

File I

Implementation

1 l3pdf implementation

```

1 <*package>
2 <@@=pdf>
3 \ProvidesExplPackage{l3pdf}{2020-08-07}{ }
4 {L3 Experimental core PDF support}

```

\s__pdf_stop Internal scan marks.

```

5 \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.

```

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

```

(End definition for \g__pdf_init_bool.)

1.1 Compression

\pdf_uncompress: Simple to do.

```

14 \cs_new_protected:Npn \pdf_uncompress:
15 {
16   \bool_if:NF \g__pdf_init_bool
17   {
18     \__pdf_backend_compresslevel:n { 0 }
19     \__pdf_backend_compress_objects:n { \c_false_bool }
20   }
21 }

```

(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 22 \cs_new_protected:Npn \pdf_object_new:nn #1#2
\pdf_object_write:nx 23 { \__pdf_backend_object_new:nn {#1} {#2} }
\pdf_object_if_exist_p:n 24 \prg_new_conditional:Npnn \pdf_object_if_exist:n #1 { p , T , F , TF }
\pdf_object_if_exist:nTF 25 {
\pdf_object_ref:n 26   \int_if_exist:cTF { c__pdf_backend_object_ \tl_to_str:n {#1} _int }
\pdf_object_now:nn 27   { \prg_return_true: }
\pdf_object_now:nx 28   { \prg_return_false: }
\pdf_object_last:

```

```

29 }
30 \cs_new_protected:Npn \pdf_object_write:nn #1#2
31 { \__pdf_backend_object_write:nn {#1} {#2} }
32 \cs_generate_variant:Nn \pdf_object_write:nn { nx }
33 \cs_new:Npn \pdf_object_ref:n #1 { \__pdf_backend_object_ref:n {#1} }
34 \cs_new_protected:Npn \pdf_object_now:nn #1#2
35 { \__pdf_backend_object_now:nn {#1} {#2} }
36 \cs_generate_variant:Nn \pdf_object_now:nn { nx }
37 \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 38 % To compare version, we need to split the given value then deal with both
__pdf_version_compare<:w 39 % major and minor version
__pdf_version_compare>:w 40 \prg_new_conditional:Npnn \pdf_version_compare:Nn #1#2 { p , T , F , TF }
41 { \use:c { __pdf_version_compare_ #1 :w } #2 . . \s__pdf_stop }
42 \cs_new:cpn { __pdf_version_compare=:w } #1 . #2 . #3 \s__pdf_stop
43 {
44   \bool_lazy_and:nnTF
45   { \int_compare_p:nNn \__pdf_backend_version_major: = {#1} }
46   { \int_compare_p:nNn \__pdf_backend_version_minor: = {#2} }
47   { \prg_return_true: }
48   { \prg_return_false: }
49 }
50 \cs_new:cpn { __pdf_version_compare<:w } #1 . #2 . #3 \s__pdf_stop
51 {
52   \bool_lazy_or:nnTF
53   { \int_compare_p:nNn \__pdf_backend_version_major: < {#1} }
54   {
55     \bool_lazy_and_p:nn
56     { \int_compare_p:nNn \__pdf_backend_version_major: = {#1} }
57     { \int_compare_p:nNn \__pdf_backend_version_minor: < {#2} }
58   }
59   { \prg_return_true: }
60   { \prg_return_false: }
61 }
62 \cs_new:cpn { __pdf_version_compare>:w } #1 . #2 . #3 \s__pdf_stop
63 {
64   \bool_lazy_or:nnTF
65   { \int_compare_p:nNn \__pdf_backend_version_major: > {#1} }
66   {
67     \bool_lazy_and_p:nn
68     { \int_compare_p:nNn \__pdf_backend_version_major: = {#1} }
69     { \int_compare_p:nNn \__pdf_backend_version_minor: > {#2} }
70   }
71   { \prg_return_true: }
72   { \prg_return_false: }
73 }

