

1	%%	1
2	%% This is file `expl3-code.tex',	2
3	%% generated with the docstrip utility.	3
4	%%	4
5	%% The original source files were:	5
6	%%	6
7	%% expl3.dtx (with options: `code')	7
8	%% l3bootstrap.dtx (with options: `code')	8
9	%% l3names.dtx (with options: `code')	9
10	%% l3basics.dtx (with options: `code')	10
11	%% l3expan.dtx (with options: `code')	11
12	%% l3quark.dtx (with options: `code')	12
13	%% l3tl.dtx (with options: `code')	13
14	%% l3tl-build.dtx (with options: `code')	14
15	%% l3str.dtx (with options: `code')	15
16	%% l3seq.dtx (with options: `code')	16
17	%% l3int.dtx (with options: `code')	17
18	%% l3flag.dtx (with options: `code')	18
19	%% l3prg.dtx (with options: `code')	19
20	%% l3sys.dtx (with options: `code')	20
21	%% l3clist.dtx (with options: `code')	21
22	%% l3token.dtx (with options: `code')	22
23	%% l3prop.dtx (with options: `code')	23
24	%% l3msg.dtx (with options: `code')	24
25	%% l3file.dtx (with options: `code')	25
26	%% l3skip.dtx (with options: `code')	26
27	%% l3keys.dtx (with options: `code')	27
28	%% l3intarray.dtx (with options: `code')	28
29	%% l3fp.dtx (with options: `code')	29
30	%% l3fp-aux.dtx (with options: `code')	30
31	%% l3fp-traps.dtx (with options: `code')	31
32	%% l3fp-round.dtx (with options: `code')	32
33	%% l3fp-parse.dtx (with options: `code')	33
34	%% l3fp-assign.dtx (with options: `code')	34
35	%% l3fp-logic.dtx (with options: `code')	35
36	%% l3fp-basics.dtx (with options: `code')	36
37	%% l3fp-extended.dtx (with options: `code')	37
38	%% l3fp-expo.dtx (with options: `code')	38
39	%% l3fp-trig.dtx (with options: `code')	39
40	%% l3fp-convert.dtx (with options: `code')	40
41	%% l3fp-random.dtx (with options: `code')	41
42	%% l3fp-types.dtx (with options: `code')	42
43	%% l3fp-symbolic.dtx (with options: `code')	43
44	%% l3fp-functions.dtx (with options: `code')	44
45	%% l3fparray.dtx (with options: `code')	45
46	%% l3bitset.dtx (with options: `code')	46

47	%% l3cctab.dtx (with options: `code')	47
48	%% l3sort.dtx (with options: `code')	48
49	%% l3str-convert.dtx (with options: `code')	49
50	%% l3tl-analysis.dtx (with options: `code')	50
51	%% l3benchmark.dtx (with options: `code')	51
52	%% l3regex.dtx (with options: `code')	52
53	%% l3box.dtx (with options: `code')	53
54	%% l3color.dtx (with options: `code')	54
55	%% l3graphics.dtx (with options: `code')	55
56	%% l3opacity.dtx (with options: `code')	56
57	%% l3pdf.dtx (with options: `code')	57
58	%% l3coffins.dtx (with options: `code')	58
59	%% l3luatex.dtx (with options: `code')	59
60	%% l3unicode.dtx (with options: `code')	60
61	%% l3text.dtx (with options: `code')	61
62	%% l3text-case.dtx (with options: `code')	62
63	%% l3text-map.dtx (with options: `code')	63
64	%% l3text-purify.dtx (with options: `code')	64
65	%% l3legacy.dtx (with options: `code')	65
66	%% l3deprecation.dtx (with options: `code')	66
67	%%	67
68	%% Copyright (C) 1990-2025 The LaTeX Project	68
69	%%	69
70	%% It may be distributed and/or modified under the conditions of	70
71	%% the LaTeX Project Public License (LPPL), either version 1.3c of	71
72	%% this license or (at your option) any later version. The latest	72
73	%% version of this license is in the file:	73
74	%%	74
75	%% https://www.latex-project.org/lppl.txt	75
76	%%	76
77	%% This file is part of the "l3kernel bundle" (The Work in LPPL)	77
78	%% and all files in that bundle must be distributed together.	78
79	%%	79
80	%% File: expl3.dtx	80
81	<u>\def\ExplFileDate{2025-04-29}%</u>	81
82	<u>\beginingroup</u>	82
83	<u>\def\next{\endgroup}%</u>	83
84	<u>\expandafter\ifx\csname PackageError\endcsname\relax</u>	84
85	<u>\beginingroup</u>	85
86	<u>\def\next{\endgroup\endgroup}%</u>	86
87	<u>\def\PackageError#1#2#3%</u>	87
88	<u>{%</u>	88
89	<u>\endgroup</u>	89
90	<u>\errhelp{#3}%</u>	90
91	<u>\errmessage{#1 Error: #2!}%</u>	91
92	<u>}%</u>	92

93	<u>\fi</u>	93
94	<u>\expandafter\ifx\csname</u> ExplLoaderFileDate <u>\endcsname\relax</u>	94
95	<u>\def\next</u>	95
96	{%	96
97	\PackageError{expl3}{No expl3 loader detected}	97
98	{%	98
99	You have attempted to use the expl3 code directly rather than using	99
100	the correct loader. Loading of expl3 will abort.	100
101	}%	101
102	<u>\endgroup</u>	102
103	<u>\endinput</u>	103
104	}	104
105	<u>\else</u>	105
106	<u>\ifx\ExplLoaderFileDate\ExplFileDate</u>	106
107	<u>\else</u>	107
108	<u>\def\next</u>	108
109	{%	109
110	\PackageError{expl3}{Mismatched expl3 files detected}	110
111	{%	111
112	You have attempted to load expl3 with mismatched files:	112
113	probably you have one or more files 'locally installed' which	113
114	are in conflict. Loading of expl3 will abort.	114
115	}%	115
116	<u>\endgroup</u>	116
117	<u>\endinput</u>	117
118	}%	118
119	<u>\fi</u>	119
120	<u>\fi</u>	120
121	<u>\next</u>	121
122	<u>\beginingroup\expandafter\expandafter\expandafter\endgroup</u>	122
123	<u>\expandafter\ifx\csname</u> ver@expl3-code.tex <u>\endcsname\relax</u>	123
124	<u>\expandafter\edef\csname</u> ver@expl3-code.tex <u>\endcsname</u>	124
125	{%	125
126	\ExplFileDate\space	126
127	L3 programming layer	127
128	}%	128
129	<u>\else</u>	129
130	<u>\expandafter\endinput</u>	130
131	<u>\fi</u>	131
132	<u>\immediate\write</u> -1 %	132
133	{%	133
134	Package: expl3	134
135	\ExplFileDate\space	135
136	L3 programming layer (code)%	136
137	}%	137
138	%% File: l3bootstrap.dtx	138

139	<u>\beginngroup\expandafter\expandafter\expandafter\endgroup</u>	139
140	<u>\expandafter\ifx\csname</u> pdfstrcmp <u>\endcsname\relax</u>	140
141	<u>\let\pdfstrcmp\strcmp</u>	141
142	<u>\fi</u>	142
143	<u>\beginngroup\expandafter\expandafter\expandafter\endgroup</u>	143
144	<u>\expandafter\ifx\csname</u> directlua <u>\endcsname\relax</u>	144
145	<u>\else</u>	145
146	<u>\ifnum\luatexversion</u> <110 %	146
147	<u>\else</u>	147
148	<u>\beginngroup\expandafter\expandafter\expandafter\endgroup</u>	148
149	<u>\expandafter\ifx\csname</u> newcatcodetable <u>\endcsname\relax</u>	149
150	<u>\input{ltluatex}%</u>	150
151	<u>\fi</u>	151
152	<u>\beginngroup\expandafter\expandafter\expandafter\endgroup</u>	152
153	<u>\expandafter\ifx\csname</u> newluabytecode <u>\endcsname\relax</u>	153
154	<u>\else</u>	154
155	<u>\newluabytecode\@expl@luadata@bytecode</u>	155
156	<u>\fi</u>	156
157	<u>\directlua{require("expl3")}%</u>	157
158	<u>\ifnum</u> 0%	158
159	<u>\directlua{</u>	159
160	if status.ini_version then	160
161	tex.write("1")	161
162	end	162
163	<u>}>0 %</u>	163
164	<u>\everyjob\expandafter{%</u>	164
165	<u>\the\expandafter\everyjob</u>	165
166	<u>\csname\detokenize{lua_now:n}\endcsname{require("expl3")}%</u>	166
167	<u>}%</u>	167
168	<u>\fi</u>	168
169	<u>\fi</u>	169
170	<u>\fi</u>	170
171	<u>\beginngroup</u>	171
172	<u>\def\next{\endgroup}%</u>	172
173	<u>\def\ShortText{Required primitives not found}%</u>	173
174	<u>\def\LongText%</u>	174
175	<u>{%</u>	175
176	The L3 programming layer requires the e-TeX primitives and the	176
177	<u>\LineBreak</u> 'pdfTeX utilities' as described in the README file.	177
178	<u>\LineBreak</u>	178
179	These are available in the engines <u>\LineBreak</u>	179
180	- pdfTeX v1.40.20 <u>\LineBreak</u>	180
181	- XeTeX v0.999991 <u>\LineBreak</u>	181
182	- LuaTeX v1.10 <u>\LineBreak</u>	182
183	- e-(u)pTeX v3.8.2 <u>\LineBreak</u>	183
184	- Prote (2021) <u>\LineBreak</u>	184

```

185 or later.\LineBreak
186 \LineBreak
187 }%
188 \ifnum0%
189 \expandafter\ifx\csname luatexversion\endcsname\relax
190 \expandafter\ifx\csname expanded\endcsname\relax\else 1\fi
191 \else
192 \ifnum\luatexversion<110 \else 1\fi
193 \fi
194 =0 %
195 \newlinechar`^^J %
196 \def\LineBreak{\noexpand\MessageBreak}%
197 \expandafter\ifx\csname PackageError\endcsname\relax
198 \def\LineBreak{^^J}%
199 \begingroup
200 \lccode`~=`\lccode`~}%
201 \lccode`T=`T\lccode`H=`H%
202 \catcode`\_ =11 %
203 \lowercase{\endgroup\def\PackageError#1#2#3{%
204 \begingroup\errorcontextlines-1\immediate\write0{\errhelp{#3}\def%
205 \_ {#1 Error: #2.^^J^^J
206 Type H <return> for immediate help}\def~{\errmessage{%
207 \_ }}~\endgroup}}%
208 \fi
209 \edef\next
210 {%
211 \noexpand\PackageError{expl3}{\ShortText}
212 {\LongText Loading of expl3 will abort!}%
213 \endgroup
214 \noexpand\endinput
215 }%
216 \fi
217 \next
218 \protected\edef\ExplSyntaxOff
219 {%
220 \protected\def\noexpand\ExplSyntaxOff{}%
221 \catcode 9 = \the\catcode 9\relax
222 \catcode 32 = \the\catcode 32\relax
223 \catcode 34 = \the\catcode 34\relax
224 \catcode 58 = \the\catcode 58\relax
225 \catcode 94 = \the\catcode 94\relax
226 \catcode 95 = \the\catcode 95\relax
227 \catcode 124 = \the\catcode 124\relax
228 \catcode 126 = \the\catcode 126\relax
229 \endlinechar = \the\endlinechar\relax
230 \chardef\csname\detokenize{1\_kernel\_expl\_bool}\endcsname = 0\relax

```

231	}%	231
232	<u>\catcode</u> 9 = 9 <u>\relax</u>	232
233	<u>\catcode</u> 32 = 9 <u>\relax</u>	233
234	<u>\catcode</u> 34 = 12 <u>\relax</u>	234
235	<u>\catcode</u> 58 = 11 <u>\relax</u>	235
236	<u>\catcode</u> 94 = 7 <u>\relax</u>	236
237	<u>\catcode</u> 95 = 11 <u>\relax</u>	237
238	<u>\catcode</u> 124 = 12 <u>\relax</u>	238
239	<u>\catcode</u> 126 = 10 <u>\relax</u>	239
240	<u>\endlinechar</u> = 32 <u>\relax</u>	240
241	<u>\global\chardef</u> \l__kernel_expl_bool = 1 <u>\relax</u>	241
242	<u>\protected \def</u> \ExplSyntaxOn	242
243	{	243
244	\bool_if:NF \l__kernel_expl_bool	244
245	{	245
246	\cs_set_protected:Npe \ExplSyntaxOff	246
247	{	247
248	\char_set_catcode:nn { 9 } { \char_value_catcode:n { 9 } }	248
249	\char_set_catcode:nn { 32 } { \char_value_catcode:n { 32 } }	249
250	\char_set_catcode:nn { 34 } { \char_value_catcode:n { 34 } }	250
251	\char_set_catcode:nn { 58 } { \char_value_catcode:n { 58 } }	251
252	\char_set_catcode:nn { 94 } { \char_value_catcode:n { 94 } }	252
253	\char_set_catcode:nn { 95 } { \char_value_catcode:n { 95 } }	253
254	\char_set_catcode:nn { 124 } { \char_value_catcode:n { 124 } }	254
255	\char_set_catcode:nn { 126 } { \char_value_catcode:n { 126 } }	255
256	\tex_endlinechar:D =	256
257	\tex_the:D \tex_endlinechar:D \scan_stop:	257
258	\bool_set_false:N \l__kernel_expl_bool	258
259	\cs_set_protected:Npn \ExplSyntaxOff { }	259
260	}	260
261	}	261
262	\char_set_catcode_ignore:n { 9 } % tab	262
263	\char_set_catcode_ignore:n { 32 } % space	263
264	\char_set_catcode_other:n { 34 } % double quote	264
265	\char_set_catcode_letter:n { 58 } % colon	265
266	\char_set_catcode_math_superscript:n { 94 } % circumflex	266
267	\char_set_catcode_letter:n { 95 } % underscore	267
268	\char_set_catcode_other:n { 124 } % pipe	268
269	\char_set_catcode_space:n { 126 } % tilde	269
270	\tex_endlinechar:D = 32 \scan_stop:	270
271	\bool_set_true:N \l__kernel_expl_bool	271
272	}	272
273	%% File: l3names.dtx	273
274	<u>\let</u> \tex_global:D \global	274
275	<u>\let</u> \tex_let:D \let	275
276	<u>\begingroup</u>	276

277	<code>\long \def _kernel_primitive:NN #1#2</code>	277
278	<code>{ \tex_global:D \tex_let:D #2 #1 }</code>	278
279	<code>_kernel_primitive:NN \</code>	<code>\tex_space:D</code> 279
280	<code>_kernel_primitive:NN \/</code>	<code>\tex_italiccorrection:D</code> 280
281	<code>_kernel_primitive:NN \-</code>	<code>\tex_hyphen:D</code> 281
282	<code>_kernel_primitive:NN \above</code>	<code>\tex_above:D</code> 282
283	<code>_kernel_primitive:NN \abovedisplayshortskip</code>	<code>\tex_abovedisplayshortskip:D</code> 283
284	<code>_kernel_primitive:NN \abovedisplayskip</code>	<code>\tex_abovedisplayskip:D</code> 284
285	<code>_kernel_primitive:NN \abovewithdelims</code>	<code>\tex_abovewithdelims:D</code> 285
286	<code>_kernel_primitive:NN \accent</code>	<code>\tex_accent:D</code> 286
287	<code>_kernel_primitive:NN \adjdemerits</code>	<code>\tex_adjdemerits:D</code> 287
288	<code>_kernel_primitive:NN \advance</code>	<code>\tex_advance:D</code> 288
289	<code>_kernel_primitive:NN \afterassignment</code>	<code>\tex_afterassignment:D</code> 289
290	<code>_kernel_primitive:NN \aftergroup</code>	<code>\tex_aftergroup:D</code> 290
291	<code>_kernel_primitive:NN \atop</code>	<code>\tex_atop:D</code> 291
292	<code>_kernel_primitive:NN \atopwithdelims</code>	<code>\tex_atopwithdelims:D</code> 292
293	<code>_kernel_primitive:NN \badness</code>	<code>\tex_badness:D</code> 293
294	<code>_kernel_primitive:NN \baselineskip</code>	<code>\tex_baselineskip:D</code> 294
295	<code>_kernel_primitive:NN \batchmode</code>	<code>\tex_batchmode:D</code> 295
296	<code>_kernel_primitive:NN \begingroup</code>	<code>\tex_begingroup:D</code> 296
297	<code>_kernel_primitive:NN \belowdisplayshortskip</code>	<code>\tex_belowdisplayshortskip:D</code> 297
298	<code>_kernel_primitive:NN \belowdisplayskip</code>	<code>\tex_belowdisplayskip:D</code> 298
299	<code>_kernel_primitive:NN \binoppenalty</code>	<code>\tex_binoppenalty:D</code> 299
300	<code>_kernel_primitive:NN \botmark</code>	<code>\tex_botmark:D</code> 300
301	<code>_kernel_primitive:NN \box</code>	<code>\tex_box:D</code> 301
302	<code>_kernel_primitive:NN \boxmaxdepth</code>	<code>\tex_boxmaxdepth:D</code> 302
303	<code>_kernel_primitive:NN \brokenpenalty</code>	<code>\tex_brokenpenalty:D</code> 303
304	<code>_kernel_primitive:NN \catcode</code>	<code>\tex_catcode:D</code> 304
305	<code>_kernel_primitive:NN \char</code>	<code>\tex_char:D</code> 305
306	<code>_kernel_primitive:NN \chardef</code>	<code>\tex_chardef:D</code> 306
307	<code>_kernel_primitive:NN \cleaders</code>	<code>\tex_cleaders:D</code> 307
308	<code>_kernel_primitive:NN \closein</code>	<code>\tex_closein:D</code> 308
309	<code>_kernel_primitive:NN \closeout</code>	<code>\tex_closeout:D</code> 309
310	<code>_kernel_primitive:NN \clubpenalty</code>	<code>\tex_clubpenalty:D</code> 310
311	<code>_kernel_primitive:NN \copy</code>	<code>\tex_copy:D</code> 311
312	<code>_kernel_primitive:NN \count</code>	<code>\tex_count:D</code> 312
313	<code>_kernel_primitive:NN \countdef</code>	<code>\tex_countdef:D</code> 313
314	<code>_kernel_primitive:NN \cr</code>	<code>\tex_cr:D</code> 314
315	<code>_kernel_primitive:NN \crcr</code>	<code>\tex_crcr:D</code> 315
316	<code>_kernel_primitive:NN \csname</code>	<code>\tex_csname:D</code> 316
317	<code>_kernel_primitive:NN \day</code>	<code>\tex_day:D</code> 317
318	<code>_kernel_primitive:NN \deadcycles</code>	<code>\tex_deadcycles:D</code> 318
319	<code>_kernel_primitive:NN \def</code>	<code>\tex_def:D</code> 319
320	<code>_kernel_primitive:NN \defaultthyphenchar</code>	<code>\tex_defaultthyphenchar:D</code> 320
321	<code>_kernel_primitive:NN \defaultskewchar</code>	<code>\tex_defaultskewchar:D</code> 321
322	<code>_kernel_primitive:NN \delcode</code>	<code>\tex_delcode:D</code> 322

323	_kernel_primitive:NN	<u>\delimiter</u>	\tex_delimiter:D	323
324	_kernel_primitive:NN	<u>\delimiterfactor</u>	\tex_delimiterfactor:D	324
325	_kernel_primitive:NN	<u>\delimitershortfall</u>	\tex_delimitershortfall:D	325
326	_kernel_primitive:NN	<u>\dimen</u>	\tex_dimen:D	326
327	_kernel_primitive:NN	<u>\dimendef</u>	\tex_dimendef:D	327
328	_kernel_primitive:NN	<u>\discretionary</u>	\tex_discretionary:D	328
329	_kernel_primitive:NN	<u>\displayindent</u>	\tex_displayindent:D	329
330	_kernel_primitive:NN	<u>\displaylimits</u>	\tex_displaylimits:D	330
331	_kernel_primitive:NN	<u>\displaystyle</u>	\tex_displaystyle:D	331
332	_kernel_primitive:NN	<u>\displaywidowpenalty</u>	\tex_displaywidowpenalty:D	332
333	_kernel_primitive:NN	<u>\displaywidth</u>	\tex_displaywidth:D	333
334	_kernel_primitive:NN	<u>\divide</u>	\tex_divide:D	334
335	_kernel_primitive:NN	<u>\doublehyphendemerits</u>	\tex_doublehyphendemerits:D	335
336	_kernel_primitive:NN	<u>\dp</u>	\tex_dp:D	336
337	_kernel_primitive:NN	<u>\dump</u>	\tex_dump:D	337
338	_kernel_primitive:NN	<u>\edef</u>	\tex_edef:D	338
339	_kernel_primitive:NN	<u>\else</u>	\tex_else:D	339
340	_kernel_primitive:NN	<u>\emergencystretch</u>	\tex_emergencystretch:D	340
341	_kernel_primitive:NN	<u>\end</u>	\tex_end:D	341
342	_kernel_primitive:NN	<u>\endcsname</u>	\tex_endcsname:D	342
343	_kernel_primitive:NN	<u>\endgroup</u>	\tex_endgroup:D	343
344	_kernel_primitive:NN	<u>\endinput</u>	\tex_endinput:D	344
345	_kernel_primitive:NN	<u>\endlinechar</u>	\tex_endlinechar:D	345
346	_kernel_primitive:NN	<u>\eqno</u>	\tex_eqno:D	346
347	_kernel_primitive:NN	<u>\errhelp</u>	\tex_errhelp:D	347
348	_kernel_primitive:NN	<u>\errmessage</u>	\tex_errmessage:D	348
349	_kernel_primitive:NN	<u>\errorcontextlines</u>	\tex_errorcontextlines:D	349
350	_kernel_primitive:NN	<u>\errorstopmode</u>	\tex_errorstopmode:D	350
351	_kernel_primitive:NN	<u>\escapechar</u>	\tex_escapechar:D	351
352	_kernel_primitive:NN	<u>\everycr</u>	\tex_everycr:D	352
353	_kernel_primitive:NN	<u>\everydisplay</u>	\tex_everydisplay:D	353
354	_kernel_primitive:NN	<u>\everyhbox</u>	\tex_everyhbox:D	354
355	_kernel_primitive:NN	<u>\everyjob</u>	\tex_everyjob:D	355
356	_kernel_primitive:NN	<u>\everymath</u>	\tex_everymath:D	356
357	_kernel_primitive:NN	<u>\everypar</u>	\tex_everypar:D	357
358	_kernel_primitive:NN	<u>\everyvbox</u>	\tex_everyvbox:D	358
359	_kernel_primitive:NN	<u>\exhyphenpenalty</u>	\tex_exhyphenpenalty:D	359
360	_kernel_primitive:NN	<u>\expandafter</u>	\tex_expandafter:D	360
361	_kernel_primitive:NN	<u>\fam</u>	\tex_fam:D	361
362	_kernel_primitive:NN	<u>\fi</u>	\tex_fi:D	362
363	_kernel_primitive:NN	<u>\finalhyphendemerits</u>	\tex_finalhyphendemerits:D	363
364	_kernel_primitive:NN	<u>\firstmark</u>	\tex_firstmark:D	364
365	_kernel_primitive:NN	<u>\floatingpenalty</u>	\tex_floatingpenalty:D	365
366	_kernel_primitive:NN	<u>\font</u>	\tex_font:D	366
367	_kernel_primitive:NN	<u>\fontdimen</u>	\tex_fontdimen:D	367
368	_kernel_primitive:NN	<u>\fontname</u>	\tex_fontname:D	368

369	_kernel_primitive:NN	<u>\futurelet</u>	\tex_futurelet:D	369
370	_kernel_primitive:NN	<u>\gdef</u>	\tex_gdef:D	370
371	_kernel_primitive:NN	<u>\global</u>	\tex_global:D	371
372	_kernel_primitive:NN	<u>\globaldefs</u>	\tex_globaldefs:D	372
373	_kernel_primitive:NN	<u>\halign</u>	\tex_halign:D	373
374	_kernel_primitive:NN	<u>\hangafter</u>	\tex_hangafter:D	374
375	_kernel_primitive:NN	<u>\hangindent</u>	\tex_hangindent:D	375
376	_kernel_primitive:NN	<u>\hbadness</u>	\tex_hbadness:D	376
377	_kernel_primitive:NN	<u>\hbox</u>	\tex_hbox:D	377
378	_kernel_primitive:NN	<u>\hfil</u>	\tex_hfil:D	378
379	_kernel_primitive:NN	<u>\hfill</u>	\tex_hfill:D	379
380	_kernel_primitive:NN	<u>\hfilneg</u>	\tex_hfilneg:D	380
381	_kernel_primitive:NN	<u>\hfuzz</u>	\tex_hfuzz:D	381
382	_kernel_primitive:NN	<u>\hoffset</u>	\tex_hoffset:D	382
383	_kernel_primitive:NN	<u>\holdinginserts</u>	\tex_holdinginserts:D	383
384	_kernel_primitive:NN	<u>\hrule</u>	\tex_hrule:D	384
385	_kernel_primitive:NN	<u>\hsize</u>	\tex_hsize:D	385
386	_kernel_primitive:NN	<u>\hskip</u>	\tex_hskip:D	386
387	_kernel_primitive:NN	<u>\hss</u>	\tex_hss:D	387
388	_kernel_primitive:NN	<u>\ht</u>	\tex_ht:D	388
389	_kernel_primitive:NN	<u>\hyphenation</u>	\tex_hyphenation:D	389
390	_kernel_primitive:NN	<u>\hyphenchar</u>	\tex_hyphenchar:D	390
391	_kernel_primitive:NN	<u>\hyphenpenalty</u>	\tex_hyphenpenalty:D	391
392	_kernel_primitive:NN	<u>\if</u>	\tex_if:D	392
393	_kernel_primitive:NN	<u>\ifcase</u>	\tex_ifcase:D	393
394	_kernel_primitive:NN	<u>\ifcat</u>	\tex_ifcat:D	394
395	_kernel_primitive:NN	<u>\ifdim</u>	\tex_ifdim:D	395
396	_kernel_primitive:NN	<u>\ifeof</u>	\tex_ifeof:D	396
397	_kernel_primitive:NN	<u>\iffalse</u>	\tex_iffalse:D	397
398	_kernel_primitive:NN	<u>\ifhbox</u>	\tex_ifhbox:D	398
399	_kernel_primitive:NN	<u>\ifhmode</u>	\tex_ifhmode:D	399
400	_kernel_primitive:NN	<u>\ifinner</u>	\tex_ifinner:D	400
401	_kernel_primitive:NN	<u>\ifmmode</u>	\tex_ifmmode:D	401
402	_kernel_primitive:NN	<u>\ifnum</u>	\tex_ifnum:D	402
403	_kernel_primitive:NN	<u>\ifodd</u>	\tex_ifodd:D	403
404	_kernel_primitive:NN	<u>\iftrue</u>	\tex_iftrue:D	404
405	_kernel_primitive:NN	<u>\ifvbox</u>	\tex_ifvbox:D	405
406	_kernel_primitive:NN	<u>\ifvmode</u>	\tex_ifvmode:D	406
407	_kernel_primitive:NN	<u>\ifvoid</u>	\tex_ifvoid:D	407
408	_kernel_primitive:NN	<u>\ifx</u>	\tex_ifx:D	408
409	_kernel_primitive:NN	<u>\ignorespaces</u>	\tex_ignorespaces:D	409
410	_kernel_primitive:NN	<u>\immediate</u>	\tex_immediate:D	410
411	_kernel_primitive:NN	<u>\indent</u>	\tex_indent:D	411
412	_kernel_primitive:NN	<u>\input</u>	\tex_input:D	412
413	_kernel_primitive:NN	<u>\inputlineno</u>	\tex_inputlineno:D	413
414	_kernel_primitive:NN	<u>\insert</u>	\tex_insert:D	414

415	<code>__kernel_primitive:NN</code>	<code>\insertpenalties</code>	<code>\tex_insertpenalties:D</code>	415
416	<code>__kernel_primitive:NN</code>	<code>\interlinepenalty</code>	<code>\tex_interlinepenalty:D</code>	416
417	<code>__kernel_primitive:NN</code>	<code>\jobname</code>	<code>\tex_jobname:D</code>	417
418	<code>__kernel_primitive:NN</code>	<code>\kern</code>	<code>\tex_kern:D</code>	418
419	<code>__kernel_primitive:NN</code>	<code>\language</code>	<code>\tex_language:D</code>	419
420	<code>__kernel_primitive:NN</code>	<code>\lastbox</code>	<code>\tex_lastbox:D</code>	420
421	<code>__kernel_primitive:NN</code>	<code>\lastkern</code>	<code>\tex_lastkern:D</code>	421
422	<code>__kernel_primitive:NN</code>	<code>\lastpenalty</code>	<code>\tex_lastpenalty:D</code>	422
423	<code>__kernel_primitive:NN</code>	<code>\lastskip</code>	<code>\tex_lastskip:D</code>	423
424	<code>__kernel_primitive:NN</code>	<code>\lccode</code>	<code>\tex_lccode:D</code>	424
425	<code>__kernel_primitive:NN</code>	<code>\leaders</code>	<code>\tex_leaders:D</code>	425
426	<code>__kernel_primitive:NN</code>	<code>\left</code>	<code>\tex_left:D</code>	426
427	<code>__kernel_primitive:NN</code>	<code>\lefthyphenmin</code>	<code>\tex_lefthyphenmin:D</code>	427
428	<code>__kernel_primitive:NN</code>	<code>\leftskip</code>	<code>\tex_leftskip:D</code>	428
429	<code>__kernel_primitive:NN</code>	<code>\leqno</code>	<code>\tex_leqno:D</code>	429
430	<code>__kernel_primitive:NN</code>	<code>\let</code>	<code>\tex_let:D</code>	430
431	<code>__kernel_primitive:NN</code>	<code>\limits</code>	<code>\tex_limits:D</code>	431
432	<code>__kernel_primitive:NN</code>	<code>\linepenalty</code>	<code>\tex_linepenalty:D</code>	432
433	<code>__kernel_primitive:NN</code>	<code>\lineskip</code>	<code>\tex_lineskip:D</code>	433
434	<code>__kernel_primitive:NN</code>	<code>\lineskiplimit</code>	<code>\tex_lineskiplimit:D</code>	434
435	<code>__kernel_primitive:NN</code>	<code>\long</code>	<code>\tex_long:D</code>	435
436	<code>__kernel_primitive:NN</code>	<code>\looseness</code>	<code>\tex_looseness:D</code>	436
437	<code>__kernel_primitive:NN</code>	<code>\lower</code>	<code>\tex_lower:D</code>	437
438	<code>__kernel_primitive:NN</code>	<code>\lowercase</code>	<code>\tex_lowercase:D</code>	438
439	<code>__kernel_primitive:NN</code>	<code>\mag</code>	<code>\tex_mag:D</code>	439
440	<code>__kernel_primitive:NN</code>	<code>\mark</code>	<code>\tex_mark:D</code>	440
441	<code>__kernel_primitive:NN</code>	<code>\mathaccent</code>	<code>\tex_mathaccent:D</code>	441
442	<code>__kernel_primitive:NN</code>	<code>\mathbin</code>	<code>\tex_mathbin:D</code>	442
443	<code>__kernel_primitive:NN</code>	<code>\mathchar</code>	<code>\tex_mathchar:D</code>	443
444	<code>__kernel_primitive:NN</code>	<code>\mathchardef</code>	<code>\tex_mathchardef:D</code>	444
445	<code>__kernel_primitive:NN</code>	<code>\mathchoice</code>	<code>\tex_mathchoice:D</code>	445
446	<code>__kernel_primitive:NN</code>	<code>\mathclose</code>	<code>\tex_mathclose:D</code>	446
447	<code>__kernel_primitive:NN</code>	<code>\mathcode</code>	<code>\tex_mathcode:D</code>	447
448	<code>__kernel_primitive:NN</code>	<code>\mathinner</code>	<code>\tex_mathinner:D</code>	448
449	<code>__kernel_primitive:NN</code>	<code>\mathop</code>	<code>\tex_mathop:D</code>	449
450	<code>__kernel_primitive:NN</code>	<code>\mathopen</code>	<code>\tex_mathopen:D</code>	450
451	<code>__kernel_primitive:NN</code>	<code>\mathord</code>	<code>\tex_mathord:D</code>	451
452	<code>__kernel_primitive:NN</code>	<code>\mathpunct</code>	<code>\tex_mathpunct:D</code>	452
453	<code>__kernel_primitive:NN</code>	<code>\mathrel</code>	<code>\tex_mathrel:D</code>	453
454	<code>__kernel_primitive:NN</code>	<code>\mathsurround</code>	<code>\tex_mathsurround:D</code>	454
455	<code>__kernel_primitive:NN</code>	<code>\maxdeadcycles</code>	<code>\tex_maxdeadcycles:D</code>	455
456	<code>__kernel_primitive:NN</code>	<code>\maxdepth</code>	<code>\tex_maxdepth:D</code>	456
457	<code>__kernel_primitive:NN</code>	<code>\meaning</code>	<code>\tex_meaning:D</code>	457
458	<code>__kernel_primitive:NN</code>	<code>\medmuskip</code>	<code>\tex_medmuskip:D</code>	458
459	<code>__kernel_primitive:NN</code>	<code>\message</code>	<code>\tex_message:D</code>	459
460	<code>__kernel_primitive:NN</code>	<code>\mkern</code>	<code>\tex_mkern:D</code>	460

461	_kernel_primitive:NN	<u>\month</u>	\tex_month:D	461
462	_kernel_primitive:NN	<u>\moveleft</u>	\tex_moveleft:D	462
463	_kernel_primitive:NN	<u>\moveright</u>	\tex_moveright:D	463
464	_kernel_primitive:NN	<u>\mskip</u>	\tex_mskip:D	464
465	_kernel_primitive:NN	<u>\multiply</u>	\tex_multiply:D	465
466	_kernel_primitive:NN	<u>\muskip</u>	\tex_muskip:D	466
467	_kernel_primitive:NN	<u>\muskipdef</u>	\tex_muskipdef:D	467
468	_kernel_primitive:NN	<u>\newlinechar</u>	\tex_newlinechar:D	468
469	_kernel_primitive:NN	<u>\noalign</u>	\tex_noalign:D	469
470	_kernel_primitive:NN	<u>\noboundary</u>	\tex_noboundary:D	470
471	_kernel_primitive:NN	<u>\noexpand</u>	\tex_noexpand:D	471
472	_kernel_primitive:NN	<u>\noindent</u>	\tex_noindent:D	472
473	_kernel_primitive:NN	<u>\nolimits</u>	\tex_nolimits:D	473
474	_kernel_primitive:NN	<u>\nonscript</u>	\tex_nonscript:D	474
475	_kernel_primitive:NN	<u>\nonstopmode</u>	\tex_nonstopmode:D	475
476	_kernel_primitive:NN	<u>\nulldelimiterspace</u>	\tex_nulldelimiterspace:D	476
477	_kernel_primitive:NN	<u>\nullfont</u>	\tex_nullfont:D	477
478	_kernel_primitive:NN	<u>\number</u>	\tex_number:D	478
479	_kernel_primitive:NN	<u>\omit</u>	\tex_omit:D	479
480	_kernel_primitive:NN	<u>\openin</u>	\tex_openin:D	480
481	_kernel_primitive:NN	<u>\openout</u>	\tex_openout:D	481
482	_kernel_primitive:NN	<u>\or</u>	\tex_or:D	482
483	_kernel_primitive:NN	<u>\outer</u>	\tex_outer:D	483
484	_kernel_primitive:NN	<u>\output</u>	\tex_output:D	484
485	_kernel_primitive:NN	<u>\outputpenalty</u>	\tex_outputpenalty:D	485
486	_kernel_primitive:NN	<u>\over</u>	\tex_over:D	486
487	_kernel_primitive:NN	<u>\overfullrule</u>	\tex_overfullrule:D	487
488	_kernel_primitive:NN	<u>\overline</u>	\tex_overline:D	488
489	_kernel_primitive:NN	<u>\overwithdelims</u>	\tex_overwithdelims:D	489
490	_kernel_primitive:NN	<u>\pagedepth</u>	\tex_pagedepth:D	490
491	_kernel_primitive:NN	<u>\pagefilllstretch</u>	\tex_pagefilllstretch:D	491
492	_kernel_primitive:NN	<u>\pagefillstretch</u>	\tex_pagefillstretch:D	492
493	_kernel_primitive:NN	<u>\pagefilstretch</u>	\tex_pagefilstretch:D	493
494	_kernel_primitive:NN	<u>\pagegoal</u>	\tex_pagegoal:D	494
495	_kernel_primitive:NN	<u>\pageshrink</u>	\tex_pageshrink:D	495
496	_kernel_primitive:NN	<u>\pagestretch</u>	\tex_pagestretch:D	496
497	_kernel_primitive:NN	<u>\pagetotal</u>	\tex_pagetotal:D	497
498	_kernel_primitive:NN	<u>\par</u>	\tex_par:D	498
499	_kernel_primitive:NN	<u>\parfillskip</u>	\tex_parfillskip:D	499
500	_kernel_primitive:NN	<u>\parindent</u>	\tex_parindent:D	500
501	_kernel_primitive:NN	<u>\parshape</u>	\tex_parshape:D	501
502	_kernel_primitive:NN	<u>\parskip</u>	\tex_parskip:D	502
503	_kernel_primitive:NN	<u>\patterns</u>	\tex_patterns:D	503
504	_kernel_primitive:NN	<u>\pausing</u>	\tex_pausing:D	504
505	_kernel_primitive:NN	<u>\penalty</u>	\tex_penalty:D	505
506	_kernel_primitive:NN	<u>\postdisplaypenalty</u>	\tex_postdisplaypenalty:D	506

507	_kernel_primitive:NN	<u>\predisplaypenalty</u>	\tex_predisplaypenalty:D	507
508	_kernel_primitive:NN	<u>\predisplaysize</u>	\tex_predisplaysize:D	508
509	_kernel_primitive:NN	<u>\pretolerance</u>	\tex_pretolerance:D	509
510	_kernel_primitive:NN	<u>\prevdepth</u>	\tex_prevdepth:D	510
511	_kernel_primitive:NN	<u>\prevgraf</u>	\tex_prevgraf:D	511
512	_kernel_primitive:NN	<u>\radical</u>	\tex_radical:D	512
513	_kernel_primitive:NN	<u>\raise</u>	\tex_raise:D	513
514	_kernel_primitive:NN	<u>\read</u>	\tex_read:D	514
515	_kernel_primitive:NN	<u>\relax</u>	\tex_relax:D	515
516	_kernel_primitive:NN	<u>\relpenalty</u>	\tex_relpenalty:D	516
517	_kernel_primitive:NN	<u>\right</u>	\tex_right:D	517
518	_kernel_primitive:NN	<u>\righthyphenmin</u>	\tex_righthyphenmin:D	518
519	_kernel_primitive:NN	<u>\rightskip</u>	\tex_rightskip:D	519
520	_kernel_primitive:NN	<u>\romannumeral</u>	\tex_romannumeral:D	520
521	_kernel_primitive:NN	<u>\scriptfont</u>	\tex_scriptfont:D	521
522	_kernel_primitive:NN	<u>\scriptscriptfont</u>	\tex_scriptscriptfont:D	522
523	_kernel_primitive:NN	<u>\scriptscriptstyle</u>	\tex_scriptscriptstyle:D	523
524	_kernel_primitive:NN	<u>\scriptspace</u>	\tex_scriptspace:D	524
525	_kernel_primitive:NN	<u>\scriptstyle</u>	\tex_scriptstyle:D	525
526	_kernel_primitive:NN	<u>\scrollmode</u>	\tex_scrollmode:D	526
527	_kernel_primitive:NN	<u>\setbox</u>	\tex_setbox:D	527
528	_kernel_primitive:NN	<u>\setlanguage</u>	\tex_setlanguage:D	528
529	_kernel_primitive:NN	<u>\sfcode</u>	\tex_sfcode:D	529
530	_kernel_primitive:NN	<u>\shipout</u>	\tex_shipout:D	530
531	_kernel_primitive:NN	<u>\show</u>	\tex_show:D	531
532	_kernel_primitive:NN	<u>\showbox</u>	\tex_showbox:D	532
533	_kernel_primitive:NN	<u>\showboxbreadth</u>	\tex_showboxbreadth:D	533
534	_kernel_primitive:NN	<u>\showboxdepth</u>	\tex_showboxdepth:D	534
535	_kernel_primitive:NN	<u>\showlists</u>	\tex_showlists:D	535
536	_kernel_primitive:NN	<u>\showthe</u>	\tex_showthe:D	536
537	_kernel_primitive:NN	<u>\skewchar</u>	\tex_skewchar:D	537
538	_kernel_primitive:NN	<u>\skip</u>	\tex_skip:D	538
539	_kernel_primitive:NN	<u>\skipdef</u>	\tex_skipdef:D	539
540	_kernel_primitive:NN	<u>\spacefactor</u>	\tex_spacefactor:D	540
541	_kernel_primitive:NN	<u>\spaceskip</u>	\tex_spaceskip:D	541
542	_kernel_primitive:NN	<u>\span</u>	\tex_span:D	542
543	_kernel_primitive:NN	<u>\special</u>	\tex_special:D	543
544	_kernel_primitive:NN	<u>\splitbotmark</u>	\tex_splitbotmark:D	544
545	_kernel_primitive:NN	<u>\splitfirstmark</u>	\tex_splitfirstmark:D	545
546	_kernel_primitive:NN	<u>\splitmaxdepth</u>	\tex_splitmaxdepth:D	546
547	_kernel_primitive:NN	<u>\splittopskip</u>	\tex_splittopskip:D	547
548	_kernel_primitive:NN	<u>\string</u>	\tex_string:D	548
549	_kernel_primitive:NN	<u>\tabskip</u>	\tex_tabskip:D	549
550	_kernel_primitive:NN	<u>\textfont</u>	\tex_textfont:D	550
551	_kernel_primitive:NN	<u>\textstyle</u>	\tex_textstyle:D	551
552	_kernel_primitive:NN	<u>\the</u>	\tex_the:D	552

553	_kernel_primitive:NN	<u>\thickmuskip</u>	\tex_thickmuskip:D	553
554	_kernel_primitive:NN	<u>\thinmuskip</u>	\tex_thinmuskip:D	554
555	_kernel_primitive:NN	<u>\time</u>	\tex_time:D	555
556	_kernel_primitive:NN	<u>\toks</u>	\tex_toks:D	556
557	_kernel_primitive:NN	<u>\toksdef</u>	\tex_toksdef:D	557
558	_kernel_primitive:NN	<u>\tolerance</u>	\tex_tolerance:D	558
559	_kernel_primitive:NN	<u>\topmark</u>	\tex_topmark:D	559
560	_kernel_primitive:NN	<u>\topskip</u>	\tex_topskip:D	560
561	_kernel_primitive:NN	<u>\tracingcommands</u>	\tex_tracingcommands:D	561
562	_kernel_primitive:NN	<u>\tracinglostchars</u>	\tex_tracinglostchars:D	562
563	_kernel_primitive:NN	<u>\tracingmacros</u>	\tex_tracingmacros:D	563
564	_kernel_primitive:NN	<u>\tracingonline</u>	\tex_tracingonline:D	564
565	_kernel_primitive:NN	<u>\tracingoutput</u>	\tex_tracingoutput:D	565
566	_kernel_primitive:NN	<u>\tracingpages</u>	\tex_tracingpages:D	566
567	_kernel_primitive:NN	<u>\tracingparagraphs</u>	\tex_tracingparagraphs:D	567
568	_kernel_primitive:NN	<u>\tracingrestores</u>	\tex_tracingrestores:D	568
569	_kernel_primitive:NN	<u>\tracingstats</u>	\tex_tracingstats:D	569
570	_kernel_primitive:NN	<u>\uccode</u>	\tex_uccode:D	570
571	_kernel_primitive:NN	<u>\uchyph</u>	\tex_uchyph:D	571
572	_kernel_primitive:NN	<u>\underline</u>	\tex_underline:D	572
573	_kernel_primitive:NN	<u>\unhbox</u>	\tex_unhbox:D	573
574	_kernel_primitive:NN	<u>\unhcopy</u>	\tex_unhcopy:D	574
575	_kernel_primitive:NN	<u>\unkern</u>	\tex_unkern:D	575
576	_kernel_primitive:NN	<u>\unpenalty</u>	\tex_unpenalty:D	576
577	_kernel_primitive:NN	<u>\unskip</u>	\tex_unskip:D	577
578	_kernel_primitive:NN	<u>\unvbox</u>	\tex_unvbox:D	578
579	_kernel_primitive:NN	<u>\unvcopy</u>	\tex_unvcopy:D	579
580	_kernel_primitive:NN	<u>\uppercase</u>	\tex_uppercase:D	580
581	_kernel_primitive:NN	<u>\vadjust</u>	\tex_vadjust:D	581
582	_kernel_primitive:NN	<u>\valign</u>	\tex_valign:D	582
583	_kernel_primitive:NN	<u>\vbadness</u>	\tex_vbadness:D	583
584	_kernel_primitive:NN	<u>\vbox</u>	\tex_vbox:D	584
585	_kernel_primitive:NN	<u>\vcenter</u>	\tex_vcenter:D	585
586	_kernel_primitive:NN	<u>\vfil</u>	\tex_vfil:D	586
587	_kernel_primitive:NN	<u>\vfill</u>	\tex_vfill:D	587
588	_kernel_primitive:NN	<u>\vfilneg</u>	\tex_vfilneg:D	588
589	_kernel_primitive:NN	<u>\vfuzz</u>	\tex_vfuzz:D	589
590	_kernel_primitive:NN	<u>\voffset</u>	\tex_voffset:D	590
591	_kernel_primitive:NN	<u>\vrule</u>	\tex_vrule:D	591
592	_kernel_primitive:NN	<u>\vsize</u>	\tex_vsize:D	592
593	_kernel_primitive:NN	<u>\vskip</u>	\tex_vskip:D	593
594	_kernel_primitive:NN	<u>\vsplit</u>	\tex_vsplit:D	594
595	_kernel_primitive:NN	<u>\vss</u>	\tex_vss:D	595
596	_kernel_primitive:NN	<u>\vtop</u>	\tex_vtop:D	596
597	_kernel_primitive:NN	<u>\wd</u>	\tex_wd:D	597
598	_kernel_primitive:NN	<u>\widowpenalty</u>	\tex_widowpenalty:D	598

599	_kernel_primitive:NN	<u>\write</u>	\tex_write:D	599
600	_kernel_primitive:NN	<u>\xdef</u>	\tex_xdef:D	600
601	_kernel_primitive:NN	<u>\xleaders</u>	\tex_xleaders:D	601
602	_kernel_primitive:NN	<u>\xspaceskip</u>	\tex_xspaceskip:D	602
603	_kernel_primitive:NN	<u>\year</u>	\tex_year:D	603
604	_kernel_primitive:NN	<u>\beginL</u>	\tex_beginL:D	604
605	_kernel_primitive:NN	<u>\beginR</u>	\tex_beginR:D	605
606	_kernel_primitive:NN	<u>\botmarks</u>	\tex_botmarks:D	606
607	_kernel_primitive:NN	<u>\clubpenalties</u>	\tex_clubpenalties:D	607
608	_kernel_primitive:NN	<u>\currentgrouplevel</u>	\tex_currentgrouplevel:D	608
609	_kernel_primitive:NN	<u>\currentgrouptype</u>	\tex_currentgrouptype:D	609
610	_kernel_primitive:NN	<u>\currentifbranch</u>	\tex_currentifbranch:D	610
611	_kernel_primitive:NN	<u>\currentiflevel</u>	\tex_currentiflevel:D	611
612	_kernel_primitive:NN	<u>\currentifttype</u>	\tex_currentifttype:D	612
613	_kernel_primitive:NN	<u>\detokenize</u>	\tex_detokenize:D	613
614	_kernel_primitive:NN	<u>\dimexpr</u>	\tex_dimexpr:D	614
615	_kernel_primitive:NN	<u>\displaywidowpenalties</u>	\tex_displaywidowpenalties:D	615
616	_kernel_primitive:NN	<u>\endL</u>	\tex_endL:D	616
617	_kernel_primitive:NN	<u>\endR</u>	\tex_endR:D	617
618	_kernel_primitive:NN	<u>\eTeXrevision</u>	\tex_eTeXrevision:D	618
619	_kernel_primitive:NN	<u>\eTeXversion</u>	\tex_eTeXversion:D	619
620	_kernel_primitive:NN	<u>\everyeof</u>	\tex_everyeof:D	620
621	_kernel_primitive:NN	<u>\firstmarks</u>	\tex_firstmarks:D	621
622	_kernel_primitive:NN	<u>\fontchardp</u>	\tex_fontchardp:D	622
623	_kernel_primitive:NN	<u>\fontcharht</u>	\tex_fontcharht:D	623
624	_kernel_primitive:NN	<u>\fontcharic</u>	\tex_fontcharic:D	624
625	_kernel_primitive:NN	<u>\fontcharwd</u>	\tex_fontcharwd:D	625
626	_kernel_primitive:NN	<u>\glueexpr</u>	\tex_glueexpr:D	626
627	_kernel_primitive:NN	<u>\glueshrink</u>	\tex_glueshrink:D	627
628	_kernel_primitive:NN	<u>\glueshrinkorder</u>	\tex_glueshrinkorder:D	628
629	_kernel_primitive:NN	<u>\gluestretch</u>	\tex_gluestretch:D	629
630	_kernel_primitive:NN	<u>\gluestretchorder</u>	\tex_gluestretchorder:D	630
631	_kernel_primitive:NN	<u>\gluetomu</u>	\tex_gluetomu:D	631
632	_kernel_primitive:NN	<u>\ifcsname</u>	\tex_ifcsname:D	632
633	_kernel_primitive:NN	<u>\ifdefined</u>	\tex_ifdefined:D	633
634	_kernel_primitive:NN	<u>\iffontchar</u>	\tex_iffontchar:D	634
635	_kernel_primitive:NN	<u>\interactionmode</u>	\tex_interactionmode:D	635
636	_kernel_primitive:NN	<u>\interlinepenalties</u>	\tex_interlinepenalties:D	636
637	_kernel_primitive:NN	<u>\lastlinefit</u>	\tex_lastlinefit:D	637
638	_kernel_primitive:NN	<u>\lastnodetype</u>	\tex_lastnodetype:D	638
639	_kernel_primitive:NN	<u>\marks</u>	\tex_marks:D	639
640	_kernel_primitive:NN	<u>\middle</u>	\tex_middle:D	640
641	_kernel_primitive:NN	<u>\muexpr</u>	\tex_muexpr:D	641
642	_kernel_primitive:NN	<u>\mutoglua</u>	\tex_mutoglua:D	642
643	_kernel_primitive:NN	<u>\numexpr</u>	\tex_numexpr:D	643
644	_kernel_primitive:NN	<u>\pagediscards</u>	\tex_pagediscards:D	644

