试很有用。默认值： false

7.2 模板

注：本节描述的模板函数与 *PDFlib* 块的各种数据处理无关。可使用 *PDF_fill_textblock()*、*PDF_fill_imageblock()* 和 *PDF_fill_pdfblock()* 填充用 *PDFlib* 块增效工具准备的块（参见第 119 页上的第 9 章 “*Personalization Functions (PPS)*”）。

C++ Java

int begin_template_ext(double width, double height, String optlist)

Perl PHP

int PDF_begin_template_ext(resource p, float width, float height, string optlist)

C

int PDF_begin_template_ext(PDF *p, double width, double height, const char *optlist)

开始模板定义。
width, height 模板定界框的尺寸（以点数为单位）。

optlist 选项列表，指定模板相关的属性。
可以使用 *PDF_load_image()* 的下列选项（参见表 7.3）：
iconname、*layer*、*metadata*、*OPI-1.3*、*OPI-2.0*
可以使用 *PDF_begin_page_ext()* 的下列选项（参见表 2.9）：
topdown

- 返回值
- 模板句柄，可用于后续的图像相关调用，尤其是 *PDF_fit_image()*。不返回错误。
- 详细说明
- 此函数将所有文本、图形和颜色状态参数重置为其缺省值，并会根据全局 *topdown* 参数建立坐标系统。定义模板期间，不能使用超文本函数或打开图像的参数，但可以使用所有文本、图形和颜色函数。
- 范围
- 文档、页面；此参数是模板范围的开头，必须与对应的 *PDF_end_template()* 调用成对使用。

C++ Java

void end_template()

Perl PHP

PDF_end_template(resource p)

C

void PDF_end_template(PDF *p)

完成模板定义。
范围 页面；此函数终结模板范围，必须与对应的 *PDF_begin_template()* 调用成对使用。

C++ Java

int begin_template(double width, double height)

Perl PHP

int PDF_begin_template(resource p, float width, float height)

C

int PDF_begin_template(PDF *p, double width, double height)

不常用，使用 *PDF_begin_template_ext()*。

7.3 缩览图

C++ Java	<code>void add_thumbnail(int image)</code>
Perl PHP	<code>PDF_add_thumbnail(resource p, int image)</code>
C	<code>void PDF_add_thumbnail(PDF *p, int image)</code>

作为缩览图，在当前页面上添加现有图像。

`image` 由 `PDF_load_image()` 获取的有效图像句柄。

详细说明 此函数可作为缩览图，在当前页面上添加提供的图像。缩览图必须遵守以下限制：

- ▶ 图像不能大于 106 x 106 像素。
- ▶ 图像必须使用灰度、RGB 或索引 RGB 色彩空间。
- ▶ 由于缩览图必须由单一的 PDF 图像对象构建，多带区 TIFF 不能构建缩览图。

此函数不会生成用于页面的缩览图图像，只提供一个挂钩，将现有图像加入缩览图中。实际的缩览图图像必须由客户端生成。客户端必须确保缩览图的颜色、高度 / 宽度比例和缩览图的实际内容与页面内容相匹配。

由于 Acrobat 5 及更高版本可即时生成缩览图（尽管 Acrobat 5 或 Adobe Reader 6 不能在浏览器中生成缩览图），且缩览图会增加生成的 PDF 的总文件大小，所以建议不添加缩览图，而由客户端生成缩览图。

范围 页面：每页只能调用一次。并非所有页面都需要附加缩览图。



8 PDF 导入函数 (PDI)

注：在本章中描述的所有函数都要求额外的 PDF 导入库 (PDI)；该 PDF 导入库要求 *PDFlib+PDI* 或 *PDFlib Personalization Server (PPS)*，但 *PDFlib Lite* 和 *PDFlib* 中并不包括这些组件。有关如何获得 PDI 的更多信息，请访问我们的网站。

8.1 文档和页面

表 8.1 列出了本节中的相关参数关键字名称（参见第 11 页上的第 2.1 节“参数处理”）。

表 8.1 *PDF_get/set_parameter()* 的 PDI 相关关键字

关键字	说明
<i>pdi¹</i>	如果附加了 PDI 库，将返回字符串 true（这不适用于 <i>PDFlib</i> 和 <i>PDFlib Lite</i> ）；否则返回 false。 范围：任意、空 ²
<i>pdiwarning</i>	不常用，使用 <i>errorpolicy</i>

1. 仅用于 *PDF_get_parameter()*
2. 可用 *PDF** 的 NULL 或 0 自变量调用

C++ Java	<i>int open_pdi_document(String filename, String optlist)</i>
Perl PHP	<i>int PDF_open_pdi_document(resource p, string filename, string optlist)</i>
C	<i>int PDF_open_pdi_document(PDF *p, const char *filename, int len, const char *optlist)</i>

打开并准备基于磁盘的或虚拟的 PDF 文档，以供稍后使用。

filename （名称字符串）PDF 文件的文件名。

optlist 选项列表，根据表 8.2 指定 PDF 打开选项。可以使用下列选项：*infomode*、*inmemory*、*errorpolicy*、*password*、*repair*、*requiredmode*

len （仅用于 C 语言绑定）UTF-16 字符串的 *filename* 长度（以字节为单位）。如果 *len = 0*，则必须提供以 null 终结的字符串。

返回值 PDI 文档句柄，可用于处理单独的文档页面，或用于查询文档属性。返回值 -1（在 PHP 中为：0）表示无法打开 PDF 文档。可同时打开任意数量的 PDF 文档。返回值在关闭文档范围结束之前一直有效。如果函数调用失败，则可以使用 *PDF_get_errmsg()* 请求失败的原因。

详细说明 默认情况下，如果至少符合下列任一条件，则会拒绝文档：

- 文档已损坏。
- 文档使用的 PDF 版本高于当前 PDF 文档版本。
- 文档已加密，但在 `password` 选项中未提供相应的口令。
- 文档不符合当前 PDF/X 或 PDF/A 输出规范级别，或使用了不兼容的输出方法。
- 文档为标签 PDF，且 `PDF_begin_document()` 中的 `tagged` 选项为 `true`。

但是，除第一个原因以外，均可使用 `infomode` 选项打开文档。这可能会对使用 `PDF_pcos_get_*` 函数查询有关 PDF 的信息很有用，例如加密、PDF/A 或 PDF/X 状态、文档信息域等。

为了获得有关 PDF 导入相关问题性质的更多详细信息（PDF 文件名错误、不支持的格式、PDF 数据错误，等），可使用 `PDF_get_errmsg()` 接收更加详细的错误消息。

PDF/A：除非 `infomode=true`，否则导入的文档必须符合当前 PDF/A 输出规范等级。

PDF/X：除非 `infomode=true`，否则导入的文档必须符合当前 PDF/X 输出规范等级，并且必须使用与生成文档相同的输出方法。

范围 对象、文档、页面；在对象 范围内，PDI 文档句柄只能用于 `PDF_pcos_get_*` 函数。

表 8.2 `PDF_open_pdi_document()` 的选项

关键字	说明
infomode	（布尔值）如果为 <code>true</code> ，将打开文档以便使用 <code>pcos</code> 接口查询信息，但不能将页面导入当前输出文档。尤其是，当 <code>infomode=true</code> 时，可以打开下列文档类型（缺省值： <code>false</code> ，如果 <code>requiredmode=full</code> ，否则为 <code>true</code> ）： <ul style="list-style-type: none">▸ 不符合当前 PDF/X 或 PDF/A 规范等级的 PDF▸ 其 PDF 版本高于当前文档版本的 PDF▸ 不知道口令的加密 PDF（例外：使用 <code>Distiller</code> 设置“对象级压缩：最大”创建的 PDF 1.6 文档）▸ <code>PDF_begin_document()</code> 中的 <code>tagged</code> 选项为 <code>true</code> 时的标签 PDF
inmemory	（布尔值）如果为 <code>true</code> ，PDI 将整个文件载入内存，并在内存中处理文件。在有些系统中（尤其是 <code>MVS</code> ），这么做能够极大地提高性能，但会加大内存消耗。如果为 <code>false</code> ，将根据需要从磁盘中读取文档的各个部分。默认值： <code>false</code>
errorpolicy	（关键字）控制出现错误时的行为（参见第 28 页上的第 2.6 节“异常处理”）
password	（最大长度为 32 个字符的字符串）打开待导入的受保护 PDF 文档所需的许可口令。如果 <code>infomode=true</code> ，只使用文档打开口令（甚至可能为空）即可查询文档信息。如果根本没有为加密文档提供口令，则只能用文档句柄查询文档的加密状态。
pdiwarning	不常用，使用 <code>errorpolicy</code>
repair	（关键字）指定如何处理受损 PDF 输入文档。修复文档比常规解析花费的时间要长，但可能允许处理某些受损 PDF。请注意，某些文档的受损程度可能超出修复范围（默认值： <code>autof0f1%</code> ） <ul style="list-style-type: none">force 无条件尝试修复文档，无论文档是否有问题。auto 仅修复在打开 PDF 时检测到的问题。none 不尝试修复文档。如果 PDF 中存在问题，函数调用将失败。
requiredmode	（关键字）打开文档时可以接受的最小 <code>pcos</code> 模式（最小 / 受限 / 完整）。如果结果 <code>pcos</code> 模式低于所需模式，调用将失败。如果调用成功，则结果 <code>pcos</code> 模式一定至少是在此选项中指定的模式。但是，也有可能是更高的模式；例如，未加密文档的 <code>requiredmode=minimum</code> 将导致 <code>full</code> 模式。默认值： <code>full</code>

C++ Java	<code>int open_pdi(String filename, String optlist, int len)</code>
Perl PHP	<code>int PDF_open_pdi(resource p, string filename, string optlist, int len)</code>
C	<code>int PDF_open_pdi(PDF *p, const char *filename, const char *optlist, int len)</code>

不常用；使用 `PDF_open_pdi_document()`。

C	<code>int PDF_open_pdi_callback(PDF *p, void *opaque, size_t filesize, size_t (*readproc)(void *opaque, void *buffer, size_t size), int (*seekproc)(void *opaque, long offset), const char *optlist)</code>
---	---

打开并准备自定义数据源中的 PDF 文档，以供稍后使用。

opaque 一个指针，指向某些可能与输入 PDF 文档相关联的用户数据。此指针将作为第一个参数传递给回调函数，并能以任何方式使用。PDI 不会以其他方式使用 **opaque** 指针。

filesize 整个 PDF 文档的大小（以字节为单位）。

readproc 一个回调函数，将 **size** 字节数复制到 **buffer** 指向的内存。如果到达文档结尾，它复制的数据可能少于要求的字节数。此函数必须返回复制的字节数。

seekproc 一个回调函数，设置文档中的当前读取位置。**offset** 指示从文档开头起的位置（0 表示第一个字节）。如果运行成功，此函数必须返回 0，否则返回 -1。

optlist 选项列表，根据表 8.2 指定 PDF 打开选项。可以使用下列选项：

infomode、*inmemory*、*password*、*pdiwarning*、*requiredmode*

返回值	PDI 文档句柄，可用于处理单独的文档页面，或用于查询文档属性。返回值 -1 表示无法打开 PDF 文档。可同时打开任意数量的 PDF 文档。返回值在关闭文档范围结束之前一直有效。如果函数调用失败，则可以使用 <code>PDF_get_errmsg()</code> 请求失败的原因。
详细说明	这是个特殊的应用程序接口，它将从某些数据源中提取任意数量的 PDF 数据，而不在磁盘文件或内存中提供 PDF 文档。
范围	对象、文档、页面；在对象范围内，只能使用 PDI 文档句柄查询 PDF 文档中的信息。
绑定	仅用于 C 语言绑定。

C++ Java	<code>void close_pdi_document(int doc)</code>
Perl PHP	<code>PDF_close_pdi_document(resource p, int doc)</code>
C	<code>void PDF_close_pdi_document(PDF *p, int doc)</code>

关闭所有打开的 PDI 页面句柄，并关闭输入 PDF 文件。

doc 由 `PDF_open_pdi_document()` 返回的有效 PDF 文档句柄。

详细说明	此函数关闭 PDF 导入文档并释放所有与文档相关的资源。所有可能打开的文档页面都将隐式关闭。在此调用后禁止使用文档句柄。如果还要导入其他页面，则不应关闭 PDF 文档。尽管可以不限次数地打开和关闭 PDF 导入文档，但这么做会无形增大 PDF 输出文件。
范围	对象、文档、页面

C++ Java

void close_pdi(int doc)

Perl PHP

PDF_close_pdi(resource p, int doc)

C

void PDF_close_pdi(PDF *p, int doc)

不常用；使用 `PDF_close_pdi_document()`。

C++ Java

int open_pdi_page(int doc, int pagenumber, String optlist)

Perl PHP

int PDF_open_pdi_page(resource p, int doc, int pagenumber, string optlist)

C

int PDF_open_pdi_page(PDF *p, int doc, int pagenumber, const char* optlist)

使用 `PDF_fit_pdi_page()` 准备页面，以供稍后使用。

doc 由 `PDF_open_pdi_document()` 返回的有效 PDF 文档句柄。

pagenumber 要打开的页面的页码。第一页的页码为 1。

optlist 选项列表，根据表 8.3 指定页面选项。可以使用下列选项：`errorpolicy`、`hypertext-encoding`、`iconname`、`infomode`、`metadata`、`pdiusebox`、

返回值 页面句柄，可与 `PDF_fit_pdi_page()` 结合使用来放置页面。返回值 -1（在 PHP 中为：o）表示无法打开页面。返回值在关闭文档范围结束之前一直有效。如果使用 `PDF_open_pdi_document()` 打开文档时，`infomode` 选项为 `true`，则该句柄不能用于 `PDF_fit_pdi_page()`。如果函数调用失败，则可以使用 `PDF_get_errmsg()` 请求失败的原因。

详细说明 此函数会把构成导入页面的所有数据复制到输出文档，但不会在输出中显示任何效果。若要将导入页面实际置于生成的输出文档内，必须使用 `PDF_fit_pdi_page()`。若要获得 PDF 导入相关问题的更多详细信息（不支持格式、PDF 数据错误等），可以调用 `PDF_get_errmsg()`。
如果导入文档的 PDF 版本号高于生成的 PDF 输出文档的 PDF 版本号，此函数将会失败。
可同时打开任意数量的页面。如果多次打开同一页面，将返回不同的句柄，但一个句柄只能关闭一次。

范围 文档、页面

表 8.3 `PDF_open_pdi_page()` 的选项

关键字	说明
errorpolicy	（关键字）控制出现错误时的行为（参见第 28 页上的第 2.6 节“异常处理”）
hypertext-encoding	（关键字）指定 <code>iconname</code> 选项的编码。空字符串等效于 <code>unicode</code> 。默认值：全局 <code>hypertextencoding</code> 参数的值
iconname	（超文本字符串）向导入页面附加一个名称，以便能够通过 <code>JavaScript</code> 引用页面，例如，将页面用作表单域的图标。
infomode	不常用；使用 <code>pCOS</code> 查询页面属性，而不用实际放置页面
layer	（图层句柄； <code>PDF 1.5</code> ）页面所属的图层。仅当相应的图层可见时，页面才可见。请注意，页面将绑定到图层。如果需要将同一页面放置到属于不同图层的不同位置上，请在放置页面之前使用 <code>PDF_begin_layer()</code> 。
metadata	（选项列表； <code>PDF 1.4</code> ）提供导入页面的元数据（请参见第 144 页上的第 12.2 节“XMP 元数据”）
pdiusebox	（关键字）指定用于确定导入页面大小的框尺寸。默认值： <code>crop</code> 。 media 使用媒体框（始终提供此框） crop 如果有，则使用裁剪框，否则使用媒体框。 bleed 如果有，则使用出血框，否则使用裁剪框。 trim 如果有，则使用裁切框，否则使用裁剪框。 art 如果有，则使用作品框，否则使用裁剪框。
pdiwarning	不常用，使用 <code>errorpolicy</code>

C++ Java	<code>void close_pdi_page(int page)</code>
Perl PHP	<code>PDF_close_pdi_page(resource p, int page)</code>
C	<code>void PDF_close_pdi_page(PDF *p, int page)</code>

关闭页面句柄，并释放所有页面相关的资源。

page 由 `PDF_open_pdi_page()` 返回的有效 PDF 页面句柄（而非页码！）。

详细说明 此函数关闭与 `page` 确定的页面句柄相关联的页面，并释放所有相关资源。在此调用后禁止使用 `page`。

范围 文档、页面

C++ Java	<code>void fit_pdi_page(int page, double x, double y, String optlist)</code>
Perl PHP	<code>PDF_fit_pdi_page(resource p, int page, float x, float y, string optlist)</code>
C	<code>void PDF_fit_pdi_page(PDF *p, int page, double x, double y, const char *optlist)</code>

根据各种选项，在页面上放置导入的 PDF 页面。

page 由 `PDF_open_pdi_page()` 返回的有效 PDF 页面句柄（而非页码！）。打开文档和页面之前，`infomode` 选项必须已设为 `false`。不能事先关闭页面句柄。

x, y 用户坐标系统中的参考点坐标；将根据各种选项，在此坐标上放置页面。

optlist 选项列表，根据表 7.4 指定详细的缩放和放置信息。可以使用下列选项：
`adjustpage`、`blind`、`boxsize`、`dpi`、`fitmethod`、`ignore-clippingpath`、`ignore-orientation`、
 匹配框、`orientate`、`position`、`rotate`、`scale`、`showborder`

详细说明 此函数类似于 `PDF_fit_image()`，但在导入的 PDF 页面上进行操作。

范围 页面、图案、模板、字形

8.2 pCOS 函数

在 PDF 文档中，所有 pCOS 函数都将与指定目标对象的路径一起使用。有关 pCOS 路径的详细信息将在 *PDFlib* 教程中讨论。

注：在评估模式下，pCOS 只接受不超过 1 MB 或 10 页的输入文档。但是，在评估模式下，也可以查询较大文档的下列元素：页面计数、页面尺寸、块详细资料 and 所有通用伪对象。

C++ Java

double pcos_get_number(int doc, string path)

Perl PHP

double PDF_pcos_get_number(resource tet, long doc, string path)

C

double PDF_pcos_get_number(PDF *p, int doc, const char *path, ...)

获取 *number* 或 *boolean* 类型的 pCOS 路径值。

doc 通过 *PDF_open_pdi_document()* 获取的有效文档句柄。

path 数字型或布尔型对象的完整 pCOS 路径。

Additional parameters （仅用于 C 语言绑定）如果 *key* 参数包含可变数量的占位符（%s 表示字符串或 %d 表示整数；使用 %% 表示一个百分比符号），则可以提供相应数量的附加参数。使用这些参数，您无须显式格式化包含可变数值或字符串值的复杂路径。客户负责确保占位符的数量和类型与提供的附加参数相匹配。

返回值 pCOS 路径标识的对象的数值。如果为 *true*，将返回布尔值 1；否则返回 0。

范围 任意

C++ Java

const string pcos_get_string(int doc, string path)

Perl PHP

string PDF_pcos_get_string(resource tet, long doc, string path)

C

const char *PDF_pcos_get_string(TET *tet, int doc, const char *path, ...)

获取 *name*、*string* 或 *boolean* 类型的 pCOS 路径值。

doc 通过 *PDF_open_pdi_document()* 获取的有效文档句柄。

path 名称、字符串或布尔型对象的完整 pCOS 路径。

Additional parameters （仅用于 C 语言绑定）如果 *key* 参数包含可变数量的占位符（%s 表示字符串或 %d 表示整数；使用 %% 表示一个百分比符号），则可以提供相应数量的附加参数。使用这些参数，您无须显示格式化包含可变数值或字符串值的复杂路径。客户负责确保占位符的数量和类型与提供的附加参数相匹配。

返回值 具有 pCOS 路径标识的对象值的字符串。对于布尔型值，将返回字符串 *true* 或 *false*。

详细说明 如果不是在完整模式下运行 pCOS 且对象的类型为 *string*，则此函数将引发异常。由于异常，如果 *nocopy=false* 或 *plainmetadata=true*，也可以在受限的 pCOS 模式下检索 */Info/** 对象（文档信息关键字）；如果 *nocopy=false*，可以在受限制的 pCOS 模式下检索 *bookmarks[...]/Title* 和 *annots[...]/contents*。

此函数假定从 PDF 文档返回的字符串为文本字符串。包含二进制数据的字符串数据应改用 *PDF_pcos_get_stream()* 检索，这样在任何情况下都不会修改数据。

范围 任意

绑定 C 和 C++ 语言绑定：将以 UTF-8 格式返回字符串。
C 绑定：返回的字符串在下次调用此函数之前一直有效。

C++ Java	<code>const unsigned char *pcos_get_stream(int doc, int *length, string optlist, string path)</code>
Perl PHP	<code>string PDF_pcos_get_stream(resource tet, long doc, string optlist, string path)</code>
C	<code>const unsigned char *PDF_pcos_get_stream(TET *tet, int doc, int *length, const char *optlist, const char *path, ...)</code>

获取 *stream*、*fstream* 或 *string* 类型的 pCOS 路径内容。

doc 通过 *PDF_open_pdi_document()* 获取的有效文档句柄。

length （仅用于 C 和 C++ 语言绑定）一个指向变量的指针，将接收返回流数据的长度（以字节为单位）。

optlist 选项列表，根据表 8.4 指定详细的缩放和放置信息。

path 流或字符串对象的完整 pCOS 路径。

Additional parameters （仅用于 C 语言绑定）如果 *key* 参数包含可变数量的占位符（*%s* 表示字符串或 *%d* 表示整数；使用 *%%* 表示一个百分比符号），则可以提供相应数量的附加参数。使用这些参数，您无须显示格式化包含可变量值或字符串值的复杂路径。客户负责确保占位符的数量和类型与提供的附加参数相匹配。

返回值 流或字符串中包含的未加密数据。如果流或字符串为空，则返回的数据将为空（在 C 和 C++ 中为：NULL）。

如果对象为 *stream* 类型，将从流内容中删除所有筛选器（即，将返回实际原始数据）。如果对象为 *fstream* 或 *string* 类型，将完全按照 PDF 文件中的格式发送数据，将被删除的 ASCII85 和 ASCIIHex 筛选器除外。

详细说明 如果不是在完全模式下运行 pCOS，此函数将引发异常。由于异常，如果 *nocopy= false* 或 *plainmetadata=true*，也可以在受限的 pCOS 模式下检索 */Root/Metadata* 对象。如果 *path* 并非指向 *stream*、*fstream* 或 *string* 类型的对象，也将引发异常。

