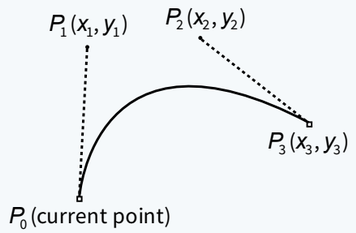
PDF Association Cheat Sheet – Graphic Operators

Vector Graphics

Path Construction

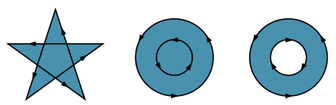
- A PDF vector *path* has a *current point* and can have multiple independent *subpaths,* each of which can have multiple *segments* (curves, lines). Subpaths can be open or closed.
- Cubic Bézier curve control points (**c** operator example):



x₁ y₁ x₂ y₂ x₃ y₃	**c**	**Curve**. Add a curved Bézier segment to the current subpath using 3 control points, as shown above. New current point is then (x_3, y_3).
	h	Close the current subpath by adding a straight-line segment from the current point to the start point of the current subpath. Next operator must be **m**, **re** to start new subpath, or a path painting operator to paint the path.
x y	**l**	**Line-to**. (Lowercase L) Add a straight-line segment from the current point to (*x, y*). New current point is then (*x, y*).
x y	**m**	**Move-to**. Begin a new subpath in the current path by setting the current point to (*x, y*).
x y width height	**re**	**Rectangle**. Add a closed rectangle subpath with lower-left corner at (*x, y*) and dimensions *width* and *height*. Next operator must be **m**, **re** to start new subpath, or a path painting operator to paint the path.
x₂ y₂ x₃ y₃	**v**	Add a curved Bézier segment to the subpath from current point to (x_3, y_3), using the current point and (x_2, y_2) as the Bézier control points. New current point is then (x_3, y_3).
x₁ y₁ x₃ y₃	**y**	Add a curved Bézier segment to the current subpath. The curve extends from current point to the point (x_3, y_3), using (x_1, y_1) and (x_3, y_3) as the Bézier control points. New current point is then (x_3, y_3).

Path Painting

Combined fill & stroke painting must be treated as a single atomic graphics object. Painting a path makes current point undefined. There is no current path until an **m** or **re** operator.



Non-zero Winding Rule

Even-odd Winding Rule

b	Close, fill, and stroke path using the non-zero winding rule. Same as: **h B**
B	Fill and stroke path using the non-zero winding rule.
b*	Close, fill, and stroke path using the even-odd winding rule. Same as: **h B***
B*	Fill and stroke path using the even-odd winding rule.
f	Close all subpaths then fill the current path using non-zero winding rule.
f*	Fill path using even-odd winding rule.
F	Fill the current path using non-zero winding number rule. (*Deprecated in PDF 2.0*)
n	End path without filling or stroking ("*no op*"). Used after **W** / **W*** operators to establish the new clipping path.
s	Close and stroke the current path. Same as **h S**.
S	Stroke the current path.

Clipping

W	Modify the current clipping path by intersecting it with the current path, using the non-zero winding number rule to determine which regions lie inside the clipping path. Initial clipping path is the page **MediaBox**.
W*	Modify the current clip path by intersecting it with the current path using the even-odd winding rule to determine which regions lie inside the clipping path. Initial clipping path is the page **MediaBox**.

Text

Text Object

BT	Begin text object. Nested.
ET	End text object. After filling/stroking and if **Tr** was set to a text clipping mode (4 - 7), then change the clipping path to the intersection of the current clipping path and any glyph-based clipping path using the Non-zero Winding rule. Must be paired with **BT** operator.

Text State

Text knockout can only be set via graphics state parameter dictionary **TK** entry.

number	**Tc**	Set character spacing in unscaled text space units to *number*. Character spacing is used by **Tj**, **TJ** and **'** text showing operators. Initial value: 0.
name size	**Tf**	Set text font and size (number) in the graphics state. *name* is the name of a font resource in the **Font** subdictionary of the current resource dictionary. Zero sized text does not mark or clip any pixels. There are no default / initial values. Equivalent to the **Font** entry (array) in the graphics state parameter dictionary.
number	**TL**	Set text leading to *number* expressed in unscaled text space units. Text leading is only used by **T***, **"** and **'** text showing operators. Initial value: 0.
mode	**Tr**	Set text rendering mode (integer). Initial value is 0 (filled text). Once set to a clipping mode (4-7), cannot change back before **ET**.
number	**Ts**	Set text rise expressed in unscaled text space units. Initial value: 0.
number	**Tw**	Set word spacing in unscaled text space units. Word spacing is used by **Tj**, **TJ** and **'** text showing operators. Initial value: 0.
number	**Tz**	Set horizontal text scaling specified as a percentage of normal width (*number* > 0). Initial value: 100 (100% is normal width).