645	_kernel_primitive:NN	<u>\parshapedimen</u>	\tex_parshapedimen:D	645
646	_kernel_primitive:NN	<u>\parshapeindent</u>	\tex_parshapeindent:D	646
647	_kernel_primitive:NN	<u>\parshapelength</u>	\tex_parshapelength:D	647
648	_kernel_primitive:NN	<u>\predisplaydirection</u>	\tex_predisplaydirection:D	648
649	_kernel_primitive:NN	<u>\protected</u>	\tex_protected:D	649
650	_kernel_primitive:NN	<u>\readline</u>	\tex_readline:D	650
651	_kernel_primitive:NN	<u>\savinghyphcodes</u>	\tex_savinghyphcodes:D	651
652	_kernel_primitive:NN	<u>\savingvdiscards</u>	\tex_savingvdiscards:D	652
653	_kernel_primitive:NN	<u>\scantokens</u>	\tex_scantokens:D	653
654	_kernel_primitive:NN	<u>\showgroups</u>	\tex_showgroups:D	654
655	_kernel_primitive:NN	<u>\showifs</u>	\tex_showifs:D	655
656	_kernel_primitive:NN	<u>\showtokens</u>	\tex_showtokens:D	656
657	_kernel_primitive:NN	<u>\splitbotmarks</u>	\tex_splitbotmarks:D	657
658	_kernel_primitive:NN	<u>\splitdiscards</u>	\tex_splitdiscards:D	658
659	_kernel_primitive:NN	<u>\splitfirstmarks</u>	\tex_splitfirstmarks:D	659
660	_kernel_primitive:NN	<u>\TeXXeTstate</u>	\tex_TeXXeTstate:D	660
661	_kernel_primitive:NN	<u>\topmarks</u>	\tex_topmarks:D	661
662	_kernel_primitive:NN	<u>\tracingassigns</u>	\tex_tracingassigns:D	662
663	_kernel_primitive:NN	<u>\tracinggroups</u>	\tex_tracinggroups:D	663
664	_kernel_primitive:NN	<u>\tracingifs</u>	\tex_tracingifs:D	664
665	_kernel_primitive:NN	<u>\tracingnesting</u>	\tex_tracingnesting:D	665
666	_kernel_primitive:NN	<u>\tracingscantokens</u>	\tex_tracingscantokens:D	666
667	_kernel_primitive:NN	<u>\unexpanded</u>	\tex_unexpanded:D	667
668	_kernel_primitive:NN	<u>\unless</u>	\tex_unless:D	668
669	_kernel_primitive:NN	<u>\widowpenalties</u>	\tex_widowpenalties:D	669
670	_kernel_primitive:NN	<u>\pdfannot</u>	\tex_pdfannot:D	670
671	_kernel_primitive:NN	<u>\pdfcatalog</u>	\tex_pdfcatalog:D	671
672	_kernel_primitive:NN	<u>\pdfcompresslevel</u>	\tex_pdfcompresslevel:D	672
673	_kernel_primitive:NN	<u>\pdfcolorstack</u>	\tex_pdfcolorstack:D	673
674	_kernel_primitive:NN	<u>\pdfcolorstackinit</u>	\tex_pdfcolorstackinit:D	674
675	_kernel_primitive:NN	<u>\pdfdecimaldigits</u>	\tex_pdfdecimaldigits:D	675
676	_kernel_primitive:NN	<u>\pdfdest</u>	\tex_pdfdest:D	676
677	_kernel_primitive:NN	<u>\pdfdestmargin</u>	\tex_pdfdestmargin:D	677
678	_kernel_primitive:NN	<u>\pdfendlink</u>	\tex_pdfendlink:D	678
679	_kernel_primitive:NN	<u>\pdfendthread</u>	\tex_pdfendthread:D	679
680	_kernel_primitive:NN	<u>\pdffakespace</u>	\tex_pdffakespace:D	680
681	_kernel_primitive:NN	<u>\pdffontattr</u>	\tex_pdffontattr:D	681
682	_kernel_primitive:NN	<u>\pdffontname</u>	\tex_pdffontname:D	682
683	_kernel_primitive:NN	<u>\pdffontobjnum</u>	\tex_pdffontobjnum:D	683
684	_kernel_primitive:NN	<u>\pdfgamma</u>	\tex_pdfgamma:D	684
685	_kernel_primitive:NN	<u>\pdfgentounicode</u>	\tex_pdfgentounicode:D	685
686	_kernel_primitive:NN	<u>\pdfglyphtounicode</u>	\tex_pdfglyphtounicode:D	686
687	_kernel_primitive:NN	<u>\pdfhorigin</u>	\tex_pdfhorigin:D	687
688	_kernel_primitive:NN	<u>\pdfimageapplygamma</u>	\tex_pdfimageapplygamma:D	688
689	_kernel_primitive:NN	<u>\pdfimagegamma</u>	\tex_pdfimagegamma:D	689
690	_kernel_primitive:NN	<u>\pdfimagehicolor</u>	\tex_pdfimagehicolor:D	690

691	_kernel_primitive:NN	<u>\pdfimageresolution</u>	\tex_pdfimageresolution:D	691
692	_kernel_primitive:NN	<u>\pdfincludechars</u>	\tex_pdfincludechars:D	692
693	_kernel_primitive:NN	<u>\pdfinclusioncopyfonts</u>	\tex_pdfinclusioncopyfonts:D	693
694	_kernel_primitive:NN	<u>\pdfinclusionerrorlevel</u>		694
695		\tex_pdfinclusionerrorlevel:D		695
696	_kernel_primitive:NN	<u>\pdfinfo</u>	\tex_pdfinfo:D	696
697	_kernel_primitive:NN	<u>\pdfinfoomitdate</u>	\tex_pdfinfoomitdate:D	697
698	_kernel_primitive:NN	<u>\pdfinterwordsoff</u>	\tex_pdfinterwordsoff:D	698
699	_kernel_primitive:NN	<u>\pdfinterwordspaceon</u>	\tex_pdfinterwordspaceon:D	699
700	_kernel_primitive:NN	<u>\pdflastannot</u>	\tex_pdflastannot:D	700
701	_kernel_primitive:NN	<u>\pdflastlink</u>	\tex_pdflastlink:D	701
702	_kernel_primitive:NN	<u>\pdflastobj</u>	\tex_pdflastobj:D	702
703	_kernel_primitive:NN	<u>\pdflastxform</u>	\tex_pdflastxform:D	703
704	_kernel_primitive:NN	<u>\pdflastximage</u>	\tex_pdflastximage:D	704
705	_kernel_primitive:NN	<u>\pdflastximagecolordepth</u>		705
706		\tex_pdflastximagecolordepth:D		706
707	_kernel_primitive:NN	<u>\pdflastximagepages</u>	\tex_pdflastximagepages:D	707
708	_kernel_primitive:NN	<u>\pdflinkmargin</u>	\tex_pdflinkmargin:D	708
709	_kernel_primitive:NN	<u>\pdfliteral</u>	\tex_pdfliteral:D	709
710	_kernel_primitive:NN	<u>\pdfmapfile</u>	\tex_pdfmapfile:D	710
711	_kernel_primitive:NN	<u>\pdfmapline</u>	\tex_pdfmapline:D	711
712	_kernel_primitive:NN	<u>\pdfmajorversion</u>	\tex_pdfmajorversion:D	712
713	_kernel_primitive:NN	<u>\pdfminorversion</u>	\tex_pdfminorversion:D	713
714	_kernel_primitive:NN	<u>\pdfnames</u>	\tex_pdfnames:D	714
715	_kernel_primitive:NN	<u>\pdfnobuiltintounicode</u>	\tex_pdfnobuiltintounicode:D	715
716	_kernel_primitive:NN	<u>\pdfobj</u>	\tex_pdfobj:D	716
717	_kernel_primitive:NN	<u>\pdfobjcompresslevel</u>	\tex_pdfobjcompresslevel:D	717
718	_kernel_primitive:NN	<u>\pdfomitcharset</u>	\tex_pdfomitcharset:D	718
719	_kernel_primitive:NN	<u>\pdfoutline</u>	\tex_pdfoutline:D	719
720	_kernel_primitive:NN	<u>\pdfoutput</u>	\tex_pdfoutput:D	720
721	_kernel_primitive:NN	<u>\pdfpageattr</u>	\tex_pdfpageattr:D	721
722	_kernel_primitive:NN	<u>\pdfpagebox</u>	\tex_pdfpagebox:D	722
723	_kernel_primitive:NN	<u>\pdfpageref</u>	\tex_pdfpageref:D	723
724	_kernel_primitive:NN	<u>\pdfpageresources</u>	\tex_pdfpageresources:D	724
725	_kernel_primitive:NN	<u>\pdfpagesattr</u>	\tex_pdfpagesattr:D	725
726	_kernel_primitive:NN	<u>\pdfptexuseunderscore</u>	\tex_pdfptexuseunderscore:D	726
727	_kernel_primitive:NN	<u>\pdfrefobj</u>	\tex_pdfrefobj:D	727
728	_kernel_primitive:NN	<u>\pdfrefxform</u>	\tex_pdfrefxform:D	728
729	_kernel_primitive:NN	<u>\pdfrefximage</u>	\tex_pdfrefximage:D	729
730	_kernel_primitive:NN	<u>\pdfrestore</u>	\tex_pdfrestore:D	730
731	_kernel_primitive:NN	<u>\pdfretval</u>	\tex_pdfretval:D	731
732	_kernel_primitive:NN	<u>\pdfrunninglinkoff</u>	\tex_pdfrunninglinkoff:D	732
733	_kernel_primitive:NN	<u>\pdfrunninglinkon</u>	\tex_pdfrunninglinkon:D	733
734	_kernel_primitive:NN	<u>\pdfsave</u>	\tex_pdfsave:D	734
735	_kernel_primitive:NN	<u>\pdfsetmatrix</u>	\tex_pdfsetmatrix:D	735
736	_kernel_primitive:NN	<u>\pdfstartlink</u>	\tex_pdfstartlink:D	736

737	<code>__kernel_primitive:NN</code>	<code>\pdfstartthread</code>	<code>\tex_pdfstartthread:D</code>	737
738	<code>__kernel_primitive:NN</code>	<code>\pdfsuppressptexinfo</code>	<code>\tex_pdfsuppressptexinfo:D</code>	738
739	<code>__kernel_primitive:NN</code>	<code>\pdfsuppresswarningdupdest</code>		739
740		<code>\tex_pdfsuppresswarningdupdest:D</code>		740
741	<code>__kernel_primitive:NN</code>	<code>\pdfsuppresswarningdupmap</code>		741
742		<code>\tex_pdfsuppresswarningdupmap:D</code>		742
743	<code>__kernel_primitive:NN</code>	<code>\pdfsuppresswarningpagegroup</code>		743
744		<code>\tex_pdfsuppresswarningpagegroup:D</code>		744
745	<code>__kernel_primitive:NN</code>	<code>\pdfthread</code>	<code>\tex_pdfthread:D</code>	745
746	<code>__kernel_primitive:NN</code>	<code>\pdfthreadmargin</code>	<code>\tex_pdfthreadmargin:D</code>	746
747	<code>__kernel_primitive:NN</code>	<code>\pdftrailer</code>	<code>\tex_pdftrailer:D</code>	747
748	<code>__kernel_primitive:NN</code>	<code>\pdftrailerid</code>	<code>\tex_pdftrailerid:D</code>	748
749	<code>__kernel_primitive:NN</code>	<code>\pdfuniqueresname</code>	<code>\tex_pdfuniqueresname:D</code>	749
750	<code>__kernel_primitive:NN</code>	<code>\pdfvorigin</code>	<code>\tex_pdfvorigin:D</code>	750
751	<code>__kernel_primitive:NN</code>	<code>\pdfxform</code>	<code>\tex_pdfxform:D</code>	751
752	<code>__kernel_primitive:NN</code>	<code>\pdfxformname</code>	<code>\tex_pdfxformname:D</code>	752
753	<code>__kernel_primitive:NN</code>	<code>\pdfximage</code>	<code>\tex_pdfximage:D</code>	753
754	<code>__kernel_primitive:NN</code>	<code>\pdfximagebbox</code>	<code>\tex_pdfximagebbox:D</code>	754
755	<code>__kernel_primitive:NN</code>	<code>\ifpdfabsdim</code>	<code>\tex_ifabsdim:D</code>	755
756	<code>__kernel_primitive:NN</code>	<code>\ifpdfabsnum</code>	<code>\tex_ifabsnum:D</code>	756
757	<code>__kernel_primitive:NN</code>	<code>\ifpdfprimitive</code>	<code>\tex_ifprimitive:D</code>	757
758	<code>__kernel_primitive:NN</code>	<code>\pdfadjustinterwordglue</code>		758
759		<code>\tex_adjustinterwordglue:D</code>		759
760	<code>__kernel_primitive:NN</code>	<code>\pdfadjustspacing</code>	<code>\tex_adjustspacing:D</code>	760
761	<code>__kernel_primitive:NN</code>	<code>\pdfappendkern</code>	<code>\tex_appendkern:D</code>	761
762	<code>__kernel_primitive:NN</code>	<code>\pdfcopyfont</code>	<code>\tex_copyfont:D</code>	762
763	<code>__kernel_primitive:NN</code>	<code>\pdfcreationdate</code>	<code>\tex_creationdate:D</code>	763
764	<code>__kernel_primitive:NN</code>	<code>\pdfdraftmode</code>	<code>\tex_draftmode:D</code>	764
765	<code>__kernel_primitive:NN</code>	<code>\pdfeachlinedepth</code>	<code>\tex_eachlinedepth:D</code>	765
766	<code>__kernel_primitive:NN</code>	<code>\pdfeachlineheight</code>	<code>\tex_eachlineheight:D</code>	766
767	<code>__kernel_primitive:NN</code>	<code>\pdfelapsedtime</code>	<code>\tex_elapsedtime:D</code>	767
768	<code>__kernel_primitive:NN</code>	<code>\pdfescapehex</code>	<code>\tex_escapehex:D</code>	768
769	<code>__kernel_primitive:NN</code>	<code>\pdfescapename</code>	<code>\tex_escapename:D</code>	769
770	<code>__kernel_primitive:NN</code>	<code>\pdfescapestring</code>	<code>\tex_escapestring:D</code>	770
771	<code>__kernel_primitive:NN</code>	<code>\pdffirstlineheight</code>	<code>\tex_firstlineheight:D</code>	771
772	<code>__kernel_primitive:NN</code>	<code>\pdffontexpand</code>	<code>\tex_fontexpand:D</code>	772
773	<code>__kernel_primitive:NN</code>	<code>\pdffontsize</code>	<code>\tex_fontsize:D</code>	773
774	<code>__kernel_primitive:NN</code>	<code>\pdfignoreddimen</code>	<code>\tex_ignoreddimen:D</code>	774
775	<code>__kernel_primitive:NN</code>	<code>\pdfinsertht</code>	<code>\tex_insertht:D</code>	775
776	<code>__kernel_primitive:NN</code>	<code>\pdflastlinedepth</code>	<code>\tex_lastlinedepth:D</code>	776
777	<code>__kernel_primitive:NN</code>	<code>\pdflastmatch</code>	<code>\tex_lastmatch:D</code>	777
778	<code>__kernel_primitive:NN</code>	<code>\pdflastxpos</code>	<code>\tex_lastxpos:D</code>	778
779	<code>__kernel_primitive:NN</code>	<code>\pdflastypos</code>	<code>\tex_lastypos:D</code>	779
780	<code>__kernel_primitive:NN</code>	<code>\pdfmatch</code>	<code>\tex_match:D</code>	780
781	<code>__kernel_primitive:NN</code>	<code>\pdfnoligatures</code>	<code>\tex_noligatures:D</code>	781
782	<code>__kernel_primitive:NN</code>	<code>\pdfnormaldeviate</code>	<code>\tex_normaldeviate:D</code>	782

783	_kernel_primitive:NN	<u>\pdfpageheight</u>	\tex_pageheight:D	783
784	_kernel_primitive:NN	<u>\pdfpagewidth</u>	\tex_pagewidth:D	784
785	_kernel_primitive:NN	<u>\pdfpkmode</u>	\tex_pkmode:D	785
786	_kernel_primitive:NN	<u>\pdfpkresolution</u>	\tex_pkresolution:D	786
787	_kernel_primitive:NN	<u>\pdfprimitive</u>	\tex_primitive:D	787
788	_kernel_primitive:NN	<u>\pdfprependkern</u>	\tex_prependkern:D	788
789	_kernel_primitive:NN	<u>\pdfprotrudechars</u>	\tex_protrudechars:D	789
790	_kernel_primitive:NN	<u>\pdfpxdimen</u>	\tex_pxdimen:D	790
791	_kernel_primitive:NN	<u>\pdfrandomseed</u>	\tex_randomseed:D	791
792	_kernel_primitive:NN	<u>\pdfresettimer</u>	\tex_resettimer:D	792
793	_kernel_primitive:NN	<u>\pdfsavepos</u>	\tex_savepos:D	793
794	_kernel_primitive:NN	<u>\pdfsetrandomseed</u>	\tex_setrandomseed:D	794
795	_kernel_primitive:NN	<u>\pdfshellescape</u>	\tex_shellescape:D	795
796	_kernel_primitive:NN	<u>\pdftracingfonts</u>	\tex_tracingfonts:D	796
797	_kernel_primitive:NN	<u>\pdfunescapehex</u>	\tex_unescapehex:D	797
798	_kernel_primitive:NN	<u>\pdfuniformdeviate</u>	\tex_uniformdeviate:D	798
799	_kernel_primitive:NN	<u>\pdftexbanner</u>	\tex_pdftexbanner:D	799
800	_kernel_primitive:NN	<u>\pdftexrevision</u>	\tex_pdftexrevision:D	800
801	_kernel_primitive:NN	<u>\pdftexversion</u>	\tex_pdftexversion:D	801
802	_kernel_primitive:NN	<u>\efcode</u>	\tex_efcode:D	802
803	_kernel_primitive:NN	<u>\ifincsname</u>	\tex_ifincsname:D	803
804	_kernel_primitive:NN	<u>\knaccode</u>	\tex_knaccode:D	804
805	_kernel_primitive:NN	<u>\knbccode</u>	\tex_knbccode:D	805
806	_kernel_primitive:NN	<u>\knbscode</u>	\tex_knbscode:D	806
807	_kernel_primitive:NN	<u>\leftmarginkern</u>	\tex_leftmarginkern:D	807
808	_kernel_primitive:NN	<u>\letterspacefont</u>	\tex_letterspacefont:D	808
809	_kernel_primitive:NN	<u>\lpcode</u>	\tex_lpcode:D	809
810	_kernel_primitive:NN	<u>\quitvmode</u>	\tex_quitvmode:D	810
811	_kernel_primitive:NN	<u>\rightmarginkern</u>	\tex_rightmarginkern:D	811
812	_kernel_primitive:NN	<u>\rpcode</u>	\tex_rpcode:D	812
813	_kernel_primitive:NN	<u>\shbscode</u>	\tex_shbscode:D	813
814	_kernel_primitive:NN	<u>\stbscode</u>	\tex_stbscode:D	814
815	_kernel_primitive:NN	<u>\synctex</u>	\tex_synctex:D	815
816	_kernel_primitive:NN	<u>\tagcode</u>	\tex_tagcode:D	816
817	\tex_long:D \tex_def:D	\use_ii:nn #1#2 {#2}		817
818	\tex_long:D \tex_def:D	\use_none:n #1 { }		818
819	\tex_long:D \tex_def:D	_kernel_primitive:NN #1#2		819
820		{		820
821		\tex_ifdefined:D #1		821
822		\tex_expandafter:D \use_ii:nn		822
823		\tex_fi:D		823
824		\use_none:n { \tex_global:D \tex_let:D #2 #1 }		824
825		}		825
826	_kernel_primitive:NN	<u>\pdfstrcmp</u>	\tex_strcmp:D	826
827	_kernel_primitive:NN	<u>\pdffilesize</u>	\tex_filesize:D	827
828	_kernel_primitive:NN	<u>\pdfmdfivesum</u>	\tex_mdfivesum:D	828

829	_kernel_primitive:NN	<u>\pdffilemoddate</u>	\tex_filemoddate:D	829
830	_kernel_primitive:NN	<u>\pdffiledump</u>	\tex_filedump:D	830
831	_kernel_primitive:NN	<u>\suppressfontnotfounderror</u>		831
832		\tex_suppressfontnotfounderror:D		832
833	_kernel_primitive:NN	<u>\XeTeXcharclass</u>	\tex_XeTeXcharclass:D	833
834	_kernel_primitive:NN	<u>\XeTeXcharglyph</u>	\tex_XeTeXcharglyph:D	834
835	_kernel_primitive:NN	<u>\XeTeXcountfeatures</u>	\tex_XeTeXcountfeatures:D	835
836	_kernel_primitive:NN	<u>\XeTeXcountglyphs</u>	\tex_XeTeXcountglyphs:D	836
837	_kernel_primitive:NN	<u>\XeTeXcountselectors</u>	\tex_XeTeXcountselectors:D	837
838	_kernel_primitive:NN	<u>\XeTeXcountvariations</u>	\tex_XeTeXcountvariations:D	838
839	_kernel_primitive:NN	<u>\XeTeXdefaultencoding</u>	\tex_XeTeXdefaultencoding:D	839
840	_kernel_primitive:NN	<u>\XeTeXdashbreakstate</u>	\tex_XeTeXdashbreakstate:D	840
841	_kernel_primitive:NN	<u>\XeTeXfeaturecode</u>	\tex_XeTeXfeaturecode:D	841
842	_kernel_primitive:NN	<u>\XeTeXfeaturename</u>	\tex_XeTeXfeaturename:D	842
843	_kernel_primitive:NN	<u>\XeTeXfindfeaturebyname</u>		843
844		\tex_XeTeXfindfeaturebyname:D		844
845	_kernel_primitive:NN	<u>\XeTeXfindselectorbyname</u>		845
846		\tex_XeTeXfindselectorbyname:D		846
847	_kernel_primitive:NN	<u>\XeTeXfindvariationbyname</u>		847
848		\tex_XeTeXfindvariationbyname:D		848
849	_kernel_primitive:NN	<u>\XeTeXfirstfontchar</u>	\tex_XeTeXfirstfontchar:D	849
850	_kernel_primitive:NN	<u>\XeTeXfonttype</u>	\tex_XeTeXfonttype:D	850
851	_kernel_primitive:NN	<u>\XeTeXgenerateactualtext</u>		851
852		\tex_XeTeXgenerateactualtext:D		852
853	_kernel_primitive:NN	<u>\XeTeXglyph</u>	\tex_XeTeXglyph:D	853
854	_kernel_primitive:NN	<u>\XeTeXglyphbounds</u>	\tex_XeTeXglyphbounds:D	854
855	_kernel_primitive:NN	<u>\XeTeXglyphindex</u>	\tex_XeTeXglyphindex:D	855
856	_kernel_primitive:NN	<u>\XeTeXglyphname</u>	\tex_XeTeXglyphname:D	856
857	_kernel_primitive:NN	<u>\XeTeXinputencoding</u>	\tex_XeTeXinputencoding:D	857
858	_kernel_primitive:NN	<u>\XeTeXinputnormalization</u>		858
859		\tex_XeTeXinputnormalization:D		859
860	_kernel_primitive:NN	<u>\XeTeXinterchartokenstate</u>		860
861		\tex_XeTeXinterchartokenstate:D		861
862	_kernel_primitive:NN	<u>\XeTeXinterchartoks</u>	\tex_XeTeXinterchartoks:D	862
863	_kernel_primitive:NN	<u>\XeTeXisdefaultselector</u>		863
864		\tex_XeTeXisdefaultselector:D		864
865	_kernel_primitive:NN	<u>\XeTeXisexclusivefeature</u>		865
866		\tex_XeTeXisexclusivefeature:D		866
867	_kernel_primitive:NN	<u>\XeTeXlastfontchar</u>	\tex_XeTeXlastfontchar:D	867
868	_kernel_primitive:NN	<u>\XeTeXlinebreakskip</u>	\tex_XeTeXlinebreakskip:D	868
869	_kernel_primitive:NN	<u>\XeTeXlinebreaklocale</u>	\tex_XeTeXlinebreaklocale:D	869
870	_kernel_primitive:NN	<u>\XeTeXlinebreakpenalty</u>	\tex_XeTeXlinebreakpenalty:D	870
871	_kernel_primitive:NN	<u>\XeTeXOTcountfeatures</u>	\tex_XeTeXOTcountfeatures:D	871
872	_kernel_primitive:NN	<u>\XeTeXOTcountlanguages</u>	\tex_XeTeXOTcountlanguages:D	872
873	_kernel_primitive:NN	<u>\XeTeXOTcountscripts</u>	\tex_XeTeXOTcountscripts:D	873
874	_kernel_primitive:NN	<u>\XeTeXOTfeaturetag</u>	\tex_XeTeXOTfeaturetag:D	874

875	_kernel_primitive:NN	<u>\XeTeXOTlanguagetag</u>	\tex_XeTeXOTlanguagetag:D	875
876	_kernel_primitive:NN	<u>\XeTeXOTscripttag</u>	\tex_XeTeXOTscripttag:D	876
877	_kernel_primitive:NN	<u>\XeTeXpdffile</u>	\tex_XeTeXpdffile:D	877
878	_kernel_primitive:NN	<u>\XeTeXpdfpagecount</u>	\tex_XeTeXpdfpagecount:D	878
879	_kernel_primitive:NN	<u>\XeTeXpicfile</u>	\tex_XeTeXpicfile:D	879
880	_kernel_primitive:NN	<u>\XeTeXrevision</u>	\tex_XeTeXrevision:D	880
881	_kernel_primitive:NN	<u>\XeTeXselectorname</u>	\tex_XeTeXselectorname:D	881
882	_kernel_primitive:NN	<u>\XeTeXtracingfonts</u>	\tex_XeTeXtracingfonts:D	882
883	_kernel_primitive:NN	<u>\XeTeXupwardsmode</u>	\tex_XeTeXupwardsmode:D	883
884	_kernel_primitive:NN	<u>\XeTeXuseglyphmetrics</u>	\tex_XeTeXuseglyphmetrics:D	884
885	_kernel_primitive:NN	<u>\XeTeXvariation</u>	\tex_XeTeXvariation:D	885
886	_kernel_primitive:NN	<u>\XeTeXvariationdefault</u>	\tex_XeTeXvariationdefault:D	886
887	_kernel_primitive:NN	<u>\XeTeXvariationmax</u>	\tex_XeTeXvariationmax:D	887
888	_kernel_primitive:NN	<u>\XeTeXvariationmin</u>	\tex_XeTeXvariationmin:D	888
889	_kernel_primitive:NN	<u>\XeTeXvariationname</u>	\tex_XeTeXvariationname:D	889
890	_kernel_primitive:NN	<u>\XeTeXversion</u>	\tex_XeTeXversion:D	890
891	_kernel_primitive:NN	<u>\XeTeXselectorcode</u>	\tex_XeTeXselectorcode:D	891
892	_kernel_primitive:NN	<u>\XeTeXinterwordspaceshaping</u>		892
893		\tex_XeTeXinterwordspaceshaping:D		893
894	_kernel_primitive:NN	<u>\XeTeXhyphenatablelength</u>		894
895		\tex_XeTeXhyphenatablelength:D		895
896	_kernel_primitive:NN	<u>\creationdate</u>	\tex_creationdate:D	896
897	_kernel_primitive:NN	<u>\elapsedtime</u>	\tex_elapsedtime:D	897
898	_kernel_primitive:NN	<u>\filedump</u>	\tex_filedump:D	898
899	_kernel_primitive:NN	<u>\filemoddate</u>	\tex_filemoddate:D	899
900	_kernel_primitive:NN	<u>\filesize</u>	\tex_filesize:D	900
901	_kernel_primitive:NN	<u>\mdfivesum</u>	\tex_mdfivesum:D	901
902	_kernel_primitive:NN	<u>\ifprimitive</u>	\tex_ifprimitive:D	902
903	_kernel_primitive:NN	<u>\primitive</u>	\tex_primitive:D	903
904	_kernel_primitive:NN	<u>\resettimer</u>	\tex_resettimer:D	904
905	_kernel_primitive:NN	<u>\shellescape</u>	\tex_shellescape:D	905
906	_kernel_primitive:NN	<u>\XeTeXprotrudechars</u>	\tex_protrudechars:D	906
907	_kernel_primitive:NN	<u>\alignmark</u>	\tex_alignmark:D	907
908	_kernel_primitive:NN	<u>\aligntab</u>	\tex_aligntab:D	908
909	_kernel_primitive:NN	<u>\attribute</u>	\tex_attribute:D	909
910	_kernel_primitive:NN	<u>\attributedef</u>	\tex_attributedef:D	910
911	_kernel_primitive:NN	<u>\automaticdiscretionary</u>		911
912		\tex_automaticdiscretionary:D		912
913	_kernel_primitive:NN	<u>\automatichyphenmode</u>	\tex_automatichyphenmode:D	913
914	_kernel_primitive:NN	<u>\automatichyphenpenalty</u>		914
915		\tex_automatichyphenpenalty:D		915
916	_kernel_primitive:NN	<u>\begincsname</u>	\tex_begincsname:D	916
917	_kernel_primitive:NN	<u>\bodydir</u>	\tex_bodydir:D	917
918	_kernel_primitive:NN	<u>\bodydirection</u>	\tex_bodydirection:D	918
919	_kernel_primitive:NN	<u>\boundary</u>	\tex_boundary:D	919
920	_kernel_primitive:NN	<u>\boxdir</u>	\tex_boxdir:D	920

921	_kernel_primitive:NN	<u>\boxdirection</u>	\tex_boxdirection:D	921
922	_kernel_primitive:NN	<u>\breakafterdirmode</u>	\tex_breakafterdirmode:D	922
923	_kernel_primitive:NN	<u>\catcodetable</u>	\tex_catcodetable:D	923
924	_kernel_primitive:NN	<u>\clearmarks</u>	\tex_clearmarks:D	924
925	_kernel_primitive:NN	<u>\crampeddisplaystyle</u>	\tex_crampeddisplaystyle:D	925
926	_kernel_primitive:NN	<u>\crampedscriptscriptstyle</u>		926
927		\tex_crampedscriptscriptstyle:D		927
928	_kernel_primitive:NN	<u>\crampedscriptstyle</u>	\tex_crampedscriptstyle:D	928
929	_kernel_primitive:NN	<u>\crampedtextstyle</u>	\tex_crampedtextstyle:D	929
930	_kernel_primitive:NN	<u>\csstring</u>	\tex_csstring:D	930
931	_kernel_primitive:NN	<u>\deferred</u>	\tex_deferred:D	931
932	_kernel_primitive:NN	<u>\discretionaryligaturemode</u>		932
933		\tex_discretionaryligaturemode:D		933
934	_kernel_primitive:NN	<u>\directlua</u>	\tex_directlua:D	934
935	_kernel_primitive:NN	<u>\dviextension</u>	\tex_dviextension:D	935
936	_kernel_primitive:NN	<u>\dvifedback</u>	\tex_dvifedback:D	936
937	_kernel_primitive:NN	<u>\dvivariable</u>	\tex_dvivariable:D	937
938	_kernel_primitive:NN	<u>\eTeXglueshrinkorder</u>	\tex_eTeXglueshrinkorder:D	938
939	_kernel_primitive:NN	<u>\eTeXgluestretchorder</u>	\tex_eTeXgluestretchorder:D	939
940	_kernel_primitive:NN	<u>\endlocalcontrol</u>	\tex_endlocalcontrol:D	940
941	_kernel_primitive:NN	<u>\etoksapp</u>	\tex_etoksapp:D	941
942	_kernel_primitive:NN	<u>\etokspre</u>	\tex_etokspre:D	942
943	_kernel_primitive:NN	<u>\exceptionpenalty</u>	\tex_exceptionpenalty:D	943
944	_kernel_primitive:NN	<u>\exhyphenchar</u>	\tex_exhyphenchar:D	944
945	_kernel_primitive:NN	<u>\explicitthyphenpenalty</u>	\tex_explicitthyphenpenalty:D	945
946	_kernel_primitive:NN	<u>\expanded</u>	\tex_expanded:D	946
947	_kernel_primitive:NN	<u>\explicitdiscretionary</u>	\tex_explicitdiscretionary:D	947
948	_kernel_primitive:NN	<u>\firstvalidlanguage</u>	\tex_firstvalidlanguage:D	948
949	_kernel_primitive:NN	<u>\fontid</u>	\tex_fontid:D	949
950	_kernel_primitive:NN	<u>\formatname</u>	\tex_formatname:D	950
951	_kernel_primitive:NN	<u>\hjcode</u>	\tex_hjcode:D	951
952	_kernel_primitive:NN	<u>\hpack</u>	\tex_hpack:D	952
953	_kernel_primitive:NN	<u>\hyphenationbounds</u>	\tex_hyphenationbounds:D	953
954	_kernel_primitive:NN	<u>\hyphenationmin</u>	\tex_hyphenationmin:D	954
955	_kernel_primitive:NN	<u>\hyphenpenaltymode</u>	\tex_hyphenpenaltymode:D	955
956	_kernel_primitive:NN	<u>\gleaders</u>	\tex_gleaders:D	956
957	_kernel_primitive:NN	<u>\glet</u>	\tex_glet:D	957
958	_kernel_primitive:NN	<u>\glyphdimensionsmode</u>	\tex_glyphdimensionsmode:D	958
959	_kernel_primitive:NN	<u>\gtoksapp</u>	\tex_gtoksapp:D	959
960	_kernel_primitive:NN	<u>\gtokspre</u>	\tex_gtokspre:D	960
961	_kernel_primitive:NN	<u>\ifcondition</u>	\tex_ifcondition:D	961
962	_kernel_primitive:NN	<u>\immediateassigned</u>	\tex_immediateassigned:D	962
963	_kernel_primitive:NN	<u>\immediateassignment</u>	\tex_immediateassignment:D	963
964	_kernel_primitive:NN	<u>\initcatcodetable</u>	\tex_initcatcodetable:D	964
965	_kernel_primitive:NN	<u>\lastnamedcs</u>	\tex_lastnamedcs:D	965
966	_kernel_primitive:NN	<u>\latelua</u>	\tex_latelua:D	966

967	__kernel_primitive:NN	<u>\lateluafunction</u>	\tex_lateluafunction:D	967
968	__kernel_primitive:NN	<u>\leftghost</u>	\tex_leftghost:D	968
969	__kernel_primitive:NN	<u>\letcharcode</u>	\tex_letcharcode:D	969
970	__kernel_primitive:NN	<u>\linedir</u>	\tex_linedir:D	970
971	__kernel_primitive:NN	<u>\linedirection</u>	\tex_linedirection:D	971
972	__kernel_primitive:NN	<u>\localbrokenpenalty</u>	\tex_localbrokenpenalty:D	972
973	__kernel_primitive:NN	<u>\localinterlinepenalty</u>	\tex_localinterlinepenalty:D	973
974	__kernel_primitive:NN	<u>\luabytecode</u>	\tex_luabytecode:D	974
975	__kernel_primitive:NN	<u>\luabytecodecall</u>	\tex_luabytecodecall:D	975
976	__kernel_primitive:NN	<u>\luacopyinputnodes</u>	\tex_luacopyinputnodes:D	976
977	__kernel_primitive:NN	<u>\luadef</u>	\tex_luadef:D	977
978	__kernel_primitive:NN	<u>\localleftbox</u>	\tex_localleftbox:D	978
979	__kernel_primitive:NN	<u>\localrightbox</u>	\tex_localrightbox:D	979
980	__kernel_primitive:NN	<u>\luaescapestring</u>	\tex_luaescapestring:D	980
981	__kernel_primitive:NN	<u>\luafunction</u>	\tex_luafunction:D	981
982	__kernel_primitive:NN	<u>\luafunctioncall</u>	\tex_luafunctioncall:D	982
983	__kernel_primitive:NN	<u>\luatexbanner</u>	\tex_luatexbanner:D	983
984	__kernel_primitive:NN	<u>\luatexrevision</u>	\tex_luatexrevision:D	984
985	__kernel_primitive:NN	<u>\luatexversion</u>	\tex_luatexversion:D	985
986	__kernel_primitive:NN	<u>\mathdefaultsmode</u>	\tex_mathdefaultsmode:D	986
987	__kernel_primitive:NN	<u>\mathdelimitersmode</u>	\tex_mathdelimitersmode:D	987
988	__kernel_primitive:NN	<u>\mathdir</u>	\tex_mathdir:D	988
989	__kernel_primitive:NN	<u>\mathdirection</u>	\tex_mathdirection:D	989
990	__kernel_primitive:NN	<u>\mathdisplayskipmode</u>	\tex_mathdisplayskipmode:D	990
991	__kernel_primitive:NN	<u>\matheqdirmode</u>	\tex_matheqdirmode:D	991
992	__kernel_primitive:NN	<u>\matheqnogapstep</u>	\tex_matheqnogapstep:D	992
993	__kernel_primitive:NN	<u>\mathemptydisplaymode</u>	\tex_mathemptydisplaymode:D	993
994	__kernel_primitive:NN	<u>\mathflattenmode</u>	\tex_mathflattenmode:D	994
995	__kernel_primitive:NN	<u>\mathitalicsmode</u>	\tex_mathitalicsmode:D	995
996	__kernel_primitive:NN	<u>\mathnolimitsmode</u>	\tex_mathnolimitsmode:D	996
997	__kernel_primitive:NN	<u>\mathoption</u>	\tex_mathoption:D	997
998	__kernel_primitive:NN	<u>\mathpenaltiesmode</u>	\tex_mathpenaltiesmode:D	998
999	__kernel_primitive:NN	<u>\mathrulesfam</u>	\tex_mathrulesfam:D	999
1000	__kernel_primitive:NN	<u>\mathscriptsmode</u>	\tex_mathscriptsmode:D	1000
1001	__kernel_primitive:NN	<u>\mathscriptboxmode</u>	\tex_mathscriptboxmode:D	1001
1002	__kernel_primitive:NN	<u>\mathscriptcharmode</u>	\tex_mathscriptcharmode:D	1002
1003	__kernel_primitive:NN	<u>\mathstyle</u>	\tex_mathstyle:D	1003
1004	__kernel_primitive:NN	<u>\mathsurroundmode</u>	\tex_mathsurroundmode:D	1004
1005	__kernel_primitive:NN	<u>\mathsurroundskip</u>	\tex_mathsurroundskip:D	1005
1006	__kernel_primitive:NN	<u>\nohrule</u>	\tex_nohrule:D	1006
1007	__kernel_primitive:NN	<u>\nokerns</u>	\tex_nokerns:D	1007
1008	__kernel_primitive:NN	<u>\noligs</u>	\tex_noligs:D	1008
1009	__kernel_primitive:NN	<u>\nospaces</u>	\tex_nospaces:D	1009
1010	__kernel_primitive:NN	<u>\novrule</u>	\tex_novrule:D	1010
1011	__kernel_primitive:NN	<u>\outputbox</u>	\tex_outputbox:D	1011
1012	__kernel_primitive:NN	<u>\pagebottomoffset</u>	\tex_pagebottomoffset:D	1012

1013	_kernel_primitive:NN	<u>\pagedir</u>	\tex_pagedir:D	1013
1014	_kernel_primitive:NN	<u>\pagedirection</u>	\tex_pagedirection:D	1014
1015	_kernel_primitive:NN	<u>\pageleftoffset</u>	\tex_pageleftoffset:D	1015
1016	_kernel_primitive:NN	<u>\pagerightoffset</u>	\tex_pagerightoffset:D	1016
1017	_kernel_primitive:NN	<u>\pagetopoffset</u>	\tex_pagetopoffset:D	1017
1018	_kernel_primitive:NN	<u>\pardir</u>	\tex_pardir:D	1018
1019	_kernel_primitive:NN	<u>\pardirection</u>	\tex_pardirection:D	1019
1020	_kernel_primitive:NN	<u>\pdfextension</u>	\tex_pdfextension:D	1020
1021	_kernel_primitive:NN	<u>\pdffeedback</u>	\tex_pdffeedback:D	1021
1022	_kernel_primitive:NN	<u>\pdfvariable</u>	\tex_pdfvariable:D	1022
1023	_kernel_primitive:NN	<u>\postexhyphenchar</u>	\tex_postexhyphenchar:D	1023
1024	_kernel_primitive:NN	<u>\posthyphenchar</u>	\tex_posthyphenchar:D	1024
1025	_kernel_primitive:NN	<u>\prebinoppenalty</u>	\tex_prebinoppenalty:D	1025
1026	_kernel_primitive:NN	<u>\predisplaygapfactor</u>	\tex_predisplaygapfactor:D	1026
1027	_kernel_primitive:NN	<u>\preexhyphenchar</u>	\tex_preexhyphenchar:D	1027
1028	_kernel_primitive:NN	<u>\prehyphenchar</u>	\tex_prehyphenchar:D	1028
1029	_kernel_primitive:NN	<u>\prerelpenalty</u>	\tex_prerelpenalty:D	1029
1030	_kernel_primitive:NN	<u>\protrusionboundary</u>	\tex_protrusionboundary:D	1030
1031	_kernel_primitive:NN	<u>\rightghost</u>	\tex_rightghost:D	1031
1032	_kernel_primitive:NN	<u>\savecatcodetable</u>	\tex_savecatcodetable:D	1032
1033	_kernel_primitive:NN	<u>\scantextokens</u>	\tex_scantextokens:D	1033
1034	_kernel_primitive:NN	<u>\setfontid</u>	\tex_setfontid:D	1034
1035	_kernel_primitive:NN	<u>\shapemode</u>	\tex_shapemode:D	1035
1036	_kernel_primitive:NN	<u>\suppressifcsnameerror</u>	\tex_suppressifcsnameerror:D	1036
1037	_kernel_primitive:NN	<u>\suppresslongerror</u>	\tex_suppresslongerror:D	1037
1038	_kernel_primitive:NN	<u>\suppressmathparerror</u>	\tex_suppressmathparerror:D	1038
1039	_kernel_primitive:NN	<u>\suppressoutererror</u>	\tex_suppressoutererror:D	1039
1040	_kernel_primitive:NN	<u>\suppressprimitiveerror</u>		1040
1041		\tex_suppressprimitiveerror:D		1041
1042	_kernel_primitive:NN	<u>\textdir</u>	\tex_textdir:D	1042
1043	_kernel_primitive:NN	<u>\textdirection</u>	\tex_textdirection:D	1043
1044	_kernel_primitive:NN	<u>\toksapp</u>	\tex_toksapp:D	1044
1045	_kernel_primitive:NN	<u>\tokspre</u>	\tex_tokspre:D	1045
1046	_kernel_primitive:NN	<u>\tpack</u>	\tex_tpack:D	1046
1047	_kernel_primitive:NN	<u>\variablefam</u>	\tex_variablefam:D	1047
1048	_kernel_primitive:NN	<u>\vpack</u>	\tex_vpack:D	1048
1049	_kernel_primitive:NN	<u>\wordboundary</u>	\tex_wordboundary:D	1049
1050	_kernel_primitive:NN	<u>\xtoksapp</u>	\tex_xtoksapp:D	1050
1051	_kernel_primitive:NN	<u>\xtokspre</u>	\tex_xtokspre:D	1051
1052	_kernel_primitive:NN	<u>\adjustspacing</u>	\tex_adjustspacing:D	1052
1053	_kernel_primitive:NN	<u>\copyfont</u>	\tex_copyfont:D	1053
1054	_kernel_primitive:NN	<u>\draftmode</u>	\tex_draftmode:D	1054
1055	_kernel_primitive:NN	<u>\expandglyphsinfont</u>	\tex_fontexpand:D	1055
1056	_kernel_primitive:NN	<u>\ifabsdim</u>	\tex_ifabsdim:D	1056
1057	_kernel_primitive:NN	<u>\ifabsnum</u>	\tex_ifabsnum:D	1057
1058	_kernel_primitive:NN	<u>\ignoreligaturesinfont</u>	\tex_ignoreligaturesinfont:D	1058

1059	_kernel_primitive:NN	<u>\insertht</u>	\tex_insertht:D	1059
1060	_kernel_primitive:NN	<u>\lastsavedboxresourceindex</u>		1060
1061		\tex_pdflastxform:D		1061
1062	_kernel_primitive:NN	<u>\lastsavedimageresourceindex</u>		1062
1063		\tex_pdflastximage:D		1063
1064	_kernel_primitive:NN	<u>\lastsavedimageresourcepages</u>		1064
1065		\tex_pdflastximagepages:D		1065
1066	_kernel_primitive:NN	<u>\lastxpos</u>	\tex_lastxpos:D	1066
1067	_kernel_primitive:NN	<u>\lasttypos</u>	\tex_lasttypos:D	1067
1068	_kernel_primitive:NN	<u>\normaldeviate</u>	\tex_normaldeviate:D	1068
1069	_kernel_primitive:NN	<u>\outputmode</u>	\tex_pdfoutput:D	1069
1070	_kernel_primitive:NN	<u>\pageheight</u>	\tex_pageheight:D	1070
1071	_kernel_primitive:NN	<u>\pagewidth</u>	\tex_pagewidth:D	1071
1072	_kernel_primitive:NN	<u>\protrudechars</u>	\tex_protrudechars:D	1072
1073	_kernel_primitive:NN	<u>\pxdimen</u>	\tex_pxdimen:D	1073
1074	_kernel_primitive:NN	<u>\randomseed</u>	\tex_randomseed:D	1074
1075	_kernel_primitive:NN	<u>\useboxresource</u>	\tex_pdfrefxform:D	1075
1076	_kernel_primitive:NN	<u>\useimageresource</u>	\tex_pdfrefximage:D	1076
1077	_kernel_primitive:NN	<u>\savepos</u>	\tex_savepos:D	1077
1078	_kernel_primitive:NN	<u>\saveboxresource</u>	\tex_pdfxform:D	1078
1079	_kernel_primitive:NN	<u>\saveimageresource</u>	\tex_pdfximage:D	1079
1080	_kernel_primitive:NN	<u>\setrandomseed</u>	\tex_setrandomseed:D	1080
1081	_kernel_primitive:NN	<u>\tracingfonts</u>	\tex_tracingfonts:D	1081
1082	_kernel_primitive:NN	<u>\uniformdeviate</u>	\tex_uniformdeviate:D	1082
1083	_kernel_primitive:NN	<u>\Uchar</u>	\tex_Uchar:D	1083
1084	_kernel_primitive:NN	<u>\Ucharcat</u>	\tex_Ucharcat:D	1084
1085	_kernel_primitive:NN	<u>\Udelcode</u>	\tex_Udelcode:D	1085
1086	_kernel_primitive:NN	<u>\Udelcodenum</u>	\tex_Udelcodenum:D	1086
1087	_kernel_primitive:NN	<u>\Udelimiter</u>	\tex_Udelimiter:D	1087
1088	_kernel_primitive:NN	<u>\Udelimiterover</u>	\tex_Udelimiterover:D	1088
1089	_kernel_primitive:NN	<u>\Udelimiterunder</u>	\tex_Udelimiterunder:D	1089
1090	_kernel_primitive:NN	<u>\Uhextensible</u>	\tex_Uhextensible:D	1090
1091	_kernel_primitive:NN	<u>\Uleft</u>	\tex_Uleft:D	1091
1092	_kernel_primitive:NN	<u>\Umathaccent</u>	\tex_Umathaccent:D	1092
1093	_kernel_primitive:NN	<u>\Umathaxis</u>	\tex_Umathaxis:D	1093
1094	_kernel_primitive:NN	<u>\Umathbinbinspacing</u>	\tex_Umathbinbinspacing:D	1094
1095	_kernel_primitive:NN	<u>\Umathbinclosespacing</u>	\tex_Umathbinclosespacing:D	1095
1096	_kernel_primitive:NN	<u>\Umathbininnerspacing</u>	\tex_Umathbininnerspacing:D	1096
1097	_kernel_primitive:NN	<u>\Umathbinopenspacing</u>	\tex_Umathbinopenspacing:D	1097
1098	_kernel_primitive:NN	<u>\Umathbinopspacing</u>	\tex_Umathbinopspacing:D	1098
1099	_kernel_primitive:NN	<u>\Umathbinordspacing</u>	\tex_Umathbinordspacing:D	1099
1100	_kernel_primitive:NN	<u>\Umathbinpunctspacing</u>	\tex_Umathbinpunctspacing:D	1100
1101	_kernel_primitive:NN	<u>\Umathbinrelspacing</u>	\tex_Umathbinrelspacing:D	1101
1102	_kernel_primitive:NN	<u>\Umathchar</u>	\tex_Umathchar:D	1102
1103	_kernel_primitive:NN	<u>\Umathcharclass</u>	\tex_Umathcharclass:D	1103
1104	_kernel_primitive:NN	<u>\Umathchardef</u>	\tex_Umathchardef:D	1104

1105	_kernel_primitive:NN	<u>\Umathcharfam</u>	\tex_Umathcharfam:D	1105
1106	_kernel_primitive:NN	<u>\Umathcharnum</u>	\tex_Umathcharnum:D	1106
1107	_kernel_primitive:NN	<u>\Umathcharnumdef</u>	\tex_Umathcharnumdef:D	1107
1108	_kernel_primitive:NN	<u>\Umathcharslot</u>	\tex_Umathcharslot:D	1108
1109	_kernel_primitive:NN	<u>\Umathclosebinspacing</u>	\tex_Umathclosebinspacing:D	1109
1110	_kernel_primitive:NN	<u>\Umathcloseclosespacing</u>		1110
1111		\tex_Umathcloseclosespacing:D		1111
1112	_kernel_primitive:NN	<u>\Umathcloseinnerspacing</u>		1112
1113		\tex_Umathcloseinnerspacing:D		1113
1114	_kernel_primitive:NN	<u>\Umathcloseopenspacing</u>	\tex_Umathcloseopenspacing:D	1114
1115	_kernel_primitive:NN	<u>\Umathcloseopspacing</u>	\tex_Umathcloseopspacing:D	1115
1116	_kernel_primitive:NN	<u>\Umathcloseordspacing</u>	\tex_Umathcloseordspacing:D	1116
1117	_kernel_primitive:NN	<u>\Umathclosepunctspacing</u>		1117
1118		\tex_Umathclosepunctspacing:D		1118
1119	_kernel_primitive:NN	<u>\Umathcloserelspacing</u>	\tex_Umathcloserelspacing:D	1119
1120	_kernel_primitive:NN	<u>\Umathcode</u>	\tex_Umathcode:D	1120
1121	_kernel_primitive:NN	<u>\Umathcodenum</u>	\tex_Umathcodenum:D	1121
1122	_kernel_primitive:NN	<u>\Umathconnectoroverlapmin</u>		1122
1123		\tex_Umathconnectoroverlapmin:D		1123
1124	_kernel_primitive:NN	<u>\Umathfractiondelsize</u>	\tex_Umathfractiondelsize:D	1124
1125	_kernel_primitive:NN	<u>\Umathfractiondenomdown</u>		1125
1126		\tex_Umathfractiondenomdown:D		1126
1127	_kernel_primitive:NN	<u>\Umathfractiondenomvgap</u>		1127
1128		\tex_Umathfractiondenomvgap:D		1128
1129	_kernel_primitive:NN	<u>\Umathfractionnumup</u>	\tex_Umathfractionnumup:D	1129
1130	_kernel_primitive:NN	<u>\Umathfractionnumvgap</u>	\tex_Umathfractionnumvgap:D	1130
1131	_kernel_primitive:NN	<u>\Umathfractionrule</u>	\tex_Umathfractionrule:D	1131
1132	_kernel_primitive:NN	<u>\Umathinnerbinspacing</u>	\tex_Umathinnerbinspacing:D	1132
1133	_kernel_primitive:NN	<u>\Umathinnerclosespacing</u>		1133
1134		\tex_Umathinnerclosespacing:D		1134
1135	_kernel_primitive:NN	<u>\Umathinnerinnerspacing</u>		1135
1136		\tex_Umathinnerinnerspacing:D		1136
1137	_kernel_primitive:NN	<u>\Umathinneropenspacing</u>	\tex_Umathinneropenspacing:D	1137
1138	_kernel_primitive:NN	<u>\Umathinneropspacing</u>	\tex_Umathinneropspacing:D	1138
1139	_kernel_primitive:NN	<u>\Umathinnerordspacing</u>	\tex_Umathinnerordspacing:D	1139
1140	_kernel_primitive:NN	<u>\Umathinnerpunctspacing</u>		1140
1141		\tex_Umathinnerpunctspacing:D		1141
1142	_kernel_primitive:NN	<u>\Umathinnerrelspacing</u>	\tex_Umathinnerrelspacing:D	1142
1143	_kernel_primitive:NN	<u>\Umathlimitabovebgap</u>	\tex_Umathlimitabovebgap:D	1143
1144	_kernel_primitive:NN	<u>\Umathlimitabovekern</u>	\tex_Umathlimitabovekern:D	1144
1145	_kernel_primitive:NN	<u>\Umathlimitabovevgap</u>	\tex_Umathlimitabovevgap:D	1145
1146	_kernel_primitive:NN	<u>\Umathlimitbelowbgap</u>	\tex_Umathlimitbelowbgap:D	1146
1147	_kernel_primitive:NN	<u>\Umathlimitbelowkern</u>	\tex_Umathlimitbelowkern:D	1147
1148	_kernel_primitive:NN	<u>\Umathlimitbelowvgap</u>	\tex_Umathlimitbelowvgap:D	1148
1149	_kernel_primitive:NN	<u>\Umathnolimitsubfactor</u>	\tex_Umathnolimitsubfactor:D	1149
1150	_kernel_primitive:NN	<u>\Umathnolimitsupfactor</u>	\tex_Umathnolimitsupfactor:D	1150