尽管其名称如此，但此函数也可用于检索 *string* 类型的对象。与将对象作为文本字符串处理的 *PDF_pcos_get_string()* 不同，此函数在任何情况下都不会修改返回数据。二进制字符串数据在 PDF 中极少使用，并且不会自动进行可靠检测。因此，用户负责选择适当的函数来检索字符串对象作为二进制数据或文本。

范围 任意

绑定 C 和 C++ 语言绑定：在下一次调用此函数之前，可以一直使用返回的数据缓存区。

注：可以使用此函数从 *PDF* 检索嵌入的字体数据。系统会提醒用户，字体应遵循相应字体供应商的许可协议，并且在未获得知识产权所有者明文许可的情况下不得使用。若要商讨相关许可协议，请与字体供应商联系。

8.3 其他 PDI 处理

C++ Java

int process_pdi(int doc, int page, String optlist)

Perl PHP

int PDF_process_pdi(resource p, int doc, int page, string optlist)

C

int PDF_process_pdi(PDF *p, int doc, int page, const char* optlist)

处理导入 PDF 文档的某些元素。

doc 由 `PDF_open_pdi_document()` 返回的有效 PDF 文档句柄。

page 如果 *optlist* 要求页面句柄（参见表 8.4），*page* 必须为通过 `PDF_open_pdi_page()` 返回的有效 PDF 页面句柄（而非页码！）。不能事先关闭页面句柄。如果 *optlist* 不要求任何页面句柄，*page* 必须为 -1。

optlist 选项列表，根据表 8.4 指定处理选项。可以使用下列选项：*action*、*errorpolicy*

- 返回值
- 如果此函数成功，则返回值 1；如果失败，则返回错误代码 -1（在 PHP 中为：o）。
- 详细说明
- PDF/X：必须使用具有 *copyoutputintent* 选项的函数，或使用 `PDF_load_iccprofile()` 设置输出方法。
PDF/A：可以使用具有 *copyoutputintent* 选项的函数，或使用 `PDF_load_iccprofile()` 设置输出方法。但是，只要文档中使用了设备无关的颜色，就不要求任何输出方法。

范围 文档

表 8.4 `PDF_process_pdi()` 的选项

关键字	说明
<i>action</i> ¹	（关键字；必选）指定 PDF 处理的种类： <i>copyoutputintent</i> 将导入文档的 PDF/X 或 PDF/A 输出方法 ICC 色彩特征描述文件复制到输出文档。第二次及后续复制输出方法尝试将被忽略。如果文档包含多个输出方法，将使用第一个输出方法。标准输出方法（没有嵌入的 ICC 色彩特征描述文件）不能使用此方法复制。
<i>errorpolicy</i>	（关键字）控制出现错误时的行为（参见第 28 页上的第 2.6 节“异常处理”）
<i>pdiwarning</i>	不常用，使用 <i>errorpolicy</i>

1. 不要求页面句柄

8.4 不常用的 PDI 参数

本节中的所有函数和参数都不常用；使用第 114 页上的第 8.2 节“pCOS 函数”中讨论的 pCOS 接口。

C++ Java

double get_pdi_value(String key, int doc, int page, int reserved)

Perl PHP

double PDF_get_pdi_value(resource p, string key, int doc, int page, int reserved)

C

double PDF_get_pdi_value(PDF *p, const char *key, int doc, int page, int reserved)

不常用，使用具有表 8.5 中列出的 pCOS 路径的 `PDF_pcos_get_number()`。

表 8.5 `PDF_get_pdi_value()` 的不常用数字参数的名称和建议的 pCOS 路径

旧有关键字	建议用来替代旧有关键字的 pCOS 路径
<code>vdp/blockcount</code>	<code>length:pages[...]/blocks</code>
<code>/Root/Pages/Count</code>	<code>length:pages</code>
<code>width, height</code>	<code>pages[...]/width, pages[...]/height</code>
<code>/Rotate</code>	<code>pages[...]/Rotate</code>
<code>version</code>	<code>pdfversion (not version!)</code>
<code>/CropBox, /BleedBox, /ArtBox, /TrimBox, /MediaBox</code>	<code>pages[...]/CropBox[0] for llx, pages[...]/CropBox[1] for lly, pages[...]/CropBox[2] for urx, pages[...]/CropBox[3] for ury, pages[...]/BleedBox[0] for llx, etc.</code>
<code>vdp/Blocks/<name>/<property></code>	<code>pages[...]/blocks/<name>/<property></code>
<code>vdp/Blocks[...]/<property></code>	<code>pages[...]/blocks[...]/<property></code>
<code>vdp/Blocks/<name>/Custom/<property></code>	<code>pages[...]/blocks/<name>/Custom/<property> or pages[...]/blocks[...]/Custom/<property></code>

C++ Java

String get_pdi_parameter(String key, int doc, int page, int reserved)

Perl PHP

string PDF_get_pdi_parameter(resource p, string key, int doc, int page, int reserved)

C

const char * PDF_get_pdi_parameter(PDF *p, const char *key, int doc, int page, int reserved, int *len)

不常用，使用具有表 8.6 中列出的 pCOS 路径的 `PDF_pcos_get_string()`。

表 8.6 `PDF_get_pdi_parameter()` 的不常用字符串参数的名称和建议的 pCOS 路径

旧有关键字	建议用来替代旧有关键字的 pCOS 路径
<code>isempty</code>	<code>pages[...]/isempty</code>
<code>filename</code>	<code>filename</code>
<code>/Info/<key></code>	<code>/Info/Title etc.</code>
<code>tagged</code>	<code>tagged</code>
<code>pdfx</code>	<code>pdfx</code>
<code>vdp/Blocks/<name>/<property></code>	<code>pages[...]/blocks/<name>/<property></code>
<code>vdp/Blocks[...]/<property></code>	<code>pages[...]/blocks[...]/<property></code>
<code>vdp/Blocks/<name>/Custom/<property></code>	<code>pages[...]/blocks/<name>/Custom/<property> or pages[...]/blocks[...]/Custom/<property></code>

9 Personalization Functions (PPS)

PDFlib Personalization Server (PPS) 提供了专用函数，来处理文本、图像和 PDF 型变量数据块。这些块必须包含在导入 PDF 页面内，但不会在生成输出中保留。使用任何块填充函数之前，导入的页面必须已经通过 `PDF_fit_pdi_page()` 放置在输出页面中。计算页面上的块位置时，块函数将考虑缩放选项，这些选项在使用 `PDF_fit_pdi_page()` 放置导入的页面时已经有效。

注：本章讨论的块处理函数要求 *PDFlib Personalization Server (PPS)*。在 *DPF* 模板中创建块要求用于 *Adobe Acrobat* 的 *PDFlib* 块增效工具。

C++ Java	<code>int fill_textblock(int page, String blockname, String text, String optlist)</code>
Perl PHP	<code>int PDF_fill_textblock(resource p, int page, string blockname, string text, string optlist)</code>
C	<code>int PDF_fill_textblock(PDF *p, int page, const char *blockname, const char *text, int len, const char *optlist)</code>

根据其属性，用变量数据填充文本块。

page 用于包含块的页面的有效 PDF 页面句柄。

blockname （名称字符串）块的名称。

text （内容字符串）要填充到块中的文本；如果使用缺省文本（如块属性的定义），则为空字符串。如果提供了 **handle** 选项且它包含一个有效的 **Textflow** 句柄，则将忽略此参数。

len （仅用于 C 语言绑定）UTF-16 字符串的 **text** 长度（以字节为单位）。如果 **len = 0**，则必须提供以 **null** 终结的字符串。

optlist 选项列表，根据表 9.1 指定详细的填充信息。可以使用下列选项：

- ▶ **boxsize**、**charref**、**encoding**、**escape-sequence**、**font**、**ignore-orientation**、**repoint**、**showborder**、**shrinklimit**、**textformat**、**textflow-handle**
- ▶ 如果未提供 **font** 选项，则可以使用 `PDF_load_font()` 的所有选项（参见表 3.4）。仅当同时提供了 **fontname** 和 **encoding** 选项，这些选项才可用。
autocidfont、**auto-subsetting**、**capheight**、**descender**、**embedding**、**fontstyle**、**keepnative**、**kerling**、**linegap**、**metadata**、**monospace**、**replace-mentchar**、**subsetlimit**、**subsetminsize**、子集化、**unicodemap**、**vertical**、**xheight**
- ▶ 如果 **textflow=true**，则 `PDF_add/create_textflow()` 的所有选项都可用（参见表 4.3）：
常规选项：**errorpolicy**
文本语义：**charclass**、**charmapping**、**hyphenchar**、**tabalignchar**、
文本格式：**alignment**、**avoid-emptybegin**、**fixedleading**、**hortabsize**、**hortabmethod**、**lastalignment**、**leader**、**leading**、**leftindent**、**minlinecount**、**parindent**、**rightindent**、**ruler**、**tabalignment**
控制分行算法：**adjustmethod**、**avoidbreak**、**maxspacing**、**minspacing**、**nofitlimit**、**shrinklimit**、**spreadlimit**
用作命令的选项：**comment**、**mark**、**nextline**、**nextparagraph**、**resetfont**、**return**、**space**
字体相关的选项：**encoding**、**fontname**
处理内嵌选项列表：**begoptlistchar**、**endoptlistchar**、**textlen**
- ▶ 如果 **textflow=true**，则 `PDF_fit_textflow()` 的下列选项可用（参见表 4.6）：
featherlimit、**firstlinedist**、**fitmethod**、**keep**、**lastlinedist**、**linespread-limit**、**maxlines**、**minfontsize**、**orientate**、**wrap**、**rotate**、**showtabs**、**verticalalign**

► *PDF_fit_textline()* 的所有外观选项（参见表 4.1）：
charref、*charspacing*、*dasharray*、*escape-sequence*、*fakebold*、*fillcolor*、字体、*fontsize*、*glyphcheck*、*horizscaling*、*italicangle*、 *Kerning*、匹配框、*overline*、*showborder*、*strikeout*、*strokecolor*、*strokewidth*、*textformat*、*textrendering*、*textrise*、*underline*、*underline-position*、*underline-width*、*wordspacing*、*fontname* 和 *encoding* 都可用于选择字体。此外，*font* 选项也可用于提供之前使用的 *PDF_load_font()* 调用创建的字体句柄。如果指定了 *font*，将忽略 *fontname* 和 *encoding* 选项。

返回值 如果页面中不存在命名的块，不能填充块（例如，由于字体问题）或者块的处理要求更高版本的 PDFlib，则返回 -1（在 PHP 中：返回 0）；如果可以成功处理块，则返回 1。如果指定了 *textflowhandle* 选项，将返回后续调用可以使用的有效 *Textflow* 句柄。

如果发现 PDF 文档受到破坏，将根据 *errorpolicy* 参数或选项引发异常，或者返回 -1。

详细说明 提供的文本将根据块的属性进行格式化，以填充至块中。如果 *text* 为空，此函数将使用块的缺省文本（如果存在）；否则，不返回任何信息。这可能对利用填充色或描边色等其他块属性很有用。

链接 *Textflow* 块：如果一个块中容纳不下 *Textflow*，则可以使用 *textflowhandle* 选项将多个块连接至块链中，以便可以容纳同一 *Textflow* 的多个部分：

- 在第一次调用时，必须提供值 -1（在 PHP 中为：0）。*PDF_fill_textblock()* 将返回内部创建的 *Textflow* 句柄，用户必须保存它。
- 下一次调用时，可以将上一步中返回的 *Textflow* 句柄提供给 *textflowhandle* 选项（在此情况下，将忽略在 *text* 参数中提供的文本，且该文本应为空）。将使用 *Textflow* 的其余部分填充该块。
- 可以对多个 *Textflow* 块重复此过程。
- 可以将返回的 *Textflow* 句柄提供给 *PDF_info_textflow()*，以便确定块填充的结果，例如文本的结束位置。

此过程可以无限制地重复。最后，用户负责使用 *PDF_delete_textflow()* 删除 *Textflow* 句柄。

范围 页面、模板

C++ Java	<code>int fill_imageblock(int page, String blockname, int image, String optlist)</code>
Perl PHP	<code>int PDF_fill_imageblock(resource p, int page, string blockname, int image, string optlist)</code>
C	<code>int PDF_fill_imageblock(PDF *p, int page, const char *blockname, int image, const char *optlist)</code>

根据其属性，用变量数据填充图像块。

page 用于包含块的页面的有效 PDF 页面句柄。

blockname （名称字符串）块的名称。

image 要填充到块中的图像的有效图像句柄；如果要使用缺省图像（如块属性的定义），将为 -1。

optlist 选项列表，根据表 9.1 指定详细的填充信息。可以使用下列选项：*boxsize*、*errorpolicy*、*ignore-orientation*、*refpoint*、*showborder*

返回值 如果页面中不存在命名的块，不能填充块或者块的处理要求更高版本的 PDFlib，则返回 -1（在 PHP 中：返回 0）；如果可以成功处理块，则返回 1。使用 *PDF_get_errmsg()* 可获得有关问题性质的更多信息。

详细说明 根据块的属性，将提供的图像句柄指向的图像置于块中。如果 *image* 为 -1（在 PHP 中为：0），此函数将使用块的缺省图像（如果存在）；否则，不返回任何信息。

如果发现 PDF 文档受到破坏，将根据 *errorpolicy* 参数或选项引发异常，或者返回 -1。

C++ Java	<i>int fill_pdfblock(int page, String blockname, int contents, String optlist)</i>
Perl PHP	<i>int PDF_fill_pdfblock(resource p, int page, string blockname, int contents, string optlist)</i>
C	<i>int PDF_fill_pdfblock(PDF *p, int page, const char *blockname, int contents, const char *optlist)</i>

根据其属性，用变量数据填充 PDF 块。

page 用于包含块的页面的有效 PDF 页面句柄。

blockname （名称字符串）块的名称。

contents 要填充到块中的 PDF 页面的有效 PDF 页面句柄；如果要使用缺省 PDF 页面（如块属性的定义），将为 -1。

optlist 选项列表，根据表 9.1 指定详细的填充信息。可以使用下列选项：*boxsize*、*encoding*、*errorpolicy*、*refpoint*、*showborder*

返回值 如果页面中不存在命名的块，不能填充块或者块的处理要求更高版本的 PDFlib，则返回 -1（在 PHP 中：返回 0）；如果可以成功处理块，则返回 1。使用 *PDF_get_errmsg()* 可获得有关问题性质的更多信息。

详细说明 根据块的属性，将提供的页面句柄 *contents* 指向的 PDF 页面置于块中。如果 *contents* 为 -1（在 PHP 中为：0）此函数将使用块的缺省 PDF 页面（如果存在）；否则，不返回任何信息。
如果发现 PDF 文档受到破坏，将根据 *errorpolicy* 参数或选项引发异常，或者返回 -1。

表 9.1 *PDF_fill_*block()* 函数的选项

关键字	说明
<i>boxsize</i>	（浮点值列表）将块的宽度和高度更改为指定的值（用当前用户坐标系中的坐标表示）。默认值：如块 <i>Rect</i> 属性中所指定的值。
<i>charref</i>	（布尔值；仅用于 <i>PDF_fill_textblock()</i> ）请参见表 4.1。
<i>encoding</i>	（字符串） <i>PDF_load_font()</i> 要求的字体编码方式。除非符合下列条件之一，否则 <i>PDF_fill_textblock()</i> 要求使用此选项： text 参数中的字符串为空且使用的是 defaulttext 属性。提供了 font 选项。
<i>errorpolicy</i>	（关键字）控制出现错误时的行为（参见第 28 页上的第 2.6 节“异常处理”）
<i>escape-sequence</i>	（布尔值；仅用于 <i>PDF_fill_textblock()</i> ）请参见表 4.1。
<i>glyphwarning</i>	不常用，使用 <i>errorpolicy</i>
<i>font</i>	（字体句柄；仅用于 <i>PDF_fill_textblock()</i> ） <i>PDF_load_font()</i> 返回的字体句柄。默认值：无；必须提供 font 或 fontname。
<i>fontwarning</i>	不常用，使用 <i>errorpolicy</i>
<i>ignore-orientation</i>	（布尔值；仅用于 <i>PDF_fill_imageblock()</i> ）如果为 true，将忽略 TIFF 图像中的方向标记。 默认值：false
<i>imagewarning</i>	不常用，使用 <i>errorpolicy</i>
<i>pdiwarning</i>	不常用，使用 <i>errorpolicy</i>
<i>refpoint</i>	（浮点值列表）将块的左下角移至使用用户坐标指定的点上。默认值：如块 <i>Rect</i> 属性中所指定的值。

表 9.1 `PDF_fill_*block()` 函数的选项（续）

关键字	说明
<code>showborder</code>	（布尔值）如果为 <code>true</code> ，将对块的边框做描边处理（使用当前图形状态）。这可能对开发和调试很有用。默认值： <code>false</code>
<code>shrinklimit</code>	（浮点值或百分比；仅用于 <code>PDF_fill_textblock()</code> ）请参见表 4.1。
<code>textformat</code>	（字符串；除非使用 <code>defaulttext</code> 属性，否则仅用于 <code>PDF_fill_textblock()</code> ）用于解释提供文本的格式。默认值： <code>auto</code>
<code>textflow-handle</code>	（ <i>Textflow</i> 句柄；仅用于 <code>PDF_fill_textblock()</code> 且 <code>textflow=true</code> ）此选项可用于 <i>Textflow</i> 块链接。必须为块链中的第一个块提供值 <code>-1</code> （在 <i>PHP</i> 中为： <code>0</code> ）；此函数返回的值在后续调用中可作为 <i>Textflow</i> 句柄提供给链中的其他块。此选项将 <code>fitmethod</code> 的默认值更改为 <code>clip</code> 。
几乎任何属性名称	<p>将用来覆盖块定义内对应名称和值的块属性名称和值。下列块属性不能覆盖： <code>Name</code>、<code>Description</code>、<code>Locked</code>、<code>Subtype</code>、<code>Type</code> <code>defaulttext</code>、<code>defaultimage</code>、<code>defaultpdf</code>、<code>defaultpdfpage</code></p> <p>除提供 <i>fontname</i> 属性以外，<code>font</code> 选项还可用于提供字体句柄（此时，将忽略 <i>fontname</i>）。</p> <p>颜色属性支持下列色彩空间关键字：<code>none</code>、<code>gray</code>、<code>rgb</code>、<code>cmk</code>、<code>spot</code>、<code>spotname</code>。</p>

10 交互功能

10.1 交互元素的参数

表 10.1 列出了交互元素的相关参数关键字名称（参见第 11 页上的第 2.1 节“参数处理”）。这些参数不能用于识别 Unicode 的语言绑定。

表 10.1 PDF_get/set_parameter() 中与字符串相关的关键字

关键字	说明
hypertextencoding	超文本字符串的编码。空字符串相当于 unicode。默认值: auto。范围: 任意
hypertextformat	超文本字符串的格式化。其可选值为 bytes、utf8、utf16、utf16le、utf16be 和 auto。默认值: auto。范围: 任意
usehypertextencoding	如果为 true, hypertextencoding 参数中指定的编码也将用于名称字符串。如果为 false, 不含 UTF-8 的名称字符串的编码是 host。默认值: false。范围: 任意
usercoordinates	如果为 false, 将预计超文本矩形的坐标使用缺省坐标系; 否则, 将使用当前用户坐标系。默认值: false。范围: 任意

10.2 动作

C++ Java	int create_action(String type, String optlist)
Perl PHP	int PDF_create_action(resource p, string type, string optlist)
C	int PDF_create_action(PDF *p, const char *type, const char *optlist)

创建一个动作，可将其提供给各种对象和事件。

类型 动作的类型，用下列一个关键字指定：

- ▶ **GoTo**: 转到当前文档中的某个目标。
此类型的特定选项: *destination*、*destname*
- ▶ **GoToR**: 转到其他（远程）文档中的某个目标。
此类型的特定选项: *destination*、*destname*、*filename*、*newwindow*
- ▶ **Launch**: （不用于 PDF/A）启动应用程序或文档。
此类型的特定选项: *defaultdir*、*filename*、*newwindow*、*operation*、*parameters*
- ▶ **URI**: 解析统一资源标识符，即：跳转到某个 Internet 地址。
此类型的特定选项: *ismap*、*url*
- ▶ **Hide**: （不适用于 PDF/A）隐藏或显示批注或表单域。
此类型的特定选项: *hide*、*namelist*
- ▶ **Named**: 执行由名称标识的 Acrobat 菜单项。
此类型的特定选项: *menuname*
- ▶ **SubmitForm**: 将数据发送到统一资源定位符，即 Internet 地址（请注意，不能在 Acrobat 中使用要求基本身份验证的提交操作）。
此类型的特定选项: *canonical-date*、*exclude*、*export-method*、*submit-emptyfields*、*url*
- ▶ **ResetForm**: （不用于 PDF/A）将某些表单域或全部表单域设置为他们的缺省值。
- ▶ **Trans**: (PDF 1.5) 使用某些可视化效果更新显示。这对在多个动作的序列中控制显示很有效。
此类型的特定选项: *duration*、*transition*

- ▶ **ImportData:**（不用于 PDF/A）从文件中导入域值。
- ▶ **JavaScript:**（不用于 PDF/A）执行用 JavaScript 代码编写的脚本。
此类型的特定选项：*script*、*scriptname*
- ▶ **SetOCGState:** (PDF 1.5) 隐藏或显示图层。
此类型的特定选项：*layerstate*、*preserve-radio*
- ▶ **GoTo3DView:** (PDF 1.6) 设置 3D 动画的当前视图。
此类型的特定选项：*3Dview*、*target*

optlist 选项列表，根据表 10.2 指定动作的属性。所有动作类型支持下列选项（有关特定于类型的附加选项，参见上文）：*errorpolicy*、*hypertext-encoding*

- 返回
值
- 动作句柄，可用来将动作附加到文档内的对象。可以在关闭的文档范围结束之前使用动作句柄。
- 详细说明
- 此函数将创建单一动作。可为一个或多个动作提供各种对象（例如，页面、表单域事件、书签等），但是必须使用单独的 *PDF_create_action()* 调用生成每个动作。允许为不同的对象多次使用一个动作。所有 PDF/X 模式都禁止使用动作。