```

(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
74 \cs_new_protected:Npn \pdf_version_gset:n #1
75 { \__pdf_version_gset:w #1 . . \s__pdf_stop }
76 \cs_new_protected:Npn \pdf_version_min_gset:n #1
77 {
78   \pdf_version_compare:NnT < {#1}
79   { \__pdf_version_gset:w #1 . . \s__pdf_stop }
80 }
81 \cs_new_protected:Npn \__pdf_version_gset:w #1 . #2 . #3\s__pdf_stop
82 {
83   \bool_if:NF \g__pdf_init_bool
84   {
85     \__pdf_backend_version_major_gset:n {#1}
86     \__pdf_backend_version_minor_gset:n {#2}
87   }
88 }

(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:
89 \cs_new:Npn \pdf_version:
90 { \__pdf_backend_version_major: . \__pdf_backend_version_minor: }
91 \cs_new:Npn \pdf_version_major: { \__pdf_backend_version_major: }
92 \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 ??.)
93 \</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.

A	
<code>\AtBeginDocument</code>	10
B	
bool commands:	
<code>\bool_gset_true:N</code>	11
<code>\bool_if:NTF</code>	16, 83
<code>\bool_lazy_and:nnTF</code>	44
<code>\bool_lazy_and_p:nn</code>	55, 67
<code>\bool_lazy_or:nnTF</code>	52, 64
<code>\bool_new:N</code>	6
<code>\c_false_bool</code>	19
C	
cs commands:	
<code>\cs_generate_variant:Nn</code>	32, 36
<code>\cs_if_exist:NTF</code>	8
<code>\cs_new:Npn</code> 33, 37, 42, 50, 62, 89, 91, 92	
<code>\cs_new_protected:Npn</code>	14, 22, 30, 34, 74, 76, 81
D	
<code>\documentclass</code>	8
I	
int commands:	
<code>\int_compare_p:nNn</code>	45, 46, 53, 56, 57, 65, 68, 69
<code>\int_if_exist:NTF</code>	26
P	
pdf commands:	
<code>\pdf_object_if_exist:nTF</code>	22
<code>\pdf_object_if_exist_p:n</code>	22
<code>\pdf_object_last:</code>	22
<code>\pdf_object_new:nn</code>	22
<code>\pdf_object_now:nn</code>	22
<code>\pdf_object_ref:n</code>	22
<code>\pdf_object_write:nn</code>	22
<code>\pdf_uncompress:</code>	14
<code>\pdf_version:</code>	89
<code>\pdf_version_compare:Nn</code>	38
<code>\pdf_version_compare:NnTF</code>	78
<code>\pdf_version_gset:n</code>	74
<code>\pdf_version_major:</code>	89
<code>\pdf_version_min_gset:n</code>	74
<code>\pdf_version_minor:</code>	89
pdf internal commands:	
<code>__pdf_backend_compress_objects:n</code>	19
<code>__pdf_backend_compresslevel:n</code>	18
<code>__pdf_backend_object_last:</code>	37
<code>__pdf_backend_object_new:nn</code>	23
<code>__pdf_backend_object_now:nn</code>	35
<code>__pdf_backend_object_ref:n</code>	33
<code>__pdf_backend_object_write:nn</code>	31
<code>__pdf_backend_version_major:</code>	45, 53, 56, 65, 68, 90, 91
<code>__pdf_backend_version_major_gset:n</code>	85
<code>__pdf_backend_version_minor:</code>	46, 57, 69, 90, 92
<code>__pdf_backend_version_minor_gset:n</code>	86
<code>\g_pdf_init_bool</code>	6, 16, 83
<code>__pdf_version_compare_<:w</code>	38
<code>__pdf_version_compare_=:w</code>	38
<code>__pdf_version_compare_>:w</code>	38
<code>__pdf_version_gset:w</code>	74
prg commands:	
<code>\prg_new_conditional:Npnn</code>	24, 40
<code>\prg_return_false:</code>	28, 48, 60, 72
<code>\prg_return_true:</code>	27, 47, 59, 71
<code>\ProvidesExplPackage</code>	3
Q	
quark internal commands:	
<code>\s__pdf_stop</code> 5, 41, 42, 50, 62, 75, 79, 81	
S	
scan commands:	
<code>\scan_new:N</code>	5
T	
tl commands:	
<code>\tl_to_str:n</code>	26
U	
use commands:	
<code>\use:N</code>	41