

# File I

## Implementation

### 1 l3pdf implementation

```

1 <*initex | package>
2 <@@=pdf>
3 <*package>
4 \ProvidesExplPackage{l3pdf}{2020-06-03}{}
5 {L3 Experimental core PDF support}
6 </package>

```

\s\_\_pdf\_stop Internal scan marks.

```

7 \scan_new:N \s__pdf_stop

```

*(End definition for \s\_\_pdf\_stop.)*

\g\_\_pdf\_init\_bool A flag so we have some chance of avoiding setting things we are not allowed to.

```

8 \bool_new:N \g__pdf_init_bool
9 <*package>
10 \cs_if_exist:NT \documentclass
11 {
12     \AtBeginDocument
13     { \bool_gset_true:N \g__pdf_init_bool }
14 }
15 </package>

```

*(End definition for \g\_\_pdf\_init\_bool.)*

#### 1.1 Compression

\pdf\_uncompress: Simple to do.

```

16 \cs_new_protected:Npn \pdf_uncompress:
17 {
18     \bool_if:NF \g__pdf_init_bool
19     {
20         \__pdf_backend_compresslevel:n { 0 }
21         \__pdf_backend_compress_objects:n { \c_false_bool }
22     }
23 }

```

*(End definition for \pdf\_uncompress:. This function is documented on page ??.)*

#### 1.2 Objects

\pdf\_object\_new:nn Simple to do.

```

\pdf_object_write:nn 24 \cs_new_protected:Npn \pdf_object_new:nn #1#2
\pdf_object_write:nx 25 { \__pdf_backend_object_new:nn {#1} {#2} }
\pdf_object_if_exist_p:n 26 \prg_new_conditional:Npnn \pdf_object_if_exist:n #1 { p , T , F , TF }
\pdf_object_if_exist:nTF 27 {
\pdf_object_ref:n 28     \int_if_exist:cTF { c__pdf_backend_object_ \tl_to_str:n {#1} _int }
\pdf_object_now:nn
\pdf_object_now:nx
\pdf_object_last:

```

```

29     { \prg_return_true: }
30     { \prg_return_false:}
31 }
32 \cs_new_protected:Npn \pdf_object_write:nn #1#2
33 { \__pdf_backend_object_write:nn {#1} {#2} }
34 \cs_generate_variant:Nn \pdf_object_write:nn { nx }
35 \cs_new:Npn \pdf_object_ref:n #1 { \__pdf_backend_object_ref:n {#1} }
36 \cs_new_protected:Npn \pdf_object_now:nn #1#2
37 { \__pdf_backend_object_now:nn {#1} {#2} }
38 \cs_generate_variant:Nn \pdf_object_now:nn { nx }
39 \cs_new:Npn \pdf_object_last: { \__pdf_backend_object_last: }

```

(End definition for \pdf\_object\_new:nn and others. These functions are documented on page ??.)

### 1.3 Version

```

\pdf_version_compare:Nn
__pdf_version_compare_=:w 40 % To compare version, we need to split the given value then deal with both
__pdf_version_compare_<:w 41 % major and minor version
__pdf_version_compare_>:w 42 \prg_new_conditional:Npnn \pdf_version_compare:Nn #1#2 { p , T , F , TF }
43 { \use:c { __pdf_version_compare_ #1 :w } #2 . . \s__pdf_stop }
44 \cs_new:cpn { __pdf_version_compare_=:w } #1 . #2 . #3 \s__pdf_stop
45 {
46   \bool_lazy_and:nnTF
47   { \int_compare_p:nNn \__pdf_backend_version_major: = {#1} }
48   { \int_compare_p:nNn \__pdf_backend_version_minor: = {#2} }
49   { \prg_return_true: }
50   { \prg_return_false: }
51 }
52 \cs_new:cpn { __pdf_version_compare_<:w } #1 . #2 . #3 \s__pdf_stop
53 {
54   \bool_lazy_or:nnTF
55   { \int_compare_p:nNn \__pdf_backend_version_major: < {#1} }
56   {
57     \bool_lazy_and_p:nn
58     { \int_compare_p:nNn \__pdf_backend_version_major: = {#1} }
59     { \int_compare_p:nNn \__pdf_backend_version_minor: < {#2} }
60   }
61   { \prg_return_true: }
62   { \prg_return_false: }
63 }
64 \cs_new:cpn { __pdf_version_compare_>:w } #1 . #2 . #3 \s__pdf_stop
65 {
66   \bool_lazy_or:nnTF
67   { \int_compare_p:nNn \__pdf_backend_version_major: > {#1} }
68   {
69     \bool_lazy_and_p:nn
70     { \int_compare_p:nNn \__pdf_backend_version_major: = {#1} }
71     { \int_compare_p:nNn \__pdf_backend_version_minor: > {#2} }
72   }
73   { \prg_return_true: }
74   { \prg_return_false: }
75 }