范围 页面、文档。在下一个 *PDF_end_document()* 调用之前，可以一直使用返回的句柄。

表 10.2 *PDF_create_action()* 的动作属性的选项

选项	说明
3Dview	（关键字或 3D 视图句柄； GoTo3DView；是必需的）选择目标 3D 注释的视图；可选用下列关键字之一： <i>first</i> 、 <i>last</i> 、 <i>next</i> 、 <i>previous</i> （参见注释的 <i>views</i> 选项中的相应项）或 <i>default</i> （参见注释的 <i>defaultview</i> 选项），或使用 <i>PDF_create_3dview()</i> 创建的 3D 视图句柄。
actionwarning	不常用，使用 <i>errorpolicy</i>
canonical-date	（布尔型； SubmitForm）如果为 <i>true</i> ，可将任何表示日期的提交域值转换为标准格式。域本身不会专门指定将域解释为日期，只会在处理域的 <i>JavaScript</i> 代码中指定。默认值： <i>false</i>
defaultdir	（字符串； Launch）设置启动程序的缺省目录。只有在 <i>Windows</i> 上运行的 <i>Acrobat</i> 支持此选项。默认值： <i>none</i>
destination	（选项列表； GoTo、 GoToR；除非提供了 <i>destname</i> ，否则是必需的）选项列表，根据表 10.3 定义要跳转到的目标。
destname	（超文本字符串； GoTo、 GoToR；除非提供了 <i>destination</i> ，否则是必需的）用 <i>PDF_add_nameddest()</i> 指定的目标名称（用于 <i>GoTo</i> ）或者远程文档中的目标的名称（用于 <i>GoToR</i> ）。
errorpolicy	（关键字）控制出现错误时的行为（参见第 28 页上的第 2.6 节“异常处理”）。
duration	（浮点型； Transf@ 设置当前页面中转换效果的持续时间，以秒为单位。默认值： <i>1</i>
exclude	（布尔型； SubmitForm）如果为 <i>true</i> ， <i>namelist</i> 选项指定要排除的域；除 <i>namelist</i> 数组中列出的域以及 <i>exportable</i> 选项为 <i>false</i> 的域之外，将提交文档中的所有域。如果为 <i>false</i> ， <i>namelist</i> 选项指定提交操作中包括的域。将同时提交指定域组中的所有成员。默认值： <i>false</i> (ResetForm) 如果为 <i>true</i> ， <i>namelist</i> 选项指定要排除的域；除 <i>namelist</i> 数组中列出的域之外，将重设文档中的所有域。如果为 <i>false</i> ， <i>namelist</i> 选项指定重设操作中包括的域。将同时重设指定域组中的所有成员。默认值： <i>false</i>

表 10.2 PDF_create_action() 的动作属性的选项（续）

选项	说明
export-method	（关键字列表； SubmitForm）控制提交域名称和域值时使用的格式。默认值： fdf。 <i>html、fdf、xfdf、pdf</i> 分别为 HTML、 FDF、 XFDF 或 PDF 格式 <i>getrequest</i> （仅用于 <i>html</i> 和 <i>pdf</i> ）使用 HTTP GET 提交；否则使用 HTTP POST <i>updates</i> （仅用于 <i>fdf</i> ）包括基础 PDF 文档中的所有增量更新。 <i>exclurl</i> （仅用于 <i>fdf</i> ）提交的 FDF 将排除 url 字符串。 <i>annotfields</i> （仅用于 <i>fdf</i> ）包括所有批注和域。 <i>onlyuser</i> （仅用于 <i>fdf</i> 和 <i>annotfields</i> ）提交操作将只包括其名称匹配当前用户名称的批注；具体由远程服务器确定。 <i>coordinate</i> （仅用于 <i>html</i> ）将作为表单数据的一部分传送导致提交动作的鼠标点击的坐标。坐标值相对于域的矩形的左上角。 选项组合的示例： <code>exportmethod {fdf updates onlyuser}</code>
filename	（字符串； GoToR、 Launch； 是必需的）触发动作时将打开的外部（PDF 或其他）文件或应用程序的名称。 （ ImportData； 必需的）：包含表单数据的外部文件的文件名。
hide	（布尔型； Hide）指示是否隐藏 (true) 或显示 (false) 批注。默认值： true
hypertext-encoding	（关键字）指定提供的文本的编码方式。空字符串相当于 unicode。默认值： 全局 hypertextencoding 参数的值
ismap	（布尔型； URI）如果为 true，解析 url 时将向目标 URI 添加鼠标位置的坐标。默认值： false
layerstate	（选项列表； SetOCGState； 是必需的）成对列表；其中，每对都由一个关键字和一个图层句柄组成。支持的关键字： <i>on</i> 启动图层 <i>off</i> 停用图层 <i>toggle</i> 变换图层的状态。如果使用此关键字， preserveradio 选项应设为 false。
menuname	（字符串； Named； 是必需的）要执行的菜单项的名称。在 PDF/A 模式下只允许使用 nextpage、prevpage、firstpage、lastpage 这些常用的名称。但也可接受其他名称。要查找其他菜单项的名称，可在 Acrobat 的 JavaScript 控制台或调试程序中执行以下代码： <pre>function MenuList(m, level) { console.println(m.cName); if (m.oChildren != null) for (var i = 0; i < m.oChildren.length; i++) MenuList(m.oChildren[i], level + 1); } var m = app.listMenuItems(); for (var i=0; i < m.length; i++) MenuList(m[i], 0);</pre>
namelist	（字符串列表； Hide； 必需的）要隐藏或显示的批注或域的名称（包括组名）。 （ SubmitForm）提交操作中包括的或者要排除的表单域的名称（包括组名），具体取决于 exclude 选项的设置。默认值：除 exportable 选项为 false 的域之外，将提交所有域。 （ ResetForm）重设操作中包括的或者要排除的表单域的名称（包括组名），具体取决于 exclude 选项的设置。默认值：将重设所有域。
newwindow	（布尔型； GoToR、 Launch）一个标志，指定是否在新窗口中打开目标文档。如果此标志为 false，目标文档将在同一窗口内替换当前文档。 Launch：如果文件不是 PDF 文档，将忽略此项。 默认值： Acrobat 的行为取决于当前用户首选项。
operation	（关键字； Launch）一个关键字，指定应用到 filename 选项中指定文档的操作。只有在 Windows 上运行的 Acrobat 支持此选项。如果 filename 选项的目标是应用程序，而非文档，将忽略此选项，并启动应用程序。默认值： open。 <i>open</i> 打开文档 <i>print</i> 打印文档

表 10.2 PDF_create_action() 的动作属性的选项（续）

选项	说明
parameters	（字符串；Launch）要传递给 filename 选项指定的应用程序的参数字符串。只有在 Windows 上运行的 Acrobat 支持此选项。可用空格字符分隔多个参数，但每个参数不能包含任何空格字符。如果 filename 的目标是文档，应省略此选项。默认值：none
preserve-radio	（布尔型；SetOCGState）如果为 true，将保留图层间的单选按钮状态关系。默认值：true
script	（超文本字符串；JavaScript；必需的）包含待执行 JavaScript 代码的字符串。
scriptname	（超文本字符串；JavaScript）如果存在，将使用提供的名称，作为文档级 JavaScript 插入 script 选项中提供的 JavaScript。如果文档中多次提供了相同的 scriptname，将使用最后一个脚本，所有其他脚本都将忽略。文档级 JavaScript 将在文档载入 Acrobat 之后执行。这可能会对用于表单域脚本很有用。
submit-emptyfields	（布尔型；SubmitForm；PDF 1.4）如果为 true，将提交 namelist 和 exclude 选项确定的所有域，而不考虑他们是否具有任何值。对于没有值的域，只传送域名称。如果为 false，将不提交没有值的域。默认值：false
target	（字符串；GoTo3DView；必需的）目标 3D 注释的名称，如 PDF_create_annotation() 的 name 选项中指定的值。
transition	（关键字；Trans）设置转换效果；有关关键字的列表，参见表 2.9。默认值：replace
url	（字符串；URI；必需的）用 7 位 ASCII 或 EBCDIC（但只能包含 ASCII 字符）编码的统一资源定位器，指定链接目标。它可指向任意（Web 或本地）资源，必须以协议标识符开头（例如 http://）。textx/texty、currentx/currenty 和 imagewidth/imageheight 参数可用来检索定位信息，以计算链接矩形尺寸。 （SubmitForm；必需的）用 URL 格式给出，将用于在 Web 服务器端处理提交操作的脚本的统一资源定位器（地址）。

10.3 命名目标

C++ Java	<code>void add_nameddest(String name, String optlist)</code>
Perl PHP	<code>PDF_add_nameddest(resource p, string name, string optlist)</code>
C	<code>void PDF_add_nameddest(PDF *p, const char *name, int len, const char *optlist)</code>

在当前文档的任意页面上创建命名目标。

name （超文本字符串）目标的名称，可用作链接、书签或其他触发器的目标。在一个文档内，目标名称必须是唯一的。如果文档中多次提供了同一名称，将只使用最后一个定义，所有其他定义都将忽略，而不返回任何信息。

len （仅用于 C 语言绑定）UTF-16 字符串的 **name** 长度（以字节为单位）。如果 **len = 0**，则必须提供以 **null** 终结的字符串。

optlist 选项列表，根据表 10.3 指定目标。空列表相当于 `{type fitwindow page 0}`。可以使用下列选项：

bottom、**group**、**hypertext-encoding**、**hypertext-format**、**left**、**页面**、**right**、**top**、**类型**、**zoom**

详细说明 目标详细信息必须在 **optlist** 中指定，且目标可位于当前文档中的任一页面之上。所提供的 **name** 可以与 `PDF_create_action()`、`PDF_create_annotation()`、`PDF_create_bookmark()`、和 `PDF_begin/end_document()` 中的 **destname** 选项一起使用。以此方式可以将目标的定义和使用分为两个单独的步骤。

另外，如果在使用时已经知道目标，可以通过使用这些函数的 **destination** 选项合并命名目标的定义和使用，并且在此情况下，不需要使用 `PDF_add_nameddest()`。

范围 文档、页面

表 10.3 `PDF_add_nameddest()` 的目标选项以及 `PDF_create_action()`、`PDF_create_annotation()`、`PDF_create_bookmark()` 和 `PDF_begin/end_document()` 中的 **destination** 选项

选项	说明
bottom	（浮点型；仅用于 type=fitrect 的情况）页面的 y 坐标，位于窗口的底边。默认值：0
group	（字符串；如果已指定了 page 选项，且文档使用页面组，则是必需的；否则，不允许使用此选项。）目标页面所属页面组的名称。
hypertext-encoding	（关键字）指定 name 参数的编码方式。空字符串相当于 unicode 。默认值：全局 hypertextencoding 参数的值
hypertext-format	（关键字）设置 name 参数的格式。其可选值为 bytes 、 utf8 、 utf16 、 utf16le 、 utf16be 和 auto 。默认值： hypertextformat 参数的值
left	（浮点型；仅用于 type=fixed 、 fitheight 、 fitrect 或 fitvisibleheight ）页面的 x 坐标，位于窗口左边缘。默认值：0
页面	（整型）目标页面的页码（第一页为 1）。目标 PDF 中必须存在此页面。在页面范围中， Page 0 表示当前页面；在文档范围中，表示第 1 页。请注意，由于存在错误， Acrobat 6.0 将忽略页码，始终都会跳转到第 1 页。 Acrobat 6.0.1 中已修复了这个错误，旧版本中亦不存在此错误。默认值：0

表 10.3 PDF_add_nameddest() 的目标选项以及 PDF_create_action()、PDF_create_annotation()、PDF_create_bookmark() 和 PDF_begin/end_document() 中的 destination 选项（续）

选项	说明
right	（仅用于 type=fitrect 的情况）页面的 x 坐标，位于窗口右边缘。默认值：1000
top	（浮点型；仅用于 type=fixed、fitwidth、fitrect 或 fitvisiblewidth 的情况）页面的 y 坐标，位于窗口顶边缘。默认值：1000
类型	（关键字）指定目标页面上的窗口位置。默认值：fitwindow。 fixed 使用 left、top 和 zoom 选项指定的固定目标视图。如果缺少任一选项，将保留其当前值。 fitwindow 将整个页面填充至窗口内。 fitwidth 根据窗口调整页面宽度，y 坐标顶部与窗口的上边缘对齐。 fitheight 根据窗口调整页面高度，x 坐标左边与窗口的左边缘对齐。 fitrect 将 left、bottom、right 和 top 指定的矩形填充至窗口内。 fitvisible 将页面的可见内容（作品框）填充至窗口内。 fitvisiblewidth 将页面的可见内容填充至窗口内，y 坐标顶部与窗口的上边缘对齐。 fitvisibleheight 将页面的可见内容填充至窗口内，x 坐标左边与窗口的左边缘对齐。
zoom	（仅用于 type=fixed 的情况）用来显示页面内容的缩放因子（1 表示 100%）。如果此选项缺失，或者为 0，将保留启用路径时生效的缩放因子。

10.4 注释

C++ Java	<code>void create_annotation(double llx, double lly, double urx, double ury, String type, String optlist)</code>
Perl PHP	<code>PDF_create_annotation(resource p, float llx, float lly, float urx, float ury, string type, string optlist)</code>
C	<code>void PDF_create_annotation(PDF *p, double llx, double lly, double urx, double ury, const char *type, const char *optlist)</code>

在当前页面上创建一个矩形注释。

`llx, lly, urx, ury` 用缺省坐标（如果 `usercoordinates` 参数或选项为 `false`）或用户坐标（如果为 `true`）确定的注释矩形左下角和右上角的 `x` 和 `y` 坐标。Acrobat 会将注释的左上角与指定矩形的左上角对齐。

请注意，注释坐标不同于 `PDF_rect()` 函数的参数。`PDF_create_annotation()` 直接设置两个角的参数，而 `PDF_rect()` 只设置一个角的坐标，及宽度和高度值。

如果已指定 `usematchbox` 选项，将忽略 `llx/lly/urx/ury` 参数。

类型 注释的类型，用下列一个关键字指定：

- ▶ **3D:** (PDF 1.6) 3D 动画模型
此类型的特定选项：`3Dactivate`、`3Ddata`、`3Dinteractive`、`3Dshared`、`3Dinitialview`
- ▶ **Circle:** 圆形注释
此类型的特定选项：`interiorcolor`
- ▶ **FileAttachment:**（不用于 PDF/A）文件附件注释。Acrobat Reader 5 不能处理文件附件，将只显示一个问号。文件附件只适用于完全版的 Acrobat 软件。
此类型的特定选项：`filename`、`iconname`、`mimetype`
- ▶ **FreeText:** 自由文本注释
此类型的特定选项：`alignment`、`fillcolor`、字体、`fontsize`、`orientate`
- ▶ **Highlight:** 高亮注释
此类型的特定选项：`polylinelist`
- ▶ **Ink:** 油墨注释
此类型的特定选项：`polylinelist`
- ▶ **Line:** 线条注释
此类型的特定选项：`endingstyles`、`interiorcolor`、`line`
- ▶ **Link:** 链接注释
此类型的特定选项：`destination`、`destname`、`highlight`
- ▶ **Polygon:** (PDF 1.5) 多边形注释（用直线连接顶点）
此类型的特定选项：`polylinelist`
- ▶ **PolyLine:** (PDF 1.5) 折线 (polyline) 注释；类似于多边形，但第一个和最后一个顶点没有连接在一起。
此类型的特定选项：`endingstyles`、`interiorcolor`、`polylinelist`
- ▶ **Popup:** 弹出式注释
此类型的特定选项：`open`、`parentname`
- ▶ **Square:** 正方形注释
此类型的特定选项：`interiorcolor`
- ▶ **Squiggly:** (PDF 1.4) 曲线下划线注释
此类型的特定选项：`polylinelist`
- ▶ **Stamp:** 橡皮图章注释
此类型的特定选项：`iconname`、`orientate`
- ▶ **StrikeOut:** 删除线注释
此类型的特定选项：`polylinelist`

- **Text**: 文本注释。在 Acrobat 中，此类型称为附注注释。
此类型的特定选项: *iconname*、*open*
- **Underline**: 下划线注释
此类型的特定选项: *polylinelist*

optlist 选项列表，根据表 10.4 指定注释属性。所有注释类型支持下列选项（有关特定于类型的附加选项，参见上文）：
action、*annotcolor*、*borderstyle*、*cloudy*、*contents*、*createdate*、*custom*、*dasharray*、*display*、*hypertext-encoding*、*layer*、*linewidth*、*locked*、*name*、*opacity*、*popup*、*readonly*、*rotate*、*subject*、*title*、*usematchbox*、*user-coordinates*、*zoom*

详细说明 在所有 PDF/X 模式下，只有注释完全位于出血框之外（如果没有出血框，则应位于裁切框 / 作品框之外），才允许使用注释。
 标签 PDF: 如果项目当前是活动的，将作为当前项目的子项插入注释。

范围 页面

表 10.4 *PDF_create_annotation()* 的选项

选项	说明
3Dactivate	（选项列表；仅用于 <i>type=3D</i> 的情况）指定应在何时激活 3D 注释，以及其在激活 / 停用时的状态。 支持的选项： enable （关键字）指定应在何时启用注释。默认值: <i>click</i> 。 open 打开页面时激活。 visible 页面开始可见时激活。 click 必须通过脚本或用户动作明确地激活注释。 enablestate （关键字）初始动画状态。默认值: <i>play</i> 。 pause 已实例化 3D 模型，但脚本动画被禁用。 play 已实例化 3D 模型；脚本动画已启用（如果存在）。 disable （关键字）指定应在何时禁用注释。默认值: <i>invisible</i> 。 close 关闭页面时停用。 invisible 页面开始不可见时停用。 click 必须通过脚本或用户动作明确地停用注释。 disablestate （关键字）注释在禁用时的状态。默认值: <i>reset</i> 。 pause 可以呈现 3D 模型，但动画被禁用。 play 可以呈现 3D 模型，并且已启用动画。 reset 3D 模型在以任何方式被使用之前的初始状态。
3Ddata	（选项列表；仅用于 <i>type=3D</i> 的情况；必需的）使用 <i>PDF_load_3ddata()</i> 创建的 3D 句柄。
3Dinteractive	（布尔型；仅用于 <i>type=3D</i> 的情况）如果为 <i>true</i> ，则 3D 模型用于交互。如果为 <i>false</i> ，将使用 <i>JavaScript</i> 操作 3D 模型。默认值: <i>true</i>
3Dshared	（布尔型；仅用于 <i>type=3D</i> 的情况）如果为 <i>true</i> ，将间接引用 3Ddata 选项中指定的 3D 数据。间接引用同一数据的多个 3D 注释将共享模型的一个运行时实例。这表示，更改将在所有此类注释中同时可见。默认值: <i>false</i>
3Dinitialview	（关键字或 3D 视图句柄）指定 3D 模型的初始视图；可选用下列关键字之一: <i>first</i> 、 <i>last</i> （参见模型的 <i>views</i> 选项中的相应项）或 <i>default</i> （参见模型的 <i>defaultview</i> 选项），或使用 <i>PDF_create_3dview()</i> 创建的 3D 视图句柄。默认值: <i>default</i>
action	（动作列表）下列事件的注释动作列表。默认值: 空列表。 activate 启动注释时执行的动作。允许所有动作类型。
alignment	（关键字；仅用于 <i>type=FreeText</i> 的情况）注释中的文本对齐方式: <i>left</i> 、 <i>center</i> 、 <i>right</i> 。此选项不能用于 <i>Acrobat 6</i> ，它一直使用 <i>left</i> 。默认值: <i>left</i>
annotcolor	（颜色）关闭时的注释图标背景颜色，注释弹出窗口的标题栏颜色以及链接注释的边框颜色。支持的色彩空间: <i>none</i> 、 <i>gray</i> 、 <i>rgb</i> 。默认值: <i>none</i> 在 <i>PDF/A</i> 模式下，只有指定了 <i>RGB</i> 输出方法时，才能使用此选项。
annotwarning	不常用，不要求使用此选项

表 10.4 PDF_create_annotation() 的选项（续）

选项	说明
borderstyle	（关键字）注释边框样式或多边形、多边线条、线条、正方形、圆形和油墨型注释的线条样式。 <i>solid</i> 、 <i>beveled</i> 、 <i>dashed</i> 、 <i>inset</i> 、 <i>underline</i> 。请注意， <i>beveled</i> 、 <i>inset</i> 和 <i>underline</i> 样式无法在 <i>Acrobat</i> 中可靠运行。默认值：solid
cloudy	（浮点型；仅用于 type=Polygon 的情况；PDF 1.5）指定用来渲染多边形的 » 云彩 « 效果的强度。其可选值为 0（无效果）、1 和 2。如果使用此选项，将忽略 borderstyle 选项。默认值：0
contents	（超文本字符串，最大长度为 65535 字节）注释显示的文本或者（如果注释不显示文本）注释内容的替代描述（必须使用用户可读的形式）。可使用回车符或换行符强制生成新段落。 对于 type=Circle、FileAttachment、FreeText、Highlight、Ink、Line、Polygon、PolyLine、Square、Squiggly、Stamp、Strikeout、Text、Underline 要求使用此选项，但可以是空字符串。如果 type=FreeText，此选项必须是字符串类型。对于 type=Link、PopUp，此选项是可选的。 在 PDF/A-1a 模式下，此选项是必需的（并且必须包含非空字符串）。
createdate	（布尔型；PDF 1.5 或更高版本）如果为 true，将为注释创建日期和时间条目。默认值：false
custom	（选项列表的列表；仅面向高级用户）可使用此选项，在注释词典内插入任意数量的专用条目，这可能对特殊的应用程序（如为数字打印机插入处理指令）。使用此选项要求用户熟悉 PDF 文件格式及目标应用程序。如果提供的值不合适，可能会生成损坏的 PDF 输出。每个列表都必须包含 3 个选项： key （字符串）词典关键字的名称（不包括 / 字符）。可指定任何非标准的 PDF 关键字，以及下列标准关键字：Contents、Name（选项 iconname）、NM（选项 name）、和 Open。此时，将忽略对应的选项。 type （关键字）对应值的类型，必须为 boolean、name 或 string 之一 value （如果 type=string，为超文本字符串；否则，为字符串）PDF 输出中显示的值；PDFlib 将自动提供字符串和名称要求的任何修饰效果。
dasharray	（浮点值列表；仅用于 borderstyle=dashed 的情况）。虚线边框的虚线和间隙长度（以缺省单位表示）（参见 PDF_setdash()）。默认值：3 3
destination	（选项列表；仅用于 type=Link 的情况；如果指定了激活动作，将被忽略）根据表 10.3 的选项列表，定义要跳转到的目标。
destname	（超文本字符串；仅用于 type=Link 的情况；如果指定了 destination 选项，将被忽略）已使用 PDF_add_nameddest() 定义的目标名称。Destination 或 destname 动作将优先于此选项。
display	（关键字）在屏幕和纸张上的可见性：visible、hidden、noview、noprint。默认值：visible
endingstyles	（关键字列表；仅用于 type=Line、PolyLine 的情况）由两个关键字组成的列表，指定直线终点样式：none、square、circle、diamond、openarrow、closedarrow。默认值：{none none}
filename	（字符串；仅用于 type=FileAttachment 的情况，要求此选项）与注释关联的文件。建议只在 filename 中使用 ASCII 字符。
fillcolor	（颜色；仅用于 type=FreeText 的情况）文本的填充色。支持的色彩空间：gray、rgb、cmyk。 默认值：{gray 0}（代表黑色） 在 PDF/A 模式下，只有指定了 RGB 或 CMYK 输出方法之后，才能使用此选项，并且必须使用对应的 rgb 或 cmyk 色彩空间。
字体	（字体句柄；仅用于 type=FreeText 的情况；是必需的）指定注释使用的字体。只允许使用 PDF 核心字体以及下列编码方式：任何 8 位编码、Unicode CMap、内置编码。
fontsize	（浮点值或选项列表；仅用于 type=FreeText 的情况；必需的）根据 usercoordinates 选项或参数，确定字体大小的单位为缺省坐标系或用户坐标系。详情见 PDF_fit_textline()。
highlight	（关键字；仅用于 type=Link 的情况）用户点击注释时显示的高亮模式：none、invert、outline、push。默认值：invert

