The l3pdf package
Core PDF support

The LaTeX Project*

Released 2020-01-29

1 l3pdf documentation

1.1 Objects

\pdf_object_new:nn
New: 2019-06-02

\pdf_object_new:nn {⟨*object*⟩} {⟨*type*⟩}

Declares ⟨*object*⟩ as a PDF object of ⟨*type*⟩, which should be one of

- `array`
- `dict`
- `fstream`
- `stream`

The object may be referenced from this point on, and written later using `\pdf_object_write:nn`.

\pdf_object_if_exist_p:n ⋆
\pdf_object_if_exist:nTF ⋆
New: 2020-05-15

\pdf_object_if_exist_p:n {⟨*object*⟩}
\pdf_object_if_exist:nTF {⟨*object*⟩}

Tests whether an object with name {⟨*object*⟩} has been defined.

\pdf_object_write:nn
\pdf_object_write:nx
New: 2019-06-02

\pdf_object_write:nn {⟨*object*⟩} {⟨*content*⟩}

Writes the ⟨*content*⟩ as content of the ⟨*object*⟩. Depending on the ⟨*type*⟩ declared for the object, the format required for the ⟨*data*⟩ will vary

`array` A space-separated list of values

`dict` Key–value pairs in the form `/`⟨*key*⟩ ⟨*value*⟩

`fstream` Two brace groups: ⟨*file name*⟩ and ⟨*file content*⟩

`stream` Two brace groups: ⟨*attributes (dictionary)*⟩ and ⟨*stream contents*⟩

*E-mail: latex-team@latex-project.org

\pdf_object_ref:n ⋆ `\pdf_object_ref:n {⟨object⟩}`

New: 2019-06-02

Inserts the appropriate information to reference the ⟨*object*⟩ in for example page resource allocation

\pdf_object_now:nn `\pdf_object_now:nn {⟨type⟩} {⟨content⟩}`
\pdf_object_now:nx

New: 2019-06-02

Writes the ⟨*content*⟩ as content of an anonymous object. Depending on the ⟨*type*⟩, the format required for the ⟨*data*⟩ will vary

`array` A space-separated list of values

`dict` Key–value pairs in the form `/`⟨`key`⟩ ⟨`value`⟩

`fstream` Two brace groups: ⟨*file name*⟩ and ⟨*file content*⟩

`stream` Two brace groups: ⟨*attributes (dictionary)*⟩ and ⟨*stream contents*⟩

\pdf_object_last: ⋆ `\pdf_object_last:`

New: 2019-06-02

Inserts the appropriate information to reference the last ⟨*object*⟩ created. This is particularly useful for anonymous objects.

1.2 Version

\pdf_version_compare_p:Nn ⋆ `\pdf_version_compare:NnTF ⟨comparator⟩ {⟨version⟩} {⟨true code⟩} {⟨false code⟩}`
\pdf_version_compare:Nn*TF* ⋆

New: 2019-06-02

Compares the version of the PDF being created with the ⟨*version*⟩ string specified, using the ⟨*comparator*⟩. Either the ⟨*true code*⟩ or ⟨*false code*⟩ will be left in the output stream.

\pdf_version_gset:n `\pdf_version_gset:n {⟨version⟩}`
\pdf_version_min_gset:n

New: 2019-06-02

Sets the ⟨*version*⟩ of the PDF being created. The `min` version will not alter the output version unless it is currently lower than the ⟨*version*⟩ requested.

This function may only be used up to the point where the PDF file is initialised.

\pdf_version: ⋆ `\pdf_version:`
\pdf_version_major: ⋆
\pdf_version_minor: ⋆

New: 2019-06-02

Expands to the currently-active PDF version.

1.3 Compression

\pdf_uncompress: `\pdf_uncompress:`

New: 2019-06-02

Disables any compression of the PDF, where possible.

This function may only be used up to the point where the PDF file is initialised.

1.4 Destinations

Destinations are the places a link jumped too. Unlike the name may suggest they don't described an exact location in the PDF. Instead a destination contains a reference to a page along with an instruction how to display this page. The normally used "XYZ *top left zoom*" for example instructs the viewer to show the page with the given *zoom* and the top left corner at the *top left* coordinates—which then gives the impression that there is an anchor at this position.

If an instruction takes a coordinate, it is calculated by the following commands relative to the location the command is issued. So to get a specific coordinate one has to move the command to the right place.

`\pdf_destination:nn` `\pdf_destination:nn {⟨name⟩} {⟨type or integer⟩}`

New: 2021-01-03

This creates a destination. `{⟨type or integer⟩}` can be one of `fit`, `fith`, `fitv`, `fitb`, `fitbh`, `fitbv`, `fitr`, `xyz` or an integer representing a scale factor in percent. `fitr` here gives only a lightweight version of `/FitR`: The backend code defines `fitr` so that it will with pdfLaTeX and LuaLaTeX use the coordinates of the surrounding box, with `dvips` and `dvipdfmx` it falls back to `fit`. For full control use `\pdf_destination:nnnn`.

The keywords match to the PDF names as described in the following tabular.

Keyword	PDF	Remarks
`fit`	`/Fit`	Fits the page to the window
`fith`	`/FitH` *top*	Fits the width of the page to the window
`fitv`	`/FitV` *left*	Fits the height of the page to the window
`fitb`	`/FitB`	Fits the page bounding box to the window
`fitbh`	`/FitBH` *top*	Fits the width of the page bounding box to the window.
`fitbv`	`/FitBV` *left*	Fits the height of the page bounding box to the window.
`fitr`	`/FitR` *left bottom right top*	Fits the rectangle specified by the four coordinates to the window (see above for the restrictions)
`xyz`	`/XYZ` *left top* null	Sets a coordinate but doesn't change the zoom.
`{`⟨*integer*⟩`}`	`/XYZ` *left top zoom*	Sets a coordinate and a zoom meaning `{`⟨*integer*⟩`}`%.

`\pdf_destination:nnnn` `\pdf_destination:nnnn {⟨name⟩} {⟨width⟩} {⟨height⟩} {⟨depth⟩}`

New: 2021-01-17

This creates a destination with `/FitR` type with the given dimensions relative to the current location. The destination is in a box of size zero, but it doesn't switch to horizontal mode.

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

P

pdf commands:

\pdf_destination:nn 3
\pdf_destination:nnnn 3, 3
\pdf_object_if_exist:nTF 1
\pdf_object_if_exist_p:n 1
\pdf_object_last: 2
\pdf_object_new:nn 1
\pdf_object_now:nn 2
\pdf_object_ref:n 2
\pdf_object_write:nn 1, 1
\pdf_uncompress: 2
\pdf_version: 2
\pdf_version_compare:NnTF 2
\pdf_version_compare_p:Nn 2
\pdf_version_gset:n 2
\pdf_version_major: 2
\pdf_version_min_gset:n 2
\pdf_version_minor: 2