1151	_kernel_primitive:NN	<u>\Umathopbinspacing</u>	\tex_Umathopbinspacing:D	1151
1152	_kernel_primitive:NN	<u>\Umathopclosespacing</u>	\tex_Umathopclosespacing:D	1152
1153	_kernel_primitive:NN	<u>\Umathopenbinspacing</u>	\tex_Umathopenbinspacing:D	1153
1154	_kernel_primitive:NN	<u>\Umathopenclosespacing</u>	\tex_Umathopenclosespacing:D	1154
1155	_kernel_primitive:NN	<u>\Umathopeninnerspacing</u>	\tex_Umathopeninnerspacing:D	1155
1156	_kernel_primitive:NN	<u>\Umathopenopenspacing</u>	\tex_Umathopenopenspacing:D	1156
1157	_kernel_primitive:NN	<u>\Umathopenopspacing</u>	\tex_Umathopenopspacing:D	1157
1158	_kernel_primitive:NN	<u>\Umathopenordspacing</u>	\tex_Umathopenordspacing:D	1158
1159	_kernel_primitive:NN	<u>\Umathopenpunctspacing</u>	\tex_Umathopenpunctspacing:D	1159
1160	_kernel_primitive:NN	<u>\Umathopenrelspacing</u>	\tex_Umathopenrelspacing:D	1160
1161	_kernel_primitive:NN	<u>\Umathoperatorsize</u>	\tex_Umathoperatorsize:D	1161
1162	_kernel_primitive:NN	<u>\Umathopinnerspacing</u>	\tex_Umathopinnerspacing:D	1162
1163	_kernel_primitive:NN	<u>\Umathopopenspacing</u>	\tex_Umathopopenspacing:D	1163
1164	_kernel_primitive:NN	<u>\Umathopopspacing</u>	\tex_Umathopopspacing:D	1164
1165	_kernel_primitive:NN	<u>\Umathopordspacing</u>	\tex_Umathopordspacing:D	1165
1166	_kernel_primitive:NN	<u>\Umathoppunctspacing</u>	\tex_Umathoppunctspacing:D	1166
1167	_kernel_primitive:NN	<u>\Umathoprelspacing</u>	\tex_Umathoprelspacing:D	1167
1168	_kernel_primitive:NN	<u>\Umathordbinspacing</u>	\tex_Umathordbinspacing:D	1168
1169	_kernel_primitive:NN	<u>\Umathordclosespacing</u>	\tex_Umathordclosespacing:D	1169
1170	_kernel_primitive:NN	<u>\Umathordinnerspacing</u>	\tex_Umathordinnerspacing:D	1170
1171	_kernel_primitive:NN	<u>\Umathordopenspacing</u>	\tex_Umathordopenspacing:D	1171
1172	_kernel_primitive:NN	<u>\Umathordopspacing</u>	\tex_Umathordopspacing:D	1172
1173	_kernel_primitive:NN	<u>\Umathordordspacing</u>	\tex_Umathordordspacing:D	1173
1174	_kernel_primitive:NN	<u>\Umathordpunctspacing</u>	\tex_Umathordpunctspacing:D	1174
1175	_kernel_primitive:NN	<u>\Umathordrelspacing</u>	\tex_Umathordrelspacing:D	1175
1176	_kernel_primitive:NN	<u>\Umathoverbarkern</u>	\tex_Umathoverbarkern:D	1176
1177	_kernel_primitive:NN	<u>\Umathoverbarrule</u>	\tex_Umathoverbarrule:D	1177
1178	_kernel_primitive:NN	<u>\Umathoverbarvgap</u>	\tex_Umathoverbarvgap:D	1178
1179	_kernel_primitive:NN	<u>\Umathoverdelimiterbgap</u>		1179
1180		\tex_Umathoverdelimiterbgap:D		1180
1181	_kernel_primitive:NN	<u>\Umathoverdelimitervgap</u>		1181
1182		\tex_Umathoverdelimitervgap:D		1182
1183	_kernel_primitive:NN	<u>\Umathpunctbinspacing</u>	\tex_Umathpunctbinspacing:D	1183
1184	_kernel_primitive:NN	<u>\Umathpunctclosespacing</u>		1184
1185		\tex_Umathpunctclosespacing:D		1185
1186	_kernel_primitive:NN	<u>\Umathpunctinnerspacing</u>		1186
1187		\tex_Umathpunctinnerspacing:D		1187
1188	_kernel_primitive:NN	<u>\Umathpunctopenspacing</u>	\tex_Umathpunctopenspacing:D	1188
1189	_kernel_primitive:NN	<u>\Umathpunctopspacing</u>	\tex_Umathpunctopspacing:D	1189
1190	_kernel_primitive:NN	<u>\Umathpunctordspacing</u>	\tex_Umathpunctordspacing:D	1190
1191	_kernel_primitive:NN	<u>\Umathpunctpunctspacing</u>		1191
1192		\tex_Umathpunctpunctspacing:D		1192
1193	_kernel_primitive:NN	<u>\Umathpunctrelspacing</u>	\tex_Umathpunctrelspacing:D	1193
1194	_kernel_primitive:NN	<u>\Umathquad</u>	\tex_Umathquad:D	1194
1195	_kernel_primitive:NN	<u>\Umathradicaldegreeafter</u>		1195
1196		\tex_Umathradicaldegreeafter:D		1196

1197	<code>__kernel_primitive:NN</code>	<code>\Umathradicaldegreebefore</code>		1197
1198		<code>\tex_Umathradicaldegreebefore:D</code>		1198
1199	<code>__kernel_primitive:NN</code>	<code>\Umathradicaldegreeraise</code>		1199
1200		<code>\tex_Umathradicaldegreeraise:D</code>		1200
1201	<code>__kernel_primitive:NN</code>	<code>\Umathradicalkern</code>	<code>\tex_Umathradicalkern:D</code>	1201
1202	<code>__kernel_primitive:NN</code>	<code>\Umathradicalrule</code>	<code>\tex_Umathradicalrule:D</code>	1202
1203	<code>__kernel_primitive:NN</code>	<code>\Umathradicalvgap</code>	<code>\tex_Umathradicalvgap:D</code>	1203
1204	<code>__kernel_primitive:NN</code>	<code>\Umathrelbinspacing</code>	<code>\tex_Umathrelbinspacing:D</code>	1204
1205	<code>__kernel_primitive:NN</code>	<code>\Umathrelclosespacing</code>	<code>\tex_Umathrelclosespacing:D</code>	1205
1206	<code>__kernel_primitive:NN</code>	<code>\Umathrelinnerspacing</code>	<code>\tex_Umathrelinnerspacing:D</code>	1206
1207	<code>__kernel_primitive:NN</code>	<code>\Umathrelopenspacing</code>	<code>\tex_Umathrelopenspacing:D</code>	1207
1208	<code>__kernel_primitive:NN</code>	<code>\Umathrelopspacing</code>	<code>\tex_Umathrelopspacing:D</code>	1208
1209	<code>__kernel_primitive:NN</code>	<code>\Umathrelordspacing</code>	<code>\tex_Umathrelordspacing:D</code>	1209
1210	<code>__kernel_primitive:NN</code>	<code>\Umathrelpunctspacing</code>	<code>\tex_Umathrelpunctspacing:D</code>	1210
1211	<code>__kernel_primitive:NN</code>	<code>\Umathrelrelspacing</code>	<code>\tex_Umathrelrelspacing:D</code>	1211
1212	<code>__kernel_primitive:NN</code>	<code>\Umathskewedfractionhgap</code>		1212
1213		<code>\tex_Umathskewedfractionhgap:D</code>		1213
1214	<code>__kernel_primitive:NN</code>	<code>\Umathskewedfractionvgap</code>		1214
1215		<code>\tex_Umathskewedfractionvgap:D</code>		1215
1216	<code>__kernel_primitive:NN</code>	<code>\Umathspaceafterscript</code>	<code>\tex_Umathspaceafterscript:D</code>	1216
1217	<code>__kernel_primitive:NN</code>	<code>\Umathstackdenomdown</code>	<code>\tex_Umathstackdenomdown:D</code>	1217
1218	<code>__kernel_primitive:NN</code>	<code>\Umathstacknumup</code>	<code>\tex_Umathstacknumup:D</code>	1218
1219	<code>__kernel_primitive:NN</code>	<code>\Umathstackvgap</code>	<code>\tex_Umathstackvgap:D</code>	1219
1220	<code>__kernel_primitive:NN</code>	<code>\Umathsubshiftdown</code>	<code>\tex_Umathsubshiftdown:D</code>	1220
1221	<code>__kernel_primitive:NN</code>	<code>\Umathsubshiftdrop</code>	<code>\tex_Umathsubshiftdrop:D</code>	1221
1222	<code>__kernel_primitive:NN</code>	<code>\Umathsubsupshiftdown</code>	<code>\tex_Umathsubsupshiftdown:D</code>	1222
1223	<code>__kernel_primitive:NN</code>	<code>\Umathsubsupvgap</code>	<code>\tex_Umathsubsupvgap:D</code>	1223
1224	<code>__kernel_primitive:NN</code>	<code>\Umathsubtopmax</code>	<code>\tex_Umathsubtopmax:D</code>	1224
1225	<code>__kernel_primitive:NN</code>	<code>\Umathsupbottommin</code>	<code>\tex_Umathsupbottommin:D</code>	1225
1226	<code>__kernel_primitive:NN</code>	<code>\Umathsupshiftdrop</code>	<code>\tex_Umathsupshiftdrop:D</code>	1226
1227	<code>__kernel_primitive:NN</code>	<code>\Umathsupshiftup</code>	<code>\tex_Umathsupshiftup:D</code>	1227
1228	<code>__kernel_primitive:NN</code>	<code>\Umathsupsubbottommax</code>	<code>\tex_Umathsupsubbottommax:D</code>	1228
1229	<code>__kernel_primitive:NN</code>	<code>\Umathunderbarkern</code>	<code>\tex_Umathunderbarkern:D</code>	1229
1230	<code>__kernel_primitive:NN</code>	<code>\Umathunderbarrule</code>	<code>\tex_Umathunderbarrule:D</code>	1230
1231	<code>__kernel_primitive:NN</code>	<code>\Umathunderbarvgap</code>	<code>\tex_Umathunderbarvgap:D</code>	1231
1232	<code>__kernel_primitive:NN</code>	<code>\Umathunderdelimiterbgap</code>		1232
1233		<code>\tex_Umathunderdelimiterbgap:D</code>		1233
1234	<code>__kernel_primitive:NN</code>	<code>\Umathunderdelimitervgap</code>		1234
1235		<code>\tex_Umathunderdelimitervgap:D</code>		1235
1236	<code>__kernel_primitive:NN</code>	<code>\Umiddle</code>	<code>\tex_Umiddle:D</code>	1236
1237	<code>__kernel_primitive:NN</code>	<code>\Unosubscript</code>	<code>\tex_Unosubscript:D</code>	1237
1238	<code>__kernel_primitive:NN</code>	<code>\Unosuperscript</code>	<code>\tex_Unosuperscript:D</code>	1238
1239	<code>__kernel_primitive:NN</code>	<code>\Uoverdelimiter</code>	<code>\tex_Uoverdelimiter:D</code>	1239
1240	<code>__kernel_primitive:NN</code>	<code>\Uradical</code>	<code>\tex_Uradical:D</code>	1240
1241	<code>__kernel_primitive:NN</code>	<code>\Uright</code>	<code>\tex_Uright:D</code>	1241
1242	<code>__kernel_primitive:NN</code>	<code>\Uroot</code>	<code>\tex_Uroot:D</code>	1242

1243	_kernel_primitive:NN	<u>\Uskewed</u>	\tex_Uskewed:D	1243
1244	_kernel_primitive:NN	<u>\Uskewedwithdelims</u>	\tex_Uskewedwithdelims:D	1244
1245	_kernel_primitive:NN	<u>\Ustack</u>	\tex_Ustack:D	1245
1246	_kernel_primitive:NN	<u>\Ustartdisplaymath</u>	\tex_Ustartdisplaymath:D	1246
1247	_kernel_primitive:NN	<u>\Ustartmath</u>	\tex_Ustartmath:D	1247
1248	_kernel_primitive:NN	<u>\Ustopdisplaymath</u>	\tex_Ustopdisplaymath:D	1248
1249	_kernel_primitive:NN	<u>\Ustopmath</u>	\tex_Ustopmath:D	1249
1250	_kernel_primitive:NN	<u>\Usubscript</u>	\tex_Usubscript:D	1250
1251	_kernel_primitive:NN	<u>\Usuperscript</u>	\tex_Usuperscript:D	1251
1252	_kernel_primitive:NN	<u>\Uunderdelimater</u>	\tex_Uunderdelimater:D	1252
1253	_kernel_primitive:NN	<u>\Uvextensible</u>	\tex_Uvextensible:D	1253
1254	_kernel_primitive:NN	<u>\autospacing</u>	\tex_autospacing:D	1254
1255	_kernel_primitive:NN	<u>\autoxspacing</u>	\tex_autoxspacing:D	1255
1256	_kernel_primitive:NN	<u>\currentcjktoken</u>	\tex_currentcjktoken:D	1256
1257	_kernel_primitive:NN	<u>\currentspacingmode</u>	\tex_currentspacingmode:D	1257
1258	_kernel_primitive:NN	<u>\currentxspacingmode</u>	\tex_currentxspacingmode:D	1258
1259	_kernel_primitive:NN	<u>\disinhibitglue</u>	\tex_disinhibitglue:D	1259
1260	_kernel_primitive:NN	<u>\dtou</u>	\tex_dtou:D	1260
1261	_kernel_primitive:NN	<u>\epTeXinputencoding</u>	\tex_epTeXinputencoding:D	1261
1262	_kernel_primitive:NN	<u>\epTeXversion</u>	\tex_epTeXversion:D	1262
1263	_kernel_primitive:NN	<u>\euc</u>	\tex_euc:D	1263
1264	_kernel_primitive:NN	<u>\hfi</u>	\tex_hfi:D	1264
1265	_kernel_primitive:NN	<u>\ifdbbox</u>	\tex_ifdbbox:D	1265
1266	_kernel_primitive:NN	<u>\ifddir</u>	\tex_ifddir:D	1266
1267	_kernel_primitive:NN	<u>\ifjfont</u>	\tex_ifjfont:D	1267
1268	_kernel_primitive:NN	<u>\ifmbox</u>	\tex_ifmbox:D	1268
1269	_kernel_primitive:NN	<u>\ifmdir</u>	\tex_ifmdir:D	1269
1270	_kernel_primitive:NN	<u>\iftbox</u>	\tex_iftbox:D	1270
1271	_kernel_primitive:NN	<u>\iftfont</u>	\tex_iftfont:D	1271
1272	_kernel_primitive:NN	<u>\iftdir</u>	\tex_iftdir:D	1272
1273	_kernel_primitive:NN	<u>\ifybox</u>	\tex_ifybox:D	1273
1274	_kernel_primitive:NN	<u>\ifydir</u>	\tex_ifydir:D	1274
1275	_kernel_primitive:NN	<u>\inhibitglue</u>	\tex_inhibitglue:D	1275
1276	_kernel_primitive:NN	<u>\inhibitxspcode</u>	\tex_inhibitxspcode:D	1276
1277	_kernel_primitive:NN	<u>\jcharwidowpenalty</u>	\tex_jcharwidowpenalty:D	1277
1278	_kernel_primitive:NN	<u>\jfam</u>	\tex_jfam:D	1278
1279	_kernel_primitive:NN	<u>\jfont</u>	\tex_jfont:D	1279
1280	_kernel_primitive:NN	<u>\jis</u>	\tex_jis:D	1280
1281	_kernel_primitive:NN	<u>\kanjiskip</u>	\tex_kanjiskip:D	1281
1282	_kernel_primitive:NN	<u>\kansuji</u>	\tex_kansuji:D	1282
1283	_kernel_primitive:NN	<u>\kansujichar</u>	\tex_kansujichar:D	1283
1284	_kernel_primitive:NN	<u>\kcatcode</u>	\tex_kcatcode:D	1284
1285	_kernel_primitive:NN	<u>\kuten</u>	\tex_kuten:D	1285
1286	_kernel_primitive:NN	<u>\lastnodechar</u>	\tex_lastnodechar:D	1286
1287	_kernel_primitive:NN	<u>\lastnodefont</u>	\tex_lastnodefont:D	1287
1288	_kernel_primitive:NN	<u>\lastnodesubtype</u>	\tex_lastnodesubtype:D	1288

1289	_kernel_primitive:NN	<u>\noautospaceing</u>	\tex_noautospaceing:D	1289
1290	_kernel_primitive:NN	<u>\noautoxspaceing</u>	\tex_noautoxspaceing:D	1290
1291	_kernel_primitive:NN	<u>\pagefistretch</u>	\tex_pagefistretch:D	1291
1292	_kernel_primitive:NN	<u>\postbreakpenalty</u>	\tex_postbreakpenalty:D	1292
1293	_kernel_primitive:NN	<u>\prebreakpenalty</u>	\tex_prebreakpenalty:D	1293
1294	_kernel_primitive:NN	<u>\ptexfontname</u>	\tex_ptexfontname:D	1294
1295	_kernel_primitive:NN	<u>\ptexlineendmode</u>	\tex_lineendmode:D	1295
1296	_kernel_primitive:NN	<u>\ptexminorversion</u>	\tex_ptexminorversion:D	1296
1297	_kernel_primitive:NN	<u>\ptexrevision</u>	\tex_ptexrevision:D	1297
1298	_kernel_primitive:NN	<u>\ptextracingfonts</u>	\tex_ptextracingfonts:D	1298
1299	_kernel_primitive:NN	<u>\ptexversion</u>	\tex_ptexversion:D	1299
1300	_kernel_primitive:NN	<u>\readpapersizespecial</u>	\tex_readpapersizespecial:D	1300
1301	_kernel_primitive:NN	<u>\scriptbaselineshiftfactor</u>		1301
1302		\tex_scriptbaselineshiftfactor:D		1302
1303	_kernel_primitive:NN	<u>\scriptscriptbaselineshiftfactor</u>		1303
1304		\tex_scriptscriptbaselineshiftfactor:D		1304
1305	_kernel_primitive:NN	<u>\showmode</u>	\tex_showmode:D	1305
1306	_kernel_primitive:NN	<u>\sjis</u>	\tex_sjis:D	1306
1307	_kernel_primitive:NN	<u>\tate</u>	\tex_tate:D	1307
1308	_kernel_primitive:NN	<u>\tbaselineshift</u>	\tex_tbaselineshift:D	1308
1309	_kernel_primitive:NN	<u>\textbaselineshiftfactor</u>		1309
1310		\tex_textbaselineshiftfactor:D		1310
1311	_kernel_primitive:NN	<u>\tfont</u>	\tex_tfont:D	1311
1312	_kernel_primitive:NN	<u>\tojis</u>	\tex_tojis:D	1312
1313	_kernel_primitive:NN	<u>\toucs</u>	\tex_toucs:D	1313
1314	_kernel_primitive:NN	<u>\ucs</u>	\tex_ucs:D	1314
1315	_kernel_primitive:NN	<u>\xkanjiskip</u>	\tex_xkanjiskip:D	1315
1316	_kernel_primitive:NN	<u>\xspcode</u>	\tex_xspcode:D	1316
1317	_kernel_primitive:NN	<u>\ybaselineshift</u>	\tex_ybaselineshift:D	1317
1318	_kernel_primitive:NN	<u>\yoko</u>	\tex_yoko:D	1318
1319	_kernel_primitive:NN	<u>\vfi</u>	\tex_vfi:D	1319
1320	_kernel_primitive:NN	<u>\disablecjktoken</u>	\tex_disablecjktoken:D	1320
1321	_kernel_primitive:NN	<u>\enablecjktoken</u>	\tex_enablecjktoken:D	1321
1322	_kernel_primitive:NN	<u>\forcecjktoken</u>	\tex_forcecjktoken:D	1322
1323	_kernel_primitive:NN	<u>\kchar</u>	\tex_kchar:D	1323
1324	_kernel_primitive:NN	<u>\kchardef</u>	\tex_kchardef:D	1324
1325	_kernel_primitive:NN	<u>\uptexrevision</u>	\tex_uptexrevision:D	1325
1326	_kernel_primitive:NN	<u>\uptexversion</u>	\tex_uptexversion:D	1326
1327	_kernel_primitive:NN	<u>\odelcode</u>	\tex_odelcode:D	1327
1328	_kernel_primitive:NN	<u>\odelimiter</u>	\tex_odelimiter:D	1328
1329	_kernel_primitive:NN	<u>\omathaccent</u>	\tex_omathaccent:D	1329
1330	_kernel_primitive:NN	<u>\omathchar</u>	\tex_omathchar:D	1330
1331	_kernel_primitive:NN	<u>\omathchardef</u>	\tex_omathchardef:D	1331
1332	_kernel_primitive:NN	<u>\omathcode</u>	\tex_omathcode:D	1332
1333	_kernel_primitive:NN	<u>\oradical</u>	\tex_oradical:D	1333
1334	_kernel_primitive:NN	<u>\ignoreprimitiveerror</u>	\tex_ignoreprimitiveerror:D	1334


```

1335 \__kernel_primitive:NN \partokencontext \tex_partokencontext:D 1335
1336 \__kernel_primitive:NN \partokenname \tex_partokenname:D 1336
1337 \__kernel_primitive:NN \showstream \tex_showstream:D 1337
1338 \__kernel_primitive:NN \tracingstacklevels \tex_tracingstacklevels:D 1338
1339 \tex_endgroup:D 1339
1340 \tex_ifdefined:D \@@end 1340
1341 \tex_let:D \tex_end:D \@@end 1341
1342 \tex_let:D \tex_input:D \@@input 1342
1343 \tex_fi:D 1343
1344 \tex_ifdefined:D \@@hyph 1344
1345 \tex_let:D \tex_everydisplay:D \frozen@everydisplay 1345
1346 \tex_let:D \tex_everymath:D \frozen@everymath 1346
1347 \tex_let:D \tex_hyphen:D \@@hyph 1347
1348 \tex_let:D \tex_italiccorrection:D \@@italiccorr 1348
1349 \tex_let:D \tex_underline:D \@@underline 1349
1350 \tex_ifdefined:D \@@shipout 1350
1351 \tex_let:D \tex_shipout:D \@@shipout 1351
1352 \tex_fi:D 1352
1353 \tex_begingroup:D 1353
1354 \tex_edef:D \l_tmpa_tl { \tex_string:D \shipout } 1354
1355 \tex_edef:D \l_tmpb_tl { \tex_meaning:D \shipout } 1355
1356 \tex_ifx:D \l_tmpa_tl \l_tmpb_tl 1356
1357 \tex_else:D 1357
1358 \tex_expandafter:D \@tfor \tex_expandafter:D \@tempa \tex_string:D := 1358
1359 \CROP@shipout 1359
1360 \dup@shipout 1360
1361 \GPTorg@shipout 1361
1362 \LL@shipout 1362
1363 \mem@oldshipout 1363
1364 \opem@shipout 1364
1365 \pgfpages@originalshipout 1365
1366 \pr@shipout 1366
1367 \Shipout 1367
1368 \verso@orig@shipout 1368
1369 \do 1369
1370 { 1370
1371 \tex_edef:D \l_tmpb_tl 1371
1372 { \tex_expandafter:D \tex_meaning:D \@tempa } 1372
1373 \tex_ifx:D \l_tmpa_tl \l_tmpb_tl 1373
1374 \tex_global:D \tex_expandafter:D \tex_let:D 1374
1375 \tex_expandafter:D \tex_shipout:D \@tempa 1375
1376 \tex_fi:D 1376
1377 } 1377
1378 \tex_fi:D 1378
1379 \tex_endgroup:D 1379
1380 \tex_let:D \tex_tracingfonts:D \tex_undefined:D 1380

```

1381	\tex_ifdefined:D \pdftracingfonts	1381
1382	\tex_let:D \tex_tracingfonts:D \pdftracingfonts	1382
1383	\tex_else:D	1383
1384	\tex_ifdefined:D \tex_directlua:D	1384
1385	\tex_directlua:D { tex.enableprimitives("@@", {"tracingfonts"}) }	1385
1386	\tex_let:D \tex_tracingfonts:D \@tracingfonts	1386
1387	\tex_fi:D	1387
1388	\tex_fi:D	1388
1389	\tex_fi:D	1389
1390	\tex_ifnum:D 0	1390
1391	\tex_ifdefined:D \tex_pdftexversion:D 1 \tex_fi:D	1391
1392	\tex_ifdefined:D \tex luatexversion:D 1 \tex_fi:D	1392
1393	= 0 %	1393
1394	\tex_let:D \tex_pdfmapfile:D \tex_undefined:D	1394
1395	\tex_let:D \tex_pdfmapline:D \tex_undefined:D	1395
1396	\tex_fi:D	1396
1397	\tex_begingroup:D	1397
1398	\tex_edef:D \l_tmpa_tl { \tex_meaning:D \tex_time:D }	1398
1399	\tex_edef:D \l_tmpb_tl { \tex_string:D \time }	1399
1400	\tex_ifx:D \l_tmpa_tl \l_tmpb_tl	1400
1401	\tex_else:D	1401
1402	\tex_global:D \tex_let:D \tex_time:D \tex_undefined:D	1402
1403	\tex_fi:D	1403
1404	\tex_edef:D \l_tmpa_tl { \tex_meaning:D \tex_day:D }	1404
1405	\tex_edef:D \l_tmpb_tl { \tex_string:D \day }	1405
1406	\tex_ifx:D \l_tmpa_tl \l_tmpb_tl	1406
1407	\tex_else:D	1407
1408	\tex_global:D \tex_let:D \tex_day:D \tex_undefined:D	1408
1409	\tex_fi:D	1409
1410	\tex_edef:D \l_tmpa_tl { \tex_meaning:D \tex_month:D }	1410
1411	\tex_edef:D \l_tmpb_tl { \tex_string:D \month }	1411
1412	\tex_ifx:D \l_tmpa_tl \l_tmpb_tl	1412
1413	\tex_else:D	1413
1414	\tex_global:D \tex_let:D \tex_month:D \tex_undefined:D	1414
1415	\tex_fi:D	1415
1416	\tex_edef:D \l_tmpa_tl { \tex_meaning:D \tex_year:D }	1416
1417	\tex_edef:D \l_tmpb_tl { \tex_string:D \year }	1417
1418	\tex_ifx:D \l_tmpa_tl \l_tmpb_tl	1418
1419	\tex_else:D	1419
1420	\tex_global:D \tex_let:D \tex_year:D \tex_undefined:D	1420
1421	\tex_fi:D	1421
1422	\tex_endgroup:D	1422
1423	\tex_ifdefined:D \orieveryjob	1423
1424	\tex_let:D \tex_everyjob:D \orieveryjob	1424
1425	\tex_fi:D	1425
1426	\tex_ifdefined:D \oripdfoutput	1426

1427	\tex_let:D \tex_pdfoutput:D \oripdfoutput	1427
1428	\tex_fi:D	1428
1429	\tex_ifdefined:D \normalend	1429
1430	\tex_let:D \tex_end:D \normalend	1430
1431	\tex_let:D \tex_everyjob:D \normaleveryjob	1431
1432	\tex_let:D \tex_input:D \normalinput	1432
1433	\tex_let:D \tex_language:D \normallanguage	1433
1434	\tex_let:D \tex_mathop:D \normalmathop	1434
1435	\tex_let:D \tex_month:D \normalmonth	1435
1436	\tex_let:D \tex_outer:D \normalouter	1436
1437	\tex_let:D \tex_over:D \normalover	1437
1438	\tex_let:D \tex_vcenter:D \normalvcenter	1438
1439	\tex_let:D \tex_unexpanded:D \normalunexpanded	1439
1440	\tex_let:D \tex_expanded:D \normalexpanded	1440
1441	\tex_fi:D	1441
1442	\tex_ifdefined:D \normalitaliccorrection	1442
1443	\tex_let:D \tex_hoffset:D \normalhoffset	1443
1444	\tex_let:D \tex_italiccorrection:D \normalitaliccorrection	1444
1445	\tex_let:D \tex_voffset:D \normalvoffset	1445
1446	\tex_let:D \tex_showtokens:D \normalshowtokens	1446
1447	\tex_let:D \tex_bodydir:D \spac_directions_normal_body_dir	1447
1448	\tex_let:D \tex_pagedir:D \spac_directions_normal_page_dir	1448
1449	\tex_fi:D	1449
1450	\tex_ifdefined:D \normalleft	1450
1451	\tex_let:D \tex_left:D \normalleft	1451
1452	\tex_let:D \tex_middle:D \normalmiddle	1452
1453	\tex_let:D \tex_right:D \normalright	1453
1454	\tex_fi:D	1454
1455	%% File: l3basics.dtx	1455
1456	\tex_global:D \tex_let:D \if_true: \tex_iftrue:D	1456
1457	\tex_global:D \tex_let:D \if_false: \tex_iffalse:D	1457
1458	\tex_global:D \tex_let:D \or: \tex_or:D	1458
1459	\tex_global:D \tex_let:D \else: \tex_else:D	1459
1460	\tex_global:D \tex_let:D \fi: \tex_fi:D	1460
1461	\tex_global:D \tex_let:D \reverse_if:N \tex_unless:D	1461
1462	\tex_global:D \tex_let:D \if:w \tex_if:D	1462
1463	\tex_global:D \tex_let:D \if_charcode:w \tex_if:D	1463
1464	\tex_global:D \tex_let:D \if_catcode:w \tex_ifcat:D	1464
1465	\tex_global:D \tex_let:D \if_meaning:w \tex_ifx:D	1465
1466	\tex_global:D \tex_let:D \if_bool:N \tex_ifodd:D	1466
1467	\tex_global:D \tex_let:D \if_mode_math: \tex_ifmmode:D	1467
1468	\tex_global:D \tex_let:D \if_mode_horizontal: \tex_ifhmode:D	1468
1469	\tex_global:D \tex_let:D \if_mode_vertical: \tex_ifvmode:D	1469
1470	\tex_global:D \tex_let:D \if_mode_inner: \tex_ifinner:D	1470
1471	\tex_global:D \tex_let:D \if_cs_exist:N \tex_ifdefined:D	1471
1472	\tex_global:D \tex_let:D \if_cs_exist:w \tex_ifcsname:D	1472

```

1473 \tex_global:D \tex_let:D \cs:w \tex_csname:D 1473
1474 \tex_global:D \tex_let:D \cs_end: \tex_endcsname:D 1474
1475 \tex_global:D \tex_let:D \exp_after:wN \tex_expandafter:D 1475
1476 \tex_global:D \tex_let:D \exp_not:N \tex_noexpand:D 1476
1477 \tex_global:D \tex_let:D \exp_not:n \tex_unexpanded:D 1477
1478 \tex_global:D \tex_let:D \exp:w \tex_romannumeral:D 1478
1479 \tex_global:D \tex_chardef:D \exp_end: = 0 ~ 1479
1480 \tex_global:D \tex_let:D \token_to_meaning:N \tex_meaning:D 1480
1481 \tex_global:D \tex_let:D \cs_meaning:N \tex_meaning:D 1481
1482 \tex_global:D \tex_let:D \tl_to_str:n \tex_detokenize:D 1482
1483 \tex_global:D \tex_let:D \token_to_str:N \tex_string:D 1483
1484 \tex_global:D \tex_let:D \__kernel_tl_to_str:w \tex_detokenize:D 1484
1485 \tex_global:D \tex_let:D \scan_stop: \tex_relax:D 1485
1486 \tex_global:D \tex_let:D \group_begin: \tex_begingroup:D 1486
1487 \tex_global:D \tex_let:D \group_end: \tex_endgroup:D 1487
1488 \tex_global:D \tex_let:D \if_int_compare:w \tex_ifnum:D 1488
1489 \tex_global:D \tex_let:D \__int_to_roman:w \tex_romannumeral:D 1489
1490 \tex_global:D \tex_let:D \group_insert_after:N \tex_aftergroup:D 1490
1491 \tex_long:D \tex_gdef:D \exp_args:Nc #1#2 1491
1492 { \exp_after:wN #1 \cs:w #2 \cs_end: } 1492
1493 \tex_long:D \tex_gdef:D \exp_args:cc #1#2 1493
1494 { \cs:w #1 \exp_after:wN \cs_end: \cs:w #2 \cs_end: } 1494
1495 \tex_gdef:D \token_to_str:c { \exp_args:Nc \token_to_str:N } 1495
1496 \tex_long:D \tex_gdef:D \cs_meaning:c #1 1496
1497 { 1497
1498 \if_cs_exist:w #1 \cs_end: 1498
1499 \exp_after:wN \use_i:nn 1499
1500 \else: 1500
1501 \exp_after:wN \use_ii:nn 1501
1502 \fi: 1502
1503 { \exp_args:Nc \cs_meaning:N {#1} } 1503
1504 { \tl_to_str:n {undefined} } 1504
1505 } 1505
1506 \tex_global:D \tex_let:D \token_to_meaning:c = \cs_meaning:c 1506
1507 \tex_global:D \tex_chardef:D \c_zero_int = 0 ~ 1507
1508 \tex_ifdefined:D \tex_luatexversion:D 1508
1509 \tex_global:D \tex_chardef:D \c_max_register_int = 65 535 ~ 1509
1510 \tex_else:D 1510
1511 \tex_ifdefined:D \tex_omathchardef:D 1511
1512 \tex_global:D \tex_omathchardef:D \c_max_register_int = 65535 ~ 1512
1513 \tex_else:D 1513
1514 \tex_global:D \tex_mathchardef:D \c_max_register_int = 32767 ~ 1514
1515 \tex_fi:D 1515
1516 \tex_fi:D 1516
1517 \tex_global:D \tex_let:D \cs_gset_nopar:Npn \tex_gdef:D 1517
1518 \tex_global:D \tex_let:D \cs_gset_nopar:Npe \tex_xdef:D 1518

```

1519	\tex_global:D \tex_let:D \cs_gset_nopar:Npx	\tex_xdef:D	1519
1520	\tex_protected:D \tex_long:D \tex_gdef:D \cs_gset:Npn		1520
1521	{ \tex_long:D \tex_gdef:D }		1521
1522	\tex_protected:D \tex_long:D \tex_gdef:D \cs_gset:Npe		1522
1523	{ \tex_long:D \tex_xdef:D }		1523
1524	\tex_global:D \tex_let:D \cs_gset:Npx \cs_gset:Npe		1524
1525	\tex_protected:D \tex_long:D \tex_gdef:D \cs_gset_protected_nopar:Npn		1525
1526	{ \tex_protected:D \tex_gdef:D }		1526
1527	\tex_protected:D \tex_long:D \tex_gdef:D \cs_gset_protected_nopar:Npe		1527
1528	{ \tex_protected:D \tex_xdef:D }		1528
1529	\tex_global:D \tex_let:D \cs_gset_protected_nopar:Npx \cs_gset_protected_nopar:Npe		1529
1530	\tex_protected:D \tex_long:D \tex_gdef:D \cs_gset_protected:Npn		1530
1531	{ \tex_protected:D \tex_long:D \tex_gdef:D }		1531
1532	\tex_protected:D \tex_long:D \tex_gdef:D \cs_gset_protected:Npe		1532
1533	{ \tex_protected:D \tex_long:D \tex_xdef:D }		1533
1534	\tex_global:D \tex_let:D \cs_gset_protected:Npx \cs_gset_protected:Npe		1534
1535	\tex_global:D \tex_let:D \cs_set_nopar:Npn	\tex_def:D	1535
1536	\tex_global:D \tex_let:D \cs_set_nopar:Npe	\tex_edef:D	1536
1537	\tex_global:D \tex_let:D \cs_set_nopar:Npx	\tex_edef:D	1537
1538	\cs_gset_protected:Npn \cs_set:Npn		1538
1539	{ \tex_long:D \tex_def:D }		1539
1540	\cs_gset_protected:Npn \cs_set:Npe		1540
1541	{ \tex_long:D \tex_edef:D }		1541
1542	\tex_global:D \tex_let:D \cs_set:Npx \cs_set:Npe		1542
1543	\cs_gset_protected:Npn \cs_set_protected_nopar:Npn		1543
1544	{ \tex_protected:D \tex_def:D }		1544
1545	\cs_gset_protected:Npn \cs_set_protected_nopar:Npe		1545
1546	{ \tex_protected:D \tex_edef:D }		1546
1547	\tex_global:D \tex_let:D \cs_set_protected_nopar:Npx \cs_set_protected_nopar:Npe		1547
1548	\cs_gset_protected:Npn \cs_set_protected:Npn		1548
1549	{ \tex_protected:D \tex_long:D \tex_def:D }		1549
1550	\cs_gset_protected:Npn \cs_set_protected:Npe		1550
1551	{ \tex_protected:D \tex_long:D \tex_edef:D }		1551
1552	\tex_global:D \tex_let:D \cs_set_protected:Npx \cs_set_protected:Npe		1552
1553	\cs_gset_nopar:Npn \l__exp_internal_tl { }		1553
1554	\cs_gset:Npn \use:c #1 { \cs:w #1 \cs_end: }		1554
1555	\cs_gset_protected:Npn \use:x #1		1555
1556	{		1556
1557	\cs_set_nopar:Npx \l__exp_internal_tl {#1}		1557
1558	\l__exp_internal_tl		1558
1559	}		1559
1560	\cs_gset:Npn \use:e #1 { \tex_expanded:D {#1} }		1560
1561	\cs_gset:Npn \use:n #1 {#1}		1561
1562	\cs_gset:Npn \use:nn #1#2 {#1#2}		1562
1563	\cs_gset:Npn \use:nnn #1#2#3 {#1#2#3}		1563
1564	\cs_gset:Npn \use:nnnn #1#2#3#4 {#1#2#3#4}		1564

1565	\cs_gset:Npn \use_i:nn	#1#2 {#1}	1565
1566	\cs_gset:Npn \use_ii:nn	#1#2 {#2}	1566
1567	\cs_gset:Npn \use_i:nnn	#1#2#3 {#1}	1567
1568	\cs_gset:Npn \use_ii:nnn	#1#2#3 {#2}	1568
1569	\cs_gset:Npn \use_iii:nnn	#1#2#3 {#3}	1569
1570	\cs_gset:Npn \use_i:nnnn	#1#2#3#4 {#1}	1570
1571	\cs_gset:Npn \use_ii:nnnn	#1#2#3#4 {#2}	1571
1572	\cs_gset:Npn \use_iii:nnnn	#1#2#3#4 {#3}	1572
1573	\cs_gset:Npn \use_iv:nnnn	#1#2#3#4 {#4}	1573
1574	\cs_gset:Npn \use_i:nnnnn	#1#2#3#4#5 {#1}	1574
1575	\cs_gset:Npn \use_ii:nnnnn	#1#2#3#4#5 {#2}	1575
1576	\cs_gset:Npn \use_iii:nnnnn	#1#2#3#4#5 {#3}	1576
1577	\cs_gset:Npn \use_iv:nnnnn	#1#2#3#4#5 {#4}	1577
1578	\cs_gset:Npn \use_v:nnnnn	#1#2#3#4#5 {#5}	1578
1579	\cs_gset:Npn \use_i:nnnnnn	#1#2#3#4#5#6 {#1}	1579
1580	\cs_gset:Npn \use_ii:nnnnnn	#1#2#3#4#5#6 {#2}	1580
1581	\cs_gset:Npn \use_iii:nnnnnn	#1#2#3#4#5#6 {#3}	1581
1582	\cs_gset:Npn \use_iv:nnnnnn	#1#2#3#4#5#6 {#4}	1582
1583	\cs_gset:Npn \use_v:nnnnnn	#1#2#3#4#5#6 {#5}	1583
1584	\cs_gset:Npn \use_vi:nnnnnn	#1#2#3#4#5#6 {#6}	1584
1585	\cs_gset:Npn \use_i:nnnnnnn	#1#2#3#4#5#6#7 {#1}	1585
1586	\cs_gset:Npn \use_ii:nnnnnnn	#1#2#3#4#5#6#7 {#2}	1586
1587	\cs_gset:Npn \use_iii:nnnnnnn	#1#2#3#4#5#6#7 {#3}	1587
1588	\cs_gset:Npn \use_iv:nnnnnnn	#1#2#3#4#5#6#7 {#4}	1588
1589	\cs_gset:Npn \use_v:nnnnnnn	#1#2#3#4#5#6#7 {#5}	1589
1590	\cs_gset:Npn \use_vi:nnnnnnn	#1#2#3#4#5#6#7 {#6}	1590
1591	\cs_gset:Npn \use_vii:nnnnnnn	#1#2#3#4#5#6#7 {#7}	1591
1592	\cs_gset:Npn \use_i:nnnnnnnn	#1#2#3#4#5#6#7#8 {#1}	1592
1593	\cs_gset:Npn \use_ii:nnnnnnnn	#1#2#3#4#5#6#7#8 {#2}	1593
1594	\cs_gset:Npn \use_iii:nnnnnnnn	#1#2#3#4#5#6#7#8 {#3}	1594
1595	\cs_gset:Npn \use_iv:nnnnnnnn	#1#2#3#4#5#6#7#8 {#4}	1595
1596	\cs_gset:Npn \use_v:nnnnnnnn	#1#2#3#4#5#6#7#8 {#5}	1596
1597	\cs_gset:Npn \use_vi:nnnnnnnn	#1#2#3#4#5#6#7#8 {#6}	1597
1598	\cs_gset:Npn \use_vii:nnnnnnnn	#1#2#3#4#5#6#7#8 {#7}	1598
1599	\cs_gset:Npn \use_viii:nnnnnnnn	#1#2#3#4#5#6#7#8 {#8}	1599
1600	\cs_gset:Npn \use_i:nnnnnnnnn	#1#2#3#4#5#6#7#8#9 {#1}	1600
1601	\cs_gset:Npn \use_ii:nnnnnnnnn	#1#2#3#4#5#6#7#8#9 {#2}	1601
1602	\cs_gset:Npn \use_iii:nnnnnnnnn	#1#2#3#4#5#6#7#8#9 {#3}	1602
1603	\cs_gset:Npn \use_iv:nnnnnnnnn	#1#2#3#4#5#6#7#8#9 {#4}	1603
1604	\cs_gset:Npn \use_v:nnnnnnnnn	#1#2#3#4#5#6#7#8#9 {#5}	1604
1605	\cs_gset:Npn \use_vi:nnnnnnnnn	#1#2#3#4#5#6#7#8#9 {#6}	1605
1606	\cs_gset:Npn \use_vii:nnnnnnnnn	#1#2#3#4#5#6#7#8#9 {#7}	1606
1607	\cs_gset:Npn \use_viii:nnnnnnnnn	#1#2#3#4#5#6#7#8#9 {#8}	1607
1608	\cs_gset:Npn \use_ix:nnnnnnnnn	#1#2#3#4#5#6#7#8#9 {#9}	1608
1609	\cs_gset:Npn \use_i_ii:nnn	#1#2#3 {#1#2}	1609
1610	\cs_gset:Npn \use_ii_i:nn	#1#2 { #2 #1 }	1610

```
1611 \cs_gset:Npn \use_none_delimit_by_q_nil:w #1 \q_nil { } 1611
1612 \cs_gset:Npn \use_none_delimit_by_q_stop:w #1 \q_stop { } 1612
1613 \cs_gset:Npn \use_none_delimit_by_q_recursion_stop:w #1 \q_recursion_stop { } 1613
1614 \cs_gset:Npn \use_i_delimit_by_q_nil:nw #1#2 \q_nil {#1} 1614
1615 \cs_gset:Npn \use_i_delimit_by_q_stop:nw #1#2 \q_stop {#1} 1615
1616 \cs_gset:Npn \use_i_delimit_by_q_recursion_stop:nw 1616
1617 #1#2 \q_recursion_stop {#1} 1617
1618 \cs_gset:Npn \use_none:n #1 { } 1618
1619 \cs_gset:Npn \use_none:nn #1#2 { } 1619
1620 \cs_gset:Npn \use_none:nnn #1#2#3 { } 1620
1621 \cs_gset:Npn \use_none:nnnn #1#2#3#4 { } 1621
1622 \cs_gset:Npn \use_none:nnnnn #1#2#3#4#5 { } 1622
1623 \cs_gset:Npn \use_none:nnnnnn #1#2#3#4#5#6 { } 1623
1624 \cs_gset:Npn \use_none:nnnnnnn #1#2#3#4#5#6#7 { } 1624
1625 \cs_gset:Npn \use_none:nnnnnnnn #1#2#3#4#5#6#7#8 { } 1625
1626 \cs_gset:Npn \use_none:nnnnnnnnn #1#2#3#4#5#6#7#8#9 { } 1626
1627 \cs_gset_protected:Npn \__kernel_if_debug:TF #1#2 {#2} 1627
1628 \cs_gset_protected:Npn \debug_on:n #1 1628
1629 { 1629
1630 \sys_load_debug: 1630
1631 \cs_if_exist:NT \__debug_all_on: 1631
1632 { \debug_on:n {#1} } 1632
1633 } 1633
1634 \cs_gset_protected:Npn \debug_off:n #1 1634
1635 { 1635
1636 \sys_load_debug: 1636
1637 \cs_if_exist:NT \__debug_all_on: 1637
1638 { \debug_off:n {#1} } 1638
1639 } 1639
1640 \cs_gset_protected:Npn \debug_suspend: { } 1640
1641 \cs_gset_protected:Npn \debug_resume: { } 1641
1642 \cs_gset_nopar:Npn \g__debug_deprecation_on_tl { } 1642
1643 \cs_gset_nopar:Npn \g__debug_deprecation_off_tl { } 1643
1644 \cs_gset_protected:Npn \__kernel_deprecation_code:nn #1#2 1644
1645 { 1645
1646 \tl_gput_right:Nn \g__debug_deprecation_on_tl {#1} 1646
1647 \tl_gput_right:Nn \g__debug_deprecation_off_tl {#2} 1647
1648 } 1648
1649 \cs_gset:Npn \prg_return_true: 1649
1650 { \exp_after:wN \use_i:nn \exp:w } 1650
1651 \cs_gset:Npn \prg_return_false: 1651
1652 { \exp_after:wN \use_ii:nn \exp:w } 1652
1653 \cs_gset:Npn \__prg_use_none_delimit_by_q_recursion_stop:w 1653
1654 #1 \q_prg_recursion_stop { } 1654
1655 \cs_gset_protected:Npn \prg_set_conditional:Npnn 1655
1656 { \__prg_generate_conditional_parm:NNNpnn \cs_set:Npn e } 1656
```



```
1657 \cs_gset_protected:Npn \prg_gset_conditional:Npnn 1657
1658 { \__prg_generate_conditional_parm:NNNpnn \cs_gset:Npn e } 1658
1659 \cs_gset_protected:Npn \prg_new_conditional:Npnn 1659
1660 { \__prg_generate_conditional_parm:NNNpnn \cs_new:Npn e } 1660
1661 \cs_gset_protected:Npn \prg_set_protected_conditional:Npnn 1661
1662 { \__prg_generate_conditional_parm:NNNpnn \cs_set_protected:Npn p } 1662
1663 \cs_gset_protected:Npn \prg_gset_protected_conditional:Npnn 1663
1664 { \__prg_generate_conditional_parm:NNNpnn \cs_gset_protected:Npn p } 1664
1665 \cs_gset_protected:Npn \prg_new_protected_conditional:Npnn 1665
1666 { \__prg_generate_conditional_parm:NNNpnn \cs_new_protected:Npn p } 1666
1667 \cs_gset_protected:Npn \__prg_generate_conditional_parm:NNNpnn #1#2#3#4# 1667
1668 { 1668
1669 \use:e 1669
1670 { 1670
1671 \__prg_generate_conditional:nnNNNnnn 1671
1672 \cs_split_function:N #3 1672
1673 } 1673
1674 #1 #2 {#4} 1674
1675 } 1675
1676 \cs_gset_protected:Npn \prg_set_conditional:Nnn 1676
1677 { \__prg_generate_conditional_count:NNNnn \cs_set:Npn e } 1677
1678 \cs_gset_protected:Npn \prg_gset_conditional:Nnn 1678
1679 { \__prg_generate_conditional_count:NNNnn \cs_gset:Npn e } 1679
1680 \cs_gset_protected:Npn \prg_new_conditional:Nnn 1680
1681 { \__prg_generate_conditional_count:NNNnn \cs_new:Npn e } 1681
1682 \cs_gset_protected:Npn \prg_set_protected_conditional:Nnn 1682
1683 { \__prg_generate_conditional_count:NNNnn \cs_set_protected:Npn p } 1683
1684 \cs_gset_protected:Npn \prg_gset_protected_conditional:Nnn 1684
1685 { \__prg_generate_conditional_count:NNNnn \cs_gset_protected:Npn p } 1685
1686 \cs_gset_protected:Npn \prg_new_protected_conditional:Nnn 1686
1687 { \__prg_generate_conditional_count:NNNnn \cs_new_protected:Npn p } 1687
1688 \cs_gset_protected:Npn \__prg_generate_conditional_count:NNNnn #1#2#3 1688
1689 { 1689
1690 \use:e 1690
1691 { 1691
1692 \__prg_generate_conditional_count:nnNNNnn 1692
1693 \cs_split_function:N #3 1693
1694 } 1694
1695 #1 #2 1695
1696 } 1696
1697 \cs_gset_protected:Npn \__prg_generate_conditional_count:nnNNNnn #1#2#3#4#5 1697
1698 { 1698
1699 \__kernel_cs_parm_from_arg_count:nnF 1699
1700 { \__prg_generate_conditional:nnNNNnnn {#1} {#2} #3 #4 #5 } 1700
1701 { \tl_count:n {#2} } 1701
1702 { 1702
```

```
1703 \msg_error:nnee { kernel } { bad-number-of-arguments } 1703
1704 { \token_to_str:c { #1 : #2 } } 1704
1705 { \tl_count:n {#2} } 1705
1706 \use_none:nn 1706
1707 } 1707
1708 } 1708
1709 \cs_gset_protected:Npn \__prg_generate_conditional:nnNNnnn #1#2#3#4#5#6#7#8 1709
1710 { 1710
1711 \if_meaning:w \c_false_bool #3 1711
1712 \msg_error:nne { kernel } { missing-colon } 1712
1713 { \token_to_str:c {#1} } 1713
1714 \exp_after:wN \use_none:nn 1714
1715 \fi: 1715
1716 \use:e 1716
1717 { 1717
1718 \exp_not:N \__prg_generate_conditional:NNnnnnNw 1718
1719 \exp_not:n { #4 #5 {#1} {#2} {#6} } 1719
1720 \__prg_generate_conditional_test:w 1720
1721 #8 \s__prg_mark 1721
1722 \__prg_generate_conditional_fast:nw 1722
1723 \prg_return_true: \else: \prg_return_false: \fi: \s__prg_mark 1723
1724 \use_none:n 1724
1725 \exp_not:n { {#8} \use_i_ii:nnn } 1725
1726 \tl_to_str:n {#7} 1726
1727 \exp_not:n { , \q__prg_recursion_tail , \q__prg_recursion_stop } 1727
1728 } 1728
1729 } 1729
1730 \cs_gset:Npn \__prg_generate_conditional_test:w 1730
1731 #1 \prg_return_true: \else: \prg_return_false: \fi: \s__prg_mark #2 1731
1732 { #2 {#1} } 1732
1733 \cs_gset:Npn \__prg_generate_conditional_fast:nw #1#2 \exp_not:n #3 1733
1734 { \exp_not:n { {#1} \use_i:nn } } 1734
1735 \cs_gset_protected:Npn \__prg_generate_conditional:NNnnnnNw #1#2#3#4#5#6#7#8 , 1735
1736 { 1736
1737 \if_meaning:w \q__prg_recursion_tail #8 1737
1738 \exp_after:wN \__prg_use_none_delimit_by_q_recursion_stop:w 1738
1739 \fi: 1739
1740 \use:c { __prg_generate_ #8 _form:wNNnnnnN } 1740
1741 \tl_if_empty:nF {#8} 1741
1742 { 1742
1743 \msg_error:nnee 1743
1744 { kernel } { conditional-form-unknown } 1744
1745 {#8} { \token_to_str:c { #3 : #4 } } 1745
1746 } 1746
1747 \use_none:nnnnnnnn 1747
1748 \s__prg_stop 1748
```

```
1749 #1 #2 {#3} {#4} {#5} {#6} #7 1749
1750 \__prg_generate_conditional:NNnnnnNw #1 #2 {#3} {#4} {#5} {#6} #7 1750
1751 } 1751
1752 \cs_gset_protected:Npn \__prg_generate_p_form:wNNnnnnN 1752
1753 #1 \s__prg_stop #2#3#4#5#6#7#8 1753
1754 { 1754
1755 \if_meaning:w e #3 1755
1756 \exp_after:wN \use_i:nn 1756
1757 \else: 1757
1758 \exp_after:wN \use_ii:nn 1758
1759 \fi: 1759
1760 { 1760
1761 #8 1761
1762 { \exp_args:Nc #2 { #4 _p: #5 } #6 } 1762
1763 { { #7 \exp_end: \c_true_bool \c_false_bool } } 1763
1764 { #7 \__prg_p_true:w \fi: \c_false_bool } 1764
1765 } 1765
1766 { 1766
1767 \msg_error:nne { kernel } { protected-predicate } 1767
1768 { \token_to_str:c { #4 _p: #5 } } 1768
1769 } 1769
1770 } 1770
1771 \cs_gset_protected:Npn \__prg_generate_T_form:wNNnnnnN 1771
1772 #1 \s__prg_stop #2#3#4#5#6#7#8 1772
1773 { 1773
1774 #8 1774
1775 { \exp_args:Nc #2 { #4 : #5 T } #6 } 1775
1776 { { #7 \exp_end: \use:n \use_none:n } } 1776
1777 { #7 \__prg_T_true:w \fi: \use_none:n } 1777
1778 } 1778
1779 \cs_gset_protected:Npn \__prg_generate_F_form:wNNnnnnN 1779
1780 #1 \s__prg_stop #2#3#4#5#6#7#8 1780
1781 { 1781
1782 #8 1782
1783 { \exp_args:Nc #2 { #4 : #5 F } #6 } 1783
1784 { { #7 \exp_end: { } } } 1784
1785 { #7 \__prg_F_true:w \fi: \use:n } 1785
1786 } 1786
1787 \cs_gset_protected:Npn \__prg_generate_TF_form:wNNnnnnN 1787
1788 #1 \s__prg_stop #2#3#4#5#6#7#8 1788
1789 { 1789
1790 #8 1790
1791 { \exp_args:Nc #2 { #4 : #5 TF } #6 } 1791
1792 { { #7 \exp_end: } } 1792
1793 { #7 \__prg_TF_true:w \fi: \use_ii:nn } 1793
1794 } 1794
```