表 10.4 PDF_create_annotation() 的选项（续）

选项	说明
hypertext-encoding	（关键字）指定提供的文本的编码方式。空字符串相当于 <i>unicode</i> 。默认值：全局 <i>hypertextencoding</i> 参数的值
iconname	（字符串；仅用于 <i>type=Text</i> 、 <i>Stamp</i> 、 <i>FileAttachment</i> 的情况）显示注释时使用的图标名称（若要创建无任何可见图标的注释，请设置 <i>opacity=0</i> ）： <i>type=Text</i> （缺省值： <i>note</i> ）： <i>comment</i>  、 <i>key</i>  、 <i>note</i>  、 <i>help</i>  、 <i>newparagraph</i>  、 <i>paragraph</i>  、 <i>insert</i>  <i>type=Stamp</i> （缺省值： <i>draft</i> ）： <i>approved</i> 、 <i>experimental</i> 、 <i>notapproved</i> 、 <i>asis</i> 、 <i>expired</i> 、 <i>notforpublicrelease</i> 、 <i>confidential</i> 、 <i>final</i> 、 <i>sold</i> 、 <i>departmental</i> 、 <i>forcomment</i> 、 <i>topsecret</i> 、 <i>draft</i> 、 <i>forpublicrelease</i> 。 <i>type=FileAttachment</i> （缺省值： <i>pushpin</i> ）： <i>graph</i>  、 <i>pushpin</i>  、 <i>paperclip</i>  、 <i>tag</i> 
interiorcolor	（颜色；仅用于 <i>type=Line</i> 、 <i>PolyLine</i> 、 <i>Square</i> 、 <i>Circle</i> 的情况）分别指定注释直线终点、矩形或椭圆形的颜色。支持的色彩空间： <i>none</i> 、 <i>gray</i> 、 <i>rgb</i> 。默认值： <i>none</i> 在 <i>PDF/A</i> 模式下，只有指定了 <i>RGB</i> 输出方法时，才能使用此选项。
layer	（图层句柄； <i>PDF 1.5</i> ）注释所属的图层。仅当对应图层可见时，注释才可见。
line	（由 4 个浮点值组成的列表；仅用于 <i>type=Line</i> 的情况；要求此选项）由 4 个数值 <i>x1</i> 、 <i>y1</i> 、 <i>x2</i> 和 <i>y2</i> 组成的列表，指定线条的起点和终点坐标；如果 <i>usercoordinates</i> 参数为 <i>false</i> ，使用缺省坐标系；如果此参数为 <i>true</i> ，使用用户坐标系。
linewidth	（整型）注释边框宽度或线条、折线、多边形、正方形、圆形和油墨型注释的线条宽度，使用缺省单位（= <i>points</i> ）。如果 <i>linewidth=0</i> ，将不显示边框。默认值： <i>1</i>
locked	（布尔型）如果为 <i>true</i> ，无法在 <i>Acrobat</i> 中编辑注释属性。默认值： <i>false</i>
mimetype	（字符串；仅用于 <i>type=FileAttachment</i> ）的情况）文件的 <i>MIME</i> 类型。启动注释时， <i>Acrobat</i> 将用它启动适当的应用程序。
name	（字符串）唯一标识注释的名称。有些动作必须具有名称，且在页面上必须是唯一的。默认值： <i>none</i>
opacity	（浮点型或百分比； <i>PDF 1.4</i> ）不透明度的常数值（ <i>0-1</i> 或 <i>0%-100%</i> ），绘制注释时会使用此常数值。默认值： <i>1</i>
open	（布尔型；仅用于 <i>type=Text</i> 、 <i>Popup</i> 的情况）如果为 <i>true</i> ，注释的初始状态显示为打开。默认值： <i>false</i>
orientate	（关键字；仅用于 <i>type=FreeText</i> 、 <i>Stamp</i> 的情况）指定注释在其矩形内的必需方向。默认值： <i>north</i> <i>north</i> 向上 <i>east</i> 指向右侧 <i>south</i> 向下 <i>west</i> 指向左侧
parentname	（字符串；仅用于 <i>type=PopUp</i> 的情况）注释的父级注释名称。
polylinelist	（包含一个或多个浮点值列表的列表；仅用于 <i>type=Polygon</i> 、 <i>PolyLine</i> 、 <i>Ink</i> 、 <i>Highlight</i> 、 <i>Underline</i> 、 <i>Squiggly</i> 、 <i>Strikeout</i> 的情况）。如果 <i>usercoordinates</i> 选项为 <i>false</i> ，将使用缺省坐标系做参照；如果此选项为 <i>true</i> ，则使用用户坐标系做参照。默认值：连接注释矩形各顶点的折线。 <i>type=Polygon</i> 、 <i>PolyLine</i> 、 <i>Ink</i> 一个包含带 <i>n</i> 线段的折线的列表（最小值： <i>n=2</i> ）。折线是一个指定成对坐标的 <i>2 x n</i> 浮点值列表。这些点将用直线连接。 <i>n=3</i> 的示例：{{10 20 30 40 50 60 }} <i>others</i> 列表包含 <i>n</i> 个子列表，各带 8 个浮点值，指定 <i>n</i> 个四边形（最小值： <i>n=1</i> ）。每个四边形都包含注释下方文本中的一个单词或一组连续单词。每个四边形的坐标都以 <i>x4 y4 x3 y3 x1 y1 x2 y2</i> 的形式给出，按逆时针方向指定四边形的顶点（ <i>x4 y4</i> 为左上角）。文本的方向取决于边缘连接（ <i>x1,y1</i> ）和（ <i>x2,y2</i> ）。 <i>n=2</i> 的示例：{{1 2 3 4 5 6 7 8} {10 20 30 40 50 60 70 80}}

表 10.4 `PDF_create_annotation()` 的选项（续）

选项	说明
popup	（字符串）弹出式注释的名称，用于输入或编辑此注释关联的文本。默认值： <i>none</i>
readonly	（布尔型）如果为 true ，将不允许注释与用户交互。可以显示或打印注释，但不能响应鼠标点击操作或者根据鼠标动作更改注释外观。默认值： false
rotate	（布尔型；在 <i>PDF/A</i> 模式下不能设置为 true ）如果为 true ，将旋转注释，以配合页面的旋转。否则，注释旋转将保持固定不变。文本注释的图标会忽略此选项。默认值：在 <i>PDF/A</i> 模式下为 false ，否则为 true
subject	（超文本字符串； <i>PDF 1.5</i> ）简要描述注释说明的主题的文本。默认值： <i>none</i>
title	（超文本字符串）打开注释或处于活动状态时，注释弹出窗口标题栏显示的文本标签（在 <i>Acrobat</i> 中，此标签将显示 » 作者 «）。标题的最大长度为 255 个单字节字符或 126 个 <i>Unicode</i> 字符。但是，建议将标题的实际限制设置为 32 个字符。此字符串与 <i>Acrobat</i> 中的 » 作者 « 域相对应。默认值： <i>none</i>
usematchbox	（字符串列表）将忽略 <i>llx/lly/urx/ury</i> 参数，改用匹配框。选项列表中的第一个元素是一个指定匹配框的名称字符串。第二个元素是一个指定所需矩形数目的整数，或者是关键字 all ，指定参考所选择的匹配框的所有矩形。如果第二个元素缺失，则默认为 all 。 如果在当前页面上不存在匹配框本身或指定的矩形，函数将返回，不创建任何注释，也不发出提示。
user-coordinates	（布尔型）如果为 false ，将预计注释坐标和字体大小将使用缺省坐标系；否则，将使用当前用户坐标系。默认值：全局 <i>usercoordinates</i> 参数的值
zoom	（布尔型；在 <i>PDF/A</i> 模式下不能设置为 true ）如果为 true ，将缩放注释，以配合页面的放大。否则，注释大小将保持固定不变。文本注释的图标会忽略此选项。默认值：在 <i>PDF/A</i> 模式下为 false ，否则为 true

10.5 表单域

C++ Java	<code>void create_field(double llx, double lly, double urx, double ury, String name, String type, String optlist)</code>
Perl PHP	<code>PDF_create_field(resource p, float llx, float lly, float urx, float ury, string name, string type, string optlist)</code>
C	<code>void PDF_create_field(PDF *p, double llx, double lly, double urx, double ury, const char *name, int len, const char *type, const char *optlist)</code>

根据各种选项，在当前页面上创建表单域。

llx, lly, urx, ury 确定的表单矩形左下角和右上角的 *x* 和 *y* 坐标；如果 *usercoordinates* 参数或选项为 *false*，用缺省坐标系；如果为 *true*），用户坐标系。








请注意，表单域坐标不同于 *PDF_rect()* 函数的参数。*PDF_create_field()* 直接设置两个角的参数，而 *PDF_rect()* 只设置一个角的坐标，及宽度和高度值。

name （超文本字符串）表单域名称，可能会附加用 *PDF_create_fieldgroup()* 创建的一个或多个组的名称前缀。组名称必须用句点 *».* *«* 字符相互隔开，并同域名称隔开。在一个页面上，域名称必须是唯一的，且不能以句点 *».* *«* 字符结尾。

len （仅用于 C 语言绑定）UTF-16 字符串的 *text* 长度（以字节为单位）。如果 *len = 0*，则必须提供以 *null* 终结的字符串。

type 根据表 10.5 提供的域类型。

表 10.5 表单域类型

类型	icon	此类型的特定选项:
pushbutton		<i>button-layout</i> 、 <i>caption</i> 、 <i>captiondown</i> 、 <i>caption-rollover</i> 、 <i>charspacing</i> 、 <i>fitmethod</i> 、 <i>icon</i> 、 <i>icondown</i> 、 <i>iconrollover</i> 、 <i>position</i> 、 <i>submitname</i>
checkbox		<i>currentvalue</i> 、 <i>itemname</i>
radiobutton		<i>buttonstyle</i> 、 <i>currentvalue</i> 、 <i>itemname</i> 、 <i>toggle</i> 、 <i>unisonselect</i> 由于单选按钮必须始终属于同一组，所以 <i>name</i> 必须附加组名称前缀。对于所有其他域类型，组关系是可选的。
listbox		<i>charspacing</i> 、 <i>currentvalue</i> 、 <i>item-namelist</i> 、 <i>itemtextlist</i> 、 <i>multiselect</i> 、 <i>sorted</i> 、 <i>topindex</i>
combobox		<i>commit-onselect</i> 、 <i>charspacing</i> 、 <i>currentvalue</i> 、 <i>editable</i> 、 <i>item-namelist</i> 、 <i>itemtextlist</i> 、 <i>sorted</i> 、 <i>spellcheck</i>
textfield		<i>comb</i> 、 <i>charspacing</i> 、 <i>currentvalue</i> 、 <i>fileselect</i> 、 <i>maxchar</i> 、 <i>multiline</i> 、 <i>password</i> 、 <i>richtext</i> 、 <i>scrollable</i> 、 <i>spellcheck</i>
signature		<i>charspacing</i> 、 <i>lockmode</i>

optlist 一个选项列表，根据表 10.6 指定域的属性。如此表所示，字符串选项将解释为超文本字符串或文本字符串。所有域类型支持下列选项（有关更多特定于类型的选项，参见表 10.5）：

action、*alignment*、*background-color*、*bordercolor*、*borderstyle*、*calcorder*、*dasharray*、*defaultvalue*、*display*、*errorpolicy*、*exportable*、*fieldtype*、*fillcolor*、字体、*fontsize*、*highlight*、*hypertext-encoding*、*hypertext-format*、*layer*、*linewidth*、*locked*、*orientate*、*readonly*、*required*、*strokecolor*、*taborder*、*tooltip*、*user-coordinates*

详细说明 缺省情况下，页面上各域的跳位顺序（按 Tab 键时，各个域接收焦点的顺序）取决于 *PDF_create_field()* 调用的顺序，但可以使用 *taborder* 选项指定不同的顺序。创建域之后，便不能修改跳位顺序。不过，可以使用 *PDF_begin/end_page_ext()* 的 *taborder* 选项覆盖此行为。

在 Acrobat 中，可以为文本域指定一个格式（数值、百分比等）。但 PDF Reference 中没有相关规定，而是使用自定义 JavaScript 实施的。将 JavaScript 动作附加到指向 Acrobat 中预定义的（而非标准化的）JavaScript 函数的表单域，可异曲同工。

不能在标签 PDF/A 模式下使用此函数。

在所有 PDF/X 模式下，只有表单域完全位于出血框之外（如果没有出血框，则应位于裁切框 / 作品框之外）时，才允许使用表单域。

标签 PDF：如果项目当前是活动的，将作为当前项目的子项插入域。

范围 页面

C++ Java

void create_fieldgroup(String name, String optlist)

Perl PHP

PDF_create_fieldgroup(resource p, string name, string optlist)

C

void PDF_create_fieldgroup(PDF *p, const char *name, int len, const char *optlist)

根据各种选项创建表单域组。

name （超文本字符串）表单域组的名称，也可作为另一组的前缀。可以任意级别嵌套域组。组名称必须用句点 ».« 字符分隔。在一个文档中，组名称必须是唯一的，且不能以句点 ».« 字符结尾。

len （仅用于 C 语言绑定）UTF-16 字符串的 *text* 长度（以字节为单位）。如果 *len = 0*，则必须提供以 null 终结的字符串。

optlist 选项列表，根据表 10.6 指定域属性。

表 10.6 *PDF_create_field()* 和 *PDF_create_fieldgroup()* 的表单属性选项

选项	说明
action	（动作列表）下列一个或多个事件的域动作列表。所有域类型都允许启动事件； <i>type=pushbutton</i> 、 <i>checkbox</i> 和 <i>radiobutton</i> 时，不允许使用其他事件。默认值：空列表。 activate 启动域时执行的动作。 keystroke 用户在文本域或组合框内键入内容或修改滚动列表框内的选定时，执行的 <i>JavaScript</i> 动作。 format 对域进行格式化以显示其当前值之前执行的 <i>JavaScript</i> 动作。这样便可以在格式化之前修改域的值。 validate 更改域的值时执行的 <i>JavaScript</i> 动作。这样便可以检查新值的有效性。 calculate 另一个域的值发生更改时执行的 <i>JavaScript</i> 动作；这样可以重新计算此域的值。 enter 鼠标进入域的区域时执行的动作。 exit 鼠标退出域的区域时执行的动作。 down 在域的区域内容下鼠标按钮时执行的动作。 up 在域的区域内容放鼠标按钮时执行的动作（这通常用来启动域）。 focus 域收到输入焦点时执行的动作。 blur 域失去输入焦点时执行的动作。
alignment	（关键字）域内文本的对齐方式： <i>left</i> 、 <i>center</i> 、 <i>right</i> 。默认值： <i>left</i>
background-color bordercolor	（颜色）域背景或边框的颜色。支持的色彩空间： <i>none</i> 、 <i>gray</i> 、 <i>rgb</i> 、 <i>cmymk</i> 。默认值： <i>none</i>
borderstyle	（关键字）域边框的样式，可以是 <i>solid</i> 、 <i>beveled</i> 、 <i>dashed</i> 、 <i>inset</i> 或 <i>underline</i> 。默认值： <i>solid</i>
button-layout	（关键字；仅用于 <i>type=pushbutton</i> 的情况）按钮标题相对于按钮图标的位置；条件是，同时指定了按钮标题和按钮图标： <i>below</i> 、 <i>above</i> 、 <i>right</i> 、 <i>left</i> 、 <i>overlaid</i> 。默认值： <i>right</i>
buttonstyle	（关键字；仅用于 <i>type=radiobutton</i> 和 <i>checkbox</i> 的情况）指定域使用的符号： <i>check</i> 、 <i>cross</i> 、 <i>diamond</i> 、 <i>circle</i> 、 <i>star</i> 、 <i>square</i> 。默认值： <i>check</i>

表 10.6 PDF_create_field() 和 PDF_create_fieldgroup() 的表单属性选项（续）

选项	说明
calcorder	（整型；仅用于域具有计算事件的 <i>JavaScript</i> 动作的情况）指定此域相对于其他域的计算顺序。将先计算较小编码的域，然后计算较大编码的域。默认值： 10 加上当前页面上使用的最大 calcorder （最初为 10 ）
caption	（内容字符串；仅用于 type=pushbutton 的情况，对于下压按钮，必须提供一个 caption 或 icon 选项）标题文本，按钮没有输入焦点时将显示此文字。如果不需要标题或图标，可使用空字符串（即 caption{} ）。默认值： none
captiondown	（内容字符串；仅用于 type=pushbutton 的情况）启动按钮时显示的标题文字。默认值： none
caption-rollover	（内容字符串；仅用于 type=pushbutton 的情况）按钮具有输入焦点时显示的标题文字。默认值： none
charspacing	（浮点型；不能用于 type=radiobutton 和 checkbox 的情况）域内文本的字符间距，用当前用户坐标系统的单位表示。 <i>Acrobat 7</i> 中会忽略此选项。默认值： 0
comb	（布尔型；仅用于 type=textfield 的情况； <i>PDF 1.5</i> ）如果为 true ， multiline 、 fileselect 和 password 选项为 false ，且为 maxchar 选项提供了一个整数，则针对具体的字符将域分割为几个等距离的子域（根据 maxchar 值）。默认值： false
commit-onselect	（布尔型；仅用于 type=listbox 和 combobox 的情况； <i>PDF 1.5</i> ）如果为 true ，选中后将立即提交在列表中选中的项目。如果为 false ，将在退出域时提交所选项目。默认值： false
currentvalue	（不能用于 type=pushbutton 和 signature 的情况）域的初始值。类型和缺省值取决于域类型。 checkbox 、 radiobutton （字符串）除 Off 之外的任何字符串均表示按钮已启动；如果指定了 itemname 并且 / 或者 unisonselect 为 true ，则 <i>Acrobat 6</i> 将显示不确定的行为。字符串 Off 表示按钮已停用。应为第一个按钮设置此选项。默认值： Off textfield 、 combobox （内容字符串）域的内容。默认值： empty listbox （整数列表） itemtextlist 内所选项目的索引。默认值： none
dasharray	（浮点值列表；仅用于 borderstyle=dashed 的情况）。虚线边框的虚线和间隙长度（以缺省单位表示）（参见 <i>PDF_setdash()</i> ）。默认值： 3 3
defaultvalue	reset 动作之后的域值。类型和缺省值与 currentvalue 选项相同。例外：对于 listbox ，只允许一个整数。
display	（关键字）在屏幕和纸张上的可见性： visible 、 hidden 、 noview 、 noprint 。默认值： visible
editable	（布尔型；仅用于 type=combobox 的情况）如果为 true ，可以编辑框中当前选中的文本。默认值： false
errorpolicy	（关键字）控制出现错误时的行为（参见第 28 页上的第 2.6 节“异常处理”）。
exportable	（布尔型）发生 SubmitForm 动作时，将导出域。默认值： true
fieldtype	（关键字；仅用于 <i>PDF_create_fieldgroup()</i> 组中包含的域的类型： mixed 、 pushbutton 、 checkbox 、 radiobutton 、 listbox 、 combobox 、 textfield 或 signature 。除非 fieldtype=mixed ，否则组中只能包含指定类型的域。如果已经为组指定了特定 fieldtype ，将在所有包含的域中同时显示当前值；即使这些域位于不同的页面上，也是如此。如果 fieldtype=radiobutton ，必须提供选项 unisonselect 。必须在域组选项中指定 itemtextlist 、 itemnamelist 、 currentvalue 和 defaultvalue 选项，而非在单独域的选项指定。默认值： mixed
fieldwarning	不常用，使用 errorpolicy
fileselect	（布尔型；仅用于 type=textfield 的情况）如果为 true ，域中的文本将视为一个文件名。默认值： false
fillcolor	（颜色）文本的填充色。支持的色彩空间： gray 、 rgb 、 cmymk 。默认值： {gray 0} （代表黑色）