Tr modes:

Mode	Description	Example
0	Filled text.	R
1	Stroked text.	R
2	Fill, then stroke text.	R
3	Invisible. Neither fill nor stroke text. Text will still be selectable/searchable.	
4	Fill text and add to path for clipping.	R
5	Stroke text and add to path for clipping.	R
6	Fill, then stroke and add to path for clipping.	R
7	Add text to path for clipping.	R

Text Positioning

	T*	Move to start of next text line.
t_x t_y	**Td**	Move to the start of the next line, offset from the start of the current line by (t_x, t_y). t_x and t_y are numbers expressed in unscaled text space units.
t_x t_y	**TD**	Move to the start of the next line, offset from the start of the current line by (t_x, t_y). As a side effect, also set the leading parameter in the text state. **TD** is equivalent to: *$-t_y$* **TL** *t_x t_y* **Td**
a b c d e f	**Tm**	Set text matrix and text line matrix.

Text Showing

string	**Tj**	Show text *string*. *string* comprises glyph IDs.
[string number …]	**TJ**	Show text allowing individual glyph positioning. Each element in array is either a string (glyph IDs), or a number representing a text adjustment that is **subtracted** from the current horizontal or vertical coordinate, depending on the writing mode.
string	**'**	Move to the next line and show text *string.*
a_w a_c string	**"**	Set word and character spacing to a_w and a_c numbers respectively, move to next line, and show text *string*.

Type 3 fonts

Must always be the first operator in a Type 3 glyph description content stream. The number w_x is the horizontal displacement and the number w_y is the vertical displacement in the glyph coordinate system.

w_x w_y	**d0**	(Ends in digit zero). Set width information for a Type 3 glyph description and declare that it specifies **both its shape and color**.
w_x w_y ll_x ll_y ur_x ur_y	**d1**	(Ends in digit 1). Set width and bounding box information for a Type 3 glyph description and it specifies **only shape** and not color.

Marked Content

Marked Content Sequences and Points

See subclause 14.6 in ISO 32000-2:2020. Introduced in PDF 1.2.
EMC must be paired with either a **BDC** or **BMC** operator and nested correctly with **BT/ET** text object, **BX/EX** compatibility operators, and **q/Q** paired operators.

name property	**BDC**	Begin marked-content sequence with property list. Nested. *name* is a name object indicating the role or significance of the sequence. *property* is either an inline dictionary or a name of a resource in the **Properties** subdictionary of the current resource dictionary.
name	**BMC**	Begin a marked-content sequence. Nested. *name* is a name indicating the role or significance of the sequence
name property	**DP**	Define a marked-content point with property list. *name* is a name object indicating the role or significance of the point. *property* is either an inline dictionary or the name of a resource in the **Properties** subdictionary of the current resource dictionary.
	EMC	End marked-content sequence.
name	**MP**	Define a marked-content point. *tag* is a name indicating the role or significance of the sequence

PDF association

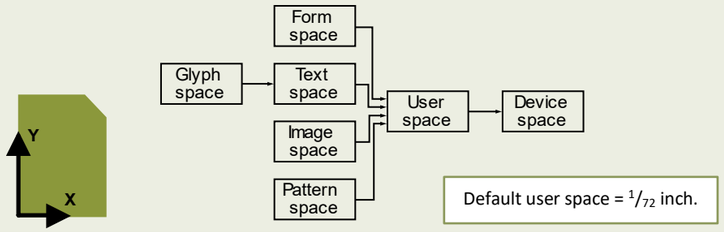
PDF association Date: 2024.10.27 11:57:07 Z

PDF Association Cheat Sheet – Graphic Operators

Graphics State

Coordinate Systems

Relationship among PDF coordinate systems:



a b c d e f	**cm**	Concatenate specified matrix to current transformation matrix. Operands are numbers. Matrix is: $\begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix}$ Initial value is a matrix that transforms default user coordinates ($^1/_{72}$ inch) into device coordinates.

Graphics State Parameters

[on off …] phase	**d**	Set stroking dash pattern to the specified dash array (*on, off, on, …* - numbers) and dash phase (number). Equivalent to the **D** entry in the graphics state parameter dictionary. Initial value is **[]** (solid undashed line).

`[3] 0`	3 on, 3 off, …
`[2 1 3] -2`	2 off, 2 on, 1 off, 3 on, 2 off, 1 on, 3 off, …

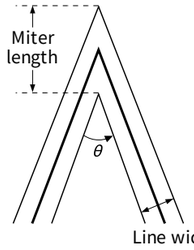
name	**gs**	(*PDF 1.2*) Set parameters from the named graphics state parameter dictionary. *name* is the name of a graphics state parameter dictionary in the **ExtGState** subdictionary of the current resource dictionary.
number	**i**	Set flatness tolerance to *number*. Equivalent to the **FL** entry in the graphics state parameter dictionary. Initial value is 1.0.
style	**j**	(Lowercase "J") Set line join style (integer). Equivalent to the **LJ** entry in the graphics state parameter dictionary. Initial value is 0 (miter join).

Style	Name	Appearance
0	Miter join	
1	Round join	
2	Bevel join	

style	**J**	(Uppercase J) Set line cap style (integer). Equivalent to the **LC** entry in the graphics state parameter dictionary. Initial value is 0 (butt cap).