1795	\cs_gset:Npn __prg_p_true:w \fi: \c_false_bool { \fi: \c_true_bool }	1795
1796	\cs_gset:Npn __prg_T_true:w \fi: \use_none:n { \fi: \use:n }	1796
1797	\cs_gset:Npn __prg_F_true:w \fi: \use:n { \fi: \use_none:n }	1797
1798	\cs_gset:Npn __prg_TF_true:w \fi: \use_ii:nn { \fi: \use_i:nn }	1798
1799	\cs_gset_protected:Npn \prg_set_eq_conditional:NNn	1799
1800	{ __prg_set_eq_conditional:NNNn \cs_set_eq:cc }	1800
1801	\cs_gset_protected:Npn \prg_gset_eq_conditional:NNn	1801
1802	{ __prg_set_eq_conditional:NNNn \cs_gset_eq:cc }	1802
1803	\cs_gset_protected:Npn \prg_new_eq_conditional:NNn	1803
1804	{ __prg_set_eq_conditional:NNNn \cs_new_eq:cc }	1804
1805	\cs_gset_protected:Npn __prg_set_eq_conditional:NNNn #1#2#3#4	1805
1806	{	1806
1807	\use:e	1807
1808	{	1808
1809	\exp_not:N __prg_set_eq_conditional:nnNnnNNw	1809
1810	\cs_split_function:N #2	1810
1811	\cs_split_function:N #3	1811
1812	\exp_not:N #1	1812
1813	\tl_to_str:n {#4}	1813
1814	\exp_not:n { , \q__prg_recursion_tail , \q__prg_recursion_stop }	1814
1815	}	1815
1816	}	1816
1817	\cs_gset_protected:Npn __prg_set_eq_conditional:nnNnnNNw #1#2#3#4#5#6	1817
1818	{	1818
1819	\if_meaning:w \c_false_bool #3	1819
1820	\msg_error:nne { kernel } { missing-colon }	1820
1821	{ \token_to_str:c {#1} }	1821
1822	\exp_after:wN __prg_use_none_delimit_by_q_recursion_stop:w	1822
1823	\fi:	1823
1824	\if_meaning:w \c_false_bool #6	1824
1825	\msg_error:nne { kernel } { missing-colon }	1825
1826	{ \token_to_str:c {#4} }	1826
1827	\exp_after:wN __prg_use_none_delimit_by_q_recursion_stop:w	1827
1828	\fi:	1828
1829	__prg_set_eq_conditional_loop:nnnnNw {#1} {#2} {#4} {#5}	1829
1830	}	1830
1831	\cs_gset_protected:Npn __prg_set_eq_conditional_loop:nnnnNw #1#2#3#4#5#6 ,	1831
1832	{	1832
1833	\if_meaning:w \q__prg_recursion_tail #6	1833
1834	\exp_after:wN __prg_use_none_delimit_by_q_recursion_stop:w	1834
1835	\fi:	1835
1836	\use:c { __prg_set_eq_conditional_ #6 _form:wNnnnn }	1836
1837	\tl_if_empty:nF {#6}	1837
1838	{	1838
1839	\msg_error:nnee	1839
1840	{ kernel } { conditional-form-unknown }	1840

```

1841      {#6} {\token_to_str:c { #1 : #2 } }
1842    }
1843    \use_none:nnnnnn
1844    \s__prg_stop
1845    #5 {#1} {#2} {#3} {#4}
1846    \__prg_set_eq_conditional_loop:nnnnNw {#1} {#2} {#3} {#4} #5
1847  }
1848  \cs_gset:Npn \__prg_set_eq_conditional_p_form:wNnnnn #1 \s__prg_stop #2#3#4#5#6
1849    { #2 { #3 _p : #4 } { #5 _p : #6 } }
1850  \cs_gset:Npn \__prg_set_eq_conditional_TF_form:wNnnnn #1 \s__prg_stop #2#3#4#5#6
1851    { #2 { #3 : #4 TF } { #5 : #6 TF } }
1852  \cs_gset:Npn \__prg_set_eq_conditional_T_form:wNnnnn #1 \s__prg_stop #2#3#4#5#6
1853    { #2 { #3 : #4 T } { #5 : #6 T } }
1854  \cs_gset:Npn \__prg_set_eq_conditional_F_form:wNnnnn #1 \s__prg_stop #2#3#4#5#6
1855    { #2 { #3 : #4 F } { #5 : #6 F } }
1856  \tex_global:D \tex_chardef:D \c_true_bool = 1 ~
1857  \tex_global:D \tex_chardef:D \c_false_bool = 0 ~
1858  \cs_gset:Npn \cs_to_str:N
1859  {
1860    \tex_romannumeral:D
1861    \if:w \token_to_str:N \_\_\cs_to_str:w \fi:
1862    \exp_after:wN \_\_\cs_to_str:N \token_to_str:N
1863  }
1864  \cs_gset:Npn \_\_\cs_to_str:N #1 { \c_zero_int }
1865  \cs_gset:Npn \_\_\cs_to_str:w #1 \_\_\cs_to_str:N
1866    { - \int_value:w \fi: \exp_after:wN \c_zero_int }
1867  \cs_gset_protected:Npn \_\_\cs_tmp:w #1
1868  {
1869    \cs_gset:Npn \cs_split_function:N ##1
1870    {
1871      \exp_after:wN \exp_after:wN \exp_after:wN
1872      \_\_\cs_split_function_auxi:w
1873      \cs_to_str:N ##1 \s__cs_mark \c_true_bool
1874      #1 \s__cs_mark \c_false_bool \s__cs_stop
1875    }
1876    \cs_gset:Npn \_\_\cs_split_function_auxi:w
1877      ##1 #1 ##2 \s__cs_mark ##3##4 \s__cs_stop
1878      { \_\_\cs_split_function_auxii:w ##1 \s__cs_mark \s__cs_stop {##2} ##3 }
1879    \cs_gset:Npn \_\_\cs_split_function_auxii:w ##1 \s__cs_mark ##2 \s__cs_stop
1880      { {##1} }
1881  }
1882  \exp_after:wN \_\_\cs_tmp:w \token_to_str:N :
1883  \prg_gset_conditional:Npnn \cs_if_exist:N #1 { p , T , F , TF }
1884  {
1885    \if_meaning:w #1 \scan_stop:
1886    \use_i:nnnn

```

```

1887 \else: 1887
1888 \fi: 1888
1889 \if_cs_exist:N #1 1889
1890 \prg_return_true: 1890
1891 \else: 1891
1892 \prg_return_false: 1892
1893 \fi: 1893
1894 } 1894
1895 \cs_if_exist:NTF \tex_lastnamedcs:D 1895
1896 { 1896
1897 \prg_gset_conditional:Npnn \cs_if_exist:c #1 { p , T , F , TF } 1897
1898 { 1898
1899 \if_cs_exist:w #1 \cs_end: 1899
1900 \__cs_if_exist_c_aux: 1900
1901 \prg_return_true: 1901
1902 \else: 1902
1903 \prg_return_false: 1903
1904 \fi: 1904
1905 } 1905
1906 \cs_gset:Npn \__cs_if_exist_c_aux: 1906
1907 { \fi: \exp_after:wN \if_meaning:w \tex_lastnamedcs:D \scan_stop: \else: } 1907
1908 } 1908
1909 { 1909
1910 \prg_gset_conditional:Npnn \cs_if_exist:c #1 { p , T , F , TF } 1910
1911 { 1911
1912 \if_cs_exist:w #1 \cs_end: 1912
1913 \__cs_if_exist_c_aux:w 1913
1914 \fi: 1914
1915 \use_none:n {#1} 1915
1916 \if_false: 1916
1917 \prg_return_true: 1917
1918 \else: 1918
1919 \prg_return_false: 1919
1920 \fi: 1920
1921 } 1921
1922 \cs_gset:Npn \__cs_if_exist_c_aux:w \fi: \use_none:n #1 \if_false: 1922
1923 { \fi: \exp_after:wN \if_meaning:w \cs:w #1 \cs_end: \scan_stop: \else: } 1923
1924 } 1924
1925 \prg_gset_conditional:Npnn \cs_if_free:N #1 { p , T , F , TF } 1925
1926 { 1926
1927 \if_cs_exist:N #1 1927
1928 \else: 1928
1929 \use_none:nnnn 1929
1930 \fi: 1930
1931 \if_meaning:w #1 \scan_stop: 1931
1932 \prg_return_true: 1932

```



```

1933     \else:
1934         \prg_return_false:
1935     \fi:
1936 }
1937 \cs_if_exist:NTF \tex_lastnamedcs:D
1938 {
1939     \prg_gset_conditional:Npnn \cs_if_free:c #1 { p , T , F , TF }
1940     {
1941         \if_cs_exist:w #1 \cs_end:
1942         \__cs_if_free_c_aux:w
1943         \fi:
1944         \if_true:
1945             \prg_return_true:
1946         \else:
1947             \prg_return_false:
1948         \fi:
1949     }
1950     \cs_gset:Npn \__cs_if_free_c_aux:w \fi: \if_true:
1951     { \fi: \exp_after:wN \if_meaning:w \tex_lastnamedcs:D \scan_stop: }
1952 }
1953 {
1954     \prg_gset_conditional:Npnn \cs_if_free:c #1 { p , T , F , TF }
1955     {
1956         \if_cs_exist:w #1 \cs_end:
1957         \__cs_if_free_c_aux:w
1958         \fi:
1959         \use_none:n {#1}
1960         \if_true:
1961             \prg_return_true:
1962         \else:
1963             \prg_return_false:
1964         \fi:
1965     }
1966     \cs_gset:Npn \__cs_if_free_c_aux:w \fi: \use_none:n #1 \if_true:
1967     { \fi: \exp_after:wN \if_meaning:w \cs:w #1 \cs_end: \scan_stop: }
1968 }
1969 \cs_gset:Npn \cs_if_exist_use:NTF #1#2
1970 { \cs_if_exist:NTF #1 { #1 #2 } }
1971 \cs_gset:Npn \cs_if_exist_use:NF #1
1972 { \cs_if_exist:NTF #1 #1 }
1973 \cs_gset:Npn \cs_if_exist_use:NT #1 #2
1974 { \cs_if_exist:NT #1 { #1 #2 } }
1975 \cs_gset:Npn \cs_if_exist_use:N #1
1976 { \cs_if_exist:NT #1 #1 }
1977 \cs_if_exist:NTF \tex_lastnamedcs:D
1978 {

```

1979	\cs_gset:Npn \cs_if_exist_use:cTF #1	1979
1980	{	1980
1981	\if_cs_exist:w #1 \cs_end:	1981
1982	__cs_if_exist_use_aux:w	1982
1983	\fi:	1983
1984	\use_ii:nn	1984
1985	}	1985
1986	\cs_gset:Npn __cs_if_exist_use_aux:w \fi: \use_ii:nn	1986
1987	{ \fi: \exp_after:wN __cs_if_exist_use_aux:Nnn \tex_lastnamedcs:D }	1987
1988	}	1988
1989	{	1989
1990	\cs_gset:Npn \cs_if_exist_use:cTF #1	1990
1991	{	1991
1992	\if_cs_exist:w #1 \cs_end:	1992
1993	__cs_if_exist_use_aux:w	1993
1994	\fi:	1994
1995	\use_iii:nnn {#1}	1995
1996	}	1996
1997	\cs_gset:Npn __cs_if_exist_use_aux:w \fi: \use_iii:nnn #1	1997
1998	{ \fi: \exp_after:wN __cs_if_exist_use_aux:Nnn \cs:w #1 \cs_end: }	1998
1999	}	1999
2000	\cs_gset:Npn __cs_if_exist_use_aux:Nnn #1#2	2000
2001	{	2001
2002	\if_meaning:w #1 \scan_stop:	2002
2003	\exp_after:wN \use_iii:nnn	2003
2004	\fi:	2004
2005	\use_i:nn { #1 #2 }	2005
2006	}	2006
2007	\cs_gset:Npn \cs_if_exist_use:cF #1	2007
2008	{ \cs_if_exist_use:cTF {#1} {} }	2008
2009	\cs_gset:Npn \cs_if_exist_use:cT #1#2	2009
2010	{ \cs_if_exist_use:cTF {#1} {#2} {} }	2010
2011	\cs_gset:Npn \cs_if_exist_use:c #1	2011
2012	{ \cs_if_exist_use:cTF {#1} {} {} }	2012
2013	\cs_gset_protected:Npn \msg_error:nnee #1#2#3#4	2013
2014	{	2014
2015	\tex_newlinechar:D = ``^^J \scan_stop:	2015
2016	\tex_errmessage:D	2016
2017	{	2017
2018	!!~! ^^J	2018
2019	Argh,~internal~LaTeX3~error! ^^J ^^J	2019
2020	Module ~ #1 , ~ message~name~"#2": ^^J	2020
2021	Arguments~'#3'~and~'#4' ^^J ^^J	2021
2022	This~is~one~for~The~LaTeX3~Project::~bailing~out	2022
2023	}	2023
2024	\tex_end:D	2024

```

2025 }
2026 \cs_gset_protected:Npn \msg_error:nne #1#2#3
2027 { \msg_error:nnee {#1} {#2} {#3} { } }
2028 \cs_gset_protected:Npn \msg_error:nn #1#2
2029 { \msg_error:nnee {#1} {#2} { } { } }
2030 \cs_gset:Npn \msg_line_context:
2031 { on~line~ \tex_the:D \tex_inputlineno:D }
2032 \cs_gset_protected:Npn \iow_log:e
2033 { \tex_immediate:D \tex_write:D -1 }
2034 \cs_gset_protected:Npn \iow_term:e
2035 { \tex_immediate:D \tex_write:D 16 }
2036 \cs_gset_protected:Npn \__kernel_chk_if_free_cs:N #1
2037 {
2038   \cs_if_free:NF #1
2039   {
2040     \msg_error:nnee { kernel } { command-already-defined }
2041     { \token_to_str:N #1 } { \token_to_meaning:N #1 }
2042   }
2043 }
2044 \cs_gset_protected:Npn \__kernel_chk_if_free_cs:c
2045 { \exp_args:Nc \__kernel_chk_if_free_cs:N }
2046 \cs_set:Npn \__cs_tmp:w #1#2
2047 {
2048   \cs_gset_protected:Npn #1 ##1
2049   {
2050     \__kernel_chk_if_free_cs:N ##1
2051     #2 ##1
2052   }
2053 }
2054 \__cs_tmp:w \cs_new_nopar:Npn \cs_gset_nopar:Npn
2055 \__cs_tmp:w \cs_new_nopar:Npe \cs_gset_nopar:Npe
2056 \__cs_tmp:w \cs_new_nopar:Npx \cs_gset_nopar:Npx
2057 \__cs_tmp:w \cs_new:Npn \cs_gset:Npn
2058 \__cs_tmp:w \cs_new:Npe \cs_gset:Npe
2059 \__cs_tmp:w \cs_new:Npx \cs_gset:Npx
2060 \__cs_tmp:w \cs_new_protected_nopar:Npn \cs_gset_protected_nopar:Npn
2061 \__cs_tmp:w \cs_new_protected_nopar:Npe \cs_gset_protected_nopar:Npe
2062 \__cs_tmp:w \cs_new_protected_nopar:Npx \cs_gset_protected_nopar:Npx
2063 \__cs_tmp:w \cs_new_protected:Npn \cs_gset_protected:Npn
2064 \__cs_tmp:w \cs_new_protected:Npe \cs_gset_protected:Npe
2065 \__cs_tmp:w \cs_new_protected:Npx \cs_gset_protected:Npx
2066 \cs_set:Npn \__cs_tmp:w #1#2
2067 { \cs_new_protected_nopar:Npn #1 { \exp_args:Nc #2 } }
2068 \__cs_tmp:w \cs_set_nopar:cpn \cs_set_nopar:Npn
2069 \__cs_tmp:w \cs_set_nopar:cpe \cs_set_nopar:Npe
2070 \__cs_tmp:w \cs_set_nopar:cpx \cs_set_nopar:Npx

```

2071	_cs_tmp:w \cs_gset_nopar:cpn \cs_gset_nopar:Npn	2071
2072	_cs_tmp:w \cs_gset_nopar:cpe \cs_gset_nopar:Npe	2072
2073	_cs_tmp:w \cs_gset_nopar:cpx \cs_gset_nopar:Npx	2073
2074	_cs_tmp:w \cs_new_nopar:cpn \cs_new_nopar:Npn	2074
2075	_cs_tmp:w \cs_new_nopar:cpe \cs_new_nopar:Npe	2075
2076	_cs_tmp:w \cs_new_nopar:cpx \cs_new_nopar:Npx	2076
2077	_cs_tmp:w \cs_set:cpn \cs_set:Npn	2077
2078	_cs_tmp:w \cs_set:cpe \cs_set:Npe	2078
2079	_cs_tmp:w \cs_set:cpx \cs_set:Npx	2079
2080	_cs_tmp:w \cs_gset:cpn \cs_gset:Npn	2080
2081	_cs_tmp:w \cs_gset:cpe \cs_gset:Npe	2081
2082	_cs_tmp:w \cs_gset:cpx \cs_gset:Npx	2082
2083	_cs_tmp:w \cs_new:cpn \cs_new:Npn	2083
2084	_cs_tmp:w \cs_new:cpe \cs_new:Npe	2084
2085	_cs_tmp:w \cs_new:cpx \cs_new:Npx	2085
2086	_cs_tmp:w \cs_set_protected_nopar:cpn \cs_set_protected_nopar:Npn	2086
2087	_cs_tmp:w \cs_set_protected_nopar:cpe \cs_set_protected_nopar:Npe	2087
2088	_cs_tmp:w \cs_set_protected_nopar:cpx \cs_set_protected_nopar:Npx	2088
2089	_cs_tmp:w \cs_gset_protected_nopar:cpn \cs_gset_protected_nopar:Npn	2089
2090	_cs_tmp:w \cs_gset_protected_nopar:cpe \cs_gset_protected_nopar:Npe	2090
2091	_cs_tmp:w \cs_gset_protected_nopar:cpx \cs_gset_protected_nopar:Npx	2091
2092	_cs_tmp:w \cs_new_protected_nopar:cpn \cs_new_protected_nopar:Npn	2092
2093	_cs_tmp:w \cs_new_protected_nopar:cpe \cs_new_protected_nopar:Npe	2093
2094	_cs_tmp:w \cs_new_protected_nopar:cpx \cs_new_protected_nopar:Npx	2094
2095	_cs_tmp:w \cs_set_protected:cpn \cs_set_protected:Npn	2095
2096	_cs_tmp:w \cs_set_protected:cpe \cs_set_protected:Npe	2096
2097	_cs_tmp:w \cs_set_protected:cpx \cs_set_protected:Npx	2097
2098	_cs_tmp:w \cs_gset_protected:cpn \cs_gset_protected:Npn	2098
2099	_cs_tmp:w \cs_gset_protected:cpe \cs_gset_protected:Npe	2099
2100	_cs_tmp:w \cs_gset_protected:cpx \cs_gset_protected:Npx	2100
2101	_cs_tmp:w \cs_new_protected:cpn \cs_new_protected:Npn	2101
2102	_cs_tmp:w \cs_new_protected:cpe \cs_new_protected:Npe	2102
2103	_cs_tmp:w \cs_new_protected:cpx \cs_new_protected:Npx	2103
2104	\cs_new_protected:Npn \cs_set_eq:NN #1 { \tex_let:D #1 =~ }	2104
2105	\cs_new_protected:Npn \cs_set_eq:cN { \exp_args:Nc \cs_set_eq:NN }	2105
2106	\cs_new_protected:Npn \cs_set_eq:Nc { \exp_args:NNc \cs_set_eq:NN }	2106
2107	\cs_new_protected:Npn \cs_set_eq:cc { \exp_args:Ncc \cs_set_eq:NN }	2107
2108	\cs_new_protected:Npn \cs_gset_eq:NN { \tex_global:D \cs_set_eq:NN }	2108
2109	\cs_new_protected:Npn \cs_gset_eq:Nc { \exp_args:NNc \cs_gset_eq:NN }	2109
2110	\cs_new_protected:Npn \cs_gset_eq:cN { \exp_args:Nc \cs_gset_eq:NN }	2110
2111	\cs_new_protected:Npn \cs_gset_eq:cc { \exp_args:Ncc \cs_gset_eq:NN }	2111
2112	\cs_new_protected:Npn \cs_new_eq:NN #1	2112
2113	{	2113
2114	_kernel_chk_if_free_cs:N #1	2114
2115	\tex_global:D \cs_set_eq:NN #1	2115
2116	}	2116

```
2117 \cs_new_protected:Npn \cs_new_eq:cN { \exp_args:Nc \cs_new_eq:NN } 2117
2118 \cs_new_protected:Npn \cs_new_eq:Nc { \exp_args:NNc \cs_new_eq:NN } 2118
2119 \cs_new_protected:Npn \cs_new_eq:cc { \exp_args:Ncc \cs_new_eq:NN } 2119
2120 \cs_new_protected:Npn \cs_undefine:N #1 2120
2121 { \cs_gset_eq:NN #1 \tex_undefined:D } 2121
2122 \cs_new_protected:Npn \cs_undefine:c #1 2122
2123 { 2123
2124 \if_cs_exist:w #1 \cs_end: 2124
2125 \else: 2125
2126 \use_i:nnnn 2126
2127 \fi: 2127
2128 \exp_args:Nc \cs_undefine:N {#1} 2128
2129 } 2129
2130 \cs_new_protected:Npn \__kernel_cs_parm_from_arg_count:nnF #1#2 2130
2131 { 2131
2132 \exp_args:Ne \__cs_parm_from_arg_count_test:nnF 2132
2133 { 2133
2134 \exp_after:wN \exp_not:n 2134
2135 \if_case:w \int_eval:n {#2} 2135
2136 { } 2136
2137 \or: { ##1 } 2137
2138 \or: { ##1##2 } 2138
2139 \or: { ##1##2##3 } 2139
2140 \or: { ##1##2##3##4 } 2140
2141 \or: { ##1##2##3##4##5 } 2141
2142 \or: { ##1##2##3##4##5##6 } 2142
2143 \or: { ##1##2##3##4##5##6##7 } 2143
2144 \or: { ##1##2##3##4##5##6##7##8 } 2144
2145 \or: { ##1##2##3##4##5##6##7##8##9 } 2145
2146 \else: { \c_false_bool } 2146
2147 \fi: 2147
2148 } 2148
2149 {#1} 2149
2150 } 2150
2151 \cs_new_protected:Npn \__cs_parm_from_arg_count_test:nnF #1#2 2151
2152 { 2152
2153 \if_meaning:w \c_false_bool #1 2153
2154 \exp_after:wN \use_ii:nn 2154
2155 \else: 2155
2156 \exp_after:wN \use_i:nn 2156
2157 \fi: 2157
2158 { #2 {#1} } 2158
2159 } 2159
2160 \cs_new:Npn \__cs_count_signature:N #1 2160
2161 { \exp_args:Nf \__cs_count_signature:n { \cs_split_function:N #1 } } 2161
2162 \cs_new:Npn \__cs_count_signature:n #1 2162
```



```
2163 { \int_eval:n { \__cs_count_signature:nnN #1 } } 2163
2164 \cs_new:Npn \__cs_count_signature:nnN #1#2#3 2164
2165 { 2165
2166   \if_meaning:w \c_true_bool #3 2166
2167   \tl_count:n {#2} 2167
2168   \else: 2168
2169   -1 2169
2170   \fi: 2170
2171 } 2171
2172 \cs_new:Npn \__cs_count_signature:c 2172
2173 { \exp_args:Nc \__cs_count_signature:N } 2173
2174 \cs_new_protected:Npn \cs_generate_from_arg_count:NNnn #1#2#3#4 2174
2175 { 2175
2176   \__kernel_cs_parm_from_arg_count:nnF { \use:nnn #2 #1 } {#3} 2176
2177   { 2177
2178     \msg_error:nnee { kernel } { bad-number-of-arguments } 2178
2179     { \token_to_str:N #1 } { \int_eval:n {#3} } 2179
2180     \use_none:n 2180
2181   } 2181
2182   {#4} 2182
2183 } 2183
2184 \cs_new_protected:Npn \cs_generate_from_arg_count:cNnn 2184
2185 { \exp_args:Nc \cs_generate_from_arg_count:NNnn } 2185
2186 \cs_new_protected:Npn \cs_generate_from_arg_count:Ncnn 2186
2187 { \exp_args:NNc \cs_generate_from_arg_count:NNnn } 2187
2188 \cs_set:Npn \__cs_tmp:w #1#2#3 2188
2189 { 2189
2190   \cs_new_protected:cpx { cs_ #1 : #2 } 2190
2191   { 2191
2192     \exp_not:N \__cs_generate_from_signature:NNn 2192
2193     \exp_after:wN \exp_not:N \cs:w cs_ #1 : #3 \cs_end: 2193
2194   } 2194
2195 } 2195
2196 \cs_new_protected:Npn \__cs_generate_from_signature:NNn #1#2 2196
2197 { 2197
2198   \use:e 2198
2199   { 2199
2200     \__cs_generate_from_signature:nnNNNn 2200
2201     \cs_split_function:N #2 2201
2202   } 2202
2203   #1 #2 2203
2204 } 2204
2205 \cs_new_protected:Npn \__cs_generate_from_signature:nnNNNn #1#2#3#4#5#6 2205
2206 { 2206
2207   \bool_if:NTF #3 2207
2208   { 2208
```

```
2209 \cs_set_nopar:Npx \__cs_tmp:w 2209
2210 { \tl_map_function:nN {#2} \__cs_generate_from_signature:n } 2210
2211 \tl_if_empty:oF \__cs_tmp:w 2211
2212 { 2212
2213 \msg_error:nneee { kernel } { non-base-function } 2213
2214 { \token_to_str:N #5 } {#2} { \__cs_tmp:w } 2214
2215 } 2215
2216 \cs_generate_from_arg_count:NNnn 2216
2217 #5 #4 { \tl_count:n {#2} } {#6} 2217
2218 } 2218
2219 { 2219
2220 \msg_error:nne { kernel } { missing-colon } 2220
2221 { \token_to_str:N #5 } 2221
2222 } 2222
2223 } 2223
2224 \cs_new:Npn \__cs_generate_from_signature:n #1 2224
2225 { 2225
2226 \if:w n #1 \else: \if:w N #1 \else: 2226
2227 \if:w T #1 \else: \if:w F #1 \else: #1 \fi: \fi: \fi: 2227
2228 } 2228
2229 \__cs_tmp:w { set } { Nn } { Npn } 2229
2230 \__cs_tmp:w { set } { Ne } { Npe } 2230
2231 \__cs_tmp:w { set } { Nx } { Npx } 2231
2232 \__cs_tmp:w { set_nopar } { Nn } { Npn } 2232
2233 \__cs_tmp:w { set_nopar } { Ne } { Npe } 2233
2234 \__cs_tmp:w { set_nopar } { Nx } { Npx } 2234
2235 \__cs_tmp:w { set_protected } { Nn } { Npn } 2235
2236 \__cs_tmp:w { set_protected } { Ne } { Npe } 2236
2237 \__cs_tmp:w { set_protected } { Nx } { Npx } 2237
2238 \__cs_tmp:w { set_protected_nopar } { Nn } { Npn } 2238
2239 \__cs_tmp:w { set_protected_nopar } { Ne } { Npe } 2239
2240 \__cs_tmp:w { set_protected_nopar } { Nx } { Npx } 2240
2241 \__cs_tmp:w { gset } { Nn } { Npn } 2241
2242 \__cs_tmp:w { gset } { Ne } { Npe } 2242
2243 \__cs_tmp:w { gset } { Nx } { Npx } 2243
2244 \__cs_tmp:w { gset_nopar } { Nn } { Npn } 2244
2245 \__cs_tmp:w { gset_nopar } { Ne } { Npe } 2245
2246 \__cs_tmp:w { gset_nopar } { Nx } { Npx } 2246
2247 \__cs_tmp:w { gset_protected } { Nn } { Npn } 2247
2248 \__cs_tmp:w { gset_protected } { Ne } { Npe } 2248
2249 \__cs_tmp:w { gset_protected } { Nx } { Npx } 2249
2250 \__cs_tmp:w { gset_protected_nopar } { Nn } { Npn } 2250
2251 \__cs_tmp:w { gset_protected_nopar } { Ne } { Npe } 2251
2252 \__cs_tmp:w { gset_protected_nopar } { Nx } { Npx } 2252
2253 \__cs_tmp:w { new } { Nn } { Npn } 2253
2254 \__cs_tmp:w { new } { Ne } { Npe } 2254
```

2255	_cs_tmp:w { new }	{ Nx }	{ Npx }	2255
2256	_cs_tmp:w { new_nopar }	{ Nn }	{ Npn }	2256
2257	_cs_tmp:w { new_nopar }	{ Ne }	{ Npe }	2257
2258	_cs_tmp:w { new_nopar }	{ Nx }	{ Npx }	2258
2259	_cs_tmp:w { new_protected }	{ Nn }	{ Npn }	2259
2260	_cs_tmp:w { new_protected }	{ Ne }	{ Npe }	2260
2261	_cs_tmp:w { new_protected }	{ Nx }	{ Npx }	2261
2262	_cs_tmp:w { new_protected_nopar }	{ Nn }	{ Npn }	2262
2263	_cs_tmp:w { new_protected_nopar }	{ Ne }	{ Npe }	2263
2264	_cs_tmp:w { new_protected_nopar }	{ Nx }	{ Npx }	2264
2265	\\cs_set:Npn _cs_tmp:w #1#2			2265
2266	{			2266
2267	\\cs_new_protected:cpx { cs_ #1 : c #2 }			2267
2268	{			2268
2269	\\exp_not:N \\exp_args:Nc			2269
2270	\\exp_after:wN \\exp_not:N \\cs:w cs_ #1 : N #2 \\cs_end:			2270
2271	}			2271
2272	}			2272
2273	_cs_tmp:w { set }	{ n }		2273
2274	_cs_tmp:w { set }	{ e }		2274
2275	_cs_tmp:w { set }	{ x }		2275
2276	_cs_tmp:w { set_nopar }	{ n }		2276
2277	_cs_tmp:w { set_nopar }	{ e }		2277
2278	_cs_tmp:w { set_nopar }	{ x }		2278
2279	_cs_tmp:w { set_protected }	{ n }		2279
2280	_cs_tmp:w { set_protected }	{ e }		2280
2281	_cs_tmp:w { set_protected }	{ x }		2281
2282	_cs_tmp:w { set_protected_nopar }	{ n }		2282
2283	_cs_tmp:w { set_protected_nopar }	{ e }		2283
2284	_cs_tmp:w { set_protected_nopar }	{ x }		2284
2285	_cs_tmp:w { gset }	{ n }		2285
2286	_cs_tmp:w { gset }	{ e }		2286
2287	_cs_tmp:w { gset }	{ x }		2287
2288	_cs_tmp:w { gset_nopar }	{ n }		2288
2289	_cs_tmp:w { gset_nopar }	{ e }		2289
2290	_cs_tmp:w { gset_nopar }	{ x }		2290
2291	_cs_tmp:w { gset_protected }	{ n }		2291
2292	_cs_tmp:w { gset_protected }	{ e }		2292
2293	_cs_tmp:w { gset_protected }	{ x }		2293
2294	_cs_tmp:w { gset_protected_nopar }	{ n }		2294
2295	_cs_tmp:w { gset_protected_nopar }	{ e }		2295
2296	_cs_tmp:w { gset_protected_nopar }	{ x }		2296
2297	_cs_tmp:w { new }	{ n }		2297
2298	_cs_tmp:w { new }	{ e }		2298
2299	_cs_tmp:w { new }	{ x }		2299
2300	_cs_tmp:w { new_nopar }	{ n }		2300

2301	_cs_tmp:w { new_nopar }	{ e }	2301
2302	_cs_tmp:w { new_nopar }	{ x }	2302
2303	_cs_tmp:w { new_protected }	{ n }	2303
2304	_cs_tmp:w { new_protected }	{ e }	2304
2305	_cs_tmp:w { new_protected }	{ x }	2305
2306	_cs_tmp:w { new_protected_nopar }	{ n }	2306
2307	_cs_tmp:w { new_protected_nopar }	{ e }	2307
2308	_cs_tmp:w { new_protected_nopar }	{ x }	2308
2309	\\prg_new_conditional:Npnn \\cs_if_eq:NN #1#2 { p , T , F , TF }		2309
2310	{		2310
2311	\\if_meaning:w #1#2		2311
2312	\\prg_return_true: \\else: \\prg_return_false: \\fi:		2312
2313	}		2313
2314	\\cs_new:Npn \\cs_if_eq_p:cN { \\exp_args:Nc \\cs_if_eq_p:NN }		2314
2315	\\cs_new:Npn \\cs_if_eq:cNTF { \\exp_args:Nc \\cs_if_eq:NNTF }		2315
2316	\\cs_new:Npn \\cs_if_eq:cNT { \\exp_args:Nc \\cs_if_eq:NNT }		2316
2317	\\cs_new:Npn \\cs_if_eq:cNF { \\exp_args:Nc \\cs_if_eq:NNF }		2317
2318	\\cs_new:Npn \\cs_if_eq_p:Nc { \\exp_args:NNc \\cs_if_eq_p:NN }		2318
2319	\\cs_new:Npn \\cs_if_eq:NcTF { \\exp_args:NNc \\cs_if_eq:NNTF }		2319
2320	\\cs_new:Npn \\cs_if_eq:NcT { \\exp_args:NNc \\cs_if_eq:NNT }		2320
2321	\\cs_new:Npn \\cs_if_eq:NcF { \\exp_args:NNc \\cs_if_eq:NNF }		2321
2322	\\cs_new:Npn \\cs_if_eq_p:cc { \\exp_args:Ncc \\cs_if_eq_p:NN }		2322
2323	\\cs_new:Npn \\cs_if_eq:ccTF { \\exp_args:Ncc \\cs_if_eq:NNTF }		2323
2324	\\cs_new:Npn \\cs_if_eq:ccT { \\exp_args:Ncc \\cs_if_eq:NNT }		2324
2325	\\cs_new:Npn \\cs_if_eq:ccF { \\exp_args:Ncc \\cs_if_eq:NNF }		2325
2326	\\cs_new_protected:Npn __kernel_chk_defined:NT #1#2		2326
2327	{		2327
2328	\\cs_if_exist:NTF #1		2328
2329	{#2}		2329
2330	{		2330
2331	\\msg_error:nne { kernel } { variable-not-defined }		2331
2332	{ \\token_to_str:N #1 }		2332
2333	}		2333
2334	}		2334
2335	\\cs_new_protected:Npn __kernel_register_show:N		2335
2336	{ __kernel_register_show_aux:NN \\tl_show:n }		2336
2337	\\cs_new_protected:Npn __kernel_register_show:c		2337
2338	{ \\exp_args:Nc __kernel_register_show:N }		2338
2339	\\cs_new_protected:Npn __kernel_register_log:N		2339
2340	{ __kernel_register_show_aux:NN \\tl_log:n }		2340
2341	\\cs_new_protected:Npn __kernel_register_log:c		2341
2342	{ \\exp_args:Nc __kernel_register_log:N }		2342
2343	\\cs_new_protected:Npn __kernel_register_show_aux:NN #1#2		2343
2344	{		2344
2345	__kernel_chk_defined:NT #2		2345
2346	{		2346

```

2347 \exp_args:No \__kernel_register_show_aux:nNN 2347
2348 { \tex_the:D #2 } #2 #1 2348
2349 } 2349
2350 } 2350
2351 \cs_new_protected:Npn \__kernel_register_show_aux:nNN #1#2#3 2351
2352 { \exp_args:No #3 { \token_to_str:N #2 = #1 } } 2352
2353 \cs_new_protected:Npn \cs_show:N { \__kernel_show:NN \tl_show:n } 2353
2354 \cs_new_protected:Npn \cs_show:c 2354
2355 { \group_begin: \exp_args:Nnc \group_end: \cs_show:N } 2355
2356 \cs_new_protected:Npn \cs_log:N { \__kernel_show:NN \tl_log:n } 2356
2357 \cs_new_protected:Npn \cs_log:c 2357
2358 { \group_begin: \exp_args:Nnc \group_end: \cs_log:N } 2358
2359 \cs_new_protected:Npn \__kernel_show:NN #1#2 2359
2360 { 2360
2361 \group_begin: 2361
2362 \int_set:Nn \tex_escapechar:D { `` } 2362
2363 \exp_args:NNe 2363
2364 \group_end: 2364
2365 #1 { \token_to_str:N #2 = \cs_meaning:N #2 } 2365
2366 } 2366
2367 \cs_new_protected:Npn \group_show_list: 2367
2368 { \__kernel_group_show:NN \use_none:n 1 } 2368
2369 \cs_new_protected:Npn \group_log_list: 2369
2370 { \__kernel_group_show:NN \int_gzero:N 0 } 2370
2371 \cs_new_protected:Npn \__kernel_group_show:NN #1#2 2371
2372 { 2372
2373 \use:e 2373
2374 { 2374
2375 #1 \tex_interactionmode:D 2375
2376 \int_set:Nn \tex_tracingonline:D {#2} 2376
2377 \int_set:Nn \tex_errorcontextlines:D { -1 } 2377
2378 \exp_not:N \exp_after:wN \scan_stop: 2378
2379 \tex_showgroups:D 2379
2380 \int_gset:Nn \tex_interactionmode:D 2380
2381 { \int_use:N \tex_interactionmode:D } 2381
2382 \int_set:Nn \tex_tracingonline:D 2382
2383 { \int_use:N \tex_tracingonline:D } 2383
2384 \int_set:Nn \tex_errorcontextlines:D 2384
2385 { \int_use:N \tex_errorcontextlines:D } 2385
2386 } 2386
2387 } 2387
2388 \use:e 2388
2389 { 2389
2390 \exp_not:n { \cs_new:Npn \__kernel_prefix_arg_replacement:wN #1 } 2390
2391 \tl_to_str:n { macro : } \exp_not:n { #2 -> #3 \s__kernel_stop #4 } 2391
2392 } 2392

```


2393	{ #4 {#1} {#2} {#3} }	2393
2394	\cs_new:Npn \cs_prefix_spec:N #1	2394
2395	{	2395
2396	\token_if_macro:NTF #1	2396
2397	{	2397
2398	\exp_after:wN __kernel_prefix_arg_replacement:wN	2398
2399	\token_to_meaning:N #1 \s__kernel_stop \use_i:nnn	2399
2400	}	2400
2401	{ \scan_stop: }	2401
2402	}	2402
2403	\cs_new:Npn \cs_parameter_spec:N #1	2403
2404	{	2404
2405	\token_if_macro:NTF #1	2405
2406	{	2406
2407	\exp_after:wN __kernel_prefix_arg_replacement:wN	2407
2408	\token_to_meaning:N #1 \s__kernel_stop \use_ii:nnn	2408
2409	}	2409
2410	{ \scan_stop: }	2410
2411	}	2411
2412	\cs_new:Npn \cs_replacement_spec:N #1	2412
2413	{	2413
2414	\token_if_macro:NTF #1	2414
2415	{	2415
2416	\exp_after:wN __kernel_prefix_arg_replacement:wN	2416
2417	\token_to_meaning:N #1 \s__kernel_stop \use_iii:nnn	2417
2418	}	2418
2419	{ \scan_stop: }	2419
2420	}	2420
2421	\cs_new:Npn \prg_do_nothing: { }	2421
2422	\cs_new_eq:NN \prg_break_point:Nn \use_ii:nn	2422
2423	\cs_new:Npn \prg_map_break:Nn #1#2#3 \prg_break_point:Nn #4#5	2423
2424	{	2424
2425	#5	2425
2426	\if_meaning:w #1 #4	2426
2427	\exp_after:wN \use_iii:nnn	2427
2428	\fi:	2428
2429	\prg_map_break:Nn #1 {#2}	2429
2430	}	2430
2431	\cs_new_eq:NN \prg_break_point: \prg_do_nothing:	2431
2432	\cs_new:Npn \prg_break: #1 \prg_break_point: { }	2432
2433	\cs_new:Npn \prg_break:n #1#2 \prg_break_point: {#1}	2433
2434	\cs_new_protected:Npn \mode_leave_vertical:	2434
2435	{	2435
2436	\if_mode_vertical:	2436
2437	\exp_after:wN \tex_indent:D	2437
2438	\fi:	2438

```

2439 }
2440 %% File: l3expan.dtx
2441 \cs_new:Npn \__exp_arg_next:nnn #1#2#3 { #2 \::: { #3 {#1} } }
2442 \cs_new:Npn \__exp_arg_next:Nnn #1#2#3 { #2 \::: { #3 #1 } }
2443 \cs_new:Npn \::: #1 {#1}
2444 \cs_new:Npn \::n #1 \::: #2#3 { #1 \::: { #2 {#3} } }
2445 \cs_new:Npn \::N #1 \::: #2#3 { #1 \::: {#2#3} }
2446 \cs_new:Npn \::p #1 \::: #2#3# { #1 \::: {#2#3} }
2447 \cs_new:Npn \::c #1 \::: #2#3
2448 { \exp_after:wN \__exp_arg_next:Nnn \cs:w #3 \cs_end: {#1} {#2} }
2449 \cs_new:Npn \::o #1 \::: #2#3
2450 { \exp_after:wN \__exp_arg_next:nnn \exp_after:wN {#3} {#1} {#2} }
2451 \cs_new:Npn \::e #1 \::: #2#3
2452 { \tex_expanded:D { \exp_not:n { #1 \::: } { \exp_not:n {#2} {#3} } } }
2453 \cs_new:Npn \::f #1 \::: #2#3
2454 {
2455   \exp_after:wN \__exp_arg_next:nnn
2456   \exp_after:wN { \exp:w \exp_end_continue_f:w #3 }
2457   {#1} {#2}
2458 }
2459 \use:nn { \cs_new_eq:NN \exp_stop_f: } { ~ }
2460 \cs_new_protected:Npn \::x #1 \::: #2#3
2461 {
2462   \cs_set_nopar:Npe \l__exp_internal_tl
2463   { \exp_not:n { #1 \::: } { \exp_not:n {#2} {#3} } }
2464   \l__exp_internal_tl
2465 }
2466 \cs_new:Npn \::V #1 \::: #2#3
2467 {
2468   \exp_after:wN \__exp_arg_next:nnn
2469   \exp_after:wN { \exp:w \__exp_eval_register:N #3 }
2470   {#1} {#2}
2471 }
2472 \cs_new:Npn \::v #1 \::: #2#3
2473 {
2474   \exp_after:wN \__exp_arg_next:nnn
2475   \exp_after:wN { \exp:w \__exp_eval_register:c {#3} }
2476   {#1} {#2}
2477 }
2478 \cs_new:Npn \__exp_eval_register:N #1
2479 {
2480   \exp_after:wN \if_meaning:w \exp_not:N #1 #1
2481   \if_meaning:w \scan_stop: #1
2482   \__exp_eval_error_msg:w
2483   \fi:
2484   \else:

```

```

2485 \exp_after:wN \use_i_ii:nnn
2486 \fi:
2487 \exp_after:wN \exp_end: \tex_the:D #1
2488 }
2489 \cs_new:Npn \__exp_eval_register:c #1
2490 { \exp_after:wN \__exp_eval_register:N \cs:w #1 \cs_end: }
2491 \cs_new:Npn \__exp_eval_error_msg:w #1 \tex_the:D #2
2492 {
2493     \fi:
2494     \fi:
2495     \msg_expandable_error:nnn { kernel } { bad-variable } {#2}
2496     \exp_end:
2497 }
2498 \cs_new:Npn \exp_args:NNc #1#2#3
2499 { \exp_after:wN #1 \exp_after:wN #2 \cs:w # 3\cs_end: }
2500 \cs_new:Npn \exp_args:Ncc #1#2#3
2501 { \exp_after:wN #1 \cs:w #2 \exp_after:wN \cs_end: \cs:w #3 \cs_end: }
2502 \cs_new:Npn \exp_args:Nccc #1#2#3#4
2503 {
2504     \exp_after:wN #1
2505     \cs:w #2 \exp_after:wN \cs_end:
2506     \cs:w #3 \exp_after:wN \cs_end:
2507     \cs:w #4 \cs_end:
2508 }
2509 \cs_new:Npn \exp_args:No #1#2 { \exp_after:wN #1 \exp_after:wN {#2} }
2510 \cs_new:Npn \exp_args:NNo #1#2#3
2511 { \exp_after:wN #1 \exp_after:wN #2 \exp_after:wN {#3} }
2512 \cs_new:Npn \exp_args:NNNo #1#2#3#4
2513 { \exp_after:wN #1 \exp_after:wN#2 \exp_after:wN #3 \exp_after:wN {#4} }
2514 \cs_new:Npn \exp_args:Ne #1#2
2515 { \exp_after:wN #1 \tex_expanded:D { {#2} } }
2516 \cs_new:Npn \exp_args:Nf #1#2
2517 { \exp_after:wN #1 \exp_after:wN { \exp:w \exp_end_continue_f:w #2 } }
2518 \cs_new:Npn \exp_args:Nv #1#2
2519 {
2520     \exp_after:wN #1 \exp_after:wN
2521     { \exp:w \__exp_eval_register:c {#2} }
2522 }
2523 \cs_new:Npn \exp_args:NV #1#2
2524 {
2525     \exp_after:wN #1 \exp_after:wN
2526     { \exp:w \__exp_eval_register:N #2 }
2527 }
2528 \cs_new:Npn \exp_args:NNV #1#2#3
2529 {
2530     \exp_after:wN #1

```

2531	\exp_after:wN #2	2531
2532	\exp_after:wN { \exp:w __exp_eval_register:N #3 }	2532
2533	}	2533
2534	\cs_new:Npn \exp_args:NNv #1#2#3	2534
2535	{	2535
2536	\exp_after:wN #1	2536
2537	\exp_after:wN #2	2537
2538	\exp_after:wN { \exp:w __exp_eval_register:c {#3} }	2538
2539	}	2539
2540	\cs_new:Npn \exp_args:NNe #1#2#3	2540
2541	{	2541
2542	\exp_after:wN #1	2542
2543	\exp_after:wN #2	2543
2544	\tex_expanded:D { {#3} }	2544
2545	}	2545
2546	\cs_new:Npn \exp_args:NNf #1#2#3	2546
2547	{	2547
2548	\exp_after:wN #1	2548
2549	\exp_after:wN #2	2549
2550	\exp_after:wN { \exp:w \exp_end_continue_f:w #3 }	2550
2551	}	2551
2552	\cs_new:Npn \exp_args:Nco #1#2#3	2552
2553	{	2553
2554	\exp_after:wN #1	2554
2555	\cs:w #2 \exp_after:wN \cs_end:	2555
2556	\exp_after:wN {#3}	2556
2557	}	2557
2558	\cs_new:Npn \exp_args:NcV #1#2#3	2558
2559	{	2559
2560	\exp_after:wN #1	2560
2561	\cs:w #2 \exp_after:wN \cs_end:	2561
2562	\exp_after:wN { \exp:w __exp_eval_register:N #3 }	2562
2563	}	2563
2564	\cs_new:Npn \exp_args:Ncv #1#2#3	2564
2565	{	2565
2566	\exp_after:wN #1	2566
2567	\cs:w #2 \exp_after:wN \cs_end:	2567
2568	\exp_after:wN { \exp:w __exp_eval_register:c {#3} }	2568
2569	}	2569
2570	\cs_new:Npn \exp_args:Ncf #1#2#3	2570
2571	{	2571
2572	\exp_after:wN #1	2572
2573	\cs:w #2 \exp_after:wN \cs_end:	2573
2574	\exp_after:wN { \exp:w \exp_end_continue_f:w #3 }	2574
2575	}	2575
2576	\cs_new:Npn \exp_args:NVV #1#2#3	2576