表 10.6 PDF_create_field() 和 PDF_create_fieldgroup() 的表单属性选项（续）

选项	说明
fitmethod	（关键字；仅用于 type=pushbutton 的情况）用按钮内的 icon、icondown 和 iconrollover 选项提供的模板的放置方法。默认值：meet。 auto 如果模板可放入按钮内，则相当于 meet；否则，相当于 clip。 nofit 相当于 clip。 clip 不缩放模板，而在域边框处剪切模板。 meet 将按比例缩放模板，使其可以置于按钮内。 slice 相当于 meet。 entire 将缩放模板，使其可以完全置于按钮内。
字体	（字体句柄；必需的，除 type=radiobutton 和 checkbox 这两种总是使用 ZapfDingbats 的情况外）。指定域使用的字体。必须已在相应的 PDF_load_font() 调用中设置了下列选项：embedding（不需要嵌入的核心字体除外）、nosubsetting、noautocidfont。只允许下列编码方式：8 位编码、任何 CMap（但仅用于标准 CJK 字体）、内置编码
fontsize	（浮点型、选项列表或关键字）字体大小以用户坐标系衡量。如果提供了关键字 auto，而非浮点值，Acrobat 将自动确定字体大小。详情见 PDF_fit_textline()。默认值：auto
highlight	（关键字）用户点击域时显示的高亮模式：none、invert、outline、push。默认值：invert
hypertext-encoding	（关键字）指定 name 参数的编码方式。空字符串相当于 unicode。默认值：全局 hypertextencoding 参数的值
hypertext-format	（关键字）设置 name 参数的格式。其可选值为 bytes、utf8、utf16、utf16le、utf16be 和 auto。默认值：hypertextformat 参数的值
icon	（模板句柄 ¹ ；仅用于 type=pushbutton 的情况，且必须为按钮提供一个 caption 或 icon 选项）模板句柄，按钮没有输入焦点时将显示此句柄。默认值：none
icondown	（模板的句柄 ¹ ；仅用于 type=pushbutton 的情况）模板的句柄，启动按钮时显示此句柄。默认值：none
iconrollover	（模板的句柄 ¹ ；仅用于 type=pushbutton 的情况）模板的句柄，按钮具有输入焦点时显示此句柄。默认值：none
itemname	（超文本字符串；仅用于 type=radiobutton 和 checkbox 的情况；如果导出值不是 Latin 1 字符串，是必需的）域的导出值。在一组中，多个单选按钮可具有相同的项目名称。Acrobat 6：即使位于不同的页面之上，一组中具有相同项目名称的复选框也会同时打开或关闭。默认值：field name
item-namelist	（超文本字符串；仅用于 type=listbox 和 combobox 的情况）列表项目的导出值。多个项目可以具有相同的导出值。默认值：none
itemtextlist	（内容字符串列表；仅用于 type=listbox 和 combobox 的情况；这些情况下，要求此选项）列表内所有项目的文本内容。如果同时指定了 itemnamelist 和 itemtextlist，则它们必须包含相同数量的字符串。
layer	（图层句柄；PDF 1.5）域所属的图层。仅当对应图层可见时，域才可见。
linewidth	（整型）域边框的线宽，用缺省单位表示（= points）。默认值：1
locked	（布尔型）如果为 true，无法在 Acrobat 中编辑域属性。默认值：false
lockmode	（关键字；仅用于 type=signature 的情况；PDF 1.5）指示在对域签名时应该锁定的一组域： all 将锁定文档中的所有域。
maxchar	（整型或关键字；仅用于 type=textfield 的情况）域内文本字符数量的上限；如果没有限制，则指定关键字 unlimited。默认值：unlimited
multiline	（布尔型；仅用于 type=textfield 的情况）如果为 true，文本将根据需要自动换行，生成多行。默认值：false
multiselect	（布尔型；仅用于 type=listbox 的情况）如果为 true，可选择列表内的多个项目。默认值：false
orientate	（关键字）域内的内容的方向：north、west、south、east。默认值：north
password	（布尔型；仅用于 type=textfield 的情况）如果为 true，将在输出时使用项目符号或星号模拟文本。默认值：false
position	（浮点值或关键字列表；仅用于 type=pushbutton 的情况）在按钮中，用 icon... 选项提供的模板的相对位置。默认值：50 50

表 10.6 PDF_create_field() 和 PDF_create_fieldgroup() 的表单属性选项（续）

选项	说明
readonly ²	（布尔型）如果为 true ，域不允许任何输入。默认值： false
required	（布尔型）如果为 true ，提交表单时，域必须包含一个值。默认值： false
richtext	（布尔型；仅用于 type=textfield 的情况；PDF 1.5）允许丰富的文本格式。如果为 true ， fontsize 不能为 0 ，且 fillcolor 不能使用 cmymk 色彩空间。默认值： false
scrollable	（布尔型；仅用于 type=textfield 的情况）如果为 true ，且文本不能完全置于域内，文本将移至域外的隐藏区域。如果为 false ，文本填满整个域时，将不能接受任何更多的输入。默认值： true
sorted	（布尔型；仅用于 type=listbox 和 combobox 的情况）如果为 true ，将对列表的内容进行排序。默认值： false
spellcheck	（布尔型；仅用于 type=textfield 和 combobox 的情况）如果为 true ，将在域中启动拼写检查器。默认值： true
strokecolor	（颜色）文本的描边色。支持的色彩空间： gray 、 rgb 、 cmymk 。默认值： {gray 0} （代表黑色）。
submitname	（超文本字符串；建议只用于 type=pushbutton 的情况）表单将提交到的 Internet 地址的 URL 编码字符串。默认值： None
taborder	（整型）指定此域相对于其他域的跳位顺序。将先移到较小编码的域，然后到达较大编码的域。默认值： 10 加上当前页面上使用的最大 taborder （页面上的第一个域为 10 ）；使用此缺省值的结果是，创建顺序将指定跳位顺序。
toggle	（布尔型；仅用于 PDF_create_fieldgroup() 和 type=radiobutton 的情况）如果为 true ，可通过单击启动或停用组内的单选按钮。如果为 false ，只能通过点击启动，通过点击另一按钮停用。默认值： false
tooltip ²	（超文本字符串）在域的工具提示中显示的文本。对于单选按钮和按钮组， Acrobat 将一直使用组中第一个按钮的工具提示；其余的将被忽略。默认值： none
topindex	（整型；仅用于 type=listbox 的情况）第一个显示的条目的索引。第一个项目的索引为 0 。缺省值： 0
unisonselect	（布尔型；仅用于 PDF_create_fieldgroup() 且 type=radiobutton 的情况和 PDF 1.5 ）如果为 true ，将同时选中具有相同域名称或项目名称的单选按钮。默认值： false
user-coordinates	（布尔型）如果为 false ，域坐标将使用缺省坐标系统；否则，将使用当前用户坐标系统。默认值：全局 usercoordinates 参数的值

1. 可用 **PDF_begin_template()** 函数创建图标模板；如果图标只包含一个图像，可向 **PDF_load_image()** 提供模板选项，以创建模板。
2. 对于 **type=radiobutton** 的情况，此选项不能与 **PDF_create_field()** 一起使用，只能与 **PDF_create_fieldgroup()** 一起使用。

详细说明 域组对镜像一个或多个其他域中的域内容很有用。如果域组的名字将作为用 **PDF_create_field()** 创建的域名称的前缀提供一个域的名称，则新域将成为此组的一部分。域组 **optlist** 中提供的所有域属性选项都将被其属下所有的域继承。

范围 页面、文档

10.6 书签

C++ Java

int create_bookmark(String text, String optlist)

Perl PHP

int PDF_create_bookmark(resource p, string text, string optlist)

C

int PDF_create_bookmark(PDF *p, const char *text, int len, const char *optlist)

根据各种选项创建书签主题。

文本 （超文本字符串）包含书签的文本。*text* 的最大长度为 255 个单字节字符（8 位编码）或 126 个 Unicode 字符。但是，建议将 *text* 的实际限制设置为 32 个字符。

len （仅用于 C 语言绑定）UTF-16 字符串的 *text* 长度（以字节为单位）。如果 *len* = 0，则必须提供以 null 终结的字符串。

optlist 一个选项列表，根据表 10.7 指定书签的属性。可以使用下列选项：
action、*destination*、*destname*、*fontstyle*、*hypertext-encoding*、*hypertext-format*、*index*、*open*、*parent*、*textcolor*

返回 生成书签的句柄，可在后续调用中与 *parent* 选项一起使用。

详细说明 此函数添加带有 *text* 的 PDF 书签。除非已指定了 *destination* 选项，否则书签将指向当前页（如果用于**文档**范围，将指向最后一页；如在第一页之前使用，将指向第一页）。
除非已经专门设置了另一种模式，否则创建书签的操作可将 *PDF_begin/end_document()* 的 *openmode* 选项设为 *bookmarks*。

范围 文档、页面

表 10.7 Options for PDF_create_bookmark() 的选项

选项	说明
<i>action</i>	（动作列表）下列事件的书签动作列表。默认值：GoTo 动作且在 <i>destination</i> 选项中指定目标。 <i>activate</i> 启动书签时执行的动作。允许所有动作类型。
<i>destination</i>	（选项列表；如果已指定激活动作，将忽略此选项）选项列表，根据表 10.3 指定书签目标。如果缺少 <i>destination</i> 、 <i>destname</i> 和 <i>action</i> ，缺省值：{type fitwindow page 0}。
<i>destname</i>	（超文本字符串；可以有空；如果指定了 <i>destination</i> 选项，将被忽略）已使用 <i>PDF_add_nameddest()</i> 定义的目标名称。 <i>Destination</i> 或 <i>destname</i> 动作将优先于此选项。如果 <i>destname</i> 为空字符串（即 {}），且未指定 <i>destination</i> 或 <i>action</i> 选项，书签将不执行任何动作；如果将书签用作分隔符，这可能很有用。
<i>fontstyle</i>	（关键字）指定书签文本的字体样式：normal、bold、italic、bolditalic。默认值：normal
<i>hypertext-encoding</i>	（关键字）指定提供的文本的编码方式。空字符串相当于 <i>unicode</i> 。默认值：全局 <i>hypertextencoding</i> 参数的值
<i>hypertext-format</i>	（关键字）设置提供的文本的格式。其可选值为 <i>bytes</i> 、 <i>utf8</i> 、 <i>utf16</i> 、 <i>utf16le</i> 、 <i>utf16be</i> 和 <i>auto</i> 。默认值：全局 <i>hypertextformat</i> 参数的值
<i>index</i>	（整型）在父级书签内插入书签的位置所处的索引。可使用处于 0 到同一级别的书签数量之间的值，在父级书签内的指定位置上插入书签。可使用值 -1，作为当前级别的最后一个插入书签。默认值：-1。但是，对于插入和恢复的页面，放置书签时会假定已经按照其物理顺序生成了所有页面（书签将反映页面顺序）。
<i>open</i>	（布尔型）如果为 <i>false</i> ，将不显示从属书签。如果为 <i>true</i> ，将展开所有子级书签。默认值：false
<i>parent</i>	（书签句柄）新书签将为句柄中指定书签的从属书签。如果 <i>parent</i> =0，将创建顶级书签。默认值：0
<i>textcolor</i>	（颜色）指定书签文本的颜色。支持的色彩空间：none、gray、rgb。默认值：rgb {0 0 0}（代表黑色）



11 多媒体功能（3D 图片）

C++ Java	<code>int load_3ddata(String filename, String optlist)</code>
Perl PHP	<code>int PDF_load_3ddata(string filename, string optlist)</code>
C	<code>int PDF_load_3ddata(PDF *p, const char *filename, int len, const char *optlist)</code>

从基于磁盘的文件或虚拟文件加载 3D 模型（要求 PDF 1.6）。

filename （名称字符串）包含 3D 模型（U3D 格式）的基于磁盘的文件或虚拟文件的名称。

len （仅用于 C 语言绑定） UTF-16 字符串的 **filename** 长度（以字节为单位）。如果 **len = 0**，则必须提供以 **null** 终结的字符串。

optlist 选项列表，根据表 11.1 指定 3D 模型的属性。可以使用下列选项：**defaultview**、**errorpolicy**、**hypertext-encoding**、**script**、**views**

返回值 可使用 **PDF_create_annotation()** 创建 3D 注释的 3D 句柄。可以在关闭的文档范围结束之前使用 3D 句柄。必须检查返回值是否为 -1（在 PHP 中为：0）；这表示出现错误。

详细说明 将加载包含 3D 数据的文件。不会对 3D 数据进行错误检查。

范围 页面、文档。在下一个 **PDF_end_document()** 调用之前，可以一直使用返回的句柄。

表 11.1 **PDF_load_3ddata()** 的选项

选项	说明
defaultview	（关键字或 3D 视图句柄）指定 3D 注释的初始视图；关键字 first 或 last 之一（指向 views 选项中的相关条目），或使用 PDF_create_3dview() 创建的 3D 视图句柄。默认值： first
errorpolicy	（关键字）控制出现错误时的行为（参见第 28 页上的第 2.6 节“异常处理”）
hypertext-encoding	（关键字）为提供的 script 指定编码。空字符串相当于 unicode 。默认值：全局 hypertextencoding 参数的值
script	（超文本字符串）包含实例化 3D 模型时执行的 JavaScript 代码的字符串。默认值：无脚本
views	（3D 视图句柄）列出 3D 模型的预定义视图。每个列表元素都是使用 PDF_create_3dview() 创建的 3D 视图句柄。默认值：空列表

C++ Java	<code>int create_3dview(String username, String optlist)</code>
Perl PHP	<code>int PDF_create_3dview(string username, string optlist)</code>
C	<code>int PDF_create_3dview(PDF *p, const char *username, int len, const char *optlist)</code>

创建 3D 视图（要求 PDF 1.6）。

username （超文本字符串） 3D 视图的用户界面名称。

len （仅用于 C 语言绑定） UTF-16 字符串的 **username** 长度（以字节为单位）。如果 **len = 0**，则必须提供以 **null** 终结的字符串。

optlist 选项列表，根据表 11.2 指定 3D 视图属性。可以使用下列选项：**background**、**camerazworld**、**camera-distance**、**errorpolicy**、**hypertext-encoding**、**name**

返回值 一个 3D 视图句柄；使用 `PDF_load_3ddata()` 中的 `views` 选项可附加到 3D 模型、使用 `PDF_create_annotation()` 创建 3D 注释、或使用 `PDF_create_action()` 创建 3D 相关的动作。可以在关闭的文档范围结束之前使用 3D 视图句柄。必须检查返回值是否为 -1（在 PHP 中为：0）；这表示出现错误。

详细说明 将创建一个指定的 3D 视图，该视图可用于将 3D 数据加载到动作

范围 页面、文档。在下一个 `PDF_end_document()` 调用之前，可以一直使用返回的句柄。

表 11.2 `PDF_create_3dview()` 的选项

选项	说明
background	（选项列表）指定 3D 模型的背景： fillcolor （色彩）背景颜色，用 RGB 色彩空间表示。默认值：white entire （布尔值）如果为 true，背景适用于整个注释；否则，它仅适用于在注释的 3Dbox 选项中指定的矩形。默认值：false
camera2world	（12 个浮点值列表）在全局坐标中指定照像机的位置和方向的 3D 转换矩阵。默认值：在 3D 数据内部定义
camera-distance	（非负数浮点值；在未指定 camera2world 时忽略）照像机和活动范围中心之间的距离。默认值：在 3D 数据内部定义
errorpolicy	（关键字）控制出现错误时的行为（参见第 28 页上的第 2.6 节“异常处理”）
hypertext-encoding	（关键字）指定提供的 name 和 username 的编码。空字符串相当于 unicode。默认值：全局 hypertextencoding 参数的值
name	（超文本字符串）3D 视图的名称，用于动作。这是一个可选的内部名称，根据所需的 username 参数单独处理。

12 文档交换

12.1 文档信息域

C++ Java	<code>void set_info(String key, String value)</code>
Perl PHP	<code>PDF_set_info(resource p, string key, string value)</code>
C	<code>void PDF_set_info(PDF *p, const char *key, const char *value)</code>
C	<code>void PDF_set_info2(PDF *p, const char *key, const char *value, int len)</code>

用 *value* 填充信息域 *key*。

key （名称字符串）文档信息域的名称，可使用任何标准名称或任意的自定义名称（参见表 12.1）。自定义域的数量没有任何限制。对于自定义文档信息域的使用和语义，建议 PDFlib 用户参阅 Dublin Core Metadata 元素集。¹

value （超文本字符串）*key* 参数将被设置的字符串。Acrobat 会将 *value* 的最大长度强制设为 255 字节。请注意，由于 Windows 版 Adobe Reader 6 中的错误，& 字符在某些信息字符串无法正确显示。

len （仅用于 *PDF_set_info2()*，并且仅用于 C 语言绑定）UTF-16 字符串的 *value* 长度（以字节为单位）。如果 *len = 0*，则必须提供以 null 终结的字符串。

详细说明 提供的信息值将只用于当前文档，不能用于在同一对象范围内生成的所有文档。如果已为 *PDF_begin/end_document()* 提供了 *autoxmp* 选项，PDFlib 将根据提供给 *PDF_set_info()* 的信息项自动创建同步的 XMP 文档元数据。

范围 对象、文档、页面。如果在对象范围内使用，提供的值将只用于下一文档。

表 12.1 文档信息域关键字的值

关键字	说明
<i>Subject</i>	文档的主题
<i>Title</i>	文档的标题
<i>Creator</i>	创建文档使用的软件（而非 PDF 输出的制作程序，它始终为 PDFlib）。Acrobat 将此条目显示为 »Application «。
<i>Author</i>	文档的作者
<i>Keywords</i>	描述文档内容的关键字
<i>Trapped</i>	指示是否已对文档应用陷印。允许值包括 True、False 和 Unknown。在 PDF/A 模式下，不允许使用 Unknown。
<i>any name other than CreationDate, Producer, and ModDate</i>	用户定义域。PDFlib 可支持任意数量的域。但是，应当只提供一次自定义域名名称。如果多次为同一域提供名称，将使用最后一次提供的名称。另请参见 <i>PDF_begin/end_document()</i> 的 <i>moddate</i> 选项。

1. 参见 dublincore.org

12.2 XMP 元数据

除文档信息域之外，PDFlib 还支持 XMP (*Extensible Metadata Platform*¹) 作为指定元数据的框架。例如，PDF/A 兼容要求 XMP，并且越来越多的应用程序支持 XMP。PDFlib 支持几种风格的 XMP，详情见下文。

文档信息关键字的 XMP 自动同步 如果 `PDF_begin/end_document()` 中的 `autoxmp` 选项为 `true`，PDFlib 将使提供给 `PDF_set_info()` 的文档信息域以及几个内部生成项（如 `CreationDate`）与文档级 XMP 元数据中的相应项同步。

与表 12.3 中任一标准 XMP 方案中已知元素对应的文档信息关键字将放置在相应的方案中。未知信息关键字将放置在扩展的 PDF (*pdfx*) 方案中。

自定义元数据流 用户可以将完整或部分 XMP 元数据流提供给各种函数的 `metadata` 选项。此选项要求提供 XMP 流，并且将对其进行验证。PDFlib 将自动生成 XDP 包头和包尾。

对于文档级元数据，PDFlib 将根据需要添加几个元素（如 `CreationDate`）。在 PDF/A 模式下，PDFlib 将用户提供的 XMP 流中的相关项与标准文档信息域同步（与将文档信息域与 XMP 同步的 `autoxmp` 模式类似）。但是，PDFlib 不会将其他 XMP 项与自定义文档信息域同步。

除文档级元数据以外，XMP 还可以提供给页面、字体、ICC 色彩特征描述文件、图像、模板和导入的 PDF 页面。表 12.2 列出了 XMP 元数据的选项。

表 12.2 `PDF_begin/end_document()`、`PDF_begin/end_page_ext()`、`PDF_load_font()`、`PDF_load_iccprofile()`、`PDF_load_image()`、`PDF_begin_template_ext()`、`PDF_open_pdi_page()` 中的 XMP 元数据选项

选项	描述
metadata	（选项列表；PDF 1.4）提供文档或其他对象的元数据。此选项列表可以包含下列选项：
inputencoding	（关键字）用来解释提供数据的编码。默认值：unicode
inputformat	（关键字）提供数据的格式。默认值：utf8（在基于 EBCDIC 的系统上为 <i>ebcdicutf8</i> ），但是，如果 <i>inputencoding</i> 为 8 位编码方式，则默认值为 <i>byte</i> 。
filename	（名称字符串；必选）包含格式正确的 XMP 元数据的基于磁盘文件或虚拟文件的文件名。 示例：metadata={filename=info.xmp inputencoding=winansi}

1. 请参见 www.adobe.com/products/xmp

标准方案和命名空间 URI PDFlib 内部已知表 12.3 中列出的用于 PDF/A 的预定义 XMP 方案，并且将使用该表中列出的命名空间 URI 和前缀（有关这些方案的详细信息，请参见 XMP 参考）。用户提供的 XMP 中的自定义方案禁止使用表 12.3 中的任何命名空间 URI。同样，两个自定义方案禁止使用同一命名空间 URI。如果要将其方案用于 PDF/A，该方案本身必须包含用于 PDF/A 扩展方案的描述方案。

表 12.3 PDFlib 内部已知的 XMP 方案

方案名称和说明	命名空间 URI	命名空间前缀
Dublin core（公共文档属性）	http://purl.org/dc/elements/1.1/	dc
XMP 基本方案	http://ns.adobe.com/xap/1.0/	xmp
XMP 权限管理方案	http://ns.adobe.com/xap/1.0/rights/	xmpRights
XMP 媒体管理方案	http://ns.adobe.com/xap/1.0/mm/	xmpMM
XMP 基本工作票证方案	http://ns.adobe.com/xap/1.0/bj	xmpBJ
XMP 分页文本方案	http://ns.adobe.com/xap/1.0/t/pg/	xmpTPg
Adobe PDF 方案	http://ns.adobe.com/pdf/1.3/	pdf
Photoshop 方案	http://ns.adobe.com/photoshop/1.0/	photoshop
TIFF 属性的 EXIF 方案	http://ns.adobe.com/tiff/1.0/	tiff
EXIF 特定的属性的 EXIF 方案	http://ns.adobe.com/exif/1.0/	exif
PDF/A 标识方案	http://www.aiim.org/pdfa/ns/id ¹	pdfaid
PDF/A 扩展方案说明方案	http://www.aiim.org/pdfa/ns/schema# ¹	pdfaSchema

1. 在第一次出版的 PDF/A 标准 (ISO 19005-1) 中，此命名空间 URI 的拼写有误。在即将于 2007 年出版的技术勘误表中将对此进行更正。

自定义 XMP 的示例 下面的示例显示自定义方案的 XMP，它可以提供给 metadata 选项：

```
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
  xmlns:ix='http://ns.adobe.com/ix/1.0/'>
  <rdf:Description rdf:about='' xmlns:xap='http://ns.adobe.com/xap/1.0/'>
    <xap:CreatorTool>ACME Reporting Tool 1.2.3</xap:CreatorTool>
    <xap:Identifier>
      <rdf:Bag>
        <rdf:li>12345</rdf:li>
      </rdf:Bag>
    </xap:Identifier>
  </rdf:Description>

  <rdf:Description rdf:about='' xmlns:acmedoc='http://ns.acme.com/acmedoc/1.0/'>
    <acmedoc:CLIENT>299</acmedoc:CLIENT>
    <acmedoc:DESTINATION>XYZ</acmedoc:DESTINATION>
    <acmedoc:OBJ>ACCOUNT</acmedoc:OBJ>
    <acmedoc:ARCHIVE>KDGTCVJEU</acmedoc:ARCHIVE>
    <acmedoc:ID>0070900328004400049208</acmedoc:ID>
  </rdf:Description>
</rdf:RDF>
```