```

(End definition for \pdf\_version\_compare:Nn and others. This function is documented on page ??.)

```

\pdf_version_gset:n Split the version and set.
\pdf_version_min_gset:n
\__pdf_version_gset:w
76 \cs_new_protected:Npn \pdf_version_gset:n #1
77 { \__pdf_version_gset:w #1 . . \s__pdf_stop }
78 \cs_new_protected:Npn \pdf_version_min_gset:n #1
79 {
80   \pdf_version_compare:NnT < {#1}
81   { \__pdf_version_gset:w #1 . . \s__pdf_stop }
82 }
83 \cs_new_protected:Npn \__pdf_version_gset:w #1 . #2 . #3\s__pdf_stop
84 {
85   \bool_if:NF \g__pdf_init_bool
86   {
87     \__pdf_backend_version_major_gset:n {#1}
88     \__pdf_backend_version_minor_gset:n {#2}
89   }
90 }

(End definition for \pdf_version_gset:n, \pdf_version_min_gset:n, and \__pdf_version_gset:w.
These functions are documented on page ??.)

\pdf_version: Wrappers.
\pdf_version_major:
\pdf_version_minor:
91 \cs_new:Npn \pdf_version:
92 { \__pdf_backend_version_major: . \__pdf_backend_version_minor: }
93 \cs_new:Npn \pdf_version_major: { \__pdf_backend_version_major: }
94 \cs_new:Npn \pdf_version_minor: { \__pdf_backend_version_minor: }

(End definition for \pdf_version:, \pdf_version_major:, and \pdf_version_minor:. These functions
are documented on page ??.)

95 </initex | package>

```

# 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.

<b>A</b>	
\AtBeginDocument	12
<b>B</b>	
bool commands:	
\bool_gset_true:N	13
\bool_if:NTF	18, 85
\bool_lazy_and:nnTF	46
\bool_lazy_and_p:nn	57, 69
\bool_lazy_or:nnTF	54, 66
\bool_new:N	8
\c_false_bool	21
<b>C</b>	
cs commands:	
\cs_generate_variant:Nn	34, 38
\cs_if_exist:NTF	10
\cs_new:Npn	35, 39, 44, 52, 64, 91, 93, 94
\cs_new_protected:Npn	16, 24, 32, 36, 76, 78, 83
<b>D</b>	
\documentclass	10
<b>I</b>	
int commands:	
\int_compare_p:nNn	47, 48, 55, 58, 59, 67, 70, 71
\int_if_exist:NTF	28
<b>P</b>	
pdf commands:	
\pdf_object_if_exist:nTF	24
\pdf_object_if_exist_p:n	24
\pdf_object_last:	24
\pdf_object_new:nn	24
\pdf_object_now:nn	24
\pdf_object_ref:n	24
\pdf_object_write:nn	24
\pdf_uncompress:	16
\pdf_version:	91
\pdf_version_compare:Nn	40
\pdf_version_compare:NnTF	80
\pdf_version_gset:n	76
\pdf_version_major:	91
\pdf_version_min_gset:n	76
\pdf_version_minor:	91
pdf internal commands:	
\__pdf_backend_compress_objects:n	21
\__pdf_backend_compresslevel:n	20
\__pdf_backend_object_last:	39
\__pdf_backend_object_new:nn	25
\__pdf_backend_object_now:nn	37
\__pdf_backend_object_ref:n	35
\__pdf_backend_object_write:nn	33
\__pdf_backend_version_major:	47, 55, 58, 67, 70, 92, 93
\__pdf_backend_version_major_gset:n	87
\__pdf_backend_version_minor:	48, 59, 71, 92, 94
\__pdf_backend_version_minor_gset:n	88
\g_pdf_init_bool	8, 18, 85
\__pdf_version_compare_<:w	40
\__pdf_version_compare_=:w	40
\__pdf_version_compare_>:w	40
\__pdf_version_gset:w	76
prg commands:	
\prg_new_conditional:Npnn	26, 42
\prg_return_false:	30, 50, 62, 74
\prg_return_true:	29, 49, 61, 73
\ProvidesExplPackage	4
<b>Q</b>	
quark internal commands:	
\s__pdf_stop	7, 43, 44, 52, 64, 77, 81, 83
<b>S</b>	
scan commands:	
\scan_new:N	7
<b>T</b>	
tl commands:	
\tl_to_str:n	28
<b>U</b>	
use commands:	
\use:N	43