2577	{	2577
2578	\exp_after:wN #1	2578
2579	\exp_after:wN { \exp:w \exp_after:wN	2579
2580	__exp_eval_register:N \exp_after:wN #2 \exp_after:wN }	2580
2581	\exp_after:wN { \exp:w __exp_eval_register:N #3 }	2581
2582	}	2582
2583	\cs_new:Npn \exp_args:NNNV #1#2#3#4	2583
2584	{	2584
2585	\exp_after:wN #1	2585
2586	\exp_after:wN #2	2586
2587	\exp_after:wN #3	2587
2588	\exp_after:wN { \exp:w __exp_eval_register:N #4 }	2588
2589	}	2589
2590	\cs_new:Npn \exp_args:NNNV #1#2#3#4	2590
2591	{	2591
2592	\exp_after:wN #1	2592
2593	\exp_after:wN #2	2593
2594	\exp_after:wN #3	2594
2595	\exp_after:wN { \exp:w __exp_eval_register:c {#4} }	2595
2596	}	2596
2597	\cs_new:Npn \exp_args:NNNe #1#2#3#4	2597
2598	{	2598
2599	\exp_after:wN #1	2599
2600	\exp_after:wN #2	2600
2601	\exp_after:wN #3	2601
2602	\tex_expanded:D { {#4} }	2602
2603	}	2603
2604	\cs_new:Npn \exp_args:NcNc #1#2#3#4	2604
2605	{	2605
2606	\exp_after:wN #1	2606
2607	\cs:w #2 \exp_after:wN \cs_end:	2607
2608	\exp_after:wN #3	2608
2609	\cs:w #4 \cs_end:	2609
2610	}	2610
2611	\cs_new:Npn \exp_args:NcNo #1#2#3#4	2611
2612	{	2612
2613	\exp_after:wN #1	2613
2614	\cs:w #2 \exp_after:wN \cs_end:	2614
2615	\exp_after:wN #3	2615
2616	\exp_after:wN {#4}	2616
2617	}	2617
2618	\cs_new:Npn \exp_args:Ncco #1#2#3#4	2618
2619	{	2619
2620	\exp_after:wN #1	2620
2621	\cs:w #2 \exp_after:wN \cs_end:	2621
2622	\cs:w #3 \exp_after:wN \cs_end:	2622

2623	\exp_after:wN {#4}	2623
2624	}	2624
2625	\cs_new_protected:Npn \exp_args:Nx #1#2	2625
2626	{ \use:x { \exp_not:N #1 {#2} } }	2626
2627	\cs_new:Npn __exp_arg_last_unbraced:nn #1#2 { #2#1 }	2627
2628	\cs_new:Npn \::o_unbraced \::: #1#2	2628
2629	{ \exp_after:wN __exp_arg_last_unbraced:nn \exp_after:wN {#2} {#1} }	2629
2630	\cs_new:Npn \::V_unbraced \::: #1#2	2630
2631	{	2631
2632	\exp_after:wN __exp_arg_last_unbraced:nn	2632
2633	\exp_after:wN { \exp:w __exp_eval_register:N #2 } {#1}	2633
2634	}	2634
2635	\cs_new:Npn \::v_unbraced \::: #1#2	2635
2636	{	2636
2637	\exp_after:wN __exp_arg_last_unbraced:nn	2637
2638	\exp_after:wN { \exp:w __exp_eval_register:c {#2} } {#1}	2638
2639	}	2639
2640	\cs_new:Npn \::e_unbraced \::: #1#2	2640
2641	{ \tex_expanded:D { \exp_not:n {#1} #2 } }	2641
2642	\cs_new:Npn \::f_unbraced \::: #1#2	2642
2643	{	2643
2644	\exp_after:wN __exp_arg_last_unbraced:nn	2644
2645	\exp_after:wN { \exp:w \exp_end_continue_f:w #2 } {#1}	2645
2646	}	2646
2647	\cs_new_protected:Npn \::x_unbraced \::: #1#2	2647
2648	{	2648
2649	\cs_set_nopar:Npe \l__exp_internal_tl { \exp_not:n {#1} #2 }	2649
2650	\l__exp_internal_tl	2650
2651	}	2651
2652	\cs_new:Npn \exp_last_unbraced:No #1#2 { \exp_after:wN #1 #2 }	2652
2653	\cs_new:Npn \exp_last_unbraced:NV #1#2	2653
2654	{ \exp_after:wN #1 \exp:w __exp_eval_register:N #2 }	2654
2655	\cs_new:Npn \exp_last_unbraced:Nv #1#2	2655
2656	{ \exp_after:wN #1 \exp:w __exp_eval_register:c {#2} }	2656
2657	\cs_new:Npn \exp_last_unbraced:Ne #1#2	2657
2658	{ \exp_after:wN #1 \tex_expanded:D {#2} }	2658
2659	\cs_new:Npn \exp_last_unbraced:Nf #1#2	2659
2660	{ \exp_after:wN #1 \exp:w \exp_end_continue_f:w #2 }	2660
2661	\cs_new:Npn \exp_last_unbraced:NNo #1#2#3	2661
2662	{ \exp_after:wN #1 \exp_after:wN #2 #3 }	2662
2663	\cs_new:Npn \exp_last_unbraced:NNV #1#2#3	2663
2664	{	2664
2665	\exp_after:wN #1	2665
2666	\exp_after:wN #2	2666
2667	\exp:w __exp_eval_register:N #3	2667
2668	}	2668

2669	\cs_new:Npn \exp_last_unbraced:NNf #1#2#3	2669
2670	{	2670
2671	\exp_after:wN #1	2671
2672	\exp_after:wN #2	2672
2673	\exp:w \exp_end_continue_f:w #3	2673
2674	}	2674
2675	\cs_new:Npn \exp_last_unbraced:Nco #1#2#3	2675
2676	{ \exp_after:wN #1 \cs:w #2 \exp_after:wN \cs_end: #3 }	2676
2677	\cs_new:Npn \exp_last_unbraced:NcV #1#2#3	2677
2678	{	2678
2679	\exp_after:wN #1	2679
2680	\cs:w #2 \exp_after:wN \cs_end:	2680
2681	\exp:w __exp_eval_register:N #3	2681
2682	}	2682
2683	\cs_new:Npn \exp_last_unbraced:NNNo #1#2#3#4	2683
2684	{ \exp_after:wN #1 \exp_after:wN #2 \exp_after:wN #3 #4 }	2684
2685	\cs_new:Npn \exp_last_unbraced:NNNV #1#2#3#4	2685
2686	{	2686
2687	\exp_after:wN #1	2687
2688	\exp_after:wN #2	2688
2689	\exp_after:wN #3	2689
2690	\exp:w __exp_eval_register:N #4	2690
2691	}	2691
2692	\cs_new:Npn \exp_last_unbraced:NNNf #1#2#3#4	2692
2693	{	2693
2694	\exp_after:wN #1	2694
2695	\exp_after:wN #2	2695
2696	\exp_after:wN #3	2696
2697	\exp:w \exp_end_continue_f:w #4	2697
2698	}	2698
2699	\cs_new:Npn \exp_last_unbraced:Nno { \::n \::o_unbraced \::: }	2699
2700	\cs_new:Npn \exp_last_unbraced:Nnf { \::n \::f_unbraced \::: }	2700
2701	\cs_new:Npn \exp_last_unbraced:Noo { \::o \::o_unbraced \::: }	2701
2702	\cs_new:Npn \exp_last_unbraced:Nfo { \::f \::o_unbraced \::: }	2702
2703	\cs_new:Npn \exp_last_unbraced:NnNo { \::n \::N \::o_unbraced \::: }	2703
2704	\cs_new:Npn \exp_last_unbraced:NNNNo #1#2#3#4#5	2704
2705	{ \exp_after:wN #1 \exp_after:wN #2 \exp_after:wN #3 \exp_after:wN #4 #5 }	2705
2706	\cs_new:Npn \exp_last_unbraced:NNNNf #1#2#3#4#5	2706
2707	{	2707
2708	\exp_after:wN #1	2708
2709	\exp_after:wN #2	2709
2710	\exp_after:wN #3	2710
2711	\exp_after:wN #4	2711
2712	\exp:w \exp_end_continue_f:w #5	2712
2713	}	2713
2714	\cs_new_protected:Npn \exp_last_unbraced:Nx { \::x_unbraced \::: }	2714

```

2715 \cs_new:Npn \exp_last_two_unbraced:Noo #1#2#3 2715
2716 { \exp_after:wN \exp_last_two_unbraced:nn \exp_after:wN {#3} {#2} #1 } 2716
2717 \cs_new:Npn \exp_last_two_unbraced:nn #1#2#3 2717
2718 { \exp_after:wN #3 #2 #1 } 2718
2719 \cs_new_eq:NN \__kernel_exp_not:w \tex_unexpanded:D 2719
2720 \cs_new:Npn \exp_not:c #1 { \exp_after:wN \exp_not:N \cs:w #1 \cs_end: } 2720
2721 \cs_new:Npn \exp_not:o #1 { \__kernel_exp_not:w \exp_after:wN {#1} } 2721
2722 \cs_new:Npn \exp_not:e #1 2722
2723 { \__kernel_exp_not:w \tex_expanded:D { {#1} } } 2723
2724 \cs_new:Npn \exp_not:f #1 2724
2725 { \__kernel_exp_not:w \exp_after:wN { \exp:w \exp_end_continue_f:w #1 } } 2725
2726 \cs_new:Npn \exp_not:V #1 2726
2727 { 2727
2728 \__kernel_exp_not:w \exp_after:wN 2728
2729 { \exp:w \exp_eval_register:N #1 } 2729
2730 } 2730
2731 \cs_new:Npn \exp_not:v #1 2731
2732 { 2732
2733 \__kernel_exp_not:w \exp_after:wN 2733
2734 { \exp:w \exp_eval_register:c {#1} } 2734
2735 } 2735
2736 \group_begin: 2736
2737 \tex_catcode:D \^^@ = 13 2737
2738 \cs_new_protected:Npn \exp_end_continue_f:w { \^^@ } 2738
2739 \if_cs_exist:N \^^@ 2739
2740 \else: 2740
2741 \cs_new:Npn \^^@ 2741
2742 { \msg_expandable_error:nn { kernel } { bad-exp-end-f } } 2742
2743 \fi: 2743
2744 \cs_new:Npn \exp_end_continue_f:nw #1 { \^^@ #1 } 2744
2745 \group_end: 2745
2746 \cs_new_eq:NN \s__cs_mark \scan_stop: 2746
2747 \cs_new_eq:NN \s__cs_stop \scan_stop: 2747
2748 \cs_new:Npn \q__cs_recursion_stop { \q__cs_recursion_stop } 2748
2749 \cs_new:Npn \__cs_use_none_delimit_by_s_stop:w #1 \s__cs_stop { } 2749
2750 \cs_new:Npn \__cs_use_i_delimit_by_s_stop:nw #1 #2 \s__cs_stop {#1} 2750
2751 \cs_new:Npn \__cs_use_none_delimit_by_q_recursion_stop:w 2751
2752 #1 \q__cs_recursion_stop { } 2752
2753 \cs_new_protected:Npn \cs_generate_variant:Nn #1#2 2753
2754 { 2754
2755 \__cs_generate_variant:N #1 2755
2756 \use:e 2756
2757 { 2757
2758 \__cs_generate_variant:nnNN 2758
2759 \cs_split_function:N #1 2759
2760 \exp_not:N #1 2760

```

```

2761         \tl_to_str:n {#2} ,
2762         \exp_not:N \scan_stop: ,
2763         \exp_not:N \q__cs_recursion_stop
2764     }
2765 }
2766 \cs_new_protected:Npn \cs_generate_variant:cn
2767 { \exp_args:Nc \cs_generate_variant:Nn }
2768 \cs_new_protected:Npe \__cs_generate_variant:N #1
2769 {
2770     \exp_not:N \exp_after:wN \exp_not:N \if_meaning:w
2771     \exp_not:N \exp_not:N #1 #1
2772     \cs_set_eq:NN \exp_not:N \__cs_tmp:w \cs_new_protected:Npe
2773     \exp_not:N \else:
2774     \exp_not:N \exp_after:wN \exp_not:N \__cs_generate_variant:ww
2775     \exp_not:N \token_to_meaning:N #1 \tl_to_str:n { ma }
2776     \s__cs_mark
2777     \s__cs_mark \cs_new_protected:Npe
2778     \tl_to_str:n { pr }
2779     \s__cs_mark \cs_new:Npe
2780     \s__cs_stop
2781     \exp_not:N \fi:
2782 }
2783 \exp_last_unbraced:NNNNNo
2784 \cs_new_protected:Npn \__cs_generate_variant:ww
2785 #1 { \tl_to_str:n { ma } } #2 \s__cs_mark
2786 { \__cs_generate_variant:wwNw #1 }
2787 \exp_last_unbraced:NNNNNo
2788 \cs_new_protected:Npn \__cs_generate_variant:wwNw
2789 #1 { \tl_to_str:n { pr } } #2 \s__cs_mark #3 #4 \s__cs_stop
2790 { \cs_set_eq:NN \__cs_tmp:w #3 }
2791 \cs_new_protected:Npn \__cs_generate_variant:nnNN #1#2#3#4
2792 {
2793     \if_meaning:w \c_false_bool #3
2794     \msg_error:nne { kernel } { missing-colon }
2795     { \token_to_str:c {#1} }
2796     \exp_after:wN \__cs_use_none_delimit_by_q_recursion_stop:w
2797     \fi:
2798     \__cs_generate_variant:Nnnw #4 {#1}{#2}
2799 }
2800 \cs_new_protected:Npn \__cs_generate_variant:Nnnw #1#2#3#4 ,
2801 {
2802     \if_meaning:w \scan_stop: #4
2803     \exp_after:wN \__cs_use_none_delimit_by_q_recursion_stop:w
2804     \fi:
2805     \use:e
2806     {

```

2807	\exp_not:N __cs_generate_variant:wwNN	2807
2808	__cs_generate_variant_loop:nNwN { }	2808
2809	#4	2809
2810	__cs_generate_variant_loop_end:nwwwNNnn	2810
2811	\s__cs_mark	2811
2812	#3 ~	2812
2813	{ ~ { } \fi: __cs_generate_variant_loop_long:wNNnn } ~	2813
2814	{ }	2814
2815	\s__cs_stop	2815
2816	\exp_not:N #1 {#2} {#4}	2816
2817	}	2817
2818	__cs_generate_variant:Nnnw #1 {#2} {#3}	2818
2819	}	2819
2820	\cs_new:Npn __cs_generate_variant_loop:nNwN #1#2#3 \s__cs_mark #4	2820
2821	{	2821
2822	\if:w #2 #4	2822
2823	\exp_after:wN __cs_generate_variant_loop_same:w	2823
2824	\else:	2824
2825	\if:w #4 __cs_generate_variant_loop_base:N #2 \else:	2825
2826	\if:w 0	2826
2827	\if:w N #4 \else: \if:w n #4 \else: 1 \fi: \fi:	2827
2828	\if:w \scan_stop: __cs_generate_variant_loop_base:N #2 1 \fi:	2828
2829	0	2829
2830	__cs_generate_variant_loop_special:NNwNNnn #4#2	2830
2831	\else:	2831
2832	__cs_generate_variant_loop_invalid:NNwNNnn #4#2	2832
2833	\fi:	2833
2834	\fi:	2834
2835	\fi:	2835
2836	#1	2836
2837	\prg_do_nothing:	2837
2838	#2	2838
2839	__cs_generate_variant_loop:nNwN { } #3 \s__cs_mark	2839
2840	}	2840
2841	\cs_new:Npn __cs_generate_variant_loop_base:N #1	2841
2842	{	2842
2843	\if:w c #1 N \else:	2843
2844	\if:w o #1 n \else:	2844
2845	\if:w V #1 n \else:	2845
2846	\if:w v #1 n \else:	2846
2847	\if:w f #1 n \else:	2847
2848	\if:w e #1 n \else:	2848
2849	\if:w x #1 n \else:	2849
2850	\if:w n #1 n \else:	2850
2851	\if:w N #1 N \else:	2851
2852	\scan_stop:	2852

2853	\fi:	2853
2854	\fi:	2854
2855	\fi:	2855
2856	\fi:	2856
2857	\fi:	2857
2858	\fi:	2858
2859	\fi:	2859
2860	\fi:	2860
2861	\fi:	2861
2862	}	2862
2863	\cs_new:Npn __cs_generate_variant_loop_same:w	2863
2864	#1 \prg_do_nothing: #2#3#4	2864
2865	{ #3 { #1 __cs_generate_variant_same:N #2 } }	2865
2866	\cs_new:Npn __cs_generate_variant_loop_end:nwwwNNnn	2866
2867	#1#2 \s__cs_mark #3 ~ #4 \s__cs_stop #5#6#7#8	2867
2868	{	2868
2869	\scan_stop: \scan_stop: \fi:	2869
2870	\s__cs_mark \s__cs_stop	2870
2871	\exp_not:N #6	2871
2872	\exp_not:c { #7 : #8 #1 #3 }	2872
2873	}	2873
2874	\cs_new:Npn __cs_generate_variant_loop_long:wNNnn #1 \s__cs_stop #2#3#4#5	2874
2875	{	2875
2876	\exp_not:n	2876
2877	{	2877
2878	\s__cs_mark	2878
2879	\msg_error:nnee { kernel } { variant-too-long }	2879
2880	{#5} { \token_to_str:N #3 }	2880
2881	\use_none:nnn	2881
2882	\s__cs_stop	2882
2883	#3	2883
2884	#3	2884
2885	}	2885
2886	}	2886
2887	\cs_new:Npn __cs_generate_variant_loop_invalid:NNwNNnn	2887
2888	#1#2 \fi: \fi: \fi: #3 \s__cs_stop #4#5#6#7	2888
2889	{	2889
2890	\fi: \fi: \fi:	2890
2891	\exp_not:n	2891
2892	{	2892
2893	\s__cs_mark	2893
2894	\msg_error:nneeee { kernel } { invalid-variant }	2894
2895	{#7} { \token_to_str:N #5 } {#1} {#2}	2895
2896	\use_none:nnn	2896
2897	\s__cs_stop	2897
2898	#5	2898

```
2899         #5
2900     }
2901 }
2902 \cs_new:Npn \__cs_generate_variant_loop_special:NNwNNnn
2903   #1#2#3 \s__cs_stop #4#5#6#7
2904 {
2905     #3 \s__cs_stop #4 #5 {#6} {#7}
2906     \exp_not:n
2907     {
2908         \msg_error:nneeee
2909         { kernel } { deprecated-variant }
2910         {#7} { \token_to_str:N #5 } {#1} {#2}
2911     }
2912 }
2913 \cs_new:Npn \__cs_generate_variant_same:N #1
2914 {
2915     \if:w N #1 #1 \else:
2916     \if:w p #1 #1 \else:
2917     \token_to_str:N n
2918     \if:w n #1 \else:
2919     \__cs_generate_variant_loop_special:NNwNNnn #1#1
2920     \fi:
2921     \fi:
2922     \fi:
2923 }
2924 \cs_new_protected:Npn \__cs_generate_variant:wwNN
2925   #1 \s__cs_mark #2 \s__cs_stop #3#4
2926 {
2927     #2
2928     \cs_if_free:NT #4
2929     {
2930         \group_begin:
2931         \__cs_generate_internal_variant:n {#1}
2932         \__cs_tmp:w #4 { \exp_not:c { exp_args:N #1 } \exp_not:N #3 }
2933         \group_end:
2934     }
2935 }
2936 \cs_new_protected:Npe \__cs_generate_internal_variant:n #1
2937 {
2938     \exp_not:N \__cs_generate_internal_variant:wwnNwn
2939     #1 \s__cs_mark
2940     { \cs_set_eq:NN \exp_not:N \__cs_tmp:w \cs_new_protected:Npe }
2941     \cs_new_protected:cpn
2942     \use:x
2943     \token_to_str:N x \s__cs_mark
2944     { }
```

```

2945         \cs_new:cpn                                2945
2946         \exp_not:N \tex_expanded:D                  2946
2947     \s__cs_stop                                     2947
2948     {#1}                                             2948
2949 }                                                    2949
2950 \exp_last_unbraced:NNNNo                            2950
2951     \cs_new_protected:Npn \__cs_generate_internal_variant:wwnNwn #1 2951
2952     { \token_to_str:N x } #2 \s__cs_mark #3#4#5#6 \s__cs_stop #7 2952
2953 {                                                    2953
2954     #3                                              2954
2955     \cs_if_free:cT { exp_args:N #7 }                2955
2956     { \__cs_generate_internal_variant:NNn #4 #5 {#7} } 2956
2957 }                                                    2957
2958 \cs_set_protected:Npn \__cs_tmp:w #1                2958
2959 {                                                    2959
2960     \cs_new_protected:Npn \__cs_generate_internal_variant:NNn ##1##2##3 2960
2961     {                                              2961
2962         \if_catcode:w X \use_none:nnnnnnnn ##3      2962
2963         \prg_do_nothing: \prg_do_nothing: \prg_do_nothing: 2963
2964         \prg_do_nothing: \prg_do_nothing: \prg_do_nothing: 2964
2965         \prg_do_nothing: \prg_do_nothing: X          2965
2966         \exp_after:wN \__cs_generate_internal_test:Nw \exp_after:wN ##2 2966
2967     \else:                                          2967
2968         \exp_after:wN \__cs_generate_internal_test_aux:w \exp_after:wN #1 2968
2969     \fi:                                           2969
2970     ##3                                             2970
2971     \s__cs_mark                                    2971
2972     {                                              2972
2973         \use:e                                       2973
2974         {                                           2974
2975             ##1 { exp_args:N ##3 }                  2975
2976             { \__cs_generate_internal_variant_loop:n ##3 { : \use_i:nn } } 2976
2977         }                                           2977
2978     }                                              2978
2979     #1                                              2979
2980     \s__cs_mark                                    2980
2981     { \exp_not:n { \__cs_generate_internal_one_go:NNn ##1 ##2 {##3} } } 2981
2982     \s__cs_stop                                    2982
2983 }                                                  2983
2984 \cs_new_protected:Npn \__cs_generate_internal_test_aux:w 2984
2985     ##1 #1 ##2 \s__cs_mark ##3 ##4 \s__cs_stop {##3} 2985
2986 \cs_new_eq:NN \__cs_generate_internal_test:Nw          2986
2987     \__cs_generate_internal_test_aux:w                 2987
2988 }                                                    2988
2989 \exp_args:No \__cs_tmp:w { \token_to_str:N p }      2989
2990 \cs_new_protected:Npn \__cs_generate_internal_one_go:NNn #1#2#3 2990

```

```
2991 {
2992     \__cs_generate_internal_loop:nwnnw
2993     { \exp_not:N ##1 } 1 . { } { }
2994     #3 { ? \__cs_generate_internal_end:w } X ;
2995     23456789 { ? \__cs_generate_internal_long:w } ;
2996     #1 #2 {#3}
2997 }
2998 \cs_new_protected:Npn \__cs_generate_internal_loop:nwnnw #1#2 . #3#4#5#6 ; #7
2999 {
3000     \use_none:n #5
3001     \use_none:n #7
3002     \cs_if_exist_use:cF { __cs_generate_internal_#5:NN }
3003     { \__cs_generate_internal_other:NN }
3004     #5 #7
3005     #7 .
3006     { #3 #1 } { #4 ## #2 }
3007     #6 ;
3008 }
3009 \cs_new_protected:Npn \__cs_generate_internal_N:NN #1#2
3010 { \__cs_generate_internal_loop:nwnnw { \exp_not:N ###2 } }
3011 \cs_new_protected:Npn \__cs_generate_internal_c:NN #1#2
3012 { \exp_args:No \__cs_generate_internal_loop:nwnnw { \exp_not:c {###2} } }
3013 \cs_new_protected:Npn \__cs_generate_internal_n:NN #1#2
3014 { \__cs_generate_internal_loop:nwnnw { { \exp_not:n {###2} } } }
3015 \cs_new_protected:Npn \__cs_generate_internal_x:NN #1#2
3016 { \__cs_generate_internal_loop:nwnnw { {###2} } }
3017 \cs_new_protected:Npn \__cs_generate_internal_other:NN #1#2
3018 {
3019     \exp_args:No \__cs_generate_internal_loop:nwnnw
3020     {
3021         \exp_after:wN
3022         {
3023             \exp:w \exp_args:NNc \exp_after:wN \exp_end:
3024             { exp_not:#1 } {###2}
3025         }
3026     }
3027 }
3028 \cs_new_protected:Npn \__cs_generate_internal_end:w #1 . #2#3#4 ; #5 ; #6#7#8
3029 { #6 { exp_args:N #8 } #3 { #7 {#2} } }
3030 \cs_new_protected:Npn \__cs_generate_internal_long:w #1 N #2#3 . #4#5#6#
3031 {
3032     \exp_args:Nx \__cs_generate_internal_long:nnnNNn
3033     { \__cs_generate_internal_variant_loop:n #2 #6 { : \use_i:nn } }
3034     {#4} {#5}
3035 }
3036 \cs_new:Npn \__cs_generate_internal_long:nnnNNn #1#2#3#4 ; ; #5#6#7
```

```
3037 { #5 { exp_args:N #7 } #3 { #6 { \exp_not:n {#1} {#2} } } } 3037
3038 \cs_new:Npn \__cs_generate_internal_variant_loop:n #1 3038
3039 { 3039
3040 \exp_after:wN \exp_not:N \cs:w :: #1 \cs_end: 3040
3041 \__cs_generate_internal_variant_loop:n 3041
3042 } 3042
3043 \cs_new_protected:Npn \prg_generate_conditional_variant:Nnn #1 3043
3044 { 3044
3045 \use:e 3045
3046 { 3046
3047 \__cs_generate_variant:nnNnn 3047
3048 \cs_split_function:N #1 3048
3049 } 3049
3050 } 3050
3051 \cs_new_protected:Npn \__cs_generate_variant:nnNnn #1#2#3#4#5 3051
3052 { 3052
3053 \if_meaning:w \c_false_bool #3 3053
3054 \msg_error:nne { kernel } { missing-colon } 3054
3055 { \token_to_str:c {#1} } 3055
3056 \__cs_use_i_delimit_by_s_stop:nw 3056
3057 \fi: 3057
3058 \exp_after:wN \__cs_generate_variant:w 3058
3059 \tl_to_str:n {#5} , \scan_stop: , \q__cs_recursion_stop 3059
3060 \__cs_use_none_delimit_by_s_stop:w \s__cs_mark {#1} {#2} {#4} \s__cs_stop 3060
3061 } 3061
3062 \cs_new_protected:Npn \__cs_generate_variant:w 3062
3063 #1 , #2 \s__cs_mark #3#4#5 3063
3064 { 3064
3065 \if_meaning:w \scan_stop: #1 \scan_stop: 3065
3066 \if_meaning:w \q__cs_nil #1 \q__cs_nil 3066
3067 \use_i:nnn 3067
3068 \fi: 3068
3069 \exp_after:wN \__cs_use_none_delimit_by_q_recursion_stop:w 3069
3070 \else: 3070
3071 \cs_if_exist_use:cTF { __cs_generate_variant_#1_form:nnn } 3071
3072 { {#3} {#4} {#5} } 3072
3073 { 3073
3074 \msg_error:nnee 3074
3075 { kernel } { conditional-form-unknown } 3075
3076 {#1} { \token_to_str:c { #3 : #4 } } 3076
3077 } 3077
3078 \fi: 3078
3079 \__cs_generate_variant:w #2 \s__cs_mark {#3} {#4} {#5} 3079
3080 } 3080
3081 \cs_new_protected:Npn \__cs_generate_variant_p_form:nnn #1#2 3081
3082 { \__cs_generate_variant_check:nn { #1 _p : #2 } } 3082
```



```
3083 \cs_new_protected:Npn \__cs_generate_variant_T_form:nnn #1#2 3083
3084 { \__cs_generate_variant_check:nn { #1 : #2 T } } 3084
3085 \cs_new_protected:Npn \__cs_generate_variant_F_form:nnn #1#2 3085
3086 { \__cs_generate_variant_check:nn { #1 : #2 F } } 3086
3087 \cs_new_protected:Npn \__cs_generate_variant_TF_form:nnn #1#2 3087
3088 { \__cs_generate_variant_check:nn { #1 : #2 TF } } 3088
3089 \cs_new_protected:Npn \__cs_generate_variant_check:nn #1#2 3089
3090 { 3090
3091 \cs_if_exist:cTF {#1} 3091
3092 { \cs_generate_variant:cn {#1} {#2} } 3092
3093 { 3093
3094 \msg_error:nne 3094
3095 { kernel } { conditional-base-undefined } 3095
3096 { \token_to_str:c {#1} } 3096
3097 } 3097
3098 } 3098
3099 \cs_new_protected:Npn \exp_args_generate:n #1 3099
3100 { 3100
3101 \exp_args:No \clist_map_inline:nn { \tl_to_str:n {#1} } 3101
3102 { 3102
3103 \str_map_inline:nn {##1} 3103
3104 { 3104
3105 \str_if_in:nnF { NnpcofeVvx } {####1} 3105
3106 { 3106
3107 \msg_error:nnnn { kernel } { invalid-exp-args } 3107
3108 {####1} {##1} 3108
3109 \str_map_break:n { \use_none:nn } 3109
3110 } 3110
3111 } 3111
3112 \__cs_generate_internal_variant:n {##1} 3112
3113 } 3113
3114 } 3114
3115 \cs_set_protected:Npn \__cs_tmp:w #1 3115
3116 { 3116
3117 \group_begin: 3117
3118 \exp_args:No \__cs_generate_internal_variant:n 3118
3119 { \tl_to_str:n {#1} } 3119
3120 \group_end: 3120
3121 } 3121
3122 \__cs_tmp:w { nc } 3122
3123 \__cs_tmp:w { no } 3123
3124 \__cs_tmp:w { nV } 3124
3125 \__cs_tmp:w { nv } 3125
3126 \__cs_tmp:w { ne } 3126
3127 \__cs_tmp:w { nf } 3127
3128 \__cs_tmp:w { oc } 3128
```

3129	_cs_tmp:w { oo }	3129
3130	_cs_tmp:w { of }	3130
3131	_cs_tmp:w { Vo }	3131
3132	_cs_tmp:w { fo }	3132
3133	_cs_tmp:w { ff }	3133
3134	_cs_tmp:w { ee }	3134
3135	_cs_tmp:w { Nx }	3135
3136	_cs_tmp:w { cx }	3136
3137	_cs_tmp:w { nx }	3137
3138	_cs_tmp:w { ox }	3138
3139	_cs_tmp:w { xo }	3139
3140	_cs_tmp:w { xx }	3140
3141	_cs_tmp:w { Ncf }	3141
3142	_cs_tmp:w { Nno }	3142
3143	_cs_tmp:w { NnV }	3143
3144	_cs_tmp:w { Noo }	3144
3145	_cs_tmp:w { NVV }	3145
3146	_cs_tmp:w { cno }	3146
3147	_cs_tmp:w { cnV }	3147
3148	_cs_tmp:w { coo }	3148
3149	_cs_tmp:w { cVV }	3149
3150	_cs_tmp:w { nnc }	3150
3151	_cs_tmp:w { nno }	3151
3152	_cs_tmp:w { nnf }	3152
3153	_cs_tmp:w { nff }	3153
3154	_cs_tmp:w { ooo }	3154
3155	_cs_tmp:w { oof }	3155
3156	_cs_tmp:w { ffo }	3156
3157	_cs_tmp:w { eee }	3157
3158	_cs_tmp:w { NNx }	3158
3159	_cs_tmp:w { Nnx }	3159
3160	_cs_tmp:w { Nox }	3160
3161	_cs_tmp:w { nnx }	3161
3162	_cs_tmp:w { nox }	3162
3163	_cs_tmp:w { ccx }	3163
3164	_cs_tmp:w { cnx }	3164
3165	_cs_tmp:w { oox }	3165
3166	\\cs_generate_variant:Nn \\cs_generate_from_arg_count:NNnn { NNno }	3166
3167	\\cs_generate_variant:Nn \\cs_replacement_spec:N { c }	3167
3168	%% File: l3quark.dtx	3168
3169	\\cs_new_protected:Npn \\quark_new:N #1	3169
3170	{	3170
3171	_kernel_chk_if_free_cs:N #1	3171
3172	\\cs_gset_nopar:Npn #1 {#1}	3172
3173	}	3173
3174	\\quark_new:N \\q_nil	3174

3175	\quark_new:N \q_mark	3175
3176	\quark_new:N \q_no_value	3176
3177	\quark_new:N \q_stop	3177
3178	\quark_new:N \q_recursion_tail	3178
3179	\quark_new:N \q_recursion_stop	3179
3180	\cs_new_eq:NN \s__quark \scan_stop:	3180
3181	\quark_new:N \q__quark_nil	3181
3182	\cs_new:Npn \quark_if_recursion_tail_stop:N #1	3182
3183	{	3183
3184	\if_meaning:w \q_recursion_tail #1	3184
3185	\exp_after:wN \use_none_delimit_by_q_recursion_stop:w	3185
3186	\fi:	3186
3187	}	3187
3188	\cs_new:Npn \quark_if_recursion_tail_stop_do:Nn #1	3188
3189	{	3189
3190	\if_meaning:w \q_recursion_tail #1	3190
3191	\exp_after:wN \use_i_delimit_by_q_recursion_stop:nw	3191
3192	\else:	3192
3193	\exp_after:wN \use_none:n	3193
3194	\fi:	3194
3195	}	3195
3196	\cs_new:Npn \quark_if_recursion_tail_stop:n #1	3196
3197	{	3197
3198	\tl_if_empty:oTF	3198
3199	{ __quark_if_recursion_tail:w {} #1 {} ?! \q_recursion_tail ??? }	3199
3200	{ \use_none_delimit_by_q_recursion_stop:w }	3200
3201	{ }	3201
3202	}	3202
3203	\cs_new:Npn \quark_if_recursion_tail_stop_do:nn #1	3203
3204	{	3204
3205	\tl_if_empty:oTF	3205
3206	{ __quark_if_recursion_tail:w {} #1 {} ?! \q_recursion_tail ??? }	3206
3207	{ \use_i_delimit_by_q_recursion_stop:nw }	3207
3208	{ \use_none:n }	3208
3209	}	3209
3210	\cs_new:Npn __quark_if_recursion_tail:w	3210
3211	#1 \q_recursion_tail #2 ? #3 ?! { #1 #2 }	3211
3212	\cs_generate_variant:Nn \quark_if_recursion_tail_stop:n { o }	3212
3213	\cs_generate_variant:Nn \quark_if_recursion_tail_stop_do:nn { o }	3213
3214	\cs_new:Npn \quark_if_recursion_tail_break:NN #1#2	3214
3215	{	3215
3216	\if_meaning:w \q_recursion_tail #1	3216
3217	\exp_after:wN #2	3217
3218	\fi:	3218
3219	}	3219
3220	\cs_new:Npn \quark_if_recursion_tail_break:nN #1#2	3220

3221	{	3221
3222	\tl_if_empty:oT	3222
3223	{ _quark_if_recursion_tail:w {} #1 {} ?! \q_recursion_tail ??? }	3223
3224	{#2}	3224
3225	}	3225
3226	\prg_new_conditional:Npnn \quark_if_nil:N #1 { p, T , F , TF }	3226
3227	{	3227
3228	\if_meaning:w \q_nil #1	3228
3229	\prg_return_true:	3229
3230	\else:	3230
3231	\prg_return_false:	3231
3232	\fi:	3232
3233	}	3233
3234	\prg_new_conditional:Npnn \quark_if_no_value:N #1 { p, T , F , TF }	3234
3235	{	3235
3236	\if_meaning:w \q_no_value #1	3236
3237	\prg_return_true:	3237
3238	\else:	3238
3239	\prg_return_false:	3239
3240	\fi:	3240
3241	}	3241
3242	\prg_generate_conditional_variant:Nnn \quark_if_no_value:N	3242
3243	{ c } { p , T , F , TF }	3243
3244	\prg_new_conditional:Npnn \quark_if_nil:n #1 { p, T , F , TF }	3244
3245	{	3245
3246	_quark_if_empty_if:o	3246
3247	{ _quark_if_nil:w {} #1 {} ? ! \q_nil ? ? ! }	3247
3248	\prg_return_true:	3248
3249	\else:	3249
3250	\prg_return_false:	3250
3251	\fi:	3251
3252	}	3252
3253	\cs_new:Npn _quark_if_nil:w #1 \q_nil #2 ? #3 ? ! { #1 #2 }	3253
3254	\prg_new_conditional:Npnn \quark_if_no_value:n #1 { p, T , F , TF }	3254
3255	{	3255
3256	_quark_if_empty_if:o	3256
3257	{ _quark_if_no_value:w {} #1 {} ? ! \q_no_value ? ? ! }	3257
3258	\prg_return_true:	3258
3259	\else:	3259
3260	\prg_return_false:	3260
3261	\fi:	3261
3262	}	3262
3263	\cs_new:Npn _quark_if_no_value:w #1 \q_no_value #2 ? #3 ? ! { #1 #2 }	3263
3264	\prg_generate_conditional_variant:Nnn \quark_if_nil:n	3264
3265	{ V , o } { p , TF , T , F }	3265
3266	\cs_new:Npn _quark_if_empty_if:o #1	3266

```
3267 { 3267
3268     \exp_after:wN \if_meaning:w \exp_after:wN \q_nil 3268
3269     \__kernel_tl_to_str:w \exp_after:wN {#1} \q_nil 3269
3270 } 3270
3271 \cs_new_protected:Npn \__kernel_quark_new_test:N #1 3271
3272 { \__quark_new_test_aux:Ne #1 { \__quark_module_name:N #1 } } 3272
3273 \cs_new_protected:Npn \__quark_new_test_aux:Nn #1 #2 3273
3274 { 3274
3275     \if_meaning:w \q_nil #2 \q_nil 3275
3276     \msg_error:nne { quark } { invalid-function } 3276
3277     { \token_to_str:N #1 } 3277
3278 \else: 3278
3279     \__quark_new_test:Nccn #1 3279
3280     { q__#2_recursion_tail } { q__#2_recursion_stop } { __#2 } 3280
3281 \fi: 3281
3282 } 3282
3283 \cs_generate_variant:Nn \__quark_new_test_aux:Nn { Ne } 3283
3284 \cs_new_protected:Npn \__quark_new_test:NNNn #1 3284
3285 { 3285
3286     \exp_last_unbraced:Nf \__quark_new_test_aux:nnNNnnnn 3286
3287     { \cs_split_function:N #1 } 3287
3288     #1 { test } 3288
3289 } 3289
3290 \cs_generate_variant:Nn \__quark_new_test:NNNn { Ncc } 3290
3291 \cs_new_protected:Npn \__kernel_quark_new_conditional:Nn #1 3291
3292 { 3292
3293     \__quark_new_conditional:Neen #1 3293
3294     { \__quark_quark_conditional_name:N #1 } 3294
3295     { \__quark_module_name:N #1 } 3295
3296 } 3296
3297 \cs_new_protected:Npn \__quark_new_conditional:Nnnn #1#2#3#4 3297
3298 { 3298
3299     \if_meaning:w \q_nil #2 \q_nil 3299
3300     \msg_error:nne { quark } { invalid-function } 3300
3301     { \token_to_str:N #1 } 3301
3302 \else: 3302
3303     \if_meaning:w \q_nil #3 \q_nil 3303
3304     \msg_error:nne { quark } { invalid-function } 3304
3305     { \token_to_str:N #1 } 3305
3306 \else: 3306
3307     \exp_last_unbraced:Nf \__quark_new_test_aux:nnNNnnnn 3307
3308     { \cs_split_function:N #1 } 3308
3309     #1 { conditional } 3309
3310     {#2} {#3} {#4} 3310
3311 \fi: 3311
3312 \fi: 3312
```



```
3313 } 3313
3314 \cs_generate_variant:Nn \__quark_new_conditional:Nnnn { Nee } 3314
3315 \cs_new_protected:Npn \__quark_new_test_aux:nnNNnnnn #1 #2 #3 #4 #5 3315
3316 { 3316
3317     \cs_if_exist_use:cTF { \__quark_new_#5_#2:Nnnn } { #4 } 3317
3318     { 3318
3319         \msg_error:nnee { quark } { invalid-function } 3319
3320         { \token_to_str:N #4 } {#2} 3320
3321         \use_none:nnn 3321
3322     } 3322
3323 } 3323
3324 \cs_new_protected:Npn \__quark_new_test_n:Nnnn #1 #2 #3 #4 3324
3325 { 3325
3326     \__quark_new_test_aux_do:nNNnnnnNNn {#4} #2 #3 { none } { } { } { } 3326
3327     \__quark_new_test_define_tl:nNnNNn #1 { } 3327
3328 } 3328
3329 \cs_new_protected:Npn \__quark_new_test_nn:Nnnn #1 #2 #3 #4 3329
3330 { 3330
3331     \__quark_new_test_aux_do:nNNnnnnNNn {#4} #2 #3 { i } { n } {##1} {##2} 3331
3332     \__quark_new_test_define_tl:nNnNNn #1 { \use_none:n } 3332
3333 } 3333
3334 \cs_new_protected:Npn \__quark_new_test_nN:Nnnn #1 #2 #3 #4 3334
3335 { 3335
3336     \__quark_new_test_aux_do:nNNnnnnNNn {#4} #2 #3 { i } { n } {##1} {##2} 3336
3337     \__quark_new_test_define_break_tl:nNNNNn #1 { } 3337
3338 } 3338
3339 \cs_new_protected:Npn \__quark_new_test_N:Nnnn #1 #2 #3 #4 3339
3340 { 3340
3341     \__quark_new_test_aux_do:nNNnnnnNNn {#4} #2 #3 { none } { } { } { } { } 3341
3342     \__quark_new_test_define_ifx:nNnNNn #1 { } 3342
3343 } 3343
3344 \cs_new_protected:Npn \__quark_new_test_Nn:Nnnn #1 #2 #3 #4 3344
3345 { 3345
3346     \__quark_new_test_aux_do:nNNnnnnNNn {#4} #2 #3 { i } { n } {##1} {##2} 3346
3347     \__quark_new_test_define_ifx:nNnNNn #1 3347
3348     { \else: \exp_after:wN \use_none:n } 3348
3349 } 3349
3350 \cs_new_protected:Npn \__quark_new_test_NN:Nnnn #1 #2 #3 #4 3350
3351 { 3351
3352     \__quark_new_test_aux_do:nNNnnnnNNn {#4} #2 #3 { i } { n } {##1} {##2} 3352
3353     \__quark_new_test_define_break_ifx:nNNNNn #1 { } 3353
3354 } 3354
3355 \cs_new_protected:Npn \__quark_new_test_aux_do:nNNnnnnNNn #1 #2 #3 #4 #5 3355
3356 { 3356
3357     \exp_args:Ncc \__quark_test_define_aux:NNNNnnNNn 3357
3358     { #1 _quark_recursion_tail:w } 3358
```

```
3359 { #1 _use_ #4 _delimit_by_q_recursion_stop: #5 w } 3359
3360 #2 #3 3360
3361 } 3361
3362 \cs_new_protected:Npn \__quark_test_define_aux:NNNNnnNNn #1 #2 #3 #4 #5 #6 #7 3362
3363 { 3363
3364 \cs_gset:Npn #1 ##1 #3 ##2 ? ##3 ?! { ##1 ##2 } 3364
3365 \cs_gset:Npn #2 ##1 #6 #4 {#5} 3365
3366 #7 {##1} #1 #2 #3 3366
3367 } 3367
3368 \cs_new_protected:Npn \__quark_new_test_define_tl:nNnNNn #1 #2 #3 #4 #5 #6 3368
3369 { 3369
3370 \cs_new:Npn #5 #1 3370
3371 { 3371
3372 \tl_if_empty:oTF 3372
3373 { #2 {} ##1 {} ?! #4 ??! } 3373
3374 {#3} {#6} 3374
3375 } 3375
3376 } 3376
3377 \cs_new_protected:Npn \__quark_new_test_define_ifx:nNnNNn #1 #2 #3 #4 #5 #6 3377
3378 { 3378
3379 \cs_new:Npn #5 #1 3379
3380 { 3380
3381 \if_meaning:w #4 ##1 3381
3382 \exp_after:wN #3 3382
3383 #6 3383
3384 \fi: 3384
3385 } 3385
3386 } 3386
3387 \cs_new_protected:Npn \__quark_new_test_define_break_tl:nNNNNn #1 #2 #3 3387
3388 { \__quark_new_test_define_tl:nNnNNn {##1##2} #2 {##2} } 3388
3389 \cs_new_protected:Npn \__quark_new_test_define_break_ifx:nNNNNn #1 #2 #3 3389
3390 { \__quark_new_test_define_ifx:nNnNNn {##1##2} #2 {##2} } 3390
3391 \cs_new_protected:Npn \__quark_new_conditional_n:Nnnn #1 #2 #3 3391
3392 { 3392
3393 \exp_args:Ncc \__quark_new_conditional_n_aux:NNNn 3393
3394 { __ #3 _if_quark_ #2 :w } { q__ #3 _ #2 } #1 3394
3395 } 3395
3396 \cs_new_protected:Npn \__quark_new_conditional_N:Nnnn #1 #2 #3 3396
3397 { 3397
3398 \exp_args:NNc \__quark_new_conditional_N_aux:NNNn 3398
3399 \prg_do_nothing: { q__ #3 _ #2 } #1 3399
3400 } 3400
3401 \cs_new_protected:Npn \__quark_new_conditional_n_aux:NNNn #1 #2 #3 #4 3401
3402 { 3402
3403 \cs_gset:Npn #1 ##1 #2 ##2 ? ##3 ?! { ##1##2 } 3403
3404 \prg_new_conditional:Npnn #3 ##1 {#4} 3404
```

```

3405 {
3406     \__quark_if_empty_if:o { #1 {} ##1 {} ?! #2 ??! }
3407     \prg_return_true:
3408 \else:
3409     \prg_return_false:
3410 \fi:
3411 }
3412 }
3413 \cs_new_protected:Npn \__quark_new_conditional_N_aux:NNNn #1 #2 #3 #4
3414 {
3415     \prg_new_conditional:Npnn #3 ##1 {#4}
3416     {
3417         \if_meaning:w #2 ##1
3418         \prg_return_true:
3419 \else:
3420         \prg_return_false:
3421 \fi:
3422     }
3423 }
3424 \cs_set:Npn \__quark_tmp:w #1#2
3425 {
3426     \cs_new:Npn \__quark_module_name:N ##1
3427     {
3428         \exp_last_unbraced:Nf \__quark_module_name:w
3429         { \cs_to_str:N ##1 } #1 \s__quark
3430     }
3431     \cs_new:Npn \__quark_module_name:w ##1 #1 ##2 \s__quark
3432     { \__quark_module_name_loop:w ##1 #2 \use_none:n { } #2 \s__quark }
3433     \cs_new:Npn \__quark_module_name_loop:w ##1 #2
3434     {
3435         \use_i_ii:nnn \if_meaning:w \prg_do_nothing:
3436         ##1 \prg_do_nothing: \prg_do_nothing:
3437         \exp_after:wN \__quark_module_name_loop:w
3438 \else:
3439         \__quark_module_name_end:w ##1
3440 \fi:
3441     }
3442     \cs_new:Npn \__quark_module_name_end:w
3443     ##1 \fi: ##2 \s__quark { \fi: ##1 }
3444 }
3445 \exp_after:wN \__quark_tmp:w \tl_to_str:n { : _ }
3446 \cs_set:Npn \__quark_tmp:w #1 #2 \s__quark
3447 {
3448     \cs_new:Npn \__quark_quark_conditional_name:N ##1
3449     {
3450         \exp_last_unbraced:Nf \__quark_quark_conditional_name:w

```

```

3451         { \cs_to_str:N ##1 } #1 #2 #1 \s__quark 3451
3452     } 3452
3453     \cs_new:Npn \__quark_quark_conditional_name:w 3453
3454         ##1 #2 ##2 #1 ##3 \s__quark {##2} 3454
3455 } 3455
3456 \exp_after:wN \__quark_tmp:w \tl_to_str:n { : _quark_if_ } \s__quark 3456
3457 \cs_new_protected:Npn \scan_new:N #1 3457
3458 { 3458
3459     \tl_if_in:NnTF \g__scan_marks_tl { #1 } 3459
3460     { 3460
3461         \msg_error:nne { scanmark } { already-defined } 3461
3462         { \token_to_str:N #1 } 3462
3463     } 3463
3464     { 3464
3465         \tl_gput_right:Nn \g__scan_marks_tl {#1} 3465
3466         \cs_new_eq:NN #1 \scan_stop: 3466
3467     } 3467
3468 } 3468
3469 \cs_new_eq:NN \s_stop \scan_stop: 3469
3470 \cs_gset_nopar:Npn \g__scan_marks_tl 3470
3471 { 3471
3472     \s_stop 3472
3473     \s__quark 3473
3474     \s__cs_mark 3474
3475     \s__cs_stop 3475
3476 } 3476
3477 \cs_new:Npn \use_none_delimit_by_s_stop:w #1 \s_stop { } 3477
3478 %% File: l3tl.dtx 3478
3479 \cs_new_eq:NN \__kernel_tl_set:Nx \cs_set_nopar:Npe 3479
3480 \cs_new_eq:NN \__kernel_tl_gset:Nx \cs_gset_nopar:Npe 3480
3481 \cs_new_protected:Npn \tl_new:N #1 3481
3482 { 3482
3483     \__kernel_chk_if_free_cs:N #1 3483
3484     \cs_gset_eq:NN #1 \c_empty_tl 3484
3485 } 3485
3486 \cs_generate_variant:Nn \tl_new:N { c } 3486
3487 \cs_new_protected:Npn \tl_const:Nn #1#2 3487
3488 { 3488
3489     \__kernel_chk_if_free_cs:N #1 3489
3490     \cs_gset_nopar:Npe #1 { \__kernel_exp_not:w {#2} } 3490
3491 } 3491
3492 \cs_generate_variant:Nn \tl_const:Nn { NV , Ne , c , cV , ce } 3492
3493 \cs_generate_variant:Nn \tl_const:Nn { Nx , cx } 3493
3494 \cs_new_protected:Npn \tl_clear:N #1 3494
3495     { \tex_let:D #1 = ~ \c_empty_tl } 3495
3496 \cs_new_protected:Npn \tl_gclear:N #1 3496