12.3 标签 PDF

若要生成标签 PDF，必须将 `PDF_begin_document()` 中的 `tagged` 选项设置为 `true`。同时，还必须提供 `lang` 选项。

如果 `PDF_begin_document()` 中的 `pdfa` 选项已设置为 `PDF/A-1a:2005`，将自动激活标签 PDF 模式。

C++ Java

int begin_item(String tag, String optlist)

Perl PHP

int PDF_begin_item(resource p, string tag, string optlist)

C

int PDF_begin_item(PDF *p, const char *tag, const char *optlist)

打开一个具有如选项所提供的属性的结构元素或其他内容项目。

tag 有关项目的元素类型，请参见表 12.4。其类型必须为当前 PDF 兼容级别所允许的一种标准结构类型或者是伪标签。

表 12.4 标准项目标签

种类	标签
分组	Document、Part、Art、Sect、Div、BlockQuote、Caption、TOC、TOCI、Index、NonStruct、Private
段落类型	P、H、H1-H6 (BLSEs)
列表	L、LI、Lb1、LBody (BLSEs)
表	Table (BLSE)、TR、TH、TD、THead ¹ 、TBody ¹ 、TFoot ¹
内嵌级别	Span、TagSuspect ² 、Quote、Note、Reference、BibEntry、Code、(ILSEs)
插图	Figure、Formula、Form
日语	Ruby ¹ (grouping)、RB ¹ 、RT ¹ 、RP ¹ 、Warichu ¹ (grouping)、WT ¹ 、WP ¹
伪标签	下列标签可创建不属于结构元素的项目： Artifact 指定要与实际的页面内容区分开来的伪像。 ASpan (可访问性范围；将作为 Span 写入 PDF，但必须区别于内嵌级项目 Span) 可用于向不属于结构元素或只与结构元素部分相似的内容附加可访问性属性。 ReversedChars 指定从右到左书写的语言的文本使用相反的字符顺序。这样便可在 Acrobat 中搜索希伯来语或阿拉伯语文本。 Clip 指定标记的剪裁顺序。该顺序只包含剪切路径或以渲染模式 7 显示的文本，而不包含显示的图形或 PDF_save() / PDF_restore()。

1. 要求 PDF 1.5 或更高版本
2. 要求 PDF 1.6 或更高版本

optlist 选项列表，根据表 12.5 指定项目的详细信息。子级元素将继承所有可继承的设置，因此无需重复设置。因为此后无法进行更改，所以必须在此处设置项目的所有属性。可以使用下列选项：

`ActualText`、`Alt`、`artifacttype`、`Attached`、`BBox`、`ColSpan`、`E`、`hypertext-encoding`、`index`、`inline`、`Lang`、`parent`、`RowSpan`、范围、`Title`

- 返回值** 项目句柄，可用于后续的项目相关调用。
- 详细说明** 此函数可生成文档的结构树，这是标签 PDF 的根本。新元素在结构树中的位置可用 *parent* 和 *index* 选项控制。结构元素可以嵌套到任意级别。常规项目不仅限于已在其上打开的页面，可在任意数量的页面上继续。
- 范围** 对于内嵌项目，范围仅限于页面；常规项目还包括文档范围；必须始终与对应的 *PDF_end_item()* 调用成对使用。只允许在标签 PDF 模式下使用此函数。

表 12.5 *PDF_begin_item()* 的结构和伪标签属性选项

选项	说明
ActualText	（超文本字符串；除具有 ASpan 的 PDF 1.5 外，不能用于伪标签；不能用于 TagSuspect；对于不兼容 Unicode 的字体中的文本，必需使用此选项）与内容项目等效的替换文本。对于以某些非标准的方式显示的文本内容（例如连字、插图中的花饰字符和首字下沉等）应提供此选项。如果在 PDF 1.4 模式下使用此选项，inline 选项必须设为 false。
Alt	（超文本字符串；除具有 ASpan 的 PDF 1.5 外，不能用于伪标签；不能用于 TagSuspect 内容项目的替代描述。对于不能识别为文本的插图和图像等应提供此选项。为了实现可访问性，必须为图像提供替代文本。如果在 PDF 1.4 模式下使用此选项，inline 选项必须设为 false。
artifacttype	（关键字；仅用于 tag=Artifact 的情况）标识伪类型的内容项目：分页、版面或页面
Attached	（关键字列表；仅用于 tag=Artifact 和 artifacttype=Pagination 的情况）一个列表，包含下列 1 到 4 个关键字：Top、Bottom、Left 和 Right
BBox	（矩形；仅用于 tag=Artifact 以及所有表和插图标签；可选，但建议重排时使用）如果 usercoordinates= false，则为默认坐标系中的伪像定界框；如果 usercoordinates=true，则为用户坐标系中的伪像定界框。如果未提供此选项，PDFlib 将自动为导入的图像和 PDF 页面创建 BBox 条目。
ColSpan	（整数值；仅用于 tag=TH 和 TD 的情况）一个单元格所跨的表列数。
E	（超文本字符串；不能用于除 ASpan 以外的伪标签；不能用于 TagSuspect；结构标签要求 PDF 1.5）内容项目的全名缩写扩展。应提供此选项，以说明缩写。Acrobat 的 Read Aloud 功能将扩展文本视为单独的词；即使缺少显式断字符，也是如此。
hypertext-encoding	（关键字）指定提供的文本的编码。空字符串等效于 unicode。默认值：对于支持 Unicode 的语言绑定，为空字符串；否则为 auto。
index	（整数值；不能用于伪标签和 TagSuspect）在父级元素内插入元素的位置索引。可以使用介于 0 到当前子级元素数量之间的值，在父级元素内的指定位置上插入项目。使用值 -1 可插入作为最后一个项目的元素。默认值：-1
inline	（布尔值；仅用于 tag=ASpan 以及除 TagSuspect 以外的所有内嵌级标签）如果为 true，将以内嵌方式写入内容项目，而不创建结构元素。默认值：true
Lang	（字符串；不能用于 TagSuspect 和除 ASpan 以外的伪标签）lang 选项的内容项目（采用表 2.3 中描述的格式）的语言标识符。此选项可用来覆盖具体内容项目的文档主要语言。
parent	（项目句柄；不能用于 TagSuspect 和伪标签）另一 PDF_begin_item() 调用返回的元素父级的项目句柄。值 0 指的是结构树的根部。-1 指的是当前页面上最早打开的元素的父级。换言之，parent=-1 可打开当前元素的同级元素。默认值：-1
RowSpan	（整数值；仅用于 tag=TH 和 TD 的情况）一个单元格所跨的表行数。
范围	（关键字；仅用于 tag=TH 的情况；要求 PDF 1.5 或更高版本）关键字 Row、Column 或 Both 之一，指示页眉单元格是否适用于其所在行、所在列或所在行和列中的其他单元格。
Title	（超文本字符串；不能用于内嵌标签和伪标签）结构元素的名称

C++ Java void end_item(int id)

Perl PHP PDF_end_item(resource p, int id)

C void PDF_end_item(PDF *p, int id)

关闭结构元素或其他内容项目。

id 项目的句柄，必须由 *PDF_begin_item()* 返回。

详细说明 所有内嵌项目都必须在页面结束之前关闭。所有常规项目都必须在文档结束之前关闭。但是，强烈建议完成操作后尽快关闭所有常规项目，以减少内存消耗。只有先关闭所有子级项目，才能关闭当前项目。在项目关闭之后，其父级项目将成为现用项目。

范围 对于内嵌项目，范围仅限于**页面**；常规项目还包括**文档**范围；必须始终与对应的 *PDF_begin_item()* 调用成对使用。只允许在标签 PDF 模式下使用此函数。

C++ Java void activate_item(int id)

Perl PHP PDF_activate_item(resource p, int id)

C void PDF_activate_item(PDF *p, int id)

激活以前创建的结构元素或其他内容项目。

id 项目的句柄，必须由 *PDF_begin_item()* 返回，并且尚未关闭。无法激活伪项目和内嵌级项目。

详细说明 存储一个结构元素并在以后激活可实现更大的灵活性，即使页面上由多个平行的结构分支（例如：中断主文本的多列版面或文本插入等），也可以高效地创建标签 PDF。

范围 文档、页面；只允许在标签 PDF 模式下使用此选项。

A 所有函数的列表

单击函数名称可跳转到相应的描述。

常规函数	文本输出函数	图形函数	<i>PDF</i> 和 <i>pCOS</i> 函数
<i>PDF_get_value</i>	<i>PDF_set_text_pos</i>	<i>PDF_setdash</i>	<i>PDF_open_pdi_document</i>
<i>PDF_set_value</i>	<i>PDF_show</i>	<i>PDF_setdashpattern</i>	<i>PDF_open_pdi_callback</i>
<i>PDF_get_parameter</i>	<i>PDF_xshow</i>	<i>PDF_setflat</i>	<i>PDF_close_pdi_document</i>
<i>PDF_set_parameter</i>	<i>PDF_show_xy</i>	<i>PDF_setlinejoin</i>	<i>PDF_open_pdi_page</i>
<i>PDF_new</i>	<i>PDF_continue_text</i>	<i>PDF_setlinecap</i>	<i>PDF_close_pdi_page</i>
<i>PDF_new2</i>	<i>PDF_stringwidth</i>	<i>PDF_setmiterlimit</i>	<i>PDF_fit_pdi_page</i>
<i>PDF_delete</i>		<i>PDF_setlinewidth</i>	<i>PDF_process_pdi</i>
<i>PDF_begin_document</i>	文本格式化函数	<i>PDF_initgraphics</i>	<i>PDF_pcos_get_number</i>
<i>PDF_begin_document_callback</i>	<i>PDF_fit_textline</i>	<i>PDF_save</i>	<i>PDF_pcos_get_string</i>
<i>PDF_end_document</i>	<i>PDF_info_textline</i>	<i>PDF_restore</i>	<i>PDF_pcos_get_stream</i>
<i>PDF_get_buffer</i>	<i>PDF_add_textflow</i>	<i>PDF_translate</i>	
<i>PDF_begin_page_ext</i>	<i>PDF_create_textflow</i>	<i>PDF_scale</i>	块填充函数 (<i>PPS</i>)
<i>PDF_end_page_ext</i>	<i>PDF_fit_textflow</i>	<i>PDF_rotate</i>	<i>PDF_fill_textblock</i>
<i>PDF_suspend_page</i>	<i>PDF_info_textflow</i>	<i>PDF_skew</i>	<i>PDF_fill_imageblock</i>
<i>PDF_resume_page</i>	<i>PDF_delete_textflow</i>	<i>PDF_concat</i>	<i>PDF_fill_pdfblock</i>
<i>PDF_create_pvf</i>		<i>PDF_setmatrix</i>	
<i>PDF_delete_pvf</i>	表格式化函数	<i>PDF_create_gstate</i>	交互功能
<i>PDF_get_errnum</i>	<i>PDF_add_table_cell</i>	<i>PDF_set_gstate</i>	<i>PDF_create_action</i>
<i>PDF_get_errmsg</i>	<i>PDF_fit_table</i>	<i>PDF_moveto</i>	<i>PDF_add_nameddest</i>
<i>PDF_get_apiname</i>	<i>PDF_info_table</i>	<i>PDF_lineto</i>	<i>PDF_create_annotation</i>
<i>PDF_get_opaque</i>	<i>PDF_delete_table</i>	<i>PDF_curveto</i>	<i>PDF_create_field</i>
	<i>PDF_info_matchbox</i>	<i>PDF_circle</i>	<i>PDF_create_fieldgroup</i>
字体函数		<i>PDF_arc</i>	<i>PDF_create_bookmark</i>
<i>PDF_load_font</i>	颜色函数	<i>PDF_arcn</i>	
<i>PDF_setfont</i>	<i>PDF_setcolor</i>	<i>PDF_rect</i>	多媒体
<i>PDF_info_font</i>	<i>PDF_makespotcolor</i>	<i>PDF_closepath</i>	<i>PDF_load_3ddata</i>
<i>PDF_begin_font</i>	<i>PDF_load_iccprofile</i>	<i>PDF_stroke</i>	<i>PDF_create_3dview</i>
<i>PDF_end_font</i>	<i>PDF_begin_pattern</i>	<i>PDF_closepath_stroke</i>	
<i>PDF_begin_glyph</i>	<i>PDF_end_pattern</i>	<i>PDF_fill</i>	文档交换
<i>PDF_end_glyph</i>	<i>PDF_shading_pattern</i>	<i>PDF_fill_stroke</i>	<i>PDF_set_info</i>
<i>PDF_encoding_set_char</i>	<i>PDF_shfill</i>	<i>PDF_closepath_fill_stroke</i>	<i>PDF_begin_item</i>
<i>PDF_utf16_to_utf8</i>	<i>PDF_shading</i>	<i>PDF_clip</i>	<i>PDF_end_item</i>
<i>PDF_utf8_to_utf16</i>		<i>PDF_endpath</i>	<i>PDF_activate_item</i>
<i>PDF_utf32_to_utf16</i>	图像函数	<i>PDF_define_layer</i>	
	<i>PDF_load_image</i>	<i>PDF_set_layer_dependency</i>	
	<i>PDF_close_image</i>	<i>PDF_begin_layer</i>	
	<i>PDF_fit_image</i>	<i>PDF_end_layer</i>	
	<i>PDF_begin_template_ext</i>		
	<i>PDF_end_template</i>		
	<i>PDF_add_thumbnail</i>		

B 所有参数的列表

本节列出了用于 `PDF_get/set_parameter()` 和 `PDF_get/set_value()` 的所有关键字。单击关键字可跳转到相应的描述。

种类	<i>PDF_get/set_parameter()</i>	<i>PDF_get/set_value()</i>
设置	<i>license</i> ¹ 、 <i>licensefile</i> 、 <i>nodemo-stamp</i> 、 <i>resourcefile</i> 、 <i>scope</i> ¹ 、 <i>SearchPath</i> 、 <i>string</i> ¹ 、 <i>asciifile</i> 、 <i>errorpolicy</i>	<i>compress</i>
日志	<i>logging</i> ¹ 、 <i>logmsg</i> ¹	
版本	<i>version</i> ¹	<i>major</i> 、 <i>minor</i> 、 <i>revision</i> ²
页面	<i>topdown</i>	<i>pagewidth</i> 、 <i>pageheight</i>
字体处理	<i>Encoding</i> 、 <i>FontAFM</i> 、 <i>FontPFM</i> 、 <i>FontOutline</i> 、 <i>HostFont</i>	<i>font</i> ² 、 <i>fontsize</i> ²
简单文本输出	<i>autospace</i> 、 <i>charref</i> 、 <i>escape-sequence</i> 、 <i>fakebold</i> 、 <i>glyphcheck</i> 、 <i> Kerning</i> 、 <i>textformat</i> 、 <i>underline</i> 、 <i>overline</i> 、 <i>strikeout</i>	<i>charspacing</i> 、 <i>horizscaling</i> 、 <i>italicangle</i> 、 <i>leading</i> 、 <i>textrendering</i> 、 <i>textrise</i> 、 <i>textx</i> ² 、 <i>texty</i> ² 、 <i>underline-position</i> 、 <i>underline-width</i> 、 <i>wordspacing</i>
图形	<i>fillrule</i>	<i>currentx</i> ² 、 <i>currenty</i> ² 、 <i>ctm_a</i> ² 、 <i>ctm_b</i> ² 、 <i>ctm_c</i> ² 、 <i>ctm_d</i> ² 、 <i>ctm_e</i> ² 、 <i>ctm_f</i> ²
色彩	<i>preserveold-pantonenames</i> 、 <i>spotcolorlookup</i>	
ICC 配置文件	<i>ICCPProfile</i> 、 <i>StandardOutputIntent</i>	<i>icccomponents</i> ² 、 <i>setcolor:icc-profilegray</i> 、 <i>setcolor:icc-profilergb</i> 、 <i>setcolor:icc-profilecmyk</i>
图形	<i>honoriccprofile</i> 、 <i>renderingintent</i>	<i>imagewidth</i> ² 、 <i>imageheight</i> ² 、 <i>image:iccprofile</i> ² 、 <i>orientation</i> ² 、 <i>resx</i> ² 、 <i>resy</i> ²
PDI	<i>pdi</i> ¹	
交互	<i>hypertextencoding</i> 、 <i>hypertextformat</i> 、 <i>usehypertextencoding</i> 、 <i>usercoordinates</i>	

1. 仅用于 `PDF_get_parameter()`
2. 仅用于 `PDF_get_value()`

C 所有选项列表

本索引按字母顺序列出了所有选项及其所应用的函数。单击选项名称可跳转到相应说明。

A

action

用于 `process_pdi()` 116

ActualText

用于 `begin_item()` 147

`add/create_textflow()` 中 `leader` 的 **alignment**
子选项 54

`add/create_textflow()` 中 `leader` 的 **encoding**
子选项 54

`add/create_textflow()` 中 `leader` 的 **fillcolor**
子选项 54

`add/create_textflow()` 中 `leader` 的 **font**
子选项 54

`add/create_textflow()` 中 `leader` 的 **fontname**
子选项 54

`add/create_textflow()` 中 `leader` 的 **fontsize**
子选项 54

`add/create_textflow()` 中 `leader` 的 **text**
子选项 54

`add/create_textflow()` 中 `leader` 的 **yposition**
子选项 54

`add/create_textflow()` 中的 **adjustmethod** 53

`add/create_textflow()` 中的 **alignment** 53

`add/create_textflow()` 中的 **avoidbreak** 53

`add/create_textflow()` 中的 **avoidemptybegin** 53

`add/create_textflow()` 中的 **charclass** 54

`add/create_textflow()` 中的 **charmapping** 54

`add/create_textflow()` 中的 **comment** 54

`add/create_textflow()` 中的 **encoding** 54

`add/create_textflow()` 中的 **errorpolicy** 54

`add/create_textflow()` 中的 **fixedleading** 54

`add/create_textflow()` 中的 **fontname** 54

`add/create_textflow()` 中的 **hortabmethod** 54

`add/create_textflow()` 中的 **hortabsize** 54

`add/create_textflow()` 中的 **hyphenchar** 54

`add/create_textflow()` 中的 **lastalignment** 54

`add/create_textflow()` 中的 **leader** 54

`add/create_textflow()` 中的 **leading** 55

`add/create_textflow()` 中的 **leftindent** 55

`add/create_textflow()` 中的 **mark** 55

`add/create_textflow()` 中的 **maxspacing** 55

`add/create_textflow()` 中的 **minlinecount** 55

`add/create_textflow()` 中的 **minspacing** 55

`add/create_textflow()` 中的 **nextline** 55

`add/create_textflow()` 中的 **nextparagraph** 55

`add/create_textflow()` 中的 **nofitlimit** 55

`add/create_textflow()` 中的 **parindent** 55

`add/create_textflow()` 中的 **resetfont** 55

`add/create_textflow()` 中的 **return** 55

`add/create_textflow()` 中的 **rightindent** 55

`add/create_textflow()` 中的 **ruler** 55

`add/create_textflow()` 中的 **shrinklimit** 55

`add/create_textflow()` 中的 **space** 55

`add/create_textflow()` 中的 **spreadlimit** 55

`add/create_textflow()` 中的 **tabalignchar** 55

`add/create_textflow()` 中的 **tabalignment** 55

`add_nameddest()` 中的 **bottom** 选项和
`create_action()`、`create_annotation()`、
`create_bookmark()` 和 `begin/end_document()` 中
`destination` 选项的子选项 127

`add_table_cell()` 中的 **checkwordsplitting** 65

`add_table_cell()` 中的 **colscalegroup** 65

`add_table_cell()` 中的 **colspan** 65

`add_table_cell()` 中的 **colwidth** 55, 65

`add_table_cell()` 中的 **errorpolicy** 65

`add_table_cell()` 中的 **fitimage** 65

`add_table_cell()` 中的 **fitdipage** 65

`add_table_cell()` 中的 **fittextflow** 65

`add_table_cell()` 中的 **fittextline** 65

`add_table_cell()` 中的 **image** 65

`add_table_cell()` 中的 **margin** 65

`add_table_cell()` 中的 **marginbottom** 65

`add_table_cell()` 中的 **marginleft** 65

`add_table_cell()` 中的 **marginright** 65

`add_table_cell()` 中的 **marginintop** 65

`add_table_cell()` 中的 **minrowheight** 65

`add_table_cell()` 中的 **pdipage** 65

`add_table_cell()` 中的 **return** 65

`add_table_cell()` 中的 **rowheight** 65

`add_table_cell()` 中的 **rowjoiningroup** 66

`add_table_cell()` 中的 **rowscalegroup** 66

`add_table_cell()` 中的 **rowspan** 66

`add_table_cell()` 中的 **textflow** 66

Alt

用于 `begin_item()` 147

artifacttype

用于 `begin_item()` 147

`ascender in info_font()` 中的 **fontsize** 子选项 35

Attached

用于 `begin_item()` 147

`attachmentpassword in begin_document()` 17

B

BBox

用于 `begin_item()` 147

`begin/end_document()` 中 `viewerpreferences` 选
项的 **centerwindow** 子选项 20

`begin/end_document()` 中 `viewerpreferences` 选
项的 **direction** 子选项 20

begin/end_document() 中 *viewerpreferences* 选项的 *displaydoctitle* 子选项 20
begin/end_document() 中 *viewerpreferences* 选项的 *fitwindow* 子选项 20
begin/end_document() 中 *viewerpreferences* 选项的 *hidemenubar* 子选项 20
begin/end_document() 中 *viewerpreferences* 选项的 *hidetoolbar* 子选项 20
begin/end_document() 中 *viewerpreferences* 选项的 *hidewindowui* 子选项 20
begin/end_document() 中 *viewerpreferences* 选项的 *nonfullscreenpagemode* 子选项 20
begin/end_document() 中 *viewerpreferences* 选项的 *printarea* 子选项 20
begin/end_document() 中 *viewerpreferences* 选项的 *printclip* 子选项 20
begin/end_document() 中 *viewerpreferences* 选项的 *printscaling* 子选项 20
begin/end_document() 中 *viewerpreferences* 选项的 *viewarea* 子选项 20
begin/end_document() 中 *viewerpreferences* 选项的 *viewclip* 子选项 20
begin/end_document() 中的 *attachments* 17
begin/end_document() 中的 *autoxmp* 17
begin/end_document() 中的 *destination* 17
begin/end_document() 中的 *hypertextencoding* 17
begin/end_document() 中的 *labels* 18
begin/end_document() 中的 *metadata* 18
begin/end_document() 中的 *moddate* 18
begin/end_document() 中的 *openmode* 18
begin/end_document() 中的 *pagelayout* 18
begin/end_document() 中的 *search* 19
begin/end_document() 中的 *uri* 19
begin/end_page_ext() 中 *separationinfo* 的 *pages* 子选项 24
begin/end_page_ext() 中 *separationinfo* 的 *spotcolor* 子选项 24
begin/end_page_ext() 中 *separationinfo* 的 *spotname* 子选项 24
begin/end_page_ext() 中的 *action* 23
begin/end_page_ext() 中的 *artbox* 23
begin/end_page_ext() 中的 *bleedbox* 23
begin/end_page_ext() 中的 *cropbox* 23
begin/end_page_ext() 中的 *defaultcmymk* 23
begin/end_page_ext() 中的 *defaultgray* 23
begin/end_page_ext() 中的 *defaultrgb* 23
begin/end_page_ext() 中的 *duration* 23
begin/end_page_ext() 中的 *height* 23
begin/end_page_ext() 中的 *label* 23
begin/end_page_ext() 中的 *mediabox* 23
begin/end_page_ext() 中的 *metadata* 23
begin/end_page_ext() 中的 *rotate* 23
begin/end_page_ext() 中的 *taborder* 24
begin/end_page_ext() 中的 *transition* 24
begin/end_page_ext() 中的 *trimbox* 24
begin/end_page_ext() 中的 *userunit* 24
begin/end_page_ext() 中的 *width* 24

begin_document() 和 *end_document()* 中的 *viewerpreferences* 19
begin_document() 中的 *compatibility* 17
begin_document() 中的 *errorpolicy* 17
begin_document() 中的 *flush* 17
begin_document() 中的 *groups* 17
begin_document() 中的 *inmemory* 18
begin_document() 中的 *lang* 18
begin_document() 中的 *linearize* 18
begin_document() 中的 *masterpassword* 18
begin_document() 中的 *optimize* 18
begin_document() 中的 *pdfa* 18
begin_document() 中的 *pdfx* 18
begin_document() 中的 *permissions* 19
begin_document() 中的 *recordsize* 19
begin_document() 中的 *tagged* 19
begin_document() 中的 *tempdirname* 19
begin_document() 中的 *tempfilenames* 19
begin_document() 中的 *userpassword* 19
begin_font() 中的 *colorized* 37
begin_font() 中的 *familyname* 37
begin_font() 中的 *stretch* 37
begin_font() 中的 *weight* 37
begin_font() 中的 *widthsonly* 37
begin_page_ext() 中的 *group* 23
begin_page_ext() 中的 *pagenumber* 23
begin_page_ext() 中的 *separationinfo* 24
begin_page_ext() 中的 *topdown* 24

