

Pixmap

The alpha channel is now optional. Its presence is controlled by a new boolean parameter (called `alpha`). This has the following consequences:

- The size of one pixel can be two different values. For e.g. colorspace RGB, this size may be 3 (no alpha) or 4 bytes. The size of a pixmap is therefore determined not only by its colorspace, but also by its alpha value.
- Handling of pixmaps needs to take the alpha parameter into account. A decision has to be made during pixmap creation. Pixmaps coming from other sources or generated by some functions may or may not contain an alpha, which therefore needs to be checked.
- In general, the alpha channel should be avoided to benefit from significant memory savings.
- The `savealpha` parameter disappeared: all methods saving pixmaps (`writePNG` and friends) now always store the complete pixmap.
- Pixmaps created from PDF-internal images may or may not contain an alpha – this solely depends on how the image was stored in the PDF.
- The `Pixmap` and `Colorspace` classes have been extended with properties that help determine their characteristics – see the following list:
 - `Pixmap.alpha` – bool
 - `Pixmap.stride` – integer containing the number of bytes of one line of the pixmap's IRect
 - `Pixmap.n` – integer containing the number of bytes per pixel
 - `Colorspace.nbytes` – integer containing the number of bytes used to determine the color (1 = DeviceGray, 3 = DeviceRGB, 4 = DeviceCMYK)
 - `Colorspace.name` – string naming the colorspace, one of DeviceGray, DeviceRGB, DeviceCMYK
 - `Pixmap.colorsname` = `Colorspace.name`
 - `Pixmap.n` - `Pixmap.alpha` = `Colorspace.nbytes`

PyMuPDF Design Decision

Where PyMuPDF constructors require an alpha parameter, we assume `alpha = False` by default.

API Change: Display List

Constructor now requires the page's mediabox.

API Change: Text Page

Constructor now requires the page's mediabox.

API Change: Links

This contains significant changes:

- Link destinations objects are no longer maintained by MuPDF and information about destinations in general has been reduced. The only two variables containing such information now are `isExternal` (a bool) and `uri`, a string.
- The base class for PyMuPDF's `linkDest`, `fz_link_dest_s` has been deleted from MuPDF. In order to maintain backward compatibility, PyMuPDF provides an own `linkDest` class from available information as closely as possible.
- Document `outline` now additionally also contains `page`, `isExternal` and `uri` properties.
- The following shows MuPDF's behavior concerning links and how we interpret this in PyMuPDF.

<code>isExternal</code>	<code>uri</code>	Links	Outlines
True	Starts with <code>file://</code>	If <code>uri</code> ends with a page number (format: <code>#page=n</code>), generate a <code>goto</code> , else as a <code>launch</code> . Strip off prefix and suffix in any case.	Behave as described in Links column. Page must be <code>-1</code> .

This is line 0.

This is line 1.

This is line 2.

This is line 3.

This is line 4.