```

3497	{ \tex_global:D \tex_let:D #1 ~ \c_empty_tl }	3497
3498	\cs_generate_variant:Nn \tl_clear:N { c }	3498
3499	\cs_generate_variant:Nn \tl_gclear:N { c }	3499
3500	\cs_new_protected:Npn \tl_clear_new:N #1	3500
3501	{ \tl_if_exist:NTF #1 { \tl_clear:N #1 } { \tl_new:N #1 } }	3501
3502	\cs_new_protected:Npn \tl_gclear_new:N #1	3502
3503	{ \tl_if_exist:NTF #1 { \tl_gclear:N #1 } { \tl_new:N #1 } }	3503
3504	\cs_generate_variant:Nn \tl_clear_new:N { c }	3504
3505	\cs_generate_variant:Nn \tl_gclear_new:N { c }	3505
3506	\cs_new_protected:Npn \tl_set_eq:NN #1#2	3506
3507	{ \tex_let:D #1 = ~ #2 }	3507
3508	\cs_new_protected:Npn \tl_gset_eq:NN #1#2	3508
3509	{ \tex_global:D \tex_let:D #1 = ~ #2 }	3509
3510	\cs_generate_variant:Nn \tl_set_eq:NN { cN, Nc, cc }	3510
3511	\cs_generate_variant:Nn \tl_gset_eq:NN { cN, Nc, cc }	3511
3512	\cs_new_protected:Npn \tl_concat:NNN #1#2#3	3512
3513	{	3513
3514	__kernel_tl_set:Nx #1	3514
3515	{	3515
3516	__kernel_exp_not:w \exp_after:wN {#2}	3516
3517	__kernel_exp_not:w \exp_after:wN {#3}	3517
3518	}	3518
3519	}	3519
3520	\cs_new_protected:Npn \tl_gconcat:NNN #1#2#3	3520
3521	{	3521
3522	__kernel_tl_gset:Nx #1	3522
3523	{	3523
3524	__kernel_exp_not:w \exp_after:wN {#2}	3524
3525	__kernel_exp_not:w \exp_after:wN {#3}	3525
3526	}	3526
3527	}	3527
3528	\cs_generate_variant:Nn \tl_concat:NNN { ccc }	3528
3529	\cs_generate_variant:Nn \tl_gconcat:NNN { ccc }	3529
3530	\prg_new_eq_conditional:NNn \tl_if_exist:N \cs_if_exist:N { TF , T , F , p }	3530
3531	\prg_new_eq_conditional:NNn \tl_if_exist:c \cs_if_exist:c { TF , T , F , p }	3531
3532	\tl_const:Nn \c_empty_tl { }	3532
3533	\group_begin:	3533
3534	\tex_catcode:D `~ = 11 ~	3534
3535	\tl_const:Ne \c_novalue_tl { - NoValue \token_to_str:N - }	3535
3536	\group_end:	3536
3537	\tl_const:Nn \c_space_tl { ~ }	3537
3538	\cs_new_protected:Npn \tl_set:Nn #1#2	3538
3539	{ __kernel_tl_set:Nx #1 { __kernel_exp_not:w {#2} } }	3539
3540	\cs_new_protected:Npn \tl_set:No #1#2	3540
3541	{ __kernel_tl_set:Nx #1 { __kernel_exp_not:w \exp_after:wN {#2} } }	3541
3542	\cs_new_protected:Npn \tl_gset:Nn #1#2	3542

```
3543 { \__kernel_tl_gset:Nx #1 { \__kernel_exp_not:w {#2} } } 3543
3544 \cs_new_protected:Npn \tl_gset:No #1#2 3544
3545 { \__kernel_tl_gset:Nx #1 { \__kernel_exp_not:w \exp_after:wN {#2} } } 3545
3546 \cs_generate_variant:Nn \tl_set:Nn { NV , Nv , Ne , Nf } 3546
3547 \cs_generate_variant:Nn \tl_set:Nn { c, cV , cv , ce , cf } 3547
3548 \cs_generate_variant:Nn \tl_set:No { c } 3548
3549 \cs_generate_variant:Nn \tl_set:Nn { Nx , cx } 3549
3550 \cs_generate_variant:Nn \tl_gset:Nn { NV , Nv , Ne , Nf } 3550
3551 \cs_generate_variant:Nn \tl_gset:Nn { c, cV , cv , ce , cf } 3551
3552 \cs_generate_variant:Nn \tl_gset:No { c } 3552
3553 \cs_generate_variant:Nn \tl_gset:Nn { Nx , cx } 3553
3554 \cs_new_protected:Npn \tl_put_left:Nn #1#2 3554
3555 { 3555
3556 \__kernel_tl_set:Nx #1 3556
3557 { \__kernel_exp_not:w {#2} \__kernel_exp_not:w \exp_after:wN {#1} } 3557
3558 } 3558
3559 \cs_new_protected:Npn \tl_put_left:NV #1#2 3559
3560 { 3560
3561 \__kernel_tl_set:Nx #1 3561
3562 { \exp_not:V #2 \__kernel_exp_not:w \exp_after:wN {#1} } 3562
3563 } 3563
3564 \cs_new_protected:Npn \tl_put_left:Nv #1#2 3564
3565 { 3565
3566 \__kernel_tl_set:Nx #1 3566
3567 { \exp_not:v {#2} \__kernel_exp_not:w \exp_after:wN {#1} } 3567
3568 } 3568
3569 \cs_new_protected:Npn \tl_put_left:Ne #1#2 3569
3570 { 3570
3571 \__kernel_tl_set:Nx #1 3571
3572 { 3572
3573 \__kernel_exp_not:w \tex_expanded:D { {#2} } 3573
3574 \__kernel_exp_not:w \exp_after:wN {#1} 3574
3575 } 3575
3576 } 3576
3577 \cs_new_protected:Npn \tl_put_left:No #1#2 3577
3578 { 3578
3579 \__kernel_tl_set:Nx #1 3579
3580 { 3580
3581 \__kernel_exp_not:w \exp_after:wN {#2} 3581
3582 \__kernel_exp_not:w \exp_after:wN {#1} 3582
3583 } 3583
3584 } 3584
3585 \cs_new_protected:Npn \tl_gput_left:Nn #1#2 3585
3586 { 3586
3587 \__kernel_tl_gset:Nx #1 3587
3588 { \__kernel_exp_not:w {#2} \__kernel_exp_not:w \exp_after:wN {#1} } 3588
```



```

3589 } 3589
3590 \cs_new_protected:Npn \tl_gput_left:NV #1#2 3590
3591 { 3591
3592     \__kernel_tl_gset:Nx #1 3592
3593     { \exp_not:V #2 \__kernel_exp_not:w \exp_after:wN {#1} } 3593
3594 } 3594
3595 \cs_new_protected:Npn \tl_gput_left:Nv #1#2 3595
3596 { 3596
3597     \__kernel_tl_gset:Nx #1 3597
3598     { \exp_not:v {#2} \__kernel_exp_not:w \exp_after:wN {#1} } 3598
3599 } 3599
3600 \cs_new_protected:Npn \tl_gput_left:Ne #1#2 3600
3601 { 3601
3602     \__kernel_tl_gset:Nx #1 3602
3603     { 3603
3604         \__kernel_exp_not:w \tex_expanded:D { {#2} } 3604
3605         \__kernel_exp_not:w \exp_after:wN {#1} 3605
3606     } 3606
3607 } 3607
3608 \cs_new_protected:Npn \tl_gput_left:No #1#2 3608
3609 { 3609
3610     \__kernel_tl_gset:Nx #1 3610
3611     { 3611
3612         \__kernel_exp_not:w \exp_after:wN {#2} 3612
3613         \__kernel_exp_not:w \exp_after:wN {#1} 3613
3614     } 3614
3615 } 3615
3616 \cs_generate_variant:Nn \tl_put_left:Nn { c } 3616
3617 \cs_generate_variant:Nn \tl_put_left:NV { c } 3617
3618 \cs_generate_variant:Nn \tl_put_left:Nv { c } 3618
3619 \cs_generate_variant:Nn \tl_put_left:Ne { c } 3619
3620 \cs_generate_variant:Nn \tl_put_left:No { c } 3620
3621 \cs_generate_variant:Nn \tl_put_left:Nn { Nx, cx } 3621
3622 \cs_generate_variant:Nn \tl_gput_left:Nn { c } 3622
3623 \cs_generate_variant:Nn \tl_gput_left:NV { c } 3623
3624 \cs_generate_variant:Nn \tl_gput_left:Nv { c } 3624
3625 \cs_generate_variant:Nn \tl_gput_left:Ne { c } 3625
3626 \cs_generate_variant:Nn \tl_gput_left:No { c } 3626
3627 \cs_generate_variant:Nn \tl_gput_left:Nn { Nx , cx } 3627
3628 \cs_new_protected:Npn \tl_put_right:Nn #1#2 3628
3629 { \__kernel_tl_set:Nx #1 { \__kernel_exp_not:w \exp_after:wN { #1 #2 } } } 3629
3630 \cs_new_protected:Npn \tl_put_right:NV #1#2 3630
3631 { 3631
3632     \__kernel_tl_set:Nx #1 3632
3633     { \__kernel_exp_not:w \exp_after:wN {#1} \exp_not:V #2 } 3633
3634 } 3634

```

3635	\cs_new_protected:Npn \tl_put_right:Nv #1#2	3635
3636	{	3636
3637	__kernel_tl_set:Nx #1	3637
3638	{ __kernel_exp_not:w \exp_after:wN {#1} \exp_not:v {#2} }	3638
3639	}	3639
3640	\cs_new_protected:Npn \tl_put_right:Ne #1#2	3640
3641	{	3641
3642	__kernel_tl_set:Nx #1	3642
3643	{	3643
3644	__kernel_exp_not:w \exp_after:wN {#1}	3644
3645	__kernel_exp_not:w \tex_expanded:D { {#2} }	3645
3646	}	3646
3647	}	3647
3648	\cs_new_protected:Npn \tl_put_right:No #1#2	3648
3649	{	3649
3650	__kernel_tl_set:Nx #1	3650
3651	{	3651
3652	__kernel_exp_not:w \exp_after:wN {#1}	3652
3653	__kernel_exp_not:w \exp_after:wN {#2}	3653
3654	}	3654
3655	}	3655
3656	\cs_new_protected:Npn \tl_gput_right:Nn #1#2	3656
3657	{ __kernel_tl_gset:Nx #1 { __kernel_exp_not:w \exp_after:wN { #1 #2 } } }	3657
3658	\cs_new_protected:Npn \tl_gput_right:NV #1#2	3658
3659	{	3659
3660	__kernel_tl_gset:Nx #1	3660
3661	{ __kernel_exp_not:w \exp_after:wN {#1} \exp_not:V #2 }	3661
3662	}	3662
3663	\cs_new_protected:Npn \tl_gput_right:Nv #1#2	3663
3664	{	3664
3665	__kernel_tl_gset:Nx #1	3665
3666	{ __kernel_exp_not:w \exp_after:wN {#1} \exp_not:v {#2} }	3666
3667	}	3667
3668	\cs_new_protected:Npn \tl_gput_right:Ne #1#2	3668
3669	{	3669
3670	__kernel_tl_gset:Nx #1	3670
3671	{	3671
3672	__kernel_exp_not:w \exp_after:wN {#1}	3672
3673	__kernel_exp_not:w \tex_expanded:D { {#2} }	3673
3674	}	3674
3675	}	3675
3676	\cs_new_protected:Npn \tl_gput_right:No #1#2	3676
3677	{	3677
3678	__kernel_tl_gset:Nx #1	3678
3679	{	3679
3680	__kernel_exp_not:w \exp_after:wN {#1}	3680

```

3681 \__kernel_exp_not:w \exp_after:wN {#2}
3682 }
3683 }
3684 \cs_generate_variant:Nn \tl_put_right:Nn { c }
3685 \cs_generate_variant:Nn \tl_put_right:NV { c }
3686 \cs_generate_variant:Nn \tl_put_right:Nv { c }
3687 \cs_generate_variant:Nn \tl_put_right:Ne { c }
3688 \cs_generate_variant:Nn \tl_put_right:No { c }
3689 \cs_generate_variant:Nn \tl_put_right:Nn { Nx , cx }
3690 \cs_generate_variant:Nn \tl_gput_right:Nn { c }
3691 \cs_generate_variant:Nn \tl_gput_right:NV { c }
3692 \cs_generate_variant:Nn \tl_gput_right:Nv { c }
3693 \cs_generate_variant:Nn \tl_gput_right:Ne { c }
3694 \cs_generate_variant:Nn \tl_gput_right:No { c }
3695 \cs_generate_variant:Nn \tl_gput_right:Nn { Nx, cx }
3696 \quark_new:N \q__tl_nil
3697 \quark_new:N \q__tl_mark
3698 \quark_new:N \q__tl_stop
3699 \quark_new:N \q__tl_recursion_tail
3700 \quark_new:N \q__tl_recursion_stop
3701 \__kernel_quark_new_test:N \__tl_if_recursion_tail_break:nN
3702 \__kernel_quark_new_conditional:Nn \__tl_quark_if_nil:n { TF }
3703 \tl_const:Ne \c__tl_rescan_marker_tl { : \token_to_str:N : }
3704 \cs_new_protected:Npn \tl_rescan:nn #1#2
3705 {
3706     \tl_set_rescan:Nnn \l__tl_internal_a_tl {#1} {#2}
3707     \exp_after:wN \__tl_rescan_aux:
3708     \l__tl_internal_a_tl
3709 }
3710 \cs_generate_variant:Nn \tl_rescan:nn { nV }
3711 \exp_args:NNo \cs_new_protected:Npn \__tl_rescan_aux:
3712 { \tl_clear:N \l__tl_internal_a_tl }
3713 \cs_new_protected:Npn \tl_set_rescan:Nnn
3714 { \__tl_set_rescan:NNnn \tl_set:No }
3715 \cs_new_protected:Npn \tl_gset_rescan:Nnn
3716 { \__tl_set_rescan:NNnn \tl_gset:No }
3717 \cs_new_protected:Npn \__tl_set_rescan:NNnn #1#2#3#4
3718 {
3719     \group_begin:
3720     \if_false: { \fi:
3721     \int_set_eq:NN \tex_tracingnesting:D \c_zero_int
3722     \int_compare:nNnT \tex_endlinechar:D = { 32 }
3723     { \int_set:Nn \tex_endlinechar:D { -1 } }
3724     \int_set_eq:NN \tex_newlinechar:D \tex_endlinechar:D
3725     #3 \scan_stop:
3726     \exp_args:No \__tl_set_rescan:nNN { \tl_to_str:n {#4} } #1 #2

```

```

3727 \if_false: } \fi:
3728 }
3729 \cs_new_protected:Npn \__tl_set_rescan_multi:nNN #1#2#3
3730 {
3731   \tex_veryeof:D \exp_after:wN { \c__tl_rescan_marker_tl }
3732   \exp_after:wN \__tl_rescan:NNw
3733   \exp_after:wN #2
3734   \exp_after:wN #3
3735   \exp_after:wN \prg_do_nothing:
3736   \tex_scantokens:D {#1}
3737 }
3738 \exp_args:Nno \use:nn
3739 { \cs_new:Npn \__tl_rescan:NNw #1#2#3 } \c__tl_rescan_marker_tl
3740 {
3741   \group_end:
3742   #1 #2 {#3}
3743 }
3744 \cs_generate_variant:Nn \tl_set_rescan:Nnn { NnV , Nne , c , cnV , cne }
3745 \cs_generate_variant:Nn \tl_set_rescan:Nnn { Nno , Nnx , cno , cnx }
3746 \cs_generate_variant:Nn \tl_gset_rescan:Nnn { NnV , Nne , c , cnV , cne }
3747 \cs_generate_variant:Nn \tl_gset_rescan:Nnn { Nno , Nnx , cno , cnx }
3748 \cs_new_protected:Npn \__tl_set_rescan:nNN #1
3749 {
3750   \int_compare:nNnTF \tex_newlinechar:D < 0
3751   { \use_ii:nn }
3752   {
3753     \exp_args:Nnf \tl_if_in:nnTF {#1}
3754     { \char_generate:nn { \tex_newlinechar:D } { 12 } }
3755   }
3756   { \__tl_set_rescan_multi:nNN }
3757   {
3758     \int_set:Nn \tex_endlinechar:D { -1 }
3759     \__tl_set_rescan_single:nNN { `` }
3760   }
3761   {#1}
3762 }
3763 \cs_new_protected:Npn \__tl_set_rescan_single:nNN #1
3764 {
3765   \int_compare:nNnTF
3766   { \char_value_catcode:n {#1} / 2 } = 6
3767   {
3768     \exp_args:Nof \__tl_set_rescan_single_aux:nnnNN
3769     \c__tl_rescan_marker_tl
3770     { \char_generate:nn {#1} { \char_value_catcode:n {#1} } }
3771   }
3772   {

```

```

3773 \int_compare:nNnTF {#1} < { `~ } 3773
3774 { 3774
3775 \exp_args:Nf \__tl_set_rescan_single:nnNN 3775
3776 { \int_eval:n { #1 + 1 } } 3776
3777 } 3777
3778 { \__tl_set_rescan_multi:nnN } 3778
3779 } 3779
3780 } 3780
3781 \cs_new_protected:Npn \__tl_set_rescan_single_aux:nnnNN #1#2#3#4#5 3781
3782 { 3782
3783 \tex_everyeof:D 3783
3784 { 3784
3785 #1 \use_none:n 3785
3786 #2 #1 { \exp:w \__tl_set_rescan_single_aux:w } 3786
3787 \s__tl_stop 3787
3788 } 3788
3789 \cs_set:Npn \__tl_rescan:NNw ##1##2##3 #2 #1 ##4 ##5 \s__tl_stop 3789
3790 { 3790
3791 \group_end: 3791
3792 ##1 ##2 { ##4 ##3 } 3792
3793 } 3793
3794 \exp_after:wN \__tl_rescan:NNw 3794
3795 \exp_after:wN #4 3795
3796 \exp_after:wN #5 3796
3797 \tex_scantokens:D { #2 #3 #2 } 3797
3798 } 3798
3799 \exp_args:Nno \use:nn 3799
3800 { \cs_new:Npn \__tl_set_rescan_single_aux:w #1 } 3800
3801 \c__tl_rescan_marker_tl #2 3801
3802 { \use_i:nn \exp_end: #1 } 3802
3803 \cs_new_protected:Npn \tl_replace_once:Nnn 3803
3804 { \__tl_replace:NnNNNnn \q__tl_mark ? \__tl_replace_wrap:w \__kernel_tl_set:Nx } 3804
3805 \cs_new_protected:Npn \tl_greplace_once:Nnn 3805
3806 { \__tl_replace:NnNNNnn \q__tl_mark ? \__tl_replace_wrap:w \__kernel_tl_gset:Nx } 3806
3807 \cs_new_protected:Npn \tl_replace_all:Nnn 3807
3808 { \__tl_replace:NnNNNnn \q__tl_mark ? \__tl_replace_next:w \__kernel_tl_set:Nx } 3808
3809 \cs_new_protected:Npn \tl_greplace_all:Nnn 3809
3810 { \__tl_replace:NnNNNnn \q__tl_mark ? \__tl_replace_next:w \__kernel_tl_gset:Nx } 3810
3811 \cs_generate_variant:Nn \tl_replace_once:Nnn 3811
3812 { NnV , Nne , NV , Ne , Nee , c , cnV , cne , cV , ce , cee } 3812
3813 \cs_generate_variant:Nn \tl_replace_once:Nnn 3813
3814 { Nx , Nnx , Nxx , cxn , cnx , cxx } 3814
3815 \cs_generate_variant:Nn \tl_greplace_once:Nnn 3815
3816 { NnV , Nne , NV , Ne , Nee , c , cnV , cne , cV , ce , cee } 3816
3817 \cs_generate_variant:Nn \tl_greplace_once:Nnn 3817
3818 { Nx , Nnx , Nxx , cxn , cnx , cxx } 3818

```

```

3819 \cs_generate_variant:Nn \tl_replace_all:Nnn
3820 { NnV , Nne , NV , Ne , Nee , c , cnV , cne , cV , ce , cee }
3821 \cs_generate_variant:Nn \tl_replace_all:Nnn
3822 { Nx , Nnx , Nxx , cxn , cnx , cxx }
3823 \cs_generate_variant:Nn \tl_greplace_all:Nnn
3824 { NnV , Nne , NV , Ne , Nee , c , cnV , cne , cV , ce , cee }
3825 \cs_generate_variant:Nn \tl_greplace_all:Nnn
3826 { Nx , Nnx , Nxx , cxn , cnx , cxx }
3827 \cs_new_protected:Npn \__tl_replace:NnNNNnn #1#2#3#4#5#6#7
3828 {
3829     \tl_if_empty:nTF {#6}
3830     {
3831         \msg_error:nne { kernel } { empty-search-pattern }
3832         { \tl_to_str:n {#7} }
3833     }
3834     {
3835         \tl_if_in:onTF { #5 #6 } {#1}
3836         {
3837             \tl_if_in:nnTF {#6} {#1}
3838             { \exp_args:Nc \__tl_replace:NnNNNnn {#2} {#2?} }
3839             {
3840                 \__tl_quark_if_nil:nTF {#6}
3841                 { \__tl_replace_auxi:NnnNNNnn #5 {#1} { #1 \q__tl_stop } }
3842                 { \__tl_replace_auxi:NnnNNNnn #5 {#1} { #1 \q__tl_nil } }
3843             }
3844         }
3845         { \__tl_replace_auxii:nNNNnn {#1} }
3846         #3#4#5 {#6} {#7}
3847     }
3848 }
3849 \cs_new_protected:Npn \__tl_replace_auxi:NnnNNNnn #1#2#3
3850 {
3851     \tl_if_in:NnTF #1 { #2 #3 #3 }
3852     { \__tl_replace_auxi:NnnNNNnn #1 { #2 #3 } {#2} }
3853     { \__tl_replace_auxii:nNNNnn { #2 #3 #3 } }
3854 }
3855 \cs_new_protected:Npn \__tl_replace_auxii:nNNNnn #1#2#3#4#5#6
3856 {
3857     \group_align_safe_begin:
3858     \cs_set:Npn \__tl_replace_wrap:w ##1 #1 ##2
3859     { \__kernel_exp_not:w \exp_after:wN { \use_none:nn ##1 } ##2 }
3860     \cs_set:Npe \__tl_replace_next:w ##1 #5
3861     {
3862         \exp_not:N \__tl_replace_wrap:w ##1
3863         \exp_not:n { #1 }
3864         \exp_not:n { \exp_not:n {#6} }

```



```
3865 \exp_not:n { #2 { } { } } 3865
3866 } 3866
3867 #3 #4 3867
3868 { 3868
3869 \exp_after:wN \__tl_replace_next_aux:w 3869
3870 #4 3870
3871 #1 3871
3872 { 3872
3873 \if_false: { \fi: } 3873
3874 \exp_after:wN \use_none:n \exp_after:wN { \if_false: } \fi: 3874
3875 } 3875
3876 #5 3876
3877 } 3877
3878 \group_align_safe_end: 3878
3879 } 3879
3880 \cs_new:Npn \__tl_replace_next_aux:w { \__tl_replace_next:w { } { } } 3880
3881 \cs_new_eq:NN \__tl_replace_wrap:w ? 3881
3882 \cs_new_eq:NN \__tl_replace_next:w ? 3882
3883 \cs_new_protected:Npn \tl_regex_replace_once:Nnn #1#2#3 3883
3884 { \regex_replace_once:nnN {#2} {#3} #1 } 3884
3885 \cs_generate_variant:Nn \tl_regex_replace_once:Nnn { c } 3885
3886 \cs_new_protected:Npn \tl_regex_replace_once:NNn #1#2#3 3886
3887 { \regex_replace_once:NnN #2 {#3} #1 } 3887
3888 \cs_generate_variant:Nn \tl_regex_replace_once:NNn { c } 3888
3889 \cs_new_protected:Npn \tl_regex_replace_all:Nnn #1#2#3 3889
3890 { \regex_replace_all:nnN {#2} {#3} #1 } 3890
3891 \cs_generate_variant:Nn \tl_regex_replace_all:Nnn { c } 3891
3892 \cs_new_protected:Npn \tl_regex_replace_all:NNn #1#2#3 3892
3893 { \regex_replace_all:NnN #2 {#3} #1 } 3893
3894 \cs_generate_variant:Nn \tl_regex_replace_all:NNn { c } 3894
3895 \group_begin: 3895
3896 \cs_set_protected:Npn \__tl_tmp:w #1#2#3 3896
3897 { 3897
3898 \cs_new_protected:cpe { tl_regex_greplac_ #1 :N #2 n } ##1##2##3 3898
3899 { 3899
3900 \group_begin: 3900
3901 \tl_set_eq:NN \exp_not:N \l__tl_internal_a_tl ##1 3901
3902 \exp_not:c { regex_replace_ #1 :Nn #2 } 3902
3903 #3 {##2} {##3} \exp_not:N \l__tl_internal_a_tl 3903
3904 \tl_gset_eq:NN ##1 \exp_not:N \l__tl_internal_a_tl 3904
3905 \group_end: 3905
3906 } 3906
3907 \cs_generate_variant:cn { tl_regex_greplac_ #1 :N #2 n } { c } 3907
3908 } 3908
3909 \__tl_tmp:w { once } n { } 3909
3910 \__tl_tmp:w { once } N \use:n 3910
```

3911	_tl_tmp:w { all } n { }	3911
3912	_tl_tmp:w { all } N \use:n	3912
3913	\group_end:	3913
3914	\cs_new_protected:Npn \tl_remove_once:Nn #1#2	3914
3915	{ \tl_replace_once:Nnn #1 {#2} { } }	3915
3916	\cs_new_protected:Npn \tl_gremove_once:Nn #1#2	3916
3917	{ \tl_greplace_once:Nnn #1 {#2} { } }	3917
3918	\cs_generate_variant:Nn \tl_remove_once:Nn { NV , Ne , c , cV , ce }	3918
3919	\cs_generate_variant:Nn \tl_gremove_once:Nn { NV , Ne , c , cV , ce }	3919
3920	\cs_new_protected:Npn \tl_remove_all:Nn #1#2	3920
3921	{ \tl_replace_all:Nnn #1 {#2} { } }	3921
3922	\cs_new_protected:Npn \tl_gremove_all:Nn #1#2	3922
3923	{ \tl_greplace_all:Nnn #1 {#2} { } }	3923
3924	\cs_generate_variant:Nn \tl_remove_all:Nn { NV , Ne , c , cV , ce }	3924
3925	\cs_generate_variant:Nn \tl_remove_all:Nn { Nx , cx }	3925
3926	\cs_generate_variant:Nn \tl_gremove_all:Nn { NV , Ne , c , cV , ce }	3926
3927	\cs_generate_variant:Nn \tl_gremove_all:Nn { Nx , cx }	3927
3928	\prg_new_conditional:Npnn \tl_if_empty:N #1 { p , T , F , TF }	3928
3929	{	3929
3930	\if_meaning:w #1 \c_empty_tl	3930
3931	\prg_return_true:	3931
3932	\else:	3932
3933	\prg_return_false:	3933
3934	\fi:	3934
3935	}	3935
3936	\prg_generate_conditional_variant:Nnn \tl_if_empty:N	3936
3937	{ c } { p , T , F , TF }	3937
3938	\prg_new_conditional:Npnn \tl_if_empty:n #1 { p , TF , T , F }	3938
3939	{	3939
3940	\if:w \scan_stop: \tl_to_str:n {#1} \scan_stop:	3940
3941	\prg_return_true:	3941
3942	\else:	3942
3943	\prg_return_false:	3943
3944	\fi:	3944
3945	}	3945
3946	\prg_generate_conditional_variant:Nnn \tl_if_empty:n	3946
3947	{ V , e } { p , TF , T , F }	3947
3948	\cs_new:Npn __tl_if_empty_if:o #1	3948
3949	{	3949
3950	\if:w \scan_stop: __kernel_tl_to_str:w \exp_after:wN {#1} \scan_stop:	3950
3951	}	3951
3952	\exp_args:Nno \use:n	3952
3953	{ \prg_new_conditional:Npnn \tl_if_empty:o #1 { p , TF , T , F } }	3953
3954	{	3954
3955	__tl_if_empty_if:o {#1}	3955
3956	\prg_return_true:	3956

```

3957 \else:
3958 \prg_return_false:
3959 \fi:
3960 }
3961 \exp_args:Nno \use:n
3962 { \prg_new_conditional:Npnn \tl_if_blank:n #1 { p , T , F , TF } }
3963 {
3964 \__tl_if_empty_if:o { \use_none:n #1 ? }
3965 \prg_return_true:
3966 \else:
3967 \prg_return_false:
3968 \fi:
3969 }
3970 \prg_generate_conditional_variant:Nnn \tl_if_blank:n
3971 { e , V , o } { p , T , F , TF }
3972 \prg_new_eq_conditional:NNn \tl_if_eq:NN \cs_if_eq:NN { p , T , F , TF }
3973 \prg_generate_conditional_variant:Nnn \tl_if_eq:NN
3974 { Nc , c , cc } { p , TF , T , F }
3975 \tl_new:N \l__tl_internal_a_tl
3976 \tl_new:N \l__tl_internal_b_tl
3977 \prg_new_protected_conditional:Npnn \tl_if_eq:Nn #1#2 { T , F , TF }
3978 {
3979 \group_begin:
3980 \tl_set:Nn \l__tl_internal_b_tl {#2}
3981 \exp_after:wN
3982 \group_end:
3983 \if_meaning:w #1 \l__tl_internal_b_tl
3984 \prg_return_true:
3985 \else:
3986 \prg_return_false:
3987 \fi:
3988 }
3989 \prg_generate_conditional_variant:Nnn \tl_if_eq:Nn { c } { TF , T , F }
3990 \prg_new_protected_conditional:Npnn \tl_if_eq:nn #1#2 { T , F , TF }
3991 {
3992 \group_begin:
3993 \tl_set:Nn \l__tl_internal_a_tl {#1}
3994 \tl_set:Nn \l__tl_internal_b_tl {#2}
3995 \exp_after:wN
3996 \group_end:
3997 \if_meaning:w \l__tl_internal_a_tl \l__tl_internal_b_tl
3998 \prg_return_true:
3999 \else:
4000 \prg_return_false:
4001 \fi:
4002 }

```

```

4003 \prg_generate_conditional_variant:Nnn \tl_if_eq:nn
4004 { nV , ne , nx , V, e , ee , x , xx }
4005 { TF , T , F }
4006 \cs_new_protected:Npn \tl_if_in:NnT { \exp_args:No \tl_if_in:nnT }
4007 \cs_new_protected:Npn \tl_if_in:NnF { \exp_args:No \tl_if_in:nnF }
4008 \cs_new_protected:Npn \tl_if_in:NnTF { \exp_args:No \tl_if_in:nnTF }
4009 \prg_generate_conditional_variant:Nnn \tl_if_in:Nn
4010 { NV , No , c , cV , co } { T , F , TF }
4011 \prg_new_protected_conditional:Npnn \tl_if_in:nn #1#2 { T , F , TF }
4012 {
4013   \scan_stop:
4014   \if_false: { \fi:
4015     \cs_set:Npn \__tl_tmp:w ##1 #2 { }
4016     \tl_if_empty:oTF { \__tl_tmp:w #1 {} {} #2 }
4017     { \prg_return_false: } { \prg_return_true: }
4018     \if_false: } \fi:
4019   }
4020 \prg_generate_conditional_variant:Nnn \tl_if_in:nn
4021 { V , VV , o , oo , nV , no } { T , F , TF }
4022 \cs_set_protected:Npn \__tl_tmp:w #1
4023 {
4024   \prg_new_conditional:Npnn \tl_if_novalue:n ##1
4025   { p , T , F , TF }
4026   {
4027     \__tl_if_empty_if:o { \__tl_if_novalue:w {} ##1 {} ? ! #1 ? ? ! }
4028     \prg_return_true:
4029     \else:
4030     \prg_return_false:
4031     \fi:
4032   }
4033   \cs_new:Npn \__tl_if_novalue:w ##1 #1 ##2 ? ##3 ? ! { ##1 ##2 }
4034 }
4035 \exp_args:No \__tl_tmp:w { \c_novalue_tl }
4036 \cs_new:Npn \tl_if_single_p:N { \exp_args:No \tl_if_single_p:n }
4037 \cs_new:Npn \tl_if_single:NT { \exp_args:No \tl_if_single:nT }
4038 \cs_new:Npn \tl_if_single:NF { \exp_args:No \tl_if_single:nF }
4039 \cs_new:Npn \tl_if_single:NTF { \exp_args:No \tl_if_single:nTF }
4040 \prg_generate_conditional_variant:Nnn \tl_if_single:N {c} { p , T , F , TF }
4041 \prg_new_conditional:Npnn \tl_if_single:n #1 { p , T , F , TF }
4042 {
4043   \if:w \scan_stop: \exp_after:wN \__tl_if_single:nnw
4044     \__kernel_tl_to_str:w
4045     \exp_after:wN { \use_none:nn #1 ?? } \scan_stop: ? \s__tl_stop
4046     \prg_return_true:
4047   \else:
4048     \prg_return_false:

```

```
4049 \fi:
4050 }
4051 \cs_new:Npn \__tl_if_single:nnw #1#2#3 \s_tl_stop {#2}
4052 \prg_new_conditional:Npnn \tl_if_single_token:n #1 { p , T , F , TF }
4053 {
4054   \tl_if_head_is_N_type:nTF {#1}
4055   { \__tl_if_empty_if:o { \use_none:n #1 } }
4056   {
4057     \tl_if_empty:nTF {#1}
4058     { \if_false: }
4059     { \__tl_if_empty_if:o { \exp:w \exp_end_continue_f:w #1 } }
4060   }
4061   \prg_return_true:
4062 \else:
4063   \prg_return_false:
4064 \fi:
4065 }
4066 \prg_new_protected_conditional:Npnn \tl_if_regex_match:nn #1#2 { TF , T , F }
4067 {
4068   \regex_match:nnTF {#2} {#1}
4069   \prg_return_true: \prg_return_false:
4070 }
4071 \prg_generate_conditional_variant:Nnn \tl_if_regex_match:nn
4072 { V } { TF , T , F }
4073 \prg_new_protected_conditional:Npnn \tl_if_regex_match:nN #1#2 { TF , T , F }
4074 {
4075   \regex_match:nNTF {#2} #1
4076   \prg_return_true: \prg_return_false:
4077 }
4078 \prg_generate_conditional_variant:Nnn \tl_if_regex_match:nN
4079 { V } { TF , T , F }
4080 \cs_new:Npn \tl_map_function:nN #1#2
4081 {
4082   \__tl_map_function:Nnnnnnnnn #2 #1
4083   \s_tl_stop \s_tl_stop \s_tl_stop \s_tl_stop
4084   \s_tl_stop \s_tl_stop \s_tl_stop \s_tl_stop
4085   \prg_break_point:Nn \tl_map_break: { }
4086 }
4087 \cs_generate_variant:Nn \tl_map_function:nN { e }
4088 \cs_new:Npn \tl_map_function:NN
4089 { \exp_args:No \tl_map_function:nN }
4090 \cs_generate_variant:Nn \tl_map_function:NN { c }
4091 \cs_new:Npn \__tl_map_function:Nnnnnnnnn #1#2#3#4#5#6#7#8#9
4092 {
4093   \__tl_use_none_delimit_by_s_stop:w
4094   #9 \__tl_map_function_end:w \s_tl_stop
```

```
4095 #1 {#2} #1 {#3} #1 {#4} #1 {#5} #1 {#6} #1 {#7} #1 {#8} #1 {#9} 4095
4096 \__tl_map_function:Nnnnnnnnn #1 4096
4097 } 4097
4098 \cs_new:Npn \__tl_map_function_end:w \s__tl_stop #1#2 4098
4099 { 4099
4100 \__tl_use_none_delimit_by_s_stop:w #2 \tl_map_break: \s__tl_stop 4100
4101 #1 {#2} 4101
4102 \__tl_map_function_end:w \s__tl_stop 4102
4103 } 4103
4104 \cs_new:Npn \__tl_use_none_delimit_by_s_stop:w #1 \s__tl_stop { } 4104
4105 \cs_new_protected:Npn \tl_map_inline:nn #1#2 4105
4106 { 4106
4107 \int_gincr:N \g__kernel_prg_map_int 4107
4108 \cs_gset_protected:cpn 4108
4109 { __tl_map_ \int_use:N \g__kernel_prg_map_int :w } ##1 {#2} 4109
4110 \exp_args:Nc \__tl_map_function:Nnnnnnnnn 4110
4111 { __tl_map_ \int_use:N \g__kernel_prg_map_int :w } 4111
4112 #1 4112
4113 \s__tl_stop \s__tl_stop \s__tl_stop \s__tl_stop 4113
4114 \s__tl_stop \s__tl_stop \s__tl_stop \s__tl_stop 4114
4115 \prg_break_point:Nn \tl_map_break: 4115
4116 { \int_gdecr:N \g__kernel_prg_map_int } 4116
4117 } 4117
4118 \cs_new_protected:Npn \tl_map_inline:Nn 4118
4119 { \exp_args:No \tl_map_inline:nn } 4119
4120 \cs_generate_variant:Nn \tl_map_inline:Nn { c } 4120
4121 \cs_new:Npn \tl_map_tokens:nn #1#2 4121
4122 { 4122
4123 \__tl_map_tokens:nnnnnnnn {#2} #1 4123
4124 \s__tl_stop \s__tl_stop \s__tl_stop \s__tl_stop 4124
4125 \s__tl_stop \s__tl_stop \s__tl_stop \s__tl_stop 4125
4126 \prg_break_point:Nn \tl_map_break: { } 4126
4127 } 4127
4128 \cs_new:Npn \tl_map_tokens:Nn 4128
4129 { \exp_args:No \tl_map_tokens:nn } 4129
4130 \cs_generate_variant:Nn \tl_map_tokens:Nn { c } 4130
4131 \cs_new:Npn \__tl_map_tokens:nnnnnnnn #1#2#3#4#5#6#7#8#9 4131
4132 { 4132
4133 \__tl_use_none_delimit_by_s_stop:w 4133
4134 #9 \__tl_map_tokens_end:w \s__tl_stop 4134
4135 \use:n {#1} {#2} \use:n {#1} {#3} \use:n {#1} {#4} \use:n {#1} {#5} 4135
4136 \use:n {#1} {#6} \use:n {#1} {#7} \use:n {#1} {#8} \use:n {#1} {#9} 4136
4137 \__tl_map_tokens:nnnnnnnn {#1} 4137
4138 } 4138
4139 \cs_new:Npn \__tl_map_tokens_end:w \s__tl_stop \use:n #1#2 4139
4140 { 4140
```



```
4141 \tl_use_none_delimit_by_s_stop:w #2 \tl_map_break: \s_tl_stop 4141
4142 #1 {#2} 4142
4143 \tl_map_tokens_end:w \s_tl_stop 4143
4144 } 4144
4145 \cs_new_protected:Npn \tl_map_variable:nNn #1#2#3 4145
4146 { \tl_map_tokens:nn {#1} { \tl_map_variable:Nnn #2 {#3} } } 4146
4147 \cs_new_protected:Npn \tl_map_variable:Nnn #1#2#3 4147
4148 { \tl_set:Nn #1 {#3} #2 } 4148
4149 \cs_new_protected:Npn \tl_map_variable:NNn 4149
4150 { \exp_args:No \tl_map_variable:nNn } 4150
4151 \cs_generate_variant:Nn \tl_map_variable:NNn { c } 4151
4152 \cs_new:Npn \tl_map_break: 4152
4153 { \prg_map_break:Nn \tl_map_break: { } } 4153
4154 \cs_new:Npn \tl_map_break:n 4154
4155 { \prg_map_break:Nn \tl_map_break: } 4155
4156 \cs_generate_variant:Nn \tl_to_str:n { o , V , v , e } 4156
4157 \cs_new:Npn \tl_to_str:N #1 { \__kernel_tl_to_str:w \exp_after:wN {#1} } 4157
4158 \cs_generate_variant:Nn \tl_to_str:N { c } 4158
4159 \cs_new:Npn \tl_use:N #1 4159
4160 { 4160
4161 \tl_if_exist:NTF #1 {#1} 4161
4162 { 4162
4163 \msg_expandable_error:nnn 4163
4164 { kernel } { bad-variable } {#1} 4164
4165 } 4165
4166 } 4166
4167 \cs_generate_variant:Nn \tl_use:N { c } 4167
4168 \cs_new:Npn \tl_count:n #1 4168
4169 { 4169
4170 \int_eval:n 4170
4171 { 0 \tl_map_function:nN {#1} \tl_count:n } 4171
4172 } 4172
4173 \cs_new:Npn \tl_count:N #1 4173
4174 { 4174
4175 \int_eval:n 4175
4176 { 0 \tl_map_function:NN #1 \tl_count:n } 4176
4177 } 4177
4178 \cs_new:Npn \tl_count:n #1 { + 1 } 4178
4179 \cs_generate_variant:Nn \tl_count:n { V , v , e , o } 4179
4180 \cs_generate_variant:Nn \tl_count:N { c } 4180
4181 \cs_new:Npn \tl_count_tokens:n #1 4181
4182 { 4182
4183 \int_eval:n 4183
4184 { 4184
4185 \tl_act:NNNn 4185
4186 \tl_act_count_normal:N 4186
```

```
4187         \__tl_act_count_group:n 4187
4188         \__tl_act_count_space: 4188
4189         {#1} 4189
4190     } 4190
4191 } 4191
4192 \cs_new:Npn \__tl_act_count_normal:N #1 { 1 + } 4192
4193 \cs_new:Npn \__tl_act_count_space: { 1 + } 4193
4194 \cs_new:Npn \__tl_act_count_group:n #1 { 2 + \tl_count_tokens:n {#1} + } 4194
4195 \cs_new:Npn \tl_reverse_items:n #1 4195
4196 { 4196
4197     \__tl_reverse_items:nwNwn #1 ? 4197
4198     \s__tl_mark \__tl_reverse_items:nwNwn 4198
4199     \s__tl_mark \__tl_reverse_items:wn 4199
4200     \s__tl_stop { } 4200
4201 } 4201
4202 \cs_new:Npn \__tl_reverse_items:nwNwn #1 #2 \s__tl_mark #3 #4 \s__tl_stop #5 4202
4203 { 4203
4204     #3 #2 4204
4205     \s__tl_mark \__tl_reverse_items:nwNwn 4205
4206     \s__tl_mark \__tl_reverse_items:wn 4206
4207     \s__tl_stop { {#1} #5 } 4207
4208 } 4208
4209 \cs_new:Npn \__tl_reverse_items:wn #1 \s__tl_stop #2 4209
4210 { \__kernel_exp_not:w \exp_after:wN { \use_none:nn #2 } } 4210
4211 \cs_new:Npn \tl_trim_spaces:n 4211
4212 { \__tl_trim_spaces:nn { \__kernel_exp_not:w \exp_after:wN } } 4212
4213 \cs_new:Npn \tl_trim_left_spaces:n 4213
4214 { \__tl_trim_left_spaces:nn { \__kernel_exp_not:w \exp_after:wN } } 4214
4215 \cs_new:Npn \tl_trim_right_spaces:n 4215
4216 { \__tl_trim_right_spaces:nn { \__kernel_exp_not:w \exp_after:wN } } 4216
4217 \cs_generate_variant:Nn \tl_trim_spaces:n { V , v , e , o } 4217
4218 \cs_generate_variant:Nn \tl_trim_left_spaces:n { V , v , e , o } 4218
4219 \cs_generate_variant:Nn \tl_trim_right_spaces:n { V , v , e , o } 4219
4220 \cs_new:Npn \tl_trim_spaces_apply:nN #1#2 4220
4221 { \__tl_trim_spaces:nn { \exp_args:No #2 } { #1 } } 4221
4222 \cs_new:Npn \tl_trim_left_spaces_apply:nN #1#2 4222
4223 { \__tl_trim_left_spaces:nn { \exp_args:No #2 } { #1 } } 4223
4224 \cs_new:Npn \tl_trim_right_spaces_apply:nN #1#2 4224
4225 { \__tl_trim_right_spaces:nn { \exp_args:No #2 } { #1 } } 4225
4226 \cs_generate_variant:Nn \tl_trim_spaces_apply:nN { o } 4226
4227 \cs_generate_variant:Nn \tl_trim_left_spaces_apply:nN { o } 4227
4228 \cs_generate_variant:Nn \tl_trim_right_spaces_apply:nN { o } 4228
4229 \cs_new_protected:Npn \tl_trim_spaces:N #1 4229
4230 { \__kernel_tl_set:Nx #1 { \exp_args:No \tl_trim_spaces:n {#1} } } 4230
4231 \cs_new_protected:Npn \tl_trim_left_spaces:N #1 4231
4232 { \__kernel_tl_set:Nx #1 { \exp_args:No \tl_trim_left_spaces:n {#1} } } 4232
```

```
4233 \cs_new_protected:Npn \tl_trim_right_spaces:N #1 4233
4234 { \__kernel_tl_set:Nx #1 { \exp_args:No \tl_trim_right_spaces:n {#1} } } 4234
4235 \cs_new_protected:Npn \tl_gtrim_spaces:N #1 4235
4236 { \__kernel_tl_gset:Nx #1 { \exp_args:No \tl_trim_spaces:n {#1} } } 4236
4237 \cs_new_protected:Npn \tl_gtrim_left_spaces:N #1 4237
4238 { \__kernel_tl_gset:Nx #1 { \exp_args:No \tl_trim_left_spaces:n {#1} } } 4238
4239 \cs_new_protected:Npn \tl_gtrim_right_spaces:N #1 4239
4240 { \__kernel_tl_gset:Nx #1 { \exp_args:No \tl_trim_right_spaces:n {#1} } } 4240
4241 \cs_generate_variant:Nn \tl_trim_spaces:N { c } 4241
4242 \cs_generate_variant:Nn \tl_trim_left_spaces:N { c } 4242
4243 \cs_generate_variant:Nn \tl_trim_right_spaces:N { c } 4243
4244 \cs_generate_variant:Nn \tl_gtrim_spaces:N { c } 4244
4245 \cs_generate_variant:Nn \tl_gtrim_left_spaces:N { c } 4245
4246 \cs_generate_variant:Nn \tl_gtrim_right_spaces:N { c } 4246
4247 \cs_set_protected:Npn \__tl_tmp:w #1 4247
4248 { 4248
4249 \cs_new:Npn \__tl_trim_spaces:nn ##1##2 4249
4250 { 4250
4251 \__tl_trim_spaces_auxi:w 4251
4252 \__tl_trim_mark: ##2 \s__tl_nil 4252
4253 \__tl_trim_mark: \__tl_trim_spaces_auxi:w 4253
4254 \__tl_trim_mark: #1 4254
4255 \__tl_trim_mark: \__tl_trim_spaces_auxii:w 4255
4256 {##1} 4256
4257 } 4257
4258 \cs_new:Npn \__tl_trim_left_spaces:nn ##1##2 4258
4259 { 4259
4260 \__tl_trim_spaces_auxi:w 4260
4261 \__tl_trim_mark: ##2 \s__tl_nil 4261
4262 \__tl_trim_mark: \__tl_trim_spaces_auxi:w 4262
4263 \__tl_trim_mark: #1 4263
4264 \__tl_trim_mark: \__tl_trim_spaces_auxv:w 4264
4265 {##1} 4265
4266 } 4266
4267 \cs_new:Npn \__tl_trim_right_spaces:nn ##1##2 4267
4268 { 4268
4269 \__tl_trim_spaces_auxiii:w 4269
4270 \__tl_trim_mark: ##2 \s__tl_nil \__tl_trim_spaces_auxiii:w 4270
4271 #1 \s__tl_nil \__tl_trim_spaces_auxiv:w 4271
4272 {##1} 4272
4273 } 4273
4274 \cs_new:Npn \__tl_trim_spaces_auxi:w 4274
4275 ##1 \__tl_trim_mark: #1 ##2 \__tl_trim_mark: ##3 4275
4276 { ##3 ##1 \__tl_trim_mark: ##2 \__tl_trim_mark: \__tl_trim_spaces_auxi:w } 4276
4277 \cs_new:Npn \__tl_trim_spaces_auxii:w 4277
4278 \__tl_trim_mark: ##1 \__tl_trim_mark: ##2 \__tl_trim_spaces_auxi:w 4278
```

```

4279 \__tl_trim_mark: \__tl_trim_mark: \__tl_trim_spaces_auxi:w 4279
4280 { 4280
4281 \__tl_trim_spaces_auxiii:w 4281
4282 \__tl_trim_mark: ##1 \__tl_trim_spaces_auxiii:w 4282
4283 #1 \s__tl_nil \__tl_trim_spaces_auxiv:w 4283
4284 } 4284
4285 \cs_new:Npn \__tl_trim_spaces_auxiii:w ##1 #1 \s__tl_nil ##2 4285
4286 { ##2 ##1 \s__tl_nil \__tl_trim_spaces_auxiii:w } 4286
4287 \cs_new:Npn \__tl_trim_spaces_auxiv:w 4287
4288 ##1 \s__tl_nil 4288
4289 \__tl_trim_spaces_auxiii:w \s__tl_nil \__tl_trim_spaces_auxiii:w 4289
4290 ##2 4290
4291 { ##2 {##1} } 4291
4292 \cs_new:Npn \__tl_trim_spaces_auxv:w 4292
4293 ##1 \s__tl_nil 4293
4294 \__tl_trim_mark: \__tl_trim_spaces_auxi:w \__tl_trim_mark: 4294
4295 \__tl_trim_mark: \__tl_trim_spaces_auxi:w ##2 4295
4296 { ##2 {##1} } 4296
4297 \cs_new:Npn \__tl_trim_mark: {} 4297
4298 } 4298
4299 \__tl_tmp:w { ~ } 4299
4300 \cs_new:Npn \tl_head:n #1 4300
4301 { 4301
4302 \__kernel_exp_not:w \tex_expanded:D 4302
4303 { { \if_false: { \fi: \__tl_head_aux:n #1 { } } } } 4303
4304 } 4304
4305 \cs_new:Npn \__tl_head_aux:n #1 4305
4306 { 4306
4307 \__kernel_exp_not:w {#1} 4307
4308 \exp_after:wN \use_none:n \exp_after:wN { \if_false: } \fi: 4308
4309 } 4309
4310 \cs_generate_variant:Nn \tl_head:n { V , v , f , e } 4310
4311 \cs_new:Npn \tl_head:w #1#2 \q_stop {#1} 4311
4312 \cs_new:Npn \__tl_tl_head:w #1#2 \s__tl_stop {#1} 4312
4313 \cs_new:Npn \tl_head:N { \exp_args:No \tl_head:n } 4313
4314 \exp_args:Nno \use:n { \cs_new:Npn \tl_tail:n #1 } 4314
4315 { 4315
4316 \exp_after:wN \__kernel_exp_not:w 4316
4317 \tl_if_blank:nTF {#1} 4317
4318 { { } } 4318
4319 { \exp_after:wN { \use_none:n #1 } } 4319
4320 } 4320
4321 \cs_generate_variant:Nn \tl_tail:n { V , v , f , e } 4321
4322 \cs_new:Npn \tl_tail:N { \exp_args:No \tl_tail:n } 4322
4323 \prg_new_conditional:Npnn \tl_if_head_eq_charcode:nN #1#2 { p , T , F , TF } 4323
4324 { 4324