Style	Name	Appearance
0	Butt cap	
1	Round cap	
2	Projecting square cap	

number	**M**	Set the miter limit ratio. When line width is zero, the miter length is zero. Equivalent to the **ML** entry in the graphics state parameter dictionary. Initial value is 10.0, for a miter cutoff below approx. $\theta = 11.5°$.



$$miter\ limit\ ratio = \frac{miter\ length}{line\ width} = \frac{1}{\sin\frac{\theta}{2}}$$

	q	Save graphics state ("**push**"). Nested. Must be paired with a **Q** operator.
	Q	Restore graphics state ("**pop**"). Nested. Must be paired with a **q** operator.
name	**ri**	Set the color render intent in the graphics state. *intent* is a name and usually one of *AbsoluteColorimetric*, *RelativeColorimetric*, *Saturation*, or *Perceptual*. Equivalent to the **RI** entry in the graphics state parameter dictionary. Initial value is *RelativeColorimetric*.
number	**w**	Set line width to *number* in user space units (*number* ≥ 0). Initial value is 1.0.

Other operators

Inline Images

Only for very small images (< 4KB). Otherwise use Image XObject and **Do** operator.

BI	Begin inline image object. Followed by Image XObject dictionary key value pairs. Certain key names and values may also be abbreviated. Abbreviated key names take precedence over full key names.
ID	Begin inline image data after a single whitespace character.
EI	Ends an inline image object.

Object painting

name	**Do**	Invoke (paint) the named XObject. *name* is the name of an XObject that is in the **XObject** subdictionary of the current resource dictionary.
name	**sh**	*(PDF 1.3)* Paint area defined by a shading pattern. *name* is the name of a shading dictionary resource in the **Shading** subdictionary of the current resource dictionary.

Compatibility Sections

Introduced in PDF 1.1.

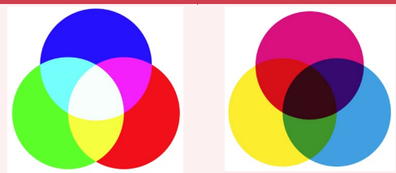
BX	Begin compatibility section. Nested. Unrecognised operators (along with all operands) will be ignored without error until the balancing **EX** operator.
EX	End compatibility section. Must be paired with **BX** operator.

All paired operators must be nested correctly: **BDC**/**EMC** or **BMC**/**EMC** marked content; **BT**/**ET** text object, **BX**/**EX** compatibility section, and **q**/**Q** graphics stack.

Color

Color Operators

UPPERCASE = stroking
lowercase = filling (non-stroking)



name *name*	**CS** **cs**	(*PDF 1.1*) Set color space for stroking (**CS**) or non-stroking (**cs**) operations. If the color space is one that can be specified by a name and no additional parameters (**DeviceGray**, **DeviceRGB**, **DeviceCMYK**, and certain cases of **Pattern**), that name may be specified directly. Otherwise, *name* is a resource in the **ColorSpace** subdictionary of the current resource dictionary. Initial color space is **DeviceGray**.
gray *gray*	**G** **g**	Set gray level for stroking (**G**) or non-stroking (**g**) operations. 0.0 (black) ≤ *gray* ≤ 1.0 (white)
c m y k *c m y k*	**K** **k**	Set CMYK levels for stroking (**K**) or non-stroking (**k**) operations. 0.0 ≤ *c, m, y, k* ≤ 1.0
r g b *r g b*	**RG** **rg**	Set RGB levels for stroking (**RG**) or non-stroking (**rg**) operations. 0.0 (no color) ≤ *r, g, b* ≤ 1.0 (max. color)
C_1 … C_n C_1 … C_n	**SC** **sc**	*(PDF 1.1)* Set the color to use for stroking operations (**SC**) or non-stroking (**sc**) to CIE-based (other than **ICCBased**), or **Indexed** color space. The number of operands required, and their interpretation depends on the current stroking/non-stroking color space: For **DeviceGray**, **CalGray**, and **Indexed** color spaces, one operand is required (*n* = 1). For **DeviceRGB**, **CalRGB**, and **Lab** color spaces, three operands are required (*n* = 3). For **DeviceCMYK**, four operands are required (*n* = 4)
C_1 … C_n C_1 … C_n *name* C_1 … C_n C_1 … C_n *name*	**SCN** **SCN** **scn** **scn**	*(PDF 1.2)* Same as **SC**/**sc** operators but also for **Pattern**, **Separation**, **DeviceN** and **ICCBased** color spaces. If the current stroking (**SCN**) or non-stroking (**scn**) color space is a **Separation**, **DeviceN**, or **ICCBased** color space, the operands C_1… C_n are numbers. The number of operands and their interpretation depends on the color space. If the current color space is a **Pattern** color space, then *name* is the name of an entry in the **Pattern** subdictionary of the current resource dictionary. For an uncolored tiling pattern (**PatternType**=1 and **PaintType**=2), C_1… C_n are component values specifying a color in the pattern's underlying color space. For other types of patterns, these operands are not specified.

PDF association