C

ColSpan 用于

begin_item() 147
create_3dview() 中的 *background* 142
create_3dview() 中的 *cameradistance* 142
create_3dview() 中的 *camerazworld* 142
create_3dview() 中的 *errorpolicy* 142
create_3dview() 中的 *hypertextencoding* 142
create_3dview() 中的 *name* 142
create_action() 中的 *3Dview* 124
create_action() 中的 *canonicaldate* 124
create_action() 中的 *defaultdir* 124
create_action() 中的 *destination* 124
create_action() 中的 *destname* 124
create_action() 中的 *duration* 124
create_action() 中的 *errorpolicy* 124
create_action() 中的 *exclude* 124
create_action() 中的 *exportmethod* 125
create_action() 中的 *filename* 125
create_action() 中的 *hide* 125
create_action() 中的 *hypertextencoding* 125
create_action() 中的 *ismap* 125
create_action() 中的 *layerstate* 125
create_action() 中的 *menuname* 125
create_action() 中的 *namelist* 125
create_action() 中的 *newwindow* 125
create_action() 中的 *operation* 125
create_action() 中的 *parameters* 126
create_action() 中的 *preserveradio* 126

`create_action()` 中的 **script** 126
`create_action()` 中的 **scriptname** 126
`create_action()` 中的 **submitemptyfields** 126
`create_action()` 中的 **target** 126
`create_action()` 中的 **transition** 126
`create_action()` 中的 **url** 126
`create_annotation()` 中 3Dactivate 的 **disable** 子选项 130
`create_annotation()` 中 3Dactivate 的 **disablestate** 子选项 130
`create_annotation()` 中 3Dactivate 的 **enable** 子选项 130
`create_annotation()` 中 3Dactivate 的 **enablestate** 子选项 130
`create_annotation()` 中 custom 的 **key** 子选项 131
`create_annotation()` 中 custom 的 **type** 子选项 131
`create_annotation()` 中 custom 的 **value** 子选项 131
`create_annotation()` 中的 **3Dactivate** 130
`create_annotation()` 中的 **3Ddata** 130
`create_annotation()` 中的 **3Dinitialview** 130
`create_annotation()` 中的 **3Dinteractive** 130
`create_annotation()` 中的 **3Dshared** 130
`create_annotation()` 中的 **action** 130
`create_annotation()` 中的 **alignment** 130
`create_annotation()` 中的 **annotcolor** 130
`create_annotation()` 中的 **borderstyle** 131
`create_annotation()` 中的 **cloudy** 131
`create_annotation()` 中的 **contents** 131
`create_annotation()` 中的 **createdate** 131
`create_annotation()` 中的 **custom** 131
`create_annotation()` 中的 **dasharray** 131
`create_annotation()` 中的 **destination** 131
`create_annotation()` 中的 **destname** 131
`create_annotation()` 中的 **display** 131
`create_annotation()` 中的 **endingstyles** 131
`create_annotation()` 中的 **filename** 131
`create_annotation()` 中的 **fillcolor** 131
`create_annotation()` 中的 **font** 131
`create_annotation()` 中的 **fontsize** 131
`create_annotation()` 中的 **highlight** 131
`create_annotation()` 中的 **hypertextencoding** 132
`create_annotation()` 中的 **iconname** 132
`create_annotation()` 中的 **interiorcolor** 132
`create_annotation()` 中的 **layer** 132
`create_annotation()` 中的 **line** 132
`create_annotation()` 中的 **linewidth** 132
`create_annotation()` 中的 **locked** 132
`create_annotation()` 中的 **mimetype** 132
`create_annotation()` 中的 **name** 132
`create_annotation()` 中的 **opacity** 132
`create_annotation()` 中的 **open** 132
`create_annotation()` 中的 **orientate** 132
`create_annotation()` 中的 **parentname** 132
`create_annotation()` 中的 **polylinelist** 132

`create_annotation()` 中的 **popup** 133
`create_annotation()` 中的 **readonly** 133
`create_annotation()` 中的 **rotate** 133
`create_annotation()` 中的 **subject** 133
`create_annotation()` 中的 **title** 133
`create_annotation()` 中的 **usematchbox** 133
`create_annotation()` 中的 **usercoordinates** 133
`create_annotation()` 中的 **zoom** 133
`create_bookmark()` 中的 **action** 139
`create_bookmark()` 中的 **destination** 139
`create_bookmark()` 中的 **destname** 139
`create_bookmark()` 中的 **fontstyle** 139
`create_bookmark()` 中的 **hypertextencoding** 139
`create_bookmark()` 中的 **hypertextformat** 139
`create_bookmark()` 中的 **index** 139
`create_bookmark()` 中的 **open** 139
`create_bookmark()` 中的 **parent** 139
`create_bookmark()` 中的 **textcolor** 139
`create_field()` 和 `create_fieldgroup()` 中的 **action** 135
`create_field()` 和 `create_fieldgroup()` 中的 **backgroundcolor** 135
`create_field()` 和 `create_fieldgroup()` 中的 **bordercolor** 135
`create_field()` 和 `create_fieldgroup()` 中的 **borderstyle** 135
`create_field()` 和 `create_fieldgroup()` 中的 **buttonlayout** 135
`create_field()` 和 `create_fieldgroup()` 中的 **buttonstyle** 135
`create_field()` 和 `create_fieldgroup()` 中的 **calcorder** 136
`create_field()` 和 `create_fieldgroup()` 中的 **caption** 136
`create_field()` 和 `create_fieldgroup()` 中的 **captiondown** 136
`create_field()` 和 `create_fieldgroup()` 中的 **captionrollover** 136
`create_field()` 和 `create_fieldgroup()` 中的 **charspacing** 136
`create_field()` 和 `create_fieldgroup()` 中的 **comb** 136
`create_field()` 和 `create_fieldgroup()` 中的 **commitonselect** 136
`create_field()` 和 `create_fieldgroup()` 中的 **currentvalue** 136
`create_field()` 和 `create_fieldgroup()` 中的 **dasharray** 136
`create_field()` 和 `create_fieldgroup()` 中的 **defaultvalue** 136
`create_field()` 和 `create_fieldgroup()` 中的 **display** 136
`create_field()` 和 `create_fieldgroup()` 中的 **editable** 136
`create_field()` 和 `create_fieldgroup()` 中的 **errorpolicy** 136
`create_field()` 和 `create_fieldgroup()` 中的 **exportable** 136

create_field() 和 *create_fieldgroup()* 中的 **fileselect** 136
create_field() 和 *create_fieldgroup()* 中的 **fillcolor** 136
create_field() 和 *create_fieldgroup()* 中的 **fitmethod** 137
create_field() 和 *create_fieldgroup()* 中的 **font** 137
create_field() 和 *create_fieldgroup()* 中的 **fontsize** 137
create_field() 和 *create_fieldgroup()* 中的 **highlight** 137
create_field() 和 *create_fieldgroup()* 中的 **hypertextencoding** 137
create_field() 和 *create_fieldgroup()* 中的 **hypertextformat** 137
create_field() 和 *create_fieldgroup()* 中的 **icon** 137
create_field() 和 *create_fieldgroup()* 中的 **icondown** 137
create_field() 和 *create_fieldgroup()* 中的 **iconrollover** 137
create_field() 和 *create_fieldgroup()* 中的 **itemname** 137
create_field() 和 *create_fieldgroup()* 中的 **itemnamelist** 137
create_field() 和 *create_fieldgroup()* 中的 **itemtextlist** 137
create_field() 和 *create_fieldgroup()* 中的 **layer** 137
create_field() 和 *create_fieldgroup()* 中的 **linewidth** 137
create_field() 和 *create_fieldgroup()* 中的 **locked** 137
create_field() 和 *create_fieldgroup()* 中的 **lockmode** 137
create_field() 和 *create_fieldgroup()* 中的 **maxchar** 137
create_field() 和 *create_fieldgroup()* 中的 **multiline** 137
create_field() 和 *create_fieldgroup()* 中的 **multiselect** 137
create_field() 和 *create_fieldgroup()* 中的 **orientate** 137
create_field() 和 *create_fieldgroup()* 中的 **password** 137
create_field() 和 *create_fieldgroup()* 中的 **position** 137
create_field() 和 *create_fieldgroup()* 中的 **readonly** 138
create_field() 和 *create_fieldgroup()* 中的 **required** 138
create_field() 和 *create_fieldgroup()* 中的 **richtext** 138
create_field() 和 *create_fieldgroup()* 中的 **scrollable** 138
create_field() 和 *create_fieldgroup()* 中的 **sorted** 138

create_field() 和 *create_fieldgroup()* 中的 **spellcheck** 138
create_field() 和 *create_fieldgroup()* 中的 **strokecolor** 138
create_field() 和 *create_fieldgroup()* 中的 **submitname** 138
create_field() 和 *create_fieldgroup()* 中的 **taborder** 138
create_field() 和 *create_fieldgroup()* 中的 **tooltip** 138
create_field() 和 *create_fieldgroup()* 中的 **topindex** 138
create_field() 和 *create_fieldgroup()* 中的 **usercoordinates** 138
create_fieldgroup() 中的 **fieldtype** 136
create_fieldgroup() 中的 **toggle** 138
create_fieldgroup() 中的 **unisonselect** 138
create_gstate() 中的 **alphaishape** 79
create_gstate() 中的 **blendmode** 79
create_gstate() 中的 **flatness** 79
create_gstate() 中的 **linecap** 79
create_gstate() 中的 **linejoin** 79
create_gstate() 中的 **linewidth** 79
create_gstate() 中的 **miterlimit** 79
create_gstate() 中的 **opacityfill** 79
create_gstate() 中的 **opacitystroke** 79
create_gstate() 中的 **overprintfill** 79
create_gstate() 中的 **overprintmode** 79
create_gstate() 中的 **overprintstroke** 79
create_gstate() 中的 **renderingintent** 79
create_gstate() 中的 **smoothness** 79
create_gstate() 中的 **strokeadjust** 79
create_gstate() 中的 **textknockout** 79
create_pvtf() 中的 **copy** 26
create_textflow() 中的 **begoptlistchar** 58
create_textflow() 中的 **endoptlistchar** 58
create_textflow() 中的 **fixedtextformat** 58
create_textflow() 中的 **textlen** 58
ctm 参数 81

D

debugshow
 in_fit_table() 67
define_layer() 中的 **children** 88
define_layer() 中的 **creatorinfo** 86
define_layer() 中的 **defaultstate** 86
define_layer() 中的 **depend** 88
define_layer() 中的 **group** 88
define_layer() 中的 **hypertextencoding** 86
define_layer() 中的 **hypertextformat** 86
define_layer() 中的 **initialexportstate** 86
define_layer() 中的 **initialprintstate** 86
define_layer() 中的 **initialviewstate** 86
define_layer() 中的 **intent** 86
define_layer() 中的 **language** 86
define_layer() 中的 **onpanel** 87
define_layer() 中的 **pageelement** 87
define_layer() 中的 **parent** 88

define_layer() 中的 **printsubtype** 87
define_layer() 中的 **zoom** 87
delete_table() 中的 **keephandles** 69
description
在 *begin/end_document()* 中用于附件的
子选项 17

E

end_document() 中的 **action** 17
end_document() 中的 **destname** 17
entirecreate_3dview() 中背景的子选项 142
errorpolicy
用于 *open_pdi_document()* 110
用于 *open_pdi_page()* 112
用于 *process_pdi()* 116
E 用于 *begin_item()* 147

F

filename

metadata 的子选项 144
在 *begin/end_document()* 中用于附件的
子选项 17
在 *begin/end_document()* 中用于搜索的
子选项 19

*fill_*block()* 中的 **boxsize** 121
*fill_*block()* 中的 **charref** 121
*fill_*block()* 中的 **encoding** 121
*fill_*block()* 中的 **errorpolicy** 121
*fill_*block()* 中的 **escapesequence** 121
*fill_*block()* 中的 **font** 121
*fill_*block()* 中的 **ignoreorientation** 121
*fill_*block()* 中的 **refpoint** 121
*fill_*block()* 中的 **showborder** 122
*fill_*block()* 中的 **shrinklimit** 122
*fill_*block()* 中的 **textformat** 122
fillcolor
 create_3dview() 中背景的子选项 142
fill_textblock() 中的 **textflowhandle** 122
fit/info_textline() 和 *add/create_textflow()* 中的
charref 48
fit/info_textline() 和 *add/create_textflow()* 中的
charspacing 48
fit/info_textline() 和 *add/create_textflow()* 中的
dasharray 48
fit/info_textline() 和 *add/create_textflow()* 中的
encoding 48
fit/info_textline() 和 *add/create_textflow()* 中的
escapesequence 48
fit/info_textline() 和 *add/create_textflow()* 中的
fakebold 48
fit/info_textline() 和 *add/create_textflow()* 中的
fillcolor 48
fit/info_textline() 和 *add/create_textflow()* 中的
font 48
fit/info_textline() 和 *add/create_textflow()* 中的
fontname 48
fit/info_textline() 和 *add/create_textflow()* 中的

fontsize 48
fit/info_textline() 和 *add/create_textflow()* 中的
glyphcheck 49
fit/info_textline() 和 *add/create_textflow()* 中的
horizscaling 49
fit/info_textline() 和 *add/create_textflow()* 中的
italicangle 49
fit/info_textline() 和 *add/create_textflow()* 中的
 Kerning 49
fit/info_textline() 和 *add/create_textflow()* 中的
matchbox 49
fit/info_textline() 和 *add/create_textflow()* 中的
overline 50
fit/info_textline() 和 *add/create_textflow()* 中的
showborder 50
fit/info_textline() 和 *add/create_textflow()* 中的
shrinklimit 50
fit/info_textline() 和 *add/create_textflow()* 中的
strikeout 50
fit/info_textline() 和 *add/create_textflow()* 中的
strokecolor 50
fit/info_textline() 和 *add/create_textflow()* 中的
strokewidth 50
fit/info_textline() 和 *add/create_textflow()* 中的
textformat 50
fit/info_textline() 和 *add/create_textflow()* 中的
textrendering 50
fit/info_textline() 和 *add/create_textflow()* 中的
textrise 50
fit/info_textline() 和 *add/create_textflow()* 中的
underline 50
fit/info_textline() 和 *add/create_textflow()* 中的
underlineposition 50
fit/info_textline() 和 *add/create_textflow()* 中的
underlinewidth 50
fit/info_textline() 和 *add/create_textflow()* 中的
wordspacing 50
fit/info_textline() 中的 **alignchar** 48
fit/info_textline() 中的 **errorpolicy** 48
fit/info_textline() 中的 **fitmethod** 48
fit/info_textline() 中的 **leader** 49
fit/info_textline() 中的 **margin** 49
fit/info_textline() 中的 **orientate** 49
fit/info_textline() 中的 **position** 50
fit/info_textline() 中的 **rotate** 50
fit/info_textline() 中的 **stamp** 50
fit/info_textline() 中的 **xadvancelist** 50
fit_image() 和 *fit_pdi_page()* 中的 **boxsize** 104
fit_image/fit_pdi_page() 中的 **dpi** 104
fit_image/pdipage() 中的 **adjustpage** 104
fit_image/pdipage() 中的 **blind** 104
fit_image/pdipage() 中的 **fitmethod** 105
fit_image/pdipage() 中的
ignoreclippingpath 105
fit_image/pdipage() 中的 **ignoreorientation** 105
fit_image/pdipage() 中的 **matchbox** 105
fit_image/pdipage() 中的 **orientate** 105
fit_image/pdipage() 中的 **position** 105

- fit_image/pdipage()* 中的 **rotate** 105
- fit_image/pdipage()* 中的 **scale** 105
- fit_image/pdipage()* 中的 **showborder** 105
- fit_table()* 中 **fill** 的 **area** 子选项 67
- fit_table()* 中 **fill** 的 **fillcolor** 子选项 67
- fit_table()* 中 **stroke** 的 **dasharray** 子选项 68
- fit_table()* 中 **stroke** 的 **dashphase** 子选项 68
- fit_table()* 中 **stroke** 的 **line** 子选项 68
- fit_table()* 中 **stroke** 的 **linewidth** 子选项 68
- fit_table()* 中 **stroke** 的 **strokecolor** 子选项 68
- fit_table()* 中的 **blind** 67
- fit_table()* 中的 **errorpolicy** 67
- fit_table()* 中的 **fill** 67
- fit_table()* 中的 **footer** 67
- fit_table()* 中的 **header** 67
- fit_table()* 中的 **horshrinklimit** 67
- fit_table()* 中的 **rewind** 67
- fit_table()* 中的 **showborder** 67
- fit_table()* 中的 **showcells** 68
- fit_table()* 中的 **showgrid** 68
- fit_table()* 中的 **stroke** 68
- fit_table()* 中的 **vertshrinklimit** 68
- fit_textflow()* 中的 **&name** 选项列表宏调用 56
- fit_textflow()* 中的 **blind** 60
- fit_textflow()* 中的 **comment** 选项列表宏定义 56
- fit_textflow()* 中的 **firstlinedist** 60
- fit_textflow()* 中的 **fitmethad** 60
- fit_textflow()* 中的 **fontscale** 60
- fit_textflow()* 中的 **lastlinedist** 60
- fit_textflow()* 中的 **linespreadlimit** 60
- fit_textflow()* 中的 **macro** 选项列表宏定义 56
- fit_textflow()* 中的 **maxlines** 60
- fit_textflow()* 中的 **minfontsize** 60
- fit_textflow()* 中的 **orientate** 60
- fit_textflow()* 中的 **rewind** 60
- fit_textflow()* 中的 **rotate** 60
- fit_textflow()* 中的 **showborder** 61
- fit_textflow()* 中的 **showtabs** 61
- fit_textflow()* 中的 **verticalalign** 61
- fit_textflow()* 中的 **wrap** 61
- fit_textline()* 中 **leader** 的 **alignment** 子选项 49
- fit_textline()* 中 **leader** 的 **encoding** 子选项 49
- fit_textline()* 中 **leader** 的 **fillcolor** 子选项 49
- fit_textline()* 中 **leader** 的 **font** 子选项 49
- fit_textline()* 中 **leader** 的 **fontname** 子选项 49
- fit_textline()* 中 **leader** 的 **fontsize** 子选项 49
- fit_textline()* 中 **leader** 的 **text** 子选项 49
- fit_textline()* 中 **leader** 的 **yposition** 子选项 49

G

group

- add_nameddest()* 中的选项和 *create_action()*、*create_annotation()*、*create_bookmark()* 和 *begin/end_document()* 中 **destination** 选项的子选项 127
- begin/end_document()* 中 **labels** 选项和 *begin/end_page_ext()* 中 **label** 选项的子选项 20

H

hypertextencoding

- begin/end_document()* 中 **labels** 选项和 *begin/end_page_ext()* 中 **label** 选项的子选项 20