```

```

4325 \if_charcode:w                                     4325
4326 \tl_if_head_is_N_type:nTF { #1 ? }                 4326
4327 { \__tl_head_exp_not:w #1 { ^ \__tl_if_head_eq_empty_arg:w } \s__tl_stop } 4327
4328 { \str_head:n {#1} }                                4328
4329 \exp_not:N #2                                       4329
4330 \prg_return_true:                                   4330
4331 \else:                                              4331
4332 \prg_return_false:                                 4332
4333 \fi:                                                4333
4334 }                                                    4334
4335 \prg_generate_conditional_variant:Nnn \tl_if_head_eq_charcode:nN 4335
4336 { V , e , f } { p , TF , T , F }                 4336
4337 \prg_new_conditional:Npnn \tl_if_head_eq_catcode:nN #1 #2 { p , T , F , TF } 4337
4338 {                                                    4338
4339 \if_catcode:w                                       4339
4340 \tl_if_head_is_N_type:nTF { #1 ? }                 4340
4341 { \__tl_head_exp_not:w #1 { ^ \__tl_if_head_eq_empty_arg:w } \s__tl_stop } 4341
4342 {                                                    4342
4343 \tl_if_head_is_group:nTF {#1}                      4343
4344 \c_group_begin_token                               4344
4345 \c_space_token                                     4345
4346 }                                                    4346
4347 \exp_not:N #2                                       4347
4348 \prg_return_true:                                   4348
4349 \else:                                              4349
4350 \prg_return_false:                                 4350
4351 \fi:                                                4351
4352 }                                                    4352
4353 \prg_generate_conditional_variant:Nnn \tl_if_head_eq_catcode:nN 4353
4354 { V , e , o } { p , TF , T , F }                 4354
4355 \prg_new_conditional:Npnn \tl_if_head_eq_meaning:nN #1#2 { p , T , F , TF } 4355
4356 {                                                    4356
4357 \tl_if_head_is_N_type:nTF { #1 ? }                 4357
4358 \__tl_if_head_eq_meaning_normal:nN                4358
4359 \__tl_if_head_eq_meaning_special:nN               4359
4360 {#1} #2                                             4360
4361 }                                                    4361
4362 \prg_generate_conditional_variant:Nnn \tl_if_head_eq_meaning:nN 4362
4363 { V , e } { p , TF , T , F }                     4363
4364 \cs_new:Npn \__tl_if_head_eq_meaning_normal:nN #1 #2 4364
4365 {                                                    4365
4366 \exp_after:wN \if_meaning:w                        4366
4367 \__tl_tl_head:w #1 { ?? \use_none:nnn } \s__tl_stop #2 4367
4368 \prg_return_true:                                   4368
4369 \else:                                              4369
4370 \prg_return_false:                                 4370

```

```

4371 \fi:
4372 }
4373 \cs_new:Npn \__tl_if_head_eq_meaning_special:nN #1 #2
4374 {
4375     \if_charcode:w \str_head:n {#1} \exp_not:N #2
4376     \exp_after:wN \use_ii:nn
4377 \else:
4378     \prg_return_false:
4379 \fi:
4380 \use_none:n
4381 {
4382     \if_catcode:w \exp_not:N #2
4383         \tl_if_head_is_group:nTF {#1}
4384         { \c_group_begin_token }
4385         { \c_space_token }
4386     \prg_return_true:
4387 \else:
4388     \prg_return_false:
4389 \fi:
4390 }
4391 }
4392 \cs_new:Npn \__tl_head_exp_not:w #1 #2 \s__tl_stop
4393 { \exp_not:N #1 }
4394 \cs_new:Npn \__tl_if_head_eq_empty_arg:w \exp_not:N #1
4395 { ? }
4396 \prg_new_conditional:Npnn \tl_if_head_is_N_type:n #1 { p , T , F , TF }
4397 {
4398     \if:w
4399         \if_false: { \fi: \__tl_if_head_is_N_type_auxi:w #1 ~ }
4400         { \exp_after:wN { \token_to_str:N #1 } }
4401         \scan_stop: \scan_stop:
4402     \prg_return_true:
4403 \else:
4404     \prg_return_false:
4405 \fi:
4406 }
4407 \exp_args:Nno \use:n { \cs_new:Npn \__tl_if_head_is_N_type_auxi:w #1 ~ }
4408 {
4409     \tl_if_empty:nTF {#1}
4410     { f \exp_after:wN \use_none:nn }
4411     { \exp_after:wN \__tl_if_head_is_N_type_auxii:n }
4412     \exp_after:wN { \if_false: } \fi:
4413 }
4414 \cs_new:Npn \__tl_if_head_is_N_type_auxii:n #1
4415 { \exp_after:wN \use_none:n \exp_after:wN }
4416 \prg_new_conditional:Npnn \tl_if_head_is_group:n #1 { p , T , F , TF }

```



```
4417 { 4417
4418 \if:w 4418
4419 \exp_after:wN \use_none:n 4419
4420 \exp_after:wN { \exp_after:wN { \token_to_str:N #1 ? } } 4420
4421 \scan_stop: \scan_stop: 4421
4422 \__tl_if_head_is_group_fi_false:w 4422
4423 \fi: 4423
4424 \if_true: 4424
4425 \prg_return_true: 4425
4426 \else: 4426
4427 \prg_return_false: 4427
4428 \fi: 4428
4429 } 4429
4430 \cs_new:Npn \__tl_if_head_is_group_fi_false:w \fi: \if_true: { \fi: \if_false: } 4430
4431 \prg_new_conditional:Npnn \tl_if_head_is_space:n #1 { p , T , F , TF } 4431
4432 { 4432
4433 \if:w 4433
4434 \if_false: { \fi: \__tl_if_head_is_space:w \prg_do_nothing: #1 ? ~ } 4434
4435 \scan_stop: \scan_stop: 4435
4436 \prg_return_true: 4436
4437 \else: 4437
4438 \prg_return_false: 4438
4439 \fi: 4439
4440 } 4440
4441 \exp_args:Nno \use:n { \cs_new:Npn \__tl_if_head_is_space:w #1 ~ } 4441
4442 { 4442
4443 \__tl_if_empty_if:o {#1} \else: f \fi: 4443
4444 \exp_after:wN \use_none:n \exp_after:wN { \if_false: } \fi: 4444
4445 } 4445
4446 \scan_new:N \s__tl_act_stop 4446
4447 \cs_set_protected:Npn \__tl_tmp:w #1 4447
4448 { 4448
4449 \cs_new:Npn \__tl_act_if_head_is_space:nTF ##1 4449
4450 { 4450
4451 \__tl_act_if_head_is_space:w 4451
4452 \s__tl_act_stop ##1 \s__tl_act_stop \__tl_act_if_head_is_space_true:w 4452
4453 \s__tl_act_stop #1 \s__tl_act_stop \use_ii:nn 4453
4454 } 4454
4455 \cs_new:Npn \__tl_act_if_head_is_space:w 4455
4456 ##1 \s__tl_act_stop #1 ##2 \s__tl_act_stop 4456
4457 {} 4457
4458 \cs_new:Npn \__tl_act_if_head_is_space_true:w 4458
4459 \s__tl_act_stop #1 \s__tl_act_stop \use_ii:nn ##1 ##2 4459
4460 {##1} 4460
4461 } 4461
4462 \__tl_tmp:w { ~ } 4462
```

```
4463 \exp_args:Nne \use:n { \cs_new:Npn \__tl_act_loop:w #1 \s__tl_act_stop } 4463
4464 { 4464
4465 \exp_not:o { \__tl_act_if_head_is_space:nTF {#1} } 4465
4466 \exp_not:N \__tl_act_space:wwNNN 4466
4467 { 4467
4468 \exp_not:o { \tl_if_head_is_group:nTF {#1} } 4468
4469 \exp_not:N \__tl_act_group:nwNNN 4469
4470 \exp_not:N \__tl_act_normal:NwNNN 4470
4471 } 4471
4472 \exp_not:n {#1} \s__tl_act_stop 4472
4473 } 4473
4474 \cs_undefine:N \__tl_act_if_head_is_space:nTF 4474
4475 \cs_new:Npn \__tl_act_normal:NwNNN #1 #2 \s__tl_act_stop #3 4475
4476 { 4476
4477 #3 #1 4477
4478 \__tl_act_loop:w #2 \s__tl_act_stop 4478
4479 #3 4479
4480 } 4480
4481 \cs_new:Npn \__tl_use_none_delimit_by_s_act_stop:w #1 \s__tl_act_stop { } 4481
4482 \cs_new:Npn \__tl_act_end:wn #1 \__tl_act_result:n #2 4482
4483 { \group_align_safe_end: \exp_end: #2 } 4483
4484 \cs_new:Npn \__tl_act_group:nwNNN #1 #2 \s__tl_act_stop #3#4#5 4484
4485 { 4485
4486 \__tl_use_none_delimit_by_s_act_stop:w #1 \__tl_act_end:wn \s__tl_act_stop 4486
4487 #5 {#1} 4487
4488 \__tl_act_loop:w #2 \s__tl_act_stop 4488
4489 #3 #4 #5 4489
4490 } 4490
4491 \exp_last_unbraced:NNo 4491
4492 \cs_new:Npn \__tl_act_space:wwNNN \c_space_tl #1 \s__tl_act_stop #2#3 4492
4493 { 4493
4494 #3 4494
4495 \__tl_act_loop:w #1 \s__tl_act_stop 4495
4496 #2 #3 4496
4497 } 4497
4498 \cs_new:Npn \__tl_act:NNNn #1#2#3#4 4498
4499 { 4499
4500 \group_align_safe_begin: 4500
4501 \__tl_act_loop:w #4 { \s__tl_act_stop } ? \s__tl_act_stop 4501
4502 #1 #3 #2 4502
4503 \__tl_act_result:n { } 4503
4504 } 4504
4505 \cs_new:Npn \__tl_act_output:n #1 #2 \__tl_act_result:n #3 4505
4506 { #2 \__tl_act_result:n { #3 #1 } } 4506
4507 \cs_new:Npn \__tl_act_reverse_output:n #1 #2 \__tl_act_result:n #3 4507
4508 { #2 \__tl_act_result:n { #1 #3 } } 4508
```

```
4509 \cs_new:Npn \tl_reverse:n #1
4510 {
4511     \__kernel_exp_not:w \exp_after:wN
4512     {
4513         \exp:w
4514         \__tl_act:NNNn
4515         \__tl_reverse_normal:N
4516         \__tl_reverse_group_preserve:n
4517         \__tl_reverse_space:
4518         {#1}
4519     }
4520 }
4521 \cs_generate_variant:Nn \tl_reverse:n { o , V , f , e }
4522 \cs_new:Npn \__tl_reverse_normal:N
4523 { \__tl_act_reverse_output:n }
4524 \cs_new:Npn \__tl_reverse_group_preserve:n #1
4525 { \__tl_act_reverse_output:n { {#1} } }
4526 \cs_new:Npn \__tl_reverse_space:
4527 { \__tl_act_reverse_output:n { ~ } }
4528 \cs_new_protected:Npn \tl_reverse:N #1
4529 { \__kernel_tl_set:Nx #1 { \exp_args:No \tl_reverse:n { #1 } } }
4530 \cs_new_protected:Npn \tl_greverse:N #1
4531 { \__kernel_tl_gset:Nx #1 { \exp_args:No \tl_reverse:n { #1 } } }
4532 \cs_generate_variant:Nn \tl_reverse:N { c }
4533 \cs_generate_variant:Nn \tl_greverse:N { c }
4534 \cs_new:Npn \tl_item:nn #1#2
4535 {
4536     \exp_args:Nf \__tl_item:nn
4537     { \exp_args:Nf \__tl_item_aux:nn { \int_eval:n {#2} } {#1} }
4538     #1
4539     \q__tl_recursion_tail
4540     \prg_break_point:
4541 }
4542 \cs_new:Npn \__tl_item_aux:nn #1#2
4543 {
4544     \int_compare:nNnTF {#1} < 0
4545     { \int_eval:n { \tl_count:n {#2} + 1 + #1 } }
4546     {#1}
4547 }
4548 \cs_new:Npn \__tl_item:nn #1#2
4549 {
4550     \__tl_if_recursion_tail_break:nN {#2} \prg_break:
4551     \int_compare:nNnTF {#1} = 1
4552     { \prg_break:n { \exp_not:n {#2} } }
4553     { \exp_args:Nf \__tl_item:nn { \int_eval:n { #1 - 1 } } }
4554 }
```

```

4555 \cs_new:Npn \tl_item:Nn { \exp_args:No \tl_item:nn }
4556 \cs_generate_variant:Nn \tl_item:Nn { c }
4557 \cs_new:Npn \tl_rand_item:n #1
4558 {
4559     \tl_if_blank:nF {#1}
4560     { \tl_item:nn {#1} { \int_rand:nn { 1 } { \tl_count:n {#1} } } }
4561 }
4562 \cs_new:Npn \tl_rand_item:N { \exp_args:No \tl_rand_item:n }
4563 \cs_generate_variant:Nn \tl_rand_item:N { c }
4564 \cs_new_eq:NN \__kernel_int_sep: \tex_right:D
4565 \cs_new_eq:NN \__tl_sep: \__kernel_int_sep:
4566 \cs_new:Npn \tl_range:Nnn { \exp_args:No \tl_range:nnn }
4567 \cs_generate_variant:Nn \tl_range:Nnn { c }
4568 \cs_new:Npn \tl_range:nnn { \__tl_range:Nnnn \__tl_range:w }
4569 \cs_new:Npn \__tl_range:Nnnn #1#2#3#4
4570 {
4571     \tl_head:f
4572     {
4573         \exp_args:Nf \__tl_range:nnnNn
4574         { \tl_count:n {#2} } {#3} {#4} #1 {#2}
4575     }
4576 }
4577 \cs_new:Npn \__tl_range:nnnNn #1#2#3
4578 {
4579     \exp_args:Nff \__tl_range:nnNn
4580     {
4581         \exp_args:Nf \__tl_range_normalize:nn
4582         { \int_eval:n { #2 - 1 } } {#1}
4583     }
4584     {
4585         \exp_args:Nf \__tl_range_normalize:nn
4586         { \int_eval:n {#3} } {#1}
4587     }
4588 }
4589 \cs_new:Npn \__tl_range:nnNn #1#2#3#4
4590 {
4591     \if_int_compare:w #2 > #1 \exp_stop_f: \else:
4592         \exp_after:wN { \exp_after:wN }
4593     \fi:
4594     \exp_after:wN #3
4595     \int_value:w \int_eval:n { #2 - #1 } \exp_after:wN \__tl_sep:
4596     \exp_after:wN { \exp:w \__tl_range_skip:w #1 \__tl_sep: { } #4 }
4597 }
4598 \cs_new:Npn \__tl_range_skip:w #1 \__tl_sep: #2
4599 {
4600     \if_int_compare:w #1 > \c_zero_int

```

```

4601     \exp_after:wN \__tl_range_skip:w
4602     \int_value:w \int_eval:n { #1 - 1 } \exp_after:wN \__tl_sep:
4603     \else:
4604     \exp_after:wN \exp_end:
4605     \fi:
4606 }
4607 \cs_new:Npn \__tl_range:w #1 \__tl_sep: #2
4608 {
4609     \exp_args:Nf \__tl_range_collect:nn
4610     { \__tl_range_skip_spaces:n {#2} } {#1}
4611 }
4612 \cs_new:Npn \__tl_range_skip_spaces:n #1
4613 {
4614     \tl_if_head_is_space:nTF {#1}
4615     { \exp_args:Nf \__tl_range_skip_spaces:n {#1} }
4616     { { } #1 }
4617 }
4618 \cs_new:Npn \__tl_range_collect:nn #1#2
4619 {
4620     \int_compare:nNnTF {#2} = 0
4621     {#1}
4622     {
4623         \exp_args:No \tl_if_head_is_space:nTF { \use_none:n #1 }
4624         {
4625             \exp_args:Nf \__tl_range_collect:nn
4626             { \__tl_range_collect_space:nw #1 }
4627             {#2}
4628         }
4629         {
4630             \__tl_range_collect:ff
4631             {
4632                 \exp_args:No \tl_if_head_is_N_type:nTF { \use_none:n #1 }
4633                 { \__tl_range_collect_N:nN }
4634                 { \__tl_range_collect_group:nn }
4635                 #1
4636             }
4637             { \int_eval:n { #2 - 1 } }
4638         }
4639     }
4640 }
4641 \cs_new:Npn \__tl_range_collect_space:nw #1 ~ { { #1 ~ } }
4642 \cs_new:Npn \__tl_range_collect_N:nN #1#2 { { #1 #2 } }
4643 \cs_new:Npn \__tl_range_collect_group:nn #1#2 { { #1 {#2} } }
4644 \cs_generate_variant:Nn \__tl_range_collect:nn { ff }
4645 \cs_new:Npn \__tl_range_normalize:nn #1#2
4646 {

```

```

4647 \int_eval:n
4648 {
4649 \if_int_compare:w #1 < \c_zero_int
4650 \if_int_compare:w #1 < -#2 \exp_stop_f:
4651 0
4652 \else:
4653 #1 + #2 + 1
4654 \fi:
4655 \else:
4656 \if_int_compare:w #1 < #2 \exp_stop_f:
4657 #1
4658 \else:
4659 #2
4660 \fi:
4661 \fi:
4662 }
4663 }
4664 \cs_new_protected:Npn \tl_show:N { \__tl_show:NN \tl_show:n }
4665 \cs_generate_variant:Nn \tl_show:N { c }
4666 \cs_new_protected:Npn \tl_log:N { \__tl_show:NN \tl_log:n }
4667 \cs_generate_variant:Nn \tl_log:N { c }
4668 \cs_new_protected:Npn \__tl_show:NN #1#2
4669 {
4670 \__kernel_chk_defined:NT #2
4671 {
4672 \exp_args:Nf \tl_if_empty:nTF
4673 { \cs_prefix_spec:N #2 \cs_parameter_spec:N #2 }
4674 {
4675 \exp_args:Ne #1
4676 { \token_to_str:N #2 = \__kernel_exp_not:w \exp_after:wN {#2} }
4677 }
4678 {
4679 \msg_error:nneee { kernel } { bad-type }
4680 { \token_to_str:N #2 } { \token_to_meaning:N #2 } { tl }
4681 }
4682 }
4683 }
4684 \cs_new_protected:Npn \tl_show:n #1
4685 { \iow_wrap:nnnN { >~ \tl_to_str:n {#1} . } { } { } { } \__tl_show:n }
4686 \cs_generate_variant:Nn \tl_show:n { e , x }
4687 \cs_new_protected:Npn \__tl_show:n #1
4688 {
4689 \tl_set:Nf \l__tl_internal_a_tl { \__tl_show:w #1 \s__tl_stop }
4690 \__kernel_iow_with:Nnn \tex_newlinechar:D { 10 }
4691 {
4692 \__kernel_iow_with:Nnn \tex_errorcontextlines:D { -1 }

```



```

4693         {
4694             \tex_showtokens:D \exp_after:wN \exp_after:wN \exp_after:wN
4695             { \exp_after:wN \l__tl_internal_a_tl }
4696         }
4697     }
4698 }
4699 \cs_new:Npn \__tl_show:w #1 > #2 . \s__tl_stop {#2}
4700 \cs_new_protected:Npn \tl_log:n #1
4701 { \iow_wrap:nnnN { > ~ \tl_to_str:n {#1} . } { } { } { } \iow_log:n }
4702 \cs_generate_variant:Nn \tl_log:n { e , x }
4703 \cs_new_protected:Npn \__kernel_chk_tl_type:NnnT #1#2#3#4
4704 {
4705     \__kernel_chk_defined:NT #1
4706     {
4707         \exp_args:Nf \tl_if_empty:nTF
4708         { \cs_prefix_spec:N #1 \cs_parameter_spec:N #1 }
4709         {
4710             \tl_set:Ne \l__tl_internal_a_tl {#3}
4711             \tl_if_eq:NNTF #1 \l__tl_internal_a_tl
4712             {#4}
4713             {
4714                 \msg_error:nneeee { kernel } { bad-type }
4715                 { \token_to_str:N #1 } { \tl_to_str:N #1 }
4716                 {#2} { \tl_to_str:N \l__tl_internal_a_tl }
4717             }
4718         }
4719     }
4720     \msg_error:nneeee { kernel } { bad-type }
4721     { \token_to_str:N #1 } { \token_to_meaning:N #1 } {#2}
4722 }
4723 }
4724 }
4725 \scan_new:N \s__tl_nil
4726 \scan_new:N \s__tl_mark
4727 \scan_new:N \s__tl_stop
4728 \tl_new:N \g_tmpa_tl
4729 \tl_new:N \g_tmpb_tl
4730 \tl_new:N \l_tmpa_tl
4731 \tl_new:N \l_tmpb_tl
4732 \cs_undefine:N \__tl_tmp:w
4733 %% File: l3tl-build.dtx
4734 \cs_new_protected:Npn \tl_build_begin:N #1
4735 { \__tl_build_begin:NN \cs_set_nopar:Npe #1 }
4736 \cs_new_protected:Npn \tl_build_gbegin:N #1
4737 { \__tl_build_begin:NN \cs_gset_nopar:Npe #1 }
4738 \cs_new_protected:Npn \__tl_build_begin:NN #1#2

```

```
4739 { \exp_args:Nc \__tl_build_begin:NNN { \cs_to_str:N #2 ' } #2 #1 } 4739
4740 \cs_new_protected:Npn \__tl_build_begin:NNN #1#2#3 4740
4741 { 4741
4742     #3 #1 { } 4742
4743     #3 #2 4743
4744     { 4744
4745         \exp_not:n { \exp_end: \exp_end: \exp_end: \exp_end: } 4745
4746         \exp_not:n { \__tl_build_last:NNn #3 #1 { } } 4746
4747     } 4747
4748 } 4748
4749 \cs_new_protected:Npn \tl_build_put_right:Nn #1#2 4749
4750 { 4750
4751     \cs_set_nopar:Npe #1 4751
4752     { \__kernel_exp_not:w \exp_after:wN { \exp:w #1 #2 } } 4752
4753 } 4753
4754 \cs_generate_variant:Nn \tl_build_put_right:Nn { Ne , Nx } 4754
4755 \cs_new_protected:Npn \tl_build_gput_right:Nn #1#2 4755
4756 { 4756
4757     \cs_gset_nopar:Npe #1 4757
4758     { \__kernel_exp_not:w \exp_after:wN { \exp:w #1 #2 } } 4758
4759 } 4759
4760 \cs_generate_variant:Nn \tl_build_gput_right:Nn { Ne , Nx } 4760
4761 \cs_new_protected:Npn \__tl_build_last:NNn #1#2 4761
4762 { 4762
4763     \if_false: { { \fi: 4763
4764         \exp_end: \exp_end: \exp_end: \exp_end: \exp_end: 4764
4765         \__tl_build_last:NNn #1 #2 { } 4765
4766     } 4766
4767 } 4767
4768 \if_meaning:w \c_empty_tl #2 4768
4769     \__tl_build_begin:NN #1 #2 4769
4770 \fi: 4770
4771 #1 #2 4771
4772 { 4772
4773     \__kernel_exp_not:w \exp_after:wN 4773
4774     { 4774
4775         \exp:w \if_false: } } \fi: 4775
4776     \exp_after:wN \__tl_build_put:nn \exp_after:wN {#2} 4776
4777 } 4777
4778 \cs_new_protected:Npn \__tl_build_put:nn #1#2 { \__tl_build_put:nw {#2} #1 } 4778
4779 \cs_new_protected:Npn \__tl_build_put:nw #1#2 \__tl_build_last:NNn #3#4#5 4779
4780     { #2 \__tl_build_last:NNn #3 #4 { #1 #5 } } 4780
4781 \cs_new_protected:Npn \tl_build_put_left:Nn #1 4781
4782     { \__tl_build_put_left:NNn \cs_set_nopar:Npe #1 } 4782
4783 \cs_generate_variant:Nn \tl_build_put_left:Nn { Ne , Nx } 4783
4784 \cs_new_protected:Npn \tl_build_gput_left:Nn #1 4784
```

```
4785 { \__tl_build_put_left:NNn \cs_gset_nopar:Npe #1 } 4785
4786 \cs_generate_variant:Nn \tl_build_gput_left:Nn { Ne , Nx } 4786
4787 \cs_new_protected:Npn \__tl_build_put_left:NNn #1#2#3 4787
4788 { 4788
4789     #1 #2 4789
4790     { 4790
4791         \__kernel_exp_not:w \exp_after:wN 4791
4792         { 4792
4793             \exp:w \exp_after:wN \__tl_build_put:nn 4793
4794             \exp_after:wN {#2} {#3} 4794
4795         } 4795
4796     } 4796
4797 } 4797
4798 \cs_new_protected:Npn \tl_build_end:N #1 4798
4799 { 4799
4800     \__tl_build_get:NNN \__kernel_tl_set:Nx #1 #1 4800
4801     \exp_args:Nc \__tl_build_end_loop:NN { \cs_to_str:N #1 ' } \tl_clear:N 4801
4802 } 4802
4803 \cs_new_protected:Npn \tl_build_gend:N #1 4803
4804 { 4804
4805     \__tl_build_get:NNN \__kernel_tl_gset:Nx #1 #1 4805
4806     \exp_args:Nc \__tl_build_end_loop:NN { \cs_to_str:N #1 ' } \tl_gclear:N 4806
4807 } 4807
4808 \cs_new_protected:Npn \__tl_build_end_loop:NN #1#2 4808
4809 { 4809
4810     \if_meaning:w \c_empty_tl #1 4810
4811     \exp_after:wN \use_none:nnnnnn 4811
4812     \fi: 4812
4813     #2 #1 4813
4814     \exp_args:Nc \__tl_build_end_loop:NN { \cs_to_str:N #1 ' } #2 4814
4815 } 4815
4816 \cs_new_protected:Npn \tl_build_get_intermediate:NN 4816
4817 { \__tl_build_get:NNN \__kernel_tl_set:Nx } 4817
4818 \cs_new_protected:Npn \__tl_build_get:NNN #1#2#3 4818
4819 { #1 #3 { \if_false: { \fi: \exp_after:wN \__tl_build_get:w #2 } } } 4819
4820 \cs_new:Npn \__tl_build_get:w #1 \__tl_build_last:NNn #2#3#4 4820
4821 { 4821
4822     \exp_not:n {#4} 4822
4823     \if_meaning:w \c_empty_tl #3 4823
4824     \exp_after:wN \__tl_build_get_end:w 4824
4825     \fi: 4825
4826     \exp_after:wN \__tl_build_get:w #3 4826
4827 } 4827
4828 \cs_new:Npn \__tl_build_get_end:w #1#2#3 4828
4829 { \__kernel_exp_not:w \exp_after:wN { \if_false: } \fi: } 4829
4830 %% File: l3str.dtx 4830
```

```
4831 \scan_new:N \s__str_mark 4831
4832 \scan_new:N \s__str_stop 4832
4833 \cs_new:Npn \__str_use_none_delimit_by_s_stop:w #1 \s__str_stop { } 4833
4834 \cs_new:Npn \__str_use_i_delimit_by_s_stop:nw #1 #2 \s__str_stop {#1} 4834
4835 \quark_new:N \q__str_recursion_tail 4835
4836 \quark_new:N \q__str_recursion_stop 4836
4837 \__kernel_quark_new_test:N \__str_if_recursion_tail_break:NN 4837
4838 \__kernel_quark_new_test:N \__str_if_recursion_tail_stop_do:Nn 4838
4839 \group_begin: 4839
4840 \cs_set_protected:Npn \__str_tmp:n #1 4840
4841 { 4841
4842 \tl_if_blank:nF {#1} 4842
4843 { 4843
4844 \cs_new_eq:cc { str_ #1 :N } { tl_ #1 :N } 4844
4845 \exp_args:Nc \cs_generate_variant:Nn { str_ #1 :N } { c } 4845
4846 \__str_tmp:n 4846
4847 } 4847
4848 } 4848
4849 \__str_tmp:n 4849
4850 { new } 4850
4851 { use } 4851
4852 { clear } 4852
4853 { gclear } 4853
4854 { clear_new } 4854
4855 { gclear_new } 4855
4856 { } 4856
4857 \group_end: 4857
4858 \cs_new_eq:NN \str_set_eq:NN \tl_set_eq:NN 4858
4859 \cs_new_eq:NN \str_gset_eq:NN \tl_gset_eq:NN 4859
4860 \cs_generate_variant:Nn \str_set_eq:NN { c , Nc , cc } 4860
4861 \cs_generate_variant:Nn \str_gset_eq:NN { c , Nc , cc } 4861
4862 \cs_new_eq:NN \str_concat:NNN \tl_concat:NNN 4862
4863 \cs_new_eq:NN \str_gconcat:NNN \tl_gconcat:NNN 4863
4864 \cs_generate_variant:Nn \str_concat:NNN { ccc } 4864
4865 \cs_generate_variant:Nn \str_gconcat:NNN { ccc } 4865
4866 \cs_new_protected:Npn \str_set:Nn #1#2 4866
4867 { \__kernel_tl_set:Nx #1 { \__kernel_tl_to_str:w {#2} } } 4867
4868 \cs_gset_protected:Npn \str_gset:Nn #1#2 4868
4869 { \__kernel_tl_gset:Nx #1 { \__kernel_tl_to_str:w {#2} } } 4869
4870 \cs_new_protected:Npn \str_const:Nn #1#2 4870
4871 { 4871
4872 \__kernel_chk_if_free_cs:N #1 4872
4873 \cs_gset_nopar:Npe #1 { \__kernel_tl_to_str:w {#2} } 4873
4874 } 4874
4875 \cs_new_protected:Npn \str_put_left:Nn #1#2 4875
4876 { 4876
```

```
4877     \__kernel_tl_set:Nx #1
4878     { \__kernel_tl_to_str:w {#2} \__kernel_exp_not:w \exp_after:wN {#1} }
4879 }
4880 \cs_new_protected:Npn \str_gput_left:Nn #1#2
4881 {
4882     \__kernel_tl_gset:Nx #1
4883     { \__kernel_tl_to_str:w {#2} \__kernel_exp_not:w \exp_after:wN {#1} }
4884 }
4885 \cs_new_protected:Npn \str_put_right:Nn #1#2
4886 {
4887     \__kernel_tl_set:Nx #1
4888     { \__kernel_exp_not:w \exp_after:wN {#1} \__kernel_tl_to_str:w {#2} }
4889 }
4890 \cs_new_protected:Npn \str_gput_right:Nn #1#2
4891 {
4892     \__kernel_tl_gset:Nx #1
4893     { \__kernel_exp_not:w \exp_after:wN {#1} \__kernel_tl_to_str:w {#2} }
4894 }
4895 \cs_generate_variant:Nn \str_set:Nn { NV , Ne , Nx , c , cV , ce , cx }
4896 \cs_generate_variant:Nn \str_gset:Nn { NV , Ne , Nx , c , cV , ce , cx }
4897 \cs_generate_variant:Nn \str_const:Nn { NV , Ne , Nx , c , cV , ce , cx }
4898 \cs_generate_variant:Nn \str_put_left:Nn { NV , Ne , Nx , c , cV , ce , cx }
4899 \cs_generate_variant:Nn \str_gput_left:Nn { NV , Ne , Nx , c , cV , ce , cx }
4900 \cs_generate_variant:Nn \str_put_right:Nn { NV , Ne , Nx , c , cV , ce , cx }
4901 \cs_generate_variant:Nn \str_gput_right:Nn { NV , Ne , Nx , c , cV , ce , cx }
4902 \cs_new_protected:Npn \str_replace_once:Nnn
4903 { \__str_replace:NNNnn \prg_do_nothing: \__kernel_tl_set:Nx }
4904 \cs_new_protected:Npn \str_greplace_once:Nnn
4905 { \__str_replace:NNNnn \prg_do_nothing: \__kernel_tl_gset:Nx }
4906 \cs_new_protected:Npn \str_replace_all:Nnn
4907 { \__str_replace:NNNnn \__str_replace_next:w \__kernel_tl_set:Nx }
4908 \cs_new_protected:Npn \str_greplace_all:Nnn
4909 { \__str_replace:NNNnn \__str_replace_next:w \__kernel_tl_gset:Nx }
4910 \cs_generate_variant:Nn \str_replace_once:Nnn { c }
4911 \cs_generate_variant:Nn \str_greplace_once:Nnn { c }
4912 \cs_generate_variant:Nn \str_replace_all:Nnn { c }
4913 \cs_generate_variant:Nn \str_greplace_all:Nnn { c }
4914 \cs_new_protected:Npn \__str_replace:NNNnn #1#2#3#4#5
4915 {
4916     \tl_if_empty:nTF {#4}
4917     {
4918         \msg_error:nne { kernel } { empty-search-pattern } {#5}
4919     }
4920     {
4921         \use:e
4922     }
```

```

4923         \exp_not:n { \__str_replace_aux:NNNnnn #1 #2 #3 }
4924         { \tl_to_str:N #3 }
4925         { \tl_to_str:n {#4} } { \tl_to_str:n {#5} }
4926     }
4927 }
4928 }
4929 \cs_new_protected:Npn \__str_replace_aux:NNNnnn #1#2#3#4#5#6
4930 {
4931     \cs_set:Npn \__str_replace_next:w ##1 #5 { ##1 #6 #1 }
4932     #2 #3
4933     {
4934         \__str_replace_next:w
4935         #4
4936         \__str_use_none_delimit_by_s_stop:w
4937         #5
4938         \s__str_stop
4939     }
4940 }
4941 \cs_new_eq:NN \__str_replace_next:w ?
4942 \cs_new_protected:Npn \str_remove_once:Nn #1#2
4943 { \str_replace_once:Nnn #1 {#2} { } }
4944 \cs_new_protected:Npn \str_gremove_once:Nn #1#2
4945 { \str_greplace_once:Nnn #1 {#2} { } }
4946 \cs_generate_variant:Nn \str_remove_once:Nn { c }
4947 \cs_generate_variant:Nn \str_gremove_once:Nn { c }
4948 \cs_new_protected:Npn \str_remove_all:Nn #1#2
4949 { \str_replace_all:Nnn #1 {#2} { } }
4950 \cs_new_protected:Npn \str_gremove_all:Nn #1#2
4951 { \str_greplace_all:Nnn #1 {#2} { } }
4952 \cs_generate_variant:Nn \str_remove_all:Nn { c }
4953 \cs_generate_variant:Nn \str_gremove_all:Nn { c }
4954 \prg_new_eq_conditional:NNn \str_if_exist:N \tl_if_exist:N
4955 { p , T , F , TF }
4956 \prg_new_eq_conditional:NNn \str_if_exist:c \tl_if_exist:c
4957 { p , T , F , TF }
4958 \prg_new_eq_conditional:NNn \str_if_empty:N \tl_if_empty:N
4959 { p , T , F , TF }
4960 \prg_new_eq_conditional:NNn \str_if_empty:c \tl_if_empty:c
4961 { p , T , F , TF }
4962 \prg_new_eq_conditional:NNn \str_if_empty:n \tl_if_empty:n
4963 { p , T , F , TF }
4964 \cs_new_eq:NN \__str_if_eq:nn \tex_strcmp:D
4965 \prg_new_conditional:Npnn \str_compare:nNn #1#2#3 { p , T , F , TF }
4966 {
4967     \if_int_compare:w
4968         \__str_if_eq:nn { \exp_not:n {#1} } { \exp_not:n {#3} }

```



```

4969         #2 \c_zero_int
4970         \prg_return_true: \else: \prg_return_false: \fi:
4971     }
4972 \prg_new_conditional:Npnn \str_compare:eNe #1#2#3 { p , T , F , TF }
4973 {
4974     \if_int_compare:w \__str_if_eq:nn {#1} {#3} #2 \c_zero_int
4975     \prg_return_true: \else: \prg_return_false: \fi:
4976 }
4977 \prg_new_conditional:Npnn \str_if_eq:nn #1#2 { p , T , F , TF }
4978 {
4979     \if:w 0 \__str_if_eq:nn { \exp_not:n {#1} } { \exp_not:n {#2} }
4980     \prg_return_true: \else: \prg_return_false: \fi:
4981 }
4982 \prg_generate_conditional_variant:Nnn \str_if_eq:nn
4983 { V , v , o , nV , no , VV , nv } { p , T , F , TF }
4984 \prg_new_conditional:Npnn \str_if_eq:ee #1#2 { p , T , F , TF }
4985 {
4986     \if:w 0 \__str_if_eq:nn {#1} {#2}
4987     \prg_return_true: \else: \prg_return_false: \fi:
4988 }
4989 \prg_new_conditional:Npnn \str_if_eq:NN #1#2 { p , TF , T , F }
4990 {
4991     \if:w 0 \__str_if_eq:nn { \tl_to_str:N #1 } { \tl_to_str:N #2 }
4992     \prg_return_true: \else: \prg_return_false: \fi:
4993 }
4994 \prg_generate_conditional_variant:Nnn \str_if_eq:NN
4995 { c , Nc , cc } { T , F , TF , p }
4996 \prg_new_protected_conditional:Npnn \str_if_in:Nn #1#2 { T , F , TF }
4997 {
4998     \use:e
4999     { \tl_if_in:nnTF { \tl_to_str:N #1 } { \tl_to_str:n {#2} } }
5000     { \prg_return_true: } { \prg_return_false: }
5001 }
5002 \prg_generate_conditional_variant:Nnn \str_if_in:Nn
5003 { c } { T , F , TF }
5004 \prg_new_protected_conditional:Npnn \str_if_in:nn #1#2 { T , F , TF }
5005 {
5006     \use:e
5007     { \tl_if_in:nnTF { \tl_to_str:n {#1} } { \tl_to_str:n {#2} } }
5008     { \prg_return_true: } { \prg_return_false: }
5009 }
5010 \cs_new:Npn \str_case:nn #1#2
5011 {
5012     \exp:w
5013     \__str_case:nnTF {#1} {#2} { } { }
5014 }

```

5015	\cs_new:Npn \str_case:nnT #1#2#3	5015
5016	{	5016
5017	\exp:w	5017
5018	__str_case:nnTF {#1} {#2} {#3} { }	5018
5019	}	5019
5020	\cs_new:Npn \str_case:nnF #1#2	5020
5021	{	5021
5022	\exp:w	5022
5023	__str_case:nnTF {#1} {#2} { }	5023
5024	}	5024
5025	\cs_new:Npn \str_case:nnTF #1#2	5025
5026	{	5026
5027	\exp:w	5027
5028	__str_case:nnTF {#1} {#2}	5028
5029	}	5029
5030	\cs_new:Npn __str_case:nnTF #1#2#3#4	5030
5031	{ __str_case:nw {#1} #2 {#1} { } \s__str_mark {#3} \s__str_mark {#4} \s__str_stop }	5031
5032	\cs_generate_variant:Nn \str_case:nn { V , o , e , nV , nv , ne }	5032
5033	\prg_generate_conditional_variant:Nnn \str_case:nn	5033
5034	{ V , o , e , nV , nv , ne } { T , F , TF }	5034
5035	\cs_new_eq:NN \str_case:Nn \str_case:Vn	5035
5036	\cs_new_eq:NN \str_case:NnT \str_case:VnT	5036
5037	\cs_new_eq:NN \str_case:NnF \str_case:VnF	5037
5038	\cs_new_eq:NN \str_case:NnTF \str_case:VnTF	5038
5039	\cs_new:Npn __str_case:nw #1#2#3	5039
5040	{	5040
5041	\str_if_eq:nnTF {#1} {#2}	5041
5042	{ __str_case_end:nw {#3} }	5042
5043	{ __str_case:nw {#1} }	5043
5044	}	5044
5045	\cs_new:Npn \str_case_e:nn #1#2	5045
5046	{	5046
5047	\exp:w	5047
5048	__str_case_e:nnTF {#1} {#2} { } { }	5048
5049	}	5049
5050	\cs_new:Npn \str_case_e:nnT #1#2#3	5050
5051	{	5051
5052	\exp:w	5052
5053	__str_case_e:nnTF {#1} {#2} {#3} { }	5053
5054	}	5054
5055	\cs_new:Npn \str_case_e:nnF #1#2	5055
5056	{	5056
5057	\exp:w	5057
5058	__str_case_e:nnTF {#1} {#2} { }	5058
5059	}	5059
5060	\cs_new:Npn \str_case_e:nnTF #1#2	5060

```
5061 { 5061
5062     \exp:w 5062
5063     \__str_case_e:nnTF {#1} {#2} 5063
5064 } 5064
5065 \cs_new:Npn \__str_case_e:nnTF #1#2#3#4 5065
5066 { \__str_case_e:nw {#1} #2 {#1} { } \s__str_mark {#3} \s__str_mark {#4} \s__str_stop } 5066
5067 \cs_generate_variant:Nn \str_case_e:nn { e } 5067
5068 \prg_generate_conditional_variant:Nnn \str_case_e:nn { e } { T , F , TF } 5068
5069 \cs_new:Npn \__str_case_e:nw #1#2#3 5069
5070 { 5070
5071     \str_if_eq:eeTF {#1} {#2} 5071
5072     { \__str_case_end:nw {#3} } 5072
5073     { \__str_case_e:nw {#1} } 5073
5074 } 5074
5075 \cs_new:Npn \__str_case_end:nw #1#2#3 \s__str_mark #4#5 \s__str_stop 5075
5076 { \exp_end: #1 #4 } 5076
5077 \cs_new:Npn \str_map_function:nN #1#2 5077
5078 { 5078
5079     \exp_after:wN \__str_map_function:w 5079
5080     \exp_after:wN \__str_map_function:nn \exp_after:wN #2 5080
5081     \__kernel_tl_to_str:w {#1} 5081
5082     \q__str_recursion_tail ? ~ 5082
5083     \prg_break_point:Nn \str_map_break: { } 5083
5084 } 5084
5085 \cs_new:Npn \str_map_function:NN 5085
5086 { \exp_args:No \str_map_function:nN } 5086
5087 \cs_new:Npn \__str_map_function:w #1 ~ 5087
5088 { #1 { ~ { ~ } } \__str_map_function:w } } 5088
5089 \cs_new:Npn \__str_map_function:nn #1#2 5089
5090 { 5090
5091     \if_meaning:w \q__str_recursion_tail #2 5091
5092     \exp_after:wN \str_map_break: 5092
5093     \fi: 5093
5094     #1 #2 \__str_map_function:nn {#1} 5094
5095 } 5095
5096 \cs_generate_variant:Nn \str_map_function:NN { c } 5096
5097 \cs_new_protected:Npn \str_map_inline:nn #1#2 5097
5098 { 5098
5099     \int_gincr:N \g__kernel_prg_map_int 5099
5100     \cs_gset_protected:cpn 5100
5101     { __str_map_ \int_use:N \g__kernel_prg_map_int :w } ##1 {#2} 5101
5102     \use:e 5102
5103     { 5103
5104         \exp_not:N \__str_map_inline:NN 5104
5105         \exp_not:c { __str_map_ \int_use:N \g__kernel_prg_map_int :w } 5105
5106         \__kernel_str_to_other_fast:n {#1} 5106
```

5107	}	5107
5108	\q__str_recursion_tail	5108
5109	\prg_break_point:Nn \str_map_break:	5109
5110	{ \int_gdecr:N \g__kernel_prg_map_int }	5110
5111	}	5111
5112	\cs_new_protected:Npn \str_map_inline:Nn	5112
5113	{ \exp_args:No \str_map_inline:nn }	5113
5114	\cs_generate_variant:Nn \str_map_inline:Nn { c }	5114
5115	\cs_new:Npn __str_map_inline:NN #1#2	5115
5116	{	5116
5117	__str_if_recursion_tail_break:NN #2 \str_map_break:	5117
5118	\exp_args:No #1 { \token_to_str:N #2 }	5118
5119	__str_map_inline:NN #1	5119
5120	}	5120
5121	\cs_new_protected:Npn \str_map_variable:nNn #1#2#3	5121
5122	{	5122
5123	\use:e	5123
5124	{	5124
5125	\exp_not:n { __str_map_variable:NnN #2 {#3} }	5125
5126	__kernel_str_to_other_fast:n {#1}	5126
5127	}	5127
5128	\q__str_recursion_tail	5128
5129	\prg_break_point:Nn \str_map_break: { }	5129
5130	}	5130
5131	\cs_new_protected:Npn \str_map_variable:NNn	5131
5132	{ \exp_args:No \str_map_variable:nNn }	5132
5133	\cs_new_protected:Npn __str_map_variable:NnN #1#2#3	5133
5134	{	5134
5135	__str_if_recursion_tail_break:NN #3 \str_map_break:	5135
5136	\str_set:Nn #1 {#3}	5136
5137	\use:n {#2}	5137
5138	__str_map_variable:NnN #1 {#2}	5138
5139	}	5139
5140	\cs_generate_variant:Nn \str_map_variable:NNn { c }	5140
5141	\cs_new:Npn \str_map_break:	5141
5142	{ \prg_map_break:Nn \str_map_break: { } }	5142
5143	\cs_new:Npn \str_map_break:n	5143
5144	{ \prg_map_break:Nn \str_map_break: }	5144
5145	\cs_new:Npn \str_map_tokens:nn #1#2	5145
5146	{	5146
5147	\exp_args:Nno \use:nn	5147
5148	{ __str_map_function:w __str_map_function:nn {#2} }	5148
5149	{ __kernel_tl_to_str:w {#1} }	5149
5150	\q__str_recursion_tail ? ~	5150
5151	\prg_break_point:Nn \str_map_break: { }	5151
5152	}	5152

```
5153 \cs_new:Npn \str_map_tokens:Nn { \exp_args:No \str_map_tokens:nn } 5153
5154 \cs_generate_variant:Nn \str_map_tokens:Nn { c } 5154
5155 \cs_new:Npn \__kernel_str_to_other:n #1 5155
5156 { 5156
5157     \exp_after:wN \__str_to_other_loop:w 5157
5158     \tl_to_str:n {#1} ~ A ~ A ~ A ~ A ~ A ~ A ~ A ~ A ~ \s__str_mark \s__str_stop 5158
5159 } 5159
5160 \group_begin: 5160
5161 \tex_lccode:D \* = \_ % 5161
5162 \tex_lccode:D \A = \A % 5162
5163 \tex_lowercase:D 5163
5164 { 5164
5165     \group_end: 5165
5166     \cs_new:Npn \__str_to_other_loop:w 5166
5167         #1 ~ #2 ~ #3 ~ #4 ~ #5 ~ #6 ~ #7 ~ #8 ~ #9 \s__str_stop 5167
5168     { 5168
5169         \if_meaning:w A #8 5169
5170         \__str_to_other_end:w 5170
5171         \fi: 5171
5172         \__str_to_other_loop:w 5172
5173         #9 #1 * #2 * #3 * #4 * #5 * #6 * #7 * #8 * \s__str_stop 5173
5174     } 5174
5175     \cs_new:Npn \__str_to_other_end:w \fi: #1 \s__str_mark #2 * A #3 \s__str_stop 5175
5176     { \fi: #2 } 5176
5177 } 5177
5178 \cs_new:Npn \__kernel_str_to_other_fast:n #1 5178
5179 { 5179
5180     \exp_after:wN \__str_to_other_fast_loop:w \tl_to_str:n {#1} ~ 5180
5181     A ~ A ~ A ~ A ~ A ~ A ~ A ~ A ~ A ~ \s__str_stop 5181
5182 } 5182
5183 \group_begin: 5183
5184 \tex_lccode:D \* = \_ % 5184
5185 \tex_lccode:D \A = \A % 5185
5186 \tex_lowercase:D 5186
5187 { 5187
5188     \group_end: 5188
5189     \cs_new:Npn \__str_to_other_fast_loop:w 5189
5190         #1 ~ #2 ~ #3 ~ #4 ~ #5 ~ #6 ~ #7 ~ #8 ~ #9 ~ 5190
5191     { 5191
5192         \if_meaning:w A #9 5192
5193         \__str_to_other_fast_end:w 5193
5194         \fi: 5194
5195         #1 * #2 * #3 * #4 * #5 * #6 * #7 * #8 * #9 5195
5196         \__str_to_other_fast_loop:w * 5196
5197     } 5197
5198     \cs_new:Npn \__str_to_other_fast_end:w #1 * A #2 \s__str_stop {#1} 5198
```

5199	}	5199
5200	\cs_new_eq:NN __str_sep: __kernel_int_sep:	5200
5201	\cs_new:Npn \str_item:Nn { \exp_args:No \str_item:nn }	5201
5202	\cs_generate_variant:Nn \str_item:Nn { c }	5202
5203	\cs_new:Npn \str_item:nn #1#2	5203
5204	{	5204
5205	\exp_args:Nf \tl_to_str:n	5205
5206	{	5206
5207	\exp_args:Nf __str_item:nn	5207
5208	{ __kernel_str_to_other:n {#1} } {#2}	5208
5209	}	5209
5210	}	5210
5211	\cs_new:Npn \str_item_ignore_spaces:nn #1	5211
5212	{ \exp_args:No __str_item:nn { \tl_to_str:n {#1} } }	5212
5213	\cs_new:Npn __str_item:nn #1#2	5213
5214	{	5214
5215	\exp_after:wN __str_item:w	5215
5216	\int_value:w \int_eval:n {#2} \exp_after:wN __str_sep:	5216
5217	\int_value:w __str_count:n {#1} __str_sep:	5217
5218	#1 \s__str_stop	5218
5219	}	5219
5220	\cs_new:Npn __str_item:w #1 __str_sep: #2 __str_sep:	5220
5221	{	5221
5222	\int_compare:nNnTF {#1} < 0	5222
5223	{	5223
5224	\int_compare:nNnTF {#1} < {-#2}	5224
5225	{ __str_use_none_delimit_by_s_stop:w }	5225
5226	{	5226
5227	\exp_after:wN __str_use_i_delimit_by_s_stop:nw	5227
5228	\exp:w \exp_after:wN __str_skip_exp_end:w	5228
5229	\int_value:w \int_eval:n { #1 + #2 } __str_sep:	5229
5230	}	5230
5231	}	5231
5232	{	5232
5233	\int_compare:nNnTF {#1} > {#2}	5233
5234	{ __str_use_none_delimit_by_s_stop:w }	5234
5235	{	5235
5236	\exp_after:wN __str_use_i_delimit_by_s_stop:nw	5236
5237	\exp:w __str_skip_exp_end:w #1 __str_sep: { }	5237
5238	}	5238
5239	}	5239
5240	}	5240
5241	\cs_new:Npn __str_skip_exp_end:w #1 __str_sep:	5241
5242	{	5242
5243	\if_int_compare:w #1 > 8 \exp_stop_f:	5243
5244	\exp_after:wN __str_skip_loop:wNNNNNNNN	5244

5245	\else:	5245
5246	\exp_after:wN __str_skip_end:w	5246
5247	\int_value:w \int_eval:w	5247
5248	\fi:	5248
5249	#1 __str_sep:	5249
5250	}	5250
5251	\cs_new:Npn __str_skip_loop:wNNNNNNNN #1 __str_sep: #2#3#4#5#6#7#8#9	5251
5252	{	5252
5253	\exp_after:wN __str_skip_exp_end:w	5253
5254	\int_value:w \int_eval:n { #1 - 8 } __str_sep:	5254
5255	}	5255
5256	\cs_new:Npn __str_skip_end:w #1 __str_sep:	5256
5257	{	5257
5258	\exp_after:wN __str_skip_end:NNNNNNNN	5258
5259	\if_case:w #1 \exp_stop_f: \or: \or: \or: \or: \or: \or: \or:	5259
5260	}	5260
5261	\cs_new:Npn __str_skip_end:NNNNNNNN #1#2#3#4#5#6#7#8 { \fi: \exp_end: }	5261
5262	\cs_new:Npn \str_range:Nnn { \exp_args:No \str_range:nnn }	5262
5263	\cs_generate_variant:Nn \str_range:Nnn { c }	5263
5264	\cs_new:Npn \str_range:nnn #1#2#3	5264
5265	{	5265
5266	\exp_args:Nf \tl_to_str:n	5266
5267	{	5267
5268	\exp_args:Nf __str_range:nnn	5268
5269	{ __kernel_str_to_other:n {#1} } {#2} {#3}	5269
5270	}	5270
5271	}	5271
5272	\cs_new:Npn \str_range_ignore_spaces:nnn #1	5272
5273	{ \exp_args:No __str_range:nnn { \tl_to_str:n {#1} } }	5273
5274	\cs_new:Npn __str_range:nnn #1#2#3	5274
5275	{	5275
5276	\exp_after:wN __str_range:w	5276
5277	\int_value:w __str_count:n {#1} \exp_after:wN __str_sep:	5277
5278	\int_value:w \int_eval:n { (#2) - 1 } \exp_after:wN __str_sep:	5278
5279	\int_value:w \int_eval:n {#3} __str_sep:	5279
5280	#1 \s__str_stop	5280
5281	}	5281
5282	\cs_new:Npn __str_range:w #1 __str_sep: #2 __str_sep: #3 __str_sep:	5282
5283	{	5283
5284	\exp_args:Nf __str_range:nnw	5284
5285	{ __str_range_normalize:nn {#2} {#1} }	5285
5286	{ __str_range_normalize:nn {#3} {#1} }	5286
5287	}	5287
5288	\cs_new:Npn __str_range:nnw #1#2	5288
5289	{	5289
5290	\exp_after:wN __str_collect_delimit_by_q_stop:w	5290