I

iconname

- 用于 *open_pdi_page()* 112

index

- 用于 *begin_item()* 147

indextype

- 在 *begin/end_document()* 中用于搜索的子选项 19

- info_font()* 中 **ascender** 的 **faked** 子选项 35
- info_font()* 中 **fontname** 的 **acrobat** 子选项 35
- info_font()* 中 **fontname** 的 **api** 子选项 35
- info_font()* 中 **fontname** 的 **full** 子选项 35
- info_font()* 中 **fontstyle** 的 **faked** 子选项 35
- info_font()* 中 **glyphid** 的 **code** 子选项 35
- info_font()* 中 **glyphid** 的 **unicode** 子选项 35
- info_font()* 中 **glyphname** 的 **code** 子选项 35
- info_font()* 中 **glyphname** 的 **glyphid** 子选项 35
- info_font()* 中 **glyphname** 的 **unicode** 子选项 35
- info_font()* 中 **unicode** 的 **cid** 子选项 36
- info_font()* 中 **unicode** 的 **code** 子选项 36
- info_font()* 中 **unicode** 的 **glyphid** 子选项 36
- info_font()* 中 **unicode** 的 **glyphname** 子选项 36
- info_font()* 中编码的 **actual** 子选项 35
- info_font()* 中编码的 **api** 子选项 35
- info_font()* 中代码的 **glyphname** 子选项 35
- info_font()* 中代码的 **unicode** 子选项 35
- info_font()* 中的 **ascender** 35
- info_font()* 中的 **capheight** 35
- info_font()* 中的 **cidfont** 35

info_font() 中的 **code** 35
info_font() 中的 **descender** 35
info_font() 中的 **encoding** 35
info_font() 中的 **fontfile** 35
info_font() 中的 **fontname** 35
info_font() 中的 **fontstyle** 35
info_font() 中的 **glyphid** 35
info_font() 中的 **glyphname** 35
info_font() 中的 **hostfont** 35
info_font() 中的 **italicangle** 35
info_font() 中的 **kerningpairs** 35
info_font() 中的 **linegap** 35
info_font() 中的 **maxcode** 35
info_font() 中的 **metricsfile** 35
info_font() 中的 **numcids** 35
info_font() 中的 **numglyphs** 35
info_font() 中的 **replacementchar** 35
info_font() 中的 **standardfont** 35
info_font() 中的 **supplement** 35
info_font() 中的 **symbolfont** 36
info_font() 中的 **unicode** 36
info_font() 中的 **unicodefont** 36
info_font() 中的 **vertical** 36
info_font() 中的 **weight** 36
info_font() 中的 **willembed** 36
info_font() 中的 **willsubset** 36
info_font() 中的 **xheight** 36
info_matchbox() 中的 **exists** 关键字 71
info_matchbox() 中的 **firstbodyrow** 关键字 71
info_matchbox() 中的 **height** 关键字 71
info_matchbox() 中的 **width** 关键字 71
info_matchbox() 中的 **x1, y1, ..., x4, y4** 关键字 71
infomode
 用于 *open_pdi_document()* 110
 用于 *open_pdi_page()* 112
info_table() 中的 **firstbodyrow** 关键字 68
info_table() 中的 **height** 关键字 68
info_table() 中的 **horboxgap** 关键字 68
info_table() 中的 **horshrinking** 关键字 68
info_table() 中的 **lastbodyrow** 关键字 68
info_table() 中的 **returnreason** 关键字 69
info_table() 中的 **rowcount** 关键字 69
info_table() 中的 **rowsplit** 关键字 69
info_table() 中的 **vertboxgap** 关键字 69
info_table() 中的 **vertshrinking** 关键字 69
info_table() 中的 **width** 关键字 69
info_table() 中的 **x1, y1, ..., x4, y4** 关键字 69
info_table() 中的 **xvrtline** 关键字 69
info_table() 中的 **yhorline** 关键字 69
info_textflow() 中的 **boxlinecount** 关键字 62
info_textflow() 中的 **firstlinedist** 关键字 62
info_textflow() 中的 **firstparalinecount** 关键字 62
info_textflow() 中的 **lastlinedist** 关键字 62
info_textflow() 中的 **lastmark** 关键字 62
info_textflow() 中的 **lastparalinecount** 关键字 62
info_textflow() 中的 **leading** 关键字 62

info_textflow() 中的 **leftlinex, leftliney** 关键字 62
info_textflow() 中的 **maxlinelength** 关键字 62
info_textflow() 中的 **maxliney** 关键字 62
info_textflow() 中的 **minlinelength** 关键字 62
info_textflow() 中的 **minliney** 关键字 62
info_textflow() 中的 **remainchars** 关键字 62
info_textflow() 中的 **returnreason** 关键字 62
info_textflow() 中的 **rightlinex, rightliney** 关键字 62
info_textflow() 中的 **split** 关键字 62
info_textflow() 中的 **textendx, textendy** 关键字 62
info_textflow() 中的 **textheight** 关键字 62
info_textflow() 中的 **textwidth** 关键字 62
info_textflow() 中的 **used** 关键字 62
info_textflow() 中的 **x1, y1, ..., x4, y4** 关键字 62
info_textline() 中的 **angle** 关键字 51
info_textline() 中的 **ascender** 关键字 51
info_textline() 中的 **capheight** 关键字 51
info_textline() 中的 **descender** 关键字 51
info_textline() 中的 **endx, endy** 关键字 51
info_textline() 中的 **height** 关键字 51
info_textline() 中的 **perpendiculardir** 关键字 51
info_textline() 中的 **scalex, scaley** 关键字 51
info_textline() 中的 **startx, starty** 关键字 51
info_textline() 中的 **unmappedglyphs** 关键字 51
info_textline() 中的 **width** 关键字 51
info_textline() 中的 **writingdirx, writingdiry** 关键字 51
info_textline() 中的 **xheight** 关键字 51
inline
 用于 *begin_item()* 147
inmemory
 用于 *open_pdi_document* 110
inputencoding
 metadata 的子选项 144
inputformat
 metadata 的子选项 144

L

Lang
 用于 *begin_item()* 147
layer
 用于 *open_pdi_page()* 112
left
 add_nameddest() 中的选项和
 create_action()、*create_annotation()*、
 create_bookmark() 和 *begin/end_document()*
 中 *destination* 选项的子选项 127
load_3d() 中的 **defaultview** 141
load_3d() 中的 **hypertextencoding** 141
load_3d() 中的 **script** 141
load_3d() 中的 **views** 141
load_3ddata() 中的 **errorpolicy** 141
load_font() 中的 **ascender** 33
load_font() 中的 **autocidfont** 33
load_font() 中的 **autosubsetting** 33

- `load_font()` 中的 **capheight** 33
- `load_font()` 中的 **descender** 33
- `load_font()` 中的 **embedding** 33
- `load_font()` 中的 **errorpolicy** 33
- `load_font()` 中的 **fontstyle** 33
- `load_font()` 中的 **keepnative** 33
- `load_font()` 中的 **kerning** 33
- `load_font()` 中的 **linecap** 34
- `load_font()` 中的 **metadata** 34
- `load_font()` 中的 **monospace** 34
- `load_font()` 中的 **subsetlimit** 34
- `load_font()` 中的 **subsetminsize** 34
- `load_font()` 中的 **subsetting** 34
- `load_font()` 中的 **unicodemap** 34
- `load_font()` 中的 **vertical** 34
- `load_font()` 中的 **xheight** 34
- `load_font()` 中的 **replacementchar** 34
- `load_iccprofile()` 中的 **description** 92
- `load_iccprofile()` 中的 **embedprofile** 92
- `load_iccprofile()` 中的 **errorpolicy** 92
- `load_iccprofile()` 中的 **metadata** 92
- `load_iccprofile()` 中的 **usage** 92
- `load_image()` 和 `begin_template()` 中的 **iconname** 102
- `load_image()` 和 `begin_template()` 中的 **layer** 102
- `load_image()` 和 `begin_template()` 中的 **OPI-1.3** 102
- `load_image()` 和 `begin_template()` 中的 **OPI-2.0** 103
- `load_image()` 和 `begin_template_ext()` 中的 **metadata** 102
- `load_image()` 中的 **bitreverse** 101
- `load_image()` 中的 **bpc** 101
- `load_image()` 中的 **clippingpathname** 101
- `load_image()` 中的 **colorize** 101
- `load_image()` 中的 **components** 101
- `load_image()` 中的 **errorpolicy** 101
- `load_image()` 中的 **height** 101
- `load_image()` 中的 **honorclippingpath** 102
- `load_image()` 中的 **honoriccprofile** 101
- `load_image()` 中的 **hypertextencoding** 102
- `load_image()` 中的 **iccprofile** 102
- `load_image()` 中的 **ignoremask** 102
- `load_image()` 中的 **ignoreorientation** 102
- `load_image()` 中的 **inline** 102
- `load_image()` 中的 **interpolate** 102
- `load_image()` 中的 **invert** 102
- `load_image()` 中的 **K** 102
- `load_image()` 中的 **mask** 102
- `load_image()` 中的 **masked** 102
- `load_image()` 中的 **page** 103
- `load_image()` 中的 **passthrough** 103
- `load_image()` 中的 **renderingintent** 103
- `load_image()` 中的 **template** 103
- `load_image()` 中的 **width** 103

M

metadata 144

用于 `open_pdi_page()` 112

P

page

`add_nameddest()` 中的选项和 `create_action()`、`create_annotation()`、`create_bookmark()` 和 `begin/end_document()` 中 **destination** 选项的子选项 127

pagenumber

`begin/end_document()` 中 **labels** 选项和 `begin/end_page_ext()` 中 **label** 选项的子选项 20

parent

用于 `begin_item()` 147

password

用于 `open_pdi_document` 110

pdiusebox

用于 `open_pdi_page()` 112

prefix

`begin/end_document()` 中 **labels** 选项和 `begin/end_page_ext()` 中 **label** 选项的子选项 20

R

repair

用于 `open_pdi_document` 110

requiredmode

用于 `open_pdi_document` 110

`resume_page()` 中的 **group** 25

`resume_page()` 中的 **pagenumber** 25

right

`add_nameddest()` 中的选项和 `create_action()`、`create_annotation()`、`create_bookmark()` 和 `begin/end_document()` 中 **destination** 选项的子选项 128

RowSpan

用于 `begin_item()` 147

日志参数中的 **classes** 30

日志参数中的 **disable** 30

日志参数中的 **enable** 30

日志参数中的 **filename** 30

日志参数中的 **flush** 30

日志参数中的 **remove** 30

日志参数中的 **stringlimit** 30

S

Scope

用于 `begin_item()` 147
`setdashpattern()` 中的 `dasharray` 73
`setdashpattern()` 中的 `dashphase` 73
`shading()` 中的 `antialias` 97
`shading()` 中的 `extend0` 97
`shading()` 中的 `extend1` 97
`shading()` 中的 `N` 97
`shading()` 中的 `ro` 97
`shading()` 中的 `r1` 97
`shading_pattern()` 中的 `gstate` 96
start

`begin/end_document()` 中 `labels` 选项和
`begin/end_page_ext()` 中 `label` 选项的
子选项 20

style

`begin/end_document()` 中 `labels` 选项和
`begin/end_page_ext()` 中 `label` 选项的
子选项 20

T

Title

用于 `begin_item()` 147

top

`add_nameddest()` 中的选项和
`create_action()`、`create_annotation()`、
`create_bookmark()` 和 `begin/end_document()`
中 `destination` 选项的子选项 128

type

`add_nameddest()` 中的选项和
`create_action()`、`create_annotation()`、
`create_bookmark()` 和 `begin/end_document()`
中 `destination` 选项的子选项 128

Y

用于 `matchbox` 选项的 `borderwidth` 子选项 70
用于 `matchbox` 选项的 `boxheight` 子选项 70
用于 `matchbox` 选项的 `boxwidth` 子选项 70
用于 `matchbox` 选项的 `clipping` 子选项 70
用于 `matchbox` 选项的 `dasharray` 子选项 70
用于 `matchbox` 选项的 `dashphase` 子选项 70
用于 `matchbox` 选项的 `drawbottom` 子选项 70
用于 `matchbox` 选项的 `drawleft` 子选项 70
用于 `matchbox` 选项的 `drawright` 子选项 70
用于 `matchbox` 选项的 `drawtop` 子选项 70
用于 `matchbox` 选项的 `end` 子选项 70
用于 `matchbox` 选项的 `fillcolor` 子选项 70
用于 `matchbox` 选项的 `innerbox` 子选项 70
用于 `matchbox` 选项的 `linecap` 子选项 70
用于 `matchbox` 选项的 `linejoin` 子选项 70
用于 `matchbox` 选项的 `margin` 子选项 70
用于 `matchbox` 选项的 `name` 子选项 71
用于 `matchbox` 选项的 `offsetbottom` 子选项 71
用于 `matchbox` 选项的 `offsetleft` 子选项 71
用于 `matchbox` 选项的 `offsetright` 子选项 71
用于 `matchbox` 选项的 `offsettop` 子选项 71
用于 `matchbox` 选项的 `openrect` 子选项 71
用于 `matchbox` 选项的 `strokecolor` 子选项 70

Z

zoom

`add_nameddest()` 中的选项和
`create_action()`、`create_annotation()`、
`create_bookmark()` 和 `begin/end_document()`
中 `destination` 选项的子选项 128
在 `fit/info_textline()` 中的 `boxsize` 48

D 修订历史记录

日期	更改
2007 年 2 月 19 日	► 有关 <i>PDFlib 7.0.1</i> 的各种更新和修正
2006 年 10 月 3 日	► 有关 <i>PDFlib 7.0.0</i> 的更新和重构；手册拆分为教程和 <i>API</i> 参考
2006 年 2 月 21 日	► 有关 <i>PDFlib 6.0.3</i> 的各种更新和修正，并增加了 <i>Ruby</i> 部分
2005 年 8 月 9 日	► 有关 <i>PDFlib 6.0.2</i> 的各种更新和修正
2004 年 11 月 17 日	► 有关 <i>PDFlib 6.0.1</i> 的小范围更新和修正 ► 第 8 章中介绍了语言特定函数范例的新格式 ► 第 3 章增加了超文本示例
2004 年 6 月 18 日	► 重要的 <i>PDFlib 6</i> 更改
2004 年 1 月 21 日	► 有关 <i>PDFlib 5.0.3</i> 的小范围内容增加和修正
2003 年 9 月 15 日	► 有关 <i>PDFlib 5.0.2</i> 的小范围内容增加和修正，并增加了块规范
2003 年 5 月 26 日	► 有关 <i>PDFlib 5.0.1</i> 的小范围更新和修正
2003 年 3 月 26 日	► 有关 <i>for PDFlib 5.0.0</i> 的小范围更改和内容改写
2002 年 6 月 14 日	► 有关 <i>PDFlib 4.0.3</i> 的小范围更改以及有关 <i>.NET</i> 绑定的扩展
2002 年 1 月 26 日	► 有关 <i>PDFlib 4.0.2</i> 的小范围更改以及有关 <i>IBM eServer</i> 版本的扩展
2001 年 5 月 17 日	► 有关 <i>PDFlib 4.0.1</i> 的小范围更改
2001 年 4 月 1 日	► 增添 <i>PDFlib 4.0.0</i> 的 <i>PDI</i> 及其他功能
2001 年 2 月 5 日	► 增添 <i>PDFlib 3.5.0</i> 中的模板和 <i>CMYK</i> 功能
2000 年 12 月 22 日	► <i>ColdFusion</i> 文档以及 <i>PDFlib 3.03</i> 相关增加内容；编制专门的 <i>COM</i> 版手册
2000 年 8 月 8 日	► <i>Delphi</i> 文档及有关 <i>PDFlib 3.02</i> 的小范围内容增加
2000 年 7 月 1 日	► 有关 <i>PDFlib 3.01</i> 的增加内容和相关澄清
2000 年 2 月 20 日	► 有关 <i>PDFlib 3.0</i> 的更改
1999 年 8 月 2 日	► 有关 <i>PDFlib 2.01</i> 的小范围更改及内容增加
1999 年 6 月 29 日	► 增添特定语言绑定的专门章节 ► 有关 <i>PDFlib 2.0</i> 的扩展
1999 年 2 月 1 日	► 有关 <i>PDFlib 1.0</i> 的小范围更改（未公开发布）
1998 年 8 月 10 日	► 有关 <i>PDFlib 0.7</i> 的扩展（仅面向一位客户）
1998 年 7 月 8 日	► 第一次尝试描述 <i>PDFlib 0.6</i> 中的 <i>PDFlib</i> 脚本支持
1998 年 2 月 25 日	► 小幅扩充手册以涵盖 <i>PDFlib 0.5</i>
1997 年 9 月 22 日	► 第一次公开发布 <i>PDFlib 0.4</i> 及本手册

索引

请注意：函数、参数和选项在单独的附录中列出。

A

All 专色名称 91
Author 域 143

B

B é zier 曲线 82
表格式化 64
标准页面大小 22
不可见文本 41

C

cmlyk 关键字 8
CMYK 颜色 89
Creator 域 143
参数处理函数 11

D

Dublin Core 143
对称 77
对齐方式 (*position* 选项) 50
对象范围 9

F

范围 9
分色色彩空间 89

G

gray 关键字 8
光栅图像函数 99

H

函数范围 9
行
 虚线和样式 73
行的虚线样式 73
横向模式 23

I

ICC 色彩特征描述文件 92
iccbasedcmlyk 关键字 8
iccbasedgray 关键字 8
iccbasedrgb 关键字 8

J

基于 *ICC* 的颜色 89
镜像 77

K

Keywords 域 143
快速 *Web* 查看 18

L

lab 关键字 8
路径绘制和剪切 84
轮廓文本 41

M

模板范围 9

N

None 专色名称 91

P

page 范围 9
path 范围 9
pCOS 函数 114
PDF 导入函数 (*PDI*) 109
PDF 的导入函数 (*PDI*) 109
PDF/A 或 *PDF/X* 输出方法 116
PDF_activate_item() 148
PDF_add_nameddest() 127
PDF_add_table_cell() 64
PDF_add_textflow() 52
PDF_add_thumbnail() 107
PDF_arc() 82
PDF_arcn() 83
PDF_begin_document() 16
PDF_begin_font() 37
PDF_begin_glyph() 38
PDF_begin_item() 146
PDF_begin_layer() 87
PDF_begin_page_ext() 22, 23
PDF_begin_pattern 95
PDF_begin_template() 106
PDF_begin_template_ext() 106
PDF_circle() 82
PDF_clip() 85
PDF_close_image() 103
PDF_close_pdi_document() 111
PDF_close_pdi_page() 113

PDF_closepath() 83
 PDF_closepath_fill_stroke() 85
 PDF_closepath_stroke() 84
 PDF_concat() 78
 PDF_continue_text() 43
 PDF_continue_text2() 43
 PDF_create_3dview() 141
 PDF_create_action() 123
 PDF_create_annotation() 129
 PDF_create_bookmark() 139
 PDF_create_field() 134
 PDF_create_fieldgroup() 135
 PDF_create_gstate() 79
 PDF_create_pvf() 26
 PDF_create_textflow() 57
 PDF_curveto() 82
 PDF_define_layer() 86
 PDF_delete() 15
 PDF_delete_dl() 15
 PDF_delete_pvf() 27
 PDF_delete_table() 69
 PDF_delete_textflow() 63
 PDF_encoding_set_char() 39
 PDF_end_document() 17
 PDF_end_font() 38
 PDF_end_glyph() 38
 PDF_end_item() 148
 PDF_end_layer() 88
 PDF_end_pattern() 95
 PDF_end_template() 106
 PDF_endpath() 85
 PDF_fill() 84
 PDF_fill_imageblock() 120
 PDF_fill_pdfblock() 121
 PDF_fill_stroke() 85
 PDF_fill_textblock() 119
 PDF_fit_image() 104
 PDF_fit_pdi_page() 113
 PDF_fit_table() 66
 PDF_fit_textflow() 59
 PDF_fit_textline() 47
 PDF_get_apiname() 29
 PDF_get_buffer() 21
 PDF_get_errmsg() 29
 PDF_get_errnum() 28
 PDF_get_opaque() 29
 PDF_get_parameter() 11
 PDF_get_value() 11
 PDF_info_font() 34
 PDF_info_matchbox() 71
 PDF_info_table() 68
 PDF_info_textflow() 62
 PDF_info_textline() 47
 PDF_initgraphics() 75
 PDF_lineto() 81
 PDF_load_3d() 141
 PDF_load_font() 31
 PDF_load_iccprofile() 92
 PDF_load_image() 100
 PDF_makespotcolor() 91
 PDF_moveto() 81
 PDF_new() 14
 PDF_new_dl() 14
 PDF_new2() 14
 PDF_open_pdi_callback() 111
 PDF_open_pdi_document() 109
 PDF_open_pdi_page() 112
 PDF_pcos_get_number() 114
 PDF_pcos_get_stream() 115
 PDF_pcos_get_string() 114
 PDF_process_pdi() 116
 PDF_rect() 83
 PDF_restore() 76
 PDF_resume_page() 25
 PDF_rotate() 77
 PDF_save() 76
 PDF_scale() 77
 PDF_set_gstate() 80
 PDF_set_info() 143
 PDF_set_info2() 143
 PDF_set_layer_dependency() 87
 PDF_set_parameter() 12
 PDF_set_text_pos() 42
 PDF_set_value() 11
 PDF_setcolor() 90
 PDF_setdash() 73
 PDF_setdashpattern() 73
 PDF_setflat() 73
 PDF_setfont() 41
 PDF_setlinecap() 74
 PDF_setlinejoin() 74
 PDF_setlinewidth() 75
 PDF_setmatrix() 78
 PDF_setmiterlimit() 75
 PDF_shading() 96
 PDF_shading_pattern() 95
 PDF_shfill() 96
 PDF_show() 42
 PDF_show_xy() 43
 PDF_show_xy2() 43
 PDF_show2() 42
 PDF_skew() 78
 PDF_stringwidth() 44
 PDF_stringwidth2() 44
 PDF_stroke() 84
 PDF_suspend_page() 24
 PDF_translate() 77
 PDF_utf16_to_utf8() 45
 PDF_utf32_to_utf16() 46
 PDF_utf8_to_utf16() 45, 46
 PDF_xshow() 42
 PDFlib Personalization Server (PPS) 119
 PDI (PDF 导入) 109
 PPS (PDFlib Personalization Server) 119

Q

倾斜 78

R

rgb 关键字 8

RGB 颜色 89

日志 30

S

spot 关键字 8

spotname 关键字 8

Subject 域 143

上标 41

设置函数 13

缩览图 107

索引色彩空间 89

T

Title 域 143

Trapped 域 143

图案范围 9

图案关键字 8

图像函数 99

图形函数 73

图形状态, 显式 79

W

网页优化 *PDF* 18

文本函数 31

文本流

内嵌选项列表 58

文本流的内嵌选项列表 58

文档范围 9

文档和页面函数 16

文档信息域 143

X

XMP 元数据 144

下标 41

显式图形状态 79

线性化 *PDF* 18

信息域 143

许可证 13

选项列表 5

选项列表中的布尔值 7

选项列表中的单字符值 7

选项列表中的动作列表 7

选项列表中的浮点型和整型值 7

选项列表中的关键字 7

选项列表中的句柄 7

选项列表中的矩形 7

选项列表中的列表值 7

选项列表中的色彩值 7

选项列表中的字符串值 7

Y

颜色函数 89

页面大小格式 22

元数据 144

Z

专色 (分色色彩空间) 89

字符串索引 13

字体范围 9

字形范围 9