5291	\int_value:w \int_eval:n { #2 - #1 } \exp_after:wN __str_sep:	5291
5292	\exp:w __str_skip_exp_end:w #1 __str_sep:	5292
5293	}	5293
5294	\cs_new:Npn __str_range_normalize:nn #1#2	5294
5295	{	5295
5296	\int_eval:n	5296
5297	{	5297
5298	\if_int_compare:w #1 < \c_zero_int	5298
5299	\if_int_compare:w #1 < -#2 \exp_stop_f:	5299
5300	0	5300
5301	\else:	5301
5302	#1 + #2 + 1	5302
5303	\fi:	5303
5304	\else:	5304
5305	\if_int_compare:w #1 < #2 \exp_stop_f:	5305
5306	#1	5306
5307	\else:	5307
5308	#2	5308
5309	\fi:	5309
5310	\fi:	5310
5311	}	5311
5312	}	5312
5313	\cs_new:Npn __str_collect_delimit_by_q_stop:w #1 __str_sep:	5313
5314	{ __str_collect_loop:wn #1 __str_sep: { } }	5314
5315	\cs_new:Npn __str_collect_loop:wn #1 __str_sep:	5315
5316	{	5316
5317	\if_int_compare:w #1 > 7 \exp_stop_f:	5317
5318	\exp_after:wN __str_collect_loop:wnNNNNNNN	5318
5319	\else:	5319
5320	\exp_after:wN __str_collect_end:wn	5320
5321	\fi:	5321
5322	#1 __str_sep:	5322
5323	}	5323
5324	\cs_new:Npn __str_collect_loop:wnNNNNNNN #1 __str_sep: #2 #3#4#5#6#7#8#9	5324
5325	{	5325
5326	\exp_after:wN __str_collect_loop:wn	5326
5327	\int_value:w \int_eval:n { #1 - 7 } __str_sep:	5327
5328	{ #2 #3#4#5#6#7#8#9 }	5328
5329	}	5329
5330	\cs_new:Npn __str_collect_end:wn #1 __str_sep:	5330
5331	{	5331
5332	\exp_after:wN __str_collect_end:nnnnnnnnw	5332
5333	\if_case:w \if_int_compare:w #1 > \c_zero_int	5333
5334	#1 \else: 0 \fi: \exp_stop_f:	5334
5335	\or: \or: \or: \or: \or: \or: \fi:	5335
5336	}	5336

```
5337 \cs_new:Npn \__str_collect_end:nnnnnnnnnw #1#2#3#4#5#6#7#8 #9 \s__str_stop 5337
5338 { #1#2#3#4#5#6#7#8 } 5338
5339 \cs_new:Npn \str_count_spaces:N 5339
5340 { \exp_args:No \str_count_spaces:n } 5340
5341 \cs_generate_variant:Nn \str_count_spaces:N { c } 5341
5342 \cs_new:Npn \str_count_spaces:n #1 5342
5343 { 5343
5344 \int_eval:n 5344
5345 { 5345
5346 \exp_after:wN \__str_count_spaces_loop:w 5346
5347 \tl_to_str:n {#1} ~ 5347
5348 X 7 ~ X 6 ~ X 5 ~ X 4 ~ X 3 ~ X 2 ~ X 1 ~ X 0 ~ X -1 ~ 5348
5349 \s__str_stop 5349
5350 } 5350
5351 } 5351
5352 \cs_new:Npn \__str_count_spaces_loop:w #1~#2~#3~#4~#5~#6~#7~#8~#9~ 5352
5353 { 5353
5354 \if_meaning:w X #9 5354
5355 \__str_use_i_delimit_by_s_stop:nw 5355
5356 \fi: 5356
5357 9 + \__str_count_spaces_loop:w 5357
5358 } 5358
5359 \cs_new:Npn \str_count:N { \exp_args:No \str_count:n } 5359
5360 \cs_generate_variant:Nn \str_count:N { c } 5360
5361 \cs_new:Npn \str_count:n #1 5361
5362 { 5362
5363 \__str_count_aux:n 5363
5364 { 5364
5365 \str_count_spaces:n {#1} 5365
5366 + \exp_after:wN \__str_count_loop:NNNNNNNNN \tl_to_str:n {#1} 5366
5367 } 5367
5368 } 5368
5369 \cs_new:Npn \__str_count:n #1 5369
5370 { 5370
5371 \__str_count_aux:n 5371
5372 { \__str_count_loop:NNNNNNNNN #1 } 5372
5373 } 5373
5374 \cs_new:Npn \str_count_ignore_spaces:n #1 5374
5375 { 5375
5376 \__str_count_aux:n 5376
5377 { \exp_after:wN \__str_count_loop:NNNNNNNNN \tl_to_str:n {#1} } 5377
5378 } 5378
5379 \cs_new:Npn \__str_count_aux:n #1 5379
5380 { 5380
5381 \int_eval:n 5381
5382 { 5382
```

```
5383         #1 5383
5384         { X 8 } { X 7 } { X 6 } 5384
5385         { X 5 } { X 4 } { X 3 } 5385
5386         { X 2 } { X 1 } { X 0 } 5386
5387         \s__str_stop 5387
5388     } 5388
5389 } 5389
5390 \cs_new:Npn \__str_count_loop:NNNNNNNNN #1#2#3#4#5#6#7#8#9 5390
5391 { 5391
5392     \if_meaning:w X #9 5392
5393     \exp_after:wN \__str_use_none_delimit_by_s_stop:w 5393
5394     \fi: 5394
5395     9 + \__str_count_loop:NNNNNNNNN 5395
5396 } 5396
5397 \cs_new:Npn \str_head:N { \exp_args:No \str_head:n } 5397
5398 \cs_generate_variant:Nn \str_head:N { c } 5398
5399 \cs_new:Npn \str_head:n #1 5399
5400 { 5400
5401     \exp_after:wN \__str_head:w 5401
5402     \tl_to_str:n {#1} 5402
5403     { { } } ~ \s__str_stop 5403
5404 } 5404
5405 \cs_new:Npn \__str_head:w #1 ~ % 5405
5406 { \__str_use_i_delimit_by_s_stop:nw #1 { ~ } } 5406
5407 \cs_new:Npn \str_head_ignore_spaces:n #1 5407
5408 { 5408
5409     \exp_after:wN \__str_use_i_delimit_by_s_stop:nw 5409
5410     \tl_to_str:n {#1} { } \s__str_stop 5410
5411 } 5411
5412 \cs_new:Npn \str_tail:N { \exp_args:No \str_tail:n } 5412
5413 \cs_generate_variant:Nn \str_tail:N { c } 5413
5414 \cs_new:Npn \str_tail:n #1 5414
5415 { 5415
5416     \exp_after:wN \__str_tail_auxi:w 5416
5417     \reverse_if:N \if_charcode:w 5417
5418     \scan_stop: \tl_to_str:n {#1} X X \s__str_stop 5418
5419 } 5419
5420 \cs_new:Npn \__str_tail_auxi:w #1 X #2 \s__str_stop { \fi: #1 } 5420
5421 \cs_new:Npn \str_tail_ignore_spaces:n #1 5421
5422 { 5422
5423     \exp_after:wN \__str_tail_auxii:w 5423
5424     \tl_to_str:n {#1} \s__str_mark \s__str_mark \s__str_stop 5424
5425 } 5425
5426 \cs_new:Npn \__str_tail_auxii:w #1 #2 \s__str_mark #3 \s__str_stop { #2 } 5426
5427 \cs_new:Npn \str_casefold:n #1 { \__str_change_case:nn {#1} { casefold } } 5427
5428 \cs_new:Npn \str_lowercase:n #1 { \__str_change_case:nn {#1} { lowercase } } 5428
```

```

5429 \cs_new:Npn \str_uppercase:n #1 { \__str_change_case:nn {#1} { uppercase } } 5429
5430 \cs_generate_variant:Nn \str_casefold:n { V } 5430
5431 \cs_generate_variant:Nn \str_lowercase:n { f } 5431
5432 \cs_generate_variant:Nn \str_uppercase:n { f } 5432
5433 \cs_new:Npn \__str_change_case:nn #1 5433
5434 { 5434
5435     \exp_after:wN \__str_change_case_aux:nn \exp_after:wN 5435
5436     { \tl_to_str:n {#1} } 5436
5437 } 5437
5438 \cs_new:Npn \__str_change_case_aux:nn #1#2 5438
5439 { 5439
5440     \__str_change_case_loop:nw {#2} #1 \q_str_recursion_tail \q_str_recursion_stop 5440
5441     \__str_change_case_result:n { } 5441
5442 } 5442
5443 \cs_new:Npn \__str_change_case_output:nw #1#2 \__str_change_case_result:n #3 5443
5444 { #2 \__str_change_case_result:n { #3 #1 } } 5444
5445 \cs_generate_variant:Nn \__str_change_case_output:nw { f } 5445
5446 \cs_new:Npn \__str_change_case_end:wn #1 \__str_change_case_result:n #2 5446
5447 { \tl_to_str:n {#2} } 5447
5448 \cs_new:Npn \__str_change_case_loop:nw #1#2 \q_str_recursion_stop 5448
5449 { 5449
5450     \tl_if_head_is_space:nTF {#2} 5450
5451     { \__str_change_case_space:n } 5451
5452     { \__str_change_case_char:nN } 5452
5453     {#1} #2 \q_str_recursion_stop 5453
5454 } 5454
5455 \exp_last_unbraced:NNNNNo 5455
5456 \cs_new:Npn \__str_change_case_space:n #1 \c_space_tl 5456
5457 { 5457
5458     \__str_change_case_output:nw { ~ } 5458
5459     \__str_change_case_loop:nw {#1} 5459
5460 } 5460
5461 \cs_new:Npn \__str_change_case_char:nN #1#2 5461
5462 { 5462
5463     \__str_if_recursion_tail_stop_do:Nn #2 5463
5464     { \__str_change_case_end:wn } 5464
5465     \__str_change_case_codepoint:nN {#1} #2 5465
5466 } 5466
5467 \if_int_compare:w 0 5467
5468 \cs_if_exist:NT \tex_XeTeXversion:D { 1 } 5468
5469 \cs_if_exist:NT \tex luatexversion:D { 1 } 5469
5470 > 0 \exp_stop_f: 5470
5471 \cs_new:Npn \__str_change_case_codepoint:nN #1#2 5471
5472 { \__str_change_case_char:fnn { \int_eval:n {`#2} } {#1} {#2} } 5472
5473 \else: 5473
5474 \cs_new:Npe \__str_change_case_codepoint:nN #1#2 5474

```

```

5475 { 5475
5476 \exp_not:N \int_compare:nNnTF {\`#2} > { "80 } 5476
5477 { 5477
5478 \cs_if_exist:NTF \tex_pdfversion:D 5478
5479 { \exp_not:N \__str_change_case_char_auxi:nN } 5479
5480 { 5480
5481 \exp_not:N \int_compare:nNnTF {\`#2} > { "FF } 5481
5482 { \exp_not:N \__str_change_case_char_auxii:nN } 5482
5483 { \exp_not:N \__str_change_case_char_auxi:nN } 5483
5484 } 5484
5485 } 5485
5486 { \exp_not:N \__str_change_case_char_auxii:nN } 5486
5487 {\#1} #2 5487
5488 } 5488
5489 \cs_new:Npn \__str_change_case_char_auxi:nN #1#2 5489
5490 { 5490
5491 \int_compare:nNnTF {\`#2} < { "E0 } 5491
5492 { \__str_change_case_codepoint:nNN } 5492
5493 { 5493
5494 \int_compare:nNnTF {\`#2} < { "F0 } 5494
5495 { \__str_change_case_codepoint:nNNN } 5495
5496 { \__str_change_case_codepoint:nNNNNN } 5496
5497 } 5497
5498 {\#1} #2 5498
5499 } 5499
5500 \cs_new:Npn \__str_change_case_char_auxii:nN #1#2 5500
5501 { \__str_change_case_char:fnn { \int_eval:n {\`#2} } {\#1} {\#2} } 5501
5502 \cs_new:Npn \__str_change_case_codepoint:nNN #1#2#3 5502
5503 { 5503
5504 \__str_change_case_char:fnn 5504
5505 { \int_eval:n { (\`#2 - "C0) * "40 + \`#3 - "80 } } 5505
5506 {\#1} {\#2#3} 5506
5507 } 5507
5508 \cs_new:Npn \__str_change_case_codepoint:nNNN #1#2#3#4 5508
5509 { 5509
5510 \__str_change_case_char:fnn 5510
5511 { 5511
5512 \int_eval:n 5512
5513 { (\`#2 - "E0) * "1000 + (\`#3 - "80) * "40 + \`#4 - "80 } 5513
5514 } 5514
5515 {\#1} {\#2#3#4} 5515
5516 } 5516
5517 \cs_new:Npn \__str_change_case_codepoint:nNNNN #1#2#3#4#5 5517
5518 { 5518
5519 \__str_change_case_char:fnn 5519
5520 { 5520

```



```

5521         \int_eval:n
5522     {
5523         (`#2 - "F0) * "40000
5524         + (`#3 - "80) * "1000
5525         + (`#4 - "80) * "40
5526         + `#5 - "80
5527     }
5528 }
5529 {#1} {#2#3#4#5}
5530 }
5531 \fi:
5532 \cs_new:Npn \__str_change_case_char:nnn #1#2#3
5533 {
5534     \__str_change_case_output:fw
5535     {
5536         \exp_args:Ne \__str_change_case_char_aux:nnn
5537             { \__kernel_codepoint_case:nn {#2} {#1} } {#1} {#3}
5538     }
5539     \__str_change_case_loop:nw {#2}
5540 }
5541 \cs_generate_variant:Nn \__str_change_case_char:nnn { f }
5542 \cs_new:Npn \__str_change_case_char_aux:nnn #1#2#3
5543 {
5544     \use:e { \__str_change_case_char:nnnnn #1 {#2} {#3} }
5545 }
5546 \cs_new:Npn \__str_change_case_char:nnnnn #1#2#3#4#5
5547 {
5548     \int_compare:nNnTF {#1} = {#4}
5549     { \tl_to_str:n {#5} }
5550     {
5551         \codepoint_str_generate:n {#1}
5552         \tl_if_blank:nF {#2}
5553         {
5554             \codepoint_str_generate:n {#2}
5555             \tl_if_blank:nF {#3}
5556             { \codepoint_str_generate:n {#3} }
5557         }
5558     }
5559 }
5560 \cs_new:Npn \str_mdfive_hash:n #1 { \tex_mdffivesum:D { \tl_to_str:n {#1} } }
5561 \cs_new:Npn \str_mdfive_hash:e #1 { \tex_mdffivesum:D {#1} }
5562 \str_const:Ne \c_ampersand_str { \cs_to_str:N \& }
5563 \str_const:Ne \c_at_sign_str { \cs_to_str:N \@ }
5564 \str_const:Ne \c_backslash_str { \cs_to_str:N \ }
5565 \str_const:Ne \c_left_brace_str { \cs_to_str:N \{ }
5566 \str_const:Ne \c_right_brace_str { \cs_to_str:N \} }

```

```
5567 \str_const:Ne \c_circumflex_str { \cs_to_str:N \^ } 5567
5568 \str_const:Ne \c_colon_str { \cs_to_str:N \: } 5568
5569 \str_const:Ne \c_dollar_str { \cs_to_str:N \$ } 5569
5570 \str_const:Ne \c_hash_str { \cs_to_str:N \# } 5570
5571 \str_const:Ne \c_percent_str { \cs_to_str:N \% } 5571
5572 \str_const:Ne \c_tilde_str { \cs_to_str:N \~ } 5572
5573 \str_const:Ne \c_underscore_str { \cs_to_str:N \_ } 5573
5574 \str_const:Ne \c_zero_str { 0 } 5574
5575 \cs_new_eq:NN \c_empty_str \c_empty_tl 5575
5576 \str_new:N \l_tmpa_str 5576
5577 \str_new:N \l_tmpb_str 5577
5578 \str_new:N \g_tmpa_str 5578
5579 \str_new:N \g_tmpb_str 5579
5580 \cs_new_eq:NN \str_show:n \tl_show:n 5580
5581 \cs_new_protected:Npn \str_show:N #1 5581
5582 { 5582
5583     \__kernel_chk_tl_type:NnnT #1 { str } { \tl_to_str:N #1 } 5583
5584     { \tl_show:N #1 } 5584
5585 } 5585
5586 \cs_generate_variant:Nn \str_show:N { c } 5586
5587 \cs_new_eq:NN \str_log:n \tl_log:n 5587
5588 \cs_new_protected:Npn \str_log:N #1 5588
5589 { 5589
5590     \__kernel_chk_tl_type:NnnT #1 { str } { \tl_to_str:N #1 } 5590
5591     { \tl_log:N #1 } 5591
5592 } 5592
5593 \cs_generate_variant:Nn \str_log:N { c } 5593
5594 %% File: l3seq.dtx 5594
5595 \scan_new:N \s__seq 5595
5596 \scan_new:N \s__seq_mark 5596
5597 \scan_new:N \s__seq_stop 5597
5598 \cs_new:Npn \__seq_item:n 5598
5599 { 5599
5600     \msg_expandable_error:nn { seq } { misused } 5600
5601     \use_none:n 5601
5602 } 5602
5603 \tl_new:N \l__seq_internal_a_tl 5603
5604 \tl_new:N \l__seq_internal_b_tl 5604
5605 \cs_new_eq:NN \__seq_tmp:w ? 5605
5606 \tl_const:Nn \c_empty_seq { \s__seq } 5606
5607 \cs_new_protected:Npn \seq_new:N #1 5607
5608 { 5608
5609     \__kernel_chk_if_free_cs:N #1 5609
5610     \cs_gset_eq:NN #1 \c_empty_seq 5610
5611 } 5611
5612 \cs_generate_variant:Nn \seq_new:N { c } 5612
```

5613	\cs_new_protected:Npn \seq_clear:N #1	5613
5614	{ \seq_set_eq:NN #1 \c_empty_seq }	5614
5615	\cs_generate_variant:Nn \seq_clear:N { c }	5615
5616	\cs_new_protected:Npn \seq_gclear:N #1	5616
5617	{ \seq_gset_eq:NN #1 \c_empty_seq }	5617
5618	\cs_generate_variant:Nn \seq_gclear:N { c }	5618
5619	\cs_new_protected:Npn \seq_clear_new:N #1	5619
5620	{ \seq_if_exist:NTF #1 { \seq_clear:N #1 } { \seq_new:N #1 } }	5620
5621	\cs_generate_variant:Nn \seq_clear_new:N { c }	5621
5622	\cs_new_protected:Npn \seq_gclear_new:N #1	5622
5623	{ \seq_if_exist:NTF #1 { \seq_gclear:N #1 } { \seq_new:N #1 } }	5623
5624	\cs_generate_variant:Nn \seq_gclear_new:N { c }	5624
5625	\cs_new_eq:NN \seq_set_eq:NN \tl_set_eq:NN	5625
5626	\cs_new_eq:NN \seq_set_eq:Nc \tl_set_eq:Nc	5626
5627	\cs_new_eq:NN \seq_set_eq:cN \tl_set_eq:cN	5627
5628	\cs_new_eq:NN \seq_set_eq:cc \tl_set_eq:cc	5628
5629	\cs_new_eq:NN \seq_gset_eq:NN \tl_gset_eq:NN	5629
5630	\cs_new_eq:NN \seq_gset_eq:Nc \tl_gset_eq:Nc	5630
5631	\cs_new_eq:NN \seq_gset_eq:cN \tl_gset_eq:cN	5631
5632	\cs_new_eq:NN \seq_gset_eq:cc \tl_gset_eq:cc	5632
5633	\cs_new_protected:Npn \seq_set_from_clist:NN #1#2	5633
5634	{	5634
5635	__kernel_tl_set:Nx #1	5635
5636	{ \s__seq \clist_map_function:NN #2 __seq_wrap_item:n }	5636
5637	}	5637
5638	\cs_new_protected:Npn \seq_set_from_clist:Nn #1#2	5638
5639	{	5639
5640	__kernel_tl_set:Nx #1	5640
5641	{ \s__seq \clist_map_function:nN {#2} __seq_wrap_item:n }	5641
5642	}	5642
5643	\cs_new_protected:Npn \seq_gset_from_clist:NN #1#2	5643
5644	{	5644
5645	__kernel_tl_gset:Nx #1	5645
5646	{ \s__seq \clist_map_function:NN #2 __seq_wrap_item:n }	5646
5647	}	5647
5648	\cs_new_protected:Npn \seq_gset_from_clist:Nn #1#2	5648
5649	{	5649
5650	__kernel_tl_gset:Nx #1	5650
5651	{ \s__seq \clist_map_function:nN {#2} __seq_wrap_item:n }	5651
5652	}	5652
5653	\cs_generate_variant:Nn \seq_set_from_clist:NN { Nc }	5653
5654	\cs_generate_variant:Nn \seq_set_from_clist:NN { c , cc }	5654
5655	\cs_generate_variant:Nn \seq_set_from_clist:Nn { c }	5655
5656	\cs_generate_variant:Nn \seq_gset_from_clist:NN { Nc }	5656
5657	\cs_generate_variant:Nn \seq_gset_from_clist:NN { c , cc }	5657
5658	\cs_generate_variant:Nn \seq_gset_from_clist:Nn { c }	5658

```

5659 \cs_new_protected:Npn \seq_const_from_clist:Nn #1#2 5659
5660 { 5660
5661     \tl_const:Nc #1 5661
5662     { \s__seq \clist_map_function:nN {#2} \__seq_wrap_item:n } 5662
5663 } 5663
5664 \cs_generate_variant:Nn \seq_const_from_clist:Nn { c } 5664
5665 \cs_new_protected:Npn \seq_set_split:Nnn 5665
5666 { \__seq_set_split:NNNnn \__kernel_tl_set:Nx \tl_trim_spaces:n } 5666
5667 \cs_new_protected:Npn \seq_gset_split:Nnn 5667
5668 { \__seq_set_split:NNNnn \__kernel_tl_gset:Nx \tl_trim_spaces:n } 5668
5669 \cs_new_protected:Npn \seq_set_split_keep_spaces:Nnn 5669
5670 { \__seq_set_split:NNNnn \__kernel_tl_set:Nx \exp_not:n } 5670
5671 \cs_new_protected:Npn \seq_gset_split_keep_spaces:Nnn 5671
5672 { \__seq_set_split:NNNnn \__kernel_tl_gset:Nx \exp_not:n } 5672
5673 \cs_new_protected:Npn \__seq_set_split:NNNnn #1#2#3#4#5 5673
5674 { 5674
5675     \tl_if_empty:nTF {#4} 5675
5676     { 5676
5677         \tl_set:Nn \l__seq_internal_a_tl 5677
5678         { \tl_map_function:nN {#5} \__seq_wrap_item:n } 5678
5679     } 5679
5680     { 5680
5681         \tl_set:Nn \l__seq_internal_a_tl 5681
5682         { 5682
5683             \__seq_set_split:Nw #2 \prg_do_nothing: 5683
5684             #5 5684
5685             \__seq_set_split_end: 5685
5686         } 5686
5687         \tl_replace_all:Nnn \l__seq_internal_a_tl {#4} 5687
5688         { 5688
5689             \__seq_set_split_end: 5689
5690             \__seq_set_split:Nw #2 \prg_do_nothing: 5690
5691         } 5691
5692         \__kernel_tl_set:Nx \l__seq_internal_a_tl { \l__seq_internal_a_tl } 5692
5693     } 5693
5694     #1 #3 { \s__seq \l__seq_internal_a_tl } 5694
5695 } 5695
5696 \cs_new:Npn \__seq_set_split:Nw #1#2 \__seq_set_split_end: 5696
5697 { 5697
5698     \exp_not:N \__seq_set_split:w 5698
5699     \exp_args:No #1 {#2} 5699
5700     \exp_not:N \__seq_set_split_end: 5700
5701 } 5701
5702 \cs_new:Npn \__seq_set_split:w #1 \__seq_set_split_end: 5702
5703 { \__seq_wrap_item:n {#1} } 5703
5704 \cs_generate_variant:Nn \seq_set_split:Nnn { NV , NnV , NVV , Nne , Nee } 5704

```

```
5705 \cs_generate_variant:Nn \seq_set_split:Nnn { Nnx , Nxx } 5705
5706 \cs_generate_variant:Nn \seq_gset_split:Nnn { NV , NnV , NVV , Nne , Nee } 5706
5707 \cs_generate_variant:Nn \seq_gset_split:Nnn { Nnx , Nxx } 5707
5708 \cs_generate_variant:Nn \seq_set_split_keep_spaces:Nnn { NnV } 5708
5709 \cs_generate_variant:Nn \seq_gset_split_keep_spaces:Nnn { NnV } 5709
5710 \cs_new_protected:Npn \seq_set_filter:NNn 5710
5711 { \__seq_set_filter:NNNn \__kernel_tl_set:Nx } 5711
5712 \cs_new_protected:Npn \seq_gset_filter:NNn 5712
5713 { \__seq_set_filter:NNNn \__kernel_tl_gset:Nx } 5713
5714 \cs_new_protected:Npn \__seq_set_filter:NNNn #1#2#3#4 5714
5715 { 5715
5716 \__seq_push_item_def:n { \bool_if:nT {#4} { \__seq_wrap_item:n {##1} } } 5716
5717 #1 #2 { #3 } 5717
5718 \__seq_pop_item_def: 5718
5719 } 5719
5720 \cs_new_protected:Npn \seq_set_regex_extract_once:Nnn #1#2#3 5720
5721 { \regex_extract_once:nnN {#2} {#3} #1 } 5721
5722 \cs_generate_variant:Nn \seq_set_regex_extract_once:Nnn { c } 5722
5723 \cs_new_protected:Npn \seq_set_regex_extract_once:NNn #1#2#3 5723
5724 { \regex_extract_once:NnN #2 {#3} #1 } 5724
5725 \cs_generate_variant:Nn \seq_set_regex_extract_once:Nnn { c } 5725
5726 \cs_new_protected:Npn \seq_set_regex_extract_all:Nnn #1#2#3 5726
5727 { \regex_extract_all:nnN {#2} {#3} #1 } 5727
5728 \cs_generate_variant:Nn \seq_set_regex_extract_all:Nnn { c } 5728
5729 \cs_new_protected:Npn \seq_set_regex_extract_all:NNn #1#2#3 5729
5730 { \regex_extract_all:NnN #2 {#3} #1 } 5730
5731 \cs_generate_variant:Nn \seq_set_regex_extract_all:NNn { c } 5731
5732 \cs_new_protected:Npn \seq_set_regex_split:Nnn #1#2#3 5732
5733 { \regex_split:nnN {#2} {#3} #1 } 5733
5734 \cs_generate_variant:Nn \seq_set_regex_split:Nnn { c } 5734
5735 \cs_new_protected:Npn \seq_set_regex_split:NNn #1#2#3 5735
5736 { \regex_split:NnN #2 {#3} #1 } 5736
5737 \cs_generate_variant:Nn \seq_set_regex_split:NNn { c } 5737
5738 \group_begin: 5738
5739 \cs_set_protected:Npn \__seq_tmp:w #1#2#3 5739
5740 { 5740
5741 \cs_new_protected:cpe { seq_gset_regex_ #1 :N #2 n } ##1##2##3 5741
5742 { 5742
5743 \group_begin: 5743
5744 \seq_set_eq:NN \exp_not:N \l__seq_tmp_seq ##1 5744
5745 \exp_not:c { regex_ #1 :Nn #2 } 5745
5746 #3 {##2} {##3} \exp_not:N \l__seq_tmp_seq 5746
5747 \seq_gset_eq:NN ##1 \exp_not:N \l__seq_tmp_seq 5747
5748 \group_end: 5748
5749 } 5749
5750 \cs_generate_variant:cn { seq_gset_regex_ #1 :N #2 n } { c } 5750
```

```

5751 }
5752 \__seq_tmp:w { extract_once } n { }
5753 \__seq_tmp:w { extract_once } N \use:n
5754 \__seq_tmp:w { extract_all } n { }
5755 \__seq_tmp:w { extract_all } N \use:n
5756 \__seq_tmp:w { split } n { }
5757 \__seq_tmp:w { split } N \use:n
5758 \group_end:
5759 \cs_new_protected:Npn \seq_concat:NNN #1#2#3
5760 { \tl_set:Nf #1 { \exp_after:wN \use_i:nn \exp_after:wN #2 #3 } }
5761 \cs_new_protected:Npn \seq_gconcat:NNN #1#2#3
5762 { \tl_gset:Nf #1 { \exp_after:wN \use_i:nn \exp_after:wN #2 #3 } }
5763 \cs_generate_variant:Nn \seq_concat:NNN { ccc }
5764 \cs_generate_variant:Nn \seq_gconcat:NNN { ccc }
5765 \prg_new_eq_conditional:NNn \seq_if_exist:N \cs_if_exist:N
5766 { TF , T , F , p }
5767 \prg_new_eq_conditional:NNn \seq_if_exist:c \cs_if_exist:c
5768 { TF , T , F , p }
5769 \cs_new_protected:Npn \seq_put_left:Nn #1#2
5770 {
5771 \__kernel_tl_set:Nx #1
5772 {
5773 \exp_not:n { \s__seq \__seq_item:n {#2} }
5774 \exp_not:f { \exp_after:wN \__seq_put_left_aux:w #1 }
5775 }
5776 }
5777 \cs_new_protected:Npn \seq_gput_left:Nn #1#2
5778 {
5779 \__kernel_tl_gset:Nx #1
5780 {
5781 \exp_not:n { \s__seq \__seq_item:n {#2} }
5782 \exp_not:f { \exp_after:wN \__seq_put_left_aux:w #1 }
5783 }
5784 }
5785 \cs_new:Npn \__seq_put_left_aux:w \s__seq { \exp_stop_f: }
5786 \cs_generate_variant:Nn \seq_put_left:Nn { NV , Nv , Ne , No , Nx }
5787 \cs_generate_variant:Nn \seq_put_left:Nn { c , cV , cv , ce , co , cx }
5788 \cs_generate_variant:Nn \seq_gput_left:Nn { NV , Nv , Ne , No , Nx }
5789 \cs_generate_variant:Nn \seq_gput_left:Nn { c , cV , cv , ce , co , cx }
5790 \cs_new_protected:Npn \seq_put_right:Nn #1#2
5791 { \tl_put_right:Nn #1 { \__seq_item:n {#2} } }
5792 \cs_new_protected:Npn \seq_gput_right:Nn #1#2
5793 { \tl_gput_right:Nn #1 { \__seq_item:n {#2} } }
5794 \cs_generate_variant:Nn \seq_put_right:Nn { NV , Nv , Ne , No , Nx }
5795 \cs_generate_variant:Nn \seq_put_right:Nn { c , cV , cv , ce , co , cx }
5796 \cs_generate_variant:Nn \seq_gput_right:Nn { NV , Nv , Ne , No , Nx }

```



```
5797 \cs_generate_variant:Nn \seq_gput_right:Nn { c , cV , cv , ce , co , cx } 5797
5798 \cs_new:Npn \__seq_wrap_item:n #1 { \exp_not:n { \__seq_item:n {#1} } } 5798
5799 \seq_new:N \l__seq_tmp_seq 5799
5800 \cs_new_protected:Npn \seq_remove_duplicates:N 5800
5801 { \__seq_remove_duplicates:NN \seq_set_eq:NN } 5801
5802 \cs_new_protected:Npn \seq_gremove_duplicates:N 5802
5803 { \__seq_remove_duplicates:NN \seq_gset_eq:NN } 5803
5804 \cs_new_protected:Npn \__seq_remove_duplicates:NN #1#2 5804
5805 { 5805
5806 \seq_clear:N \l__seq_tmp_seq 5806
5807 \seq_map_inline:Nn #2 5807
5808 { 5808
5809 \seq_if_in:NnF \l__seq_tmp_seq {##1} 5809
5810 { \seq_put_right:Nn \l__seq_tmp_seq {##1} } 5810
5811 } 5811
5812 #1 #2 \l__seq_tmp_seq 5812
5813 } 5813
5814 \cs_generate_variant:Nn \seq_remove_duplicates:N { c } 5814
5815 \cs_generate_variant:Nn \seq_gremove_duplicates:N { c } 5815
5816 \cs_new_protected:Npn \seq_remove_all:Nn 5816
5817 { \__seq_remove_all_aux:NNn \__kernel_tl_set:Nx } 5817
5818 \cs_new_protected:Npn \seq_gremove_all:Nn 5818
5819 { \__seq_remove_all_aux:NNn \__kernel_tl_gset:Nx } 5819
5820 \cs_new_protected:Npn \__seq_remove_all_aux:NNn #1#2#3 5820
5821 { 5821
5822 \__seq_push_item_def:n 5822
5823 { 5823
5824 \str_if_eq:nnT {##1} {#3} 5824
5825 { 5825
5826 \if_false: { \fi: } 5826
5827 \tl_set:Nn \l__seq_internal_b_tl {##1} 5827
5828 #1 #2 5828
5829 { \if_false: } \fi: 5829
5830 \exp_not:o {#2} 5830
5831 \tl_if_eq:NNT \l__seq_internal_a_tl \l__seq_internal_b_tl 5831
5832 { \use_none:nn } 5832
5833 } 5833
5834 \__seq_wrap_item:n {##1} 5834
5835 } 5835
5836 \tl_set:Nn \l__seq_internal_a_tl {#3} 5836
5837 #1 #2 {#2} 5837
5838 \__seq_pop_item_def: 5838
5839 } 5839
5840 \cs_generate_variant:Nn \seq_remove_all:Nn { NV , Ne , c , cV , ce } 5840
5841 \cs_generate_variant:Nn \seq_remove_all:Nn { Nx , cx } 5841
5842 \cs_generate_variant:Nn \seq_gremove_all:Nn { NV , Ne , c , cV , ce } 5842
```

```
5843 \cs_generate_variant:Nn \seq_gremove_all:Nn { Nx , cx } 5843
5844 \cs_new_eq:NN \__seq_int_eval:w \tex_numexpr:D 5844
5845 \cs_new_protected:Npn \seq_set_item:Nnn #1#2#3 5845
5846 { \__seq_set_item:NnnNN #1 {#2} {#3} \__kernel_tl_set:Nx \use_i:nn } 5846
5847 \cs_new_protected:Npn \seq_gset_item:Nnn #1#2#3 5847
5848 { \__seq_set_item:NnnNN #1 {#2} {#3} \__kernel_tl_gset:Nx \use_i:nn } 5848
5849 \cs_generate_variant:Nn \seq_set_item:Nnn { c } 5849
5850 \cs_generate_variant:Nn \seq_gset_item:Nnn { c } 5850
5851 \prg_new_protected_conditional:Npnn \seq_set_item:Nnn #1#2#3 { TF , T , F } 5851
5852 { \__seq_set_item:NnnNN #1 {#2} {#3} \__kernel_tl_set:Nx \use_ii:nn } 5852
5853 \prg_new_protected_conditional:Npnn \seq_gset_item:Nnn #1#2#3 { TF , T , F } 5853
5854 { \__seq_set_item:NnnNN #1 {#2} {#3} \__kernel_tl_gset:Nx \use_ii:nn } 5854
5855 \prg_generate_conditional_variant:Nnn \seq_set_item:Nnn { c } { TF , T , F } 5855
5856 \prg_generate_conditional_variant:Nnn \seq_gset_item:Nnn { c } { TF , T , F } 5856
5857 \cs_new_protected:Npn \__seq_set_item:NnnNN #1#2#3 5857
5858 { 5858
5859 \tl_set:Nn \l__seq_internal_a_tl { \__seq_item:n {#3} } 5859
5860 \exp_args:Nff \__seq_set_item:nnNNNN 5860
5861 { \int_eval:n {#2} } { \seq_count:N #1 } #1 \use_none:nn 5861
5862 } 5862
5863 \cs_new_protected:Npn \__seq_set_item:nnNNNN #1#2 5863
5864 { 5864
5865 \int_compare:nNnTF {#1} > 0 5865
5866 { \int_compare:nNnF {#1} > {#2} { \__seq_set_item:nNnnNNNN { #1 - 1 } } } 5866
5867 { 5867
5868 \int_compare:nNnF {#1} < {-#2} 5868
5869 { 5869
5870 \int_compare:nNnF {#1} = 0 5870
5871 { \__seq_set_item:nNnnNNNN { #2 + #1 } } 5871
5872 } 5872
5873 } 5873
5874 \__seq_set_item_false:nnNNNN {#1} {#2} 5874
5875 } 5875
5876 \cs_new_protected:Npn \__seq_set_item_false:nnNNNN #1#2#3#4#5#6 5876
5877 { 5877
5878 #6 5878
5879 { 5879
5880 \msg_error:nneee { seq } { item-too-large } 5880
5881 { \token_to_str:N #3 } {#2} {#1} 5881
5882 } 5882
5883 { \prg_return_false: } 5883
5884 } 5884
5885 \cs_new_protected:Npn \__seq_set_item:nNnnNNNN #1#2#3#4#5#6#7#8 5885
5886 { 5886
5887 #7 #5 5887
5888 { 5888
```

```

5889         \s__seq                                     5889
5890         \exp_after:wN \__seq_set_item:wn           5890
5891         \int_value:w \__seq_int_eval:w #1          5891
5892         #5 \s__seq_stop #6                          5892
5893     }                                                5893
5894     #8 { } { \prg_return_true: }                    5894
5895 }                                                    5895
5896 \cs_new:Npn \__seq_set_item:wn #1 \__seq_item:n #2 5896
5897 {                                                  5897
5898     \if_meaning:w 0 #1 \__seq_set_item_end:w \fi:  5898
5899     \exp_not:n { \__seq_item:n {#2} }              5899
5900     \exp_after:wN \__seq_set_item:wn              5900
5901     \int_value:w \__seq_int_eval:w #1 - 1 \s__seq  5901
5902 }                                                  5902
5903 \cs_new:Npn \__seq_set_item_end:w #1 \exp_not:n #2 #3 \s__seq #4 \s__seq_stop #5 5903
5904 {                                                  5904
5905     #1                                              5905
5906     \exp_not:o \l__seq_internal_a_tl              5906
5907     \exp_not:n {#4}                                5907
5908     #5 #2                                           5908
5909 }                                                  5909
5910 \cs_new_protected:Npn \seq_reverse:N               5910
5911 { \__seq_reverse:NN \__kernel_tl_set:Nx }         5911
5912 \cs_new_protected:Npn \seq_greverse:N              5912
5913 { \__seq_reverse:NN \__kernel_tl_gset:Nx }        5913
5914 \cs_new_protected:Npn \__seq_reverse:NN #1 #2      5914
5915 {                                                  5915
5916     \cs_set_eq:NN \__seq_tmp:w \__seq_item:n      5916
5917     \cs_set_eq:NN \__seq_item:n \__seq_reverse_item:nwn 5917
5918     #1 #2 { #2 \exp_not:n { } }                   5918
5919     \cs_set_eq:NN \__seq_item:n \__seq_tmp:w      5919
5920 }                                                  5920
5921 \cs_new:Npn \__seq_reverse_item:nwn #1 #2 \exp_not:n #3 5921
5922 {                                                  5922
5923     #2                                              5923
5924     \exp_not:n { \__seq_item:n {#1} #3 }          5924
5925 }                                                  5925
5926 \cs_generate_variant:Nn \seq_reverse:N { c }      5926
5927 \cs_generate_variant:Nn \seq_greverse:N { c }     5927
5928 \prg_new_conditional:Npnn \seq_if_empty:N #1 { p , T , F , TF } 5928
5929 {                                                  5929
5930     \if_meaning:w #1 \c_empty_seq                 5930
5931     \prg_return_true:                             5931
5932     \else:                                         5932
5933     \prg_return_false:                            5933
5934     \fi:                                           5934

```

```

5935 }
5936 \prg_generate_conditional_variant:Nnn \seq_if_empty:N
5937 { c } { p , T , F , TF }
5938 \seq_new:N \g__seq_internal_seq
5939 \cs_new_protected:Npn \seq_shuffle:N { \__seq_shuffle:NN \seq_set_eq:NN }
5940 \cs_new_protected:Npn \seq_gshuffle:N { \__seq_shuffle:NN \seq_gset_eq:NN }
5941 \cs_new_protected:Npn \__seq_shuffle:NN #1#2
5942 {
5943   \int_compare:nNnTF { \seq_count:N #2 } > \c_max_register_int
5944   {
5945     \msg_error:nne { seq } { shuffle-too-large }
5946     { \token_to_str:N #2 }
5947   }
5948   {
5949     \group_begin:
5950     \int_zero:N \l__seq_internal_a_int
5951     \__seq_push_item_def:
5952     \cs_gset_eq:NN \__seq_item:n \__seq_shuffle_item:n
5953     #2
5954     \__seq_pop_item_def:
5955     \seq_gclear:N \g__seq_internal_seq
5956     \int_step_inline:nn \l__seq_internal_a_int
5957     {
5958       \seq_gput_right:Ne \g__seq_internal_seq
5959       { \tex_the:D \tex_toks:D ##1 }
5960     }
5961     \group_end:
5962     #1 #2 \g__seq_internal_seq
5963     \seq_gclear:N \g__seq_internal_seq
5964   }
5965 }
5966 \cs_new_protected:Npn \__seq_shuffle_item:n
5967 {
5968   \int_incr:N \l__seq_internal_a_int
5969   \int_set:Nn \l__seq_internal_b_int
5970   { 1 + \tex_uniformdeviate:D \l__seq_internal_a_int }
5971   \tex_toks:D \l__seq_internal_a_int
5972   = \tex_toks:D \l__seq_internal_b_int
5973   \tex_toks:D \l__seq_internal_b_int
5974 }
5975 \cs_generate_variant:Nn \seq_shuffle:N { c }
5976 \cs_generate_variant:Nn \seq_gshuffle:N { c }
5977 \prg_new_protected_conditional:Npnn \seq_if_in:Nn #1#2
5978 { T , F , TF }
5979 {
5980   \group_begin:

```

```

5981      \tl_set:Nn \l__seq_internal_a_tl {#2}
5982      \cs_set_protected:Npn \__seq_item:n ##1
5983      {
5984          \tl_set:Nn \l__seq_internal_b_tl {##1}
5985          \if_meaning:w \l__seq_internal_a_tl \l__seq_internal_b_tl
5986              \exp_after:wN \__seq_if_in:
5987          \fi:
5988      }
5989      #1
5990  \group_end:
5991  \prg_return_false:
5992  \prg_break_point:
5993  }
5994  \cs_new:Npn \__seq_if_in:
5995      { \prg_break:n { \group_end: \prg_return_true: } }
5996  \prg_generate_conditional_variant:Nnn \seq_if_in:Nn
5997      { NV , Nv , Ne , No , Nx , c , cV , cv , ce , co , cx } { T , F , TF }
5998  \cs_new_protected:Npn \__seq_pop:NNNN #1#2#3#4
5999      {
6000          \if_meaning:w #3 \c_empty_seq
6001              \tl_set:Nn #4 { \q_no_value }
6002          \else:
6003              #1#2#3#4
6004          \fi:
6005      }
6006  \cs_new_protected:Npn \__seq_pop_TF:NNNN #1#2#3#4
6007      {
6008          \if_meaning:w #3 \c_empty_seq
6009              % \tl_set:Nn #4 { \q_no_value }
6010              \prg_return_false:
6011          \else:
6012              #1#2#3#4
6013              \prg_return_true:
6014          \fi:
6015      }
6016  \cs_new_protected:Npn \seq_get_left:NN #1#2
6017      {
6018          \__kernel_tl_set:Nx #2
6019          {
6020              \exp_after:wN \__seq_get_left:wnw
6021              #1 \__seq_item:n { \q_no_value } \s__seq_stop
6022          }
6023      }
6024  \cs_new:Npn \__seq_get_left:wnw #1 \__seq_item:n #2#3 \s__seq_stop
6025      { \exp_not:n {#2} }
6026  \cs_generate_variant:Nn \seq_get_left:NN { c }

```

6027	\cs_new_protected:Npn \seq_pop_left:NN	6027
6028	{ __seq_pop:NNNN __seq_pop_left:NNN \tl_set:Nn }	6028
6029	\cs_new_protected:Npn \seq_gpop_left:NN	6029
6030	{ __seq_pop:NNNN __seq_pop_left:NNN \tl_gset:Nn }	6030
6031	\cs_new_protected:Npn __seq_pop_left:NNN #1#2#3	6031
6032	{ \exp_after:wN __seq_pop_left:wnwNNN #2 \s__seq_stop #1#2#3 }	6032
6033	\cs_new_protected:Npn __seq_pop_left:wnwNNN	6033
6034	#1 __seq_item:n #2#3 \s__seq_stop #4#5#6	6034
6035	{	6035
6036	#4 #5 { #1 #3 }	6036
6037	\tl_set:Nn #6 {#2}	6037
6038	}	6038
6039	\cs_generate_variant:Nn \seq_pop_left:NN { c }	6039
6040	\cs_generate_variant:Nn \seq_gpop_left:NN { c }	6040
6041	\cs_new_protected:Npn \seq_get_right:NN #1#2	6041
6042	{	6042
6043	__kernel_tl_set:Nx #2	6043
6044	{	6044
6045	\exp_after:wN \use_i_ii:nnn	6045
6046	\exp_after:wN __seq_get_right_loop:nw	6046
6047	\exp_after:wN \q_no_value	6047
6048	#1	6048
6049	__seq_get_right_end:NnN __seq_item:n	6049
6050	}	6050
6051	}	6051
6052	\cs_new:Npn __seq_get_right_loop:nw #1#2 __seq_item:n	6052
6053	{	6053
6054	#2 \use_none:n {#1}	6054
6055	__seq_get_right_loop:nw	6055
6056	}	6056
6057	\cs_new:Npn __seq_get_right_end:NnN #1#2#3 { \exp_not:n {#2} }	6057
6058	\cs_generate_variant:Nn \seq_get_right:NN { c }	6058
6059	\cs_new_protected:Npn \seq_pop_right:NN	6059
6060	{ __seq_pop:NNNN __seq_pop_right:NNN __kernel_tl_set:Nx }	6060
6061	\cs_new_protected:Npn \seq_gpop_right:NN	6061
6062	{ __seq_pop:NNNN __seq_pop_right:NNN __kernel_tl_gset:Nx }	6062
6063	\cs_new_protected:Npn __seq_pop_right:NNN #1#2#3	6063
6064	{	6064
6065	\cs_set_eq:NN __seq_tmp:w __seq_item:n	6065
6066	\cs_set_eq:NN __seq_item:n \scan_stop:	6066
6067	#1 #2	6067
6068	{ \if_false: } \fi: \s__seq	6068
6069	\exp_after:wN \use_i:nnn	6069
6070	\exp_after:wN __seq_pop_right_loop:nn	6070
6071	#2	6071
6072	{	6072


```

6073         \if_false: { \fi: }
6074         \__kernel_tl_set:Nx #3
6075     }
6076     { } \use_none:nn
6077     \cs_set_eq:NN \__seq_item:n \__seq_tmp:w
6078 }
6079 \cs_new:Npn \__seq_pop_right_loop:nn #1#2
6080 {
6081     #2 { \exp_not:n {#1} }
6082     \__seq_pop_right_loop:nn
6083 }
6084 \cs_generate_variant:Nn \seq_pop_right:NN { c }
6085 \cs_generate_variant:Nn \seq_gpop_right:NN { c }
6086 \prg_new_protected_conditional:Npnn \seq_get_left:NN #1#2 { T , F , TF }
6087 { \__seq_pop_TF:NNNN \prg_do_nothing: \seq_get_left:NN #1#2 }
6088 \prg_new_protected_conditional:Npnn \seq_get_right:NN #1#2 { T , F , TF }
6089 { \__seq_pop_TF:NNNN \prg_do_nothing: \seq_get_right:NN #1#2 }
6090 \prg_generate_conditional_variant:Nnn \seq_get_left:NN
6091 { c } { T , F , TF }
6092 \prg_generate_conditional_variant:Nnn \seq_get_right:NN
6093 { c } { T , F , TF }
6094 \prg_new_protected_conditional:Npnn \seq_pop_left:NN #1#2
6095 { T , F , TF }
6096 { \__seq_pop_TF:NNNN \__seq_pop_left:NNN \tl_set:Nn #1 #2 }
6097 \prg_new_protected_conditional:Npnn \seq_gpop_left:NN #1#2
6098 { T , F , TF }
6099 { \__seq_pop_TF:NNNN \__seq_pop_left:NNN \tl_gset:Nn #1 #2 }
6100 \prg_new_protected_conditional:Npnn \seq_pop_right:NN #1#2
6101 { T , F , TF }
6102 { \__seq_pop_TF:NNNN \__seq_pop_right:NNN \__kernel_tl_set:Nx #1 #2 }
6103 \prg_new_protected_conditional:Npnn \seq_gpop_right:NN #1#2
6104 { T , F , TF }
6105 { \__seq_pop_TF:NNNN \__seq_pop_right:NNN \__kernel_tl_gset:Nx #1 #2 }
6106 \prg_generate_conditional_variant:Nnn \seq_pop_left:NN { c }
6107 { T , F , TF }
6108 \prg_generate_conditional_variant:Nnn \seq_gpop_left:NN { c }
6109 { T , F , TF }
6110 \prg_generate_conditional_variant:Nnn \seq_pop_right:NN { c }
6111 { T , F , TF }
6112 \prg_generate_conditional_variant:Nnn \seq_gpop_right:NN { c }
6113 { T , F , TF }
6114 \cs_new:Npn \seq_item:Nn #1
6115 { \exp_after:wN \__seq_item:wNn #1 \s__seq_stop #1 }
6116 \cs_new:Npn \__seq_item:wNn \s__seq #1 \s__seq_stop #2#3
6117 {
6118     \exp_args:Nf \__seq_item:nwn

```

```
6119 { \exp_args:Nf \__seq_item:nN { \int_eval:n {#3} } #2 } 6119
6120 #1 6120
6121 \prg_break: \__seq_item:n { } 6121
6122 \prg_break_point: 6122
6123 } 6123
6124 \cs_new:Npn \__seq_item:nN #1#2 6124
6125 { 6125
6126 \int_compare:nNnTF {#1} < 0 6126
6127 { \int_eval:n { \seq_count:N #2 + 1 + #1 } } 6127
6128 {#1} 6128
6129 } 6129
6130 \cs_new:Npn \__seq_item:nwn #1#2 \__seq_item:n #3 6130
6131 { 6131
6132 #2 6132
6133 \int_compare:nNnTF {#1} = 1 6133
6134 { \prg_break:n { \exp_not:n {#3} } } 6134
6135 { \exp_args:Nf \__seq_item:nwn { \int_eval:n { #1 - 1 } } } 6135
6136 } 6136
6137 \cs_generate_variant:Nn \seq_item:Nn { NV , Ne , c , cV , ce } 6137
6138 \cs_new:Npn \seq_rand_item:N #1 6138
6139 { 6139
6140 \seq_if_empty:NF #1 6140
6141 { \seq_item:Nn #1 { \int_rand:nn { 1 } { \seq_count:N #1 } } } 6141
6142 } 6142
6143 \cs_generate_variant:Nn \seq_rand_item:N { c } 6143
6144 \cs_new:Npn \seq_map_break: 6144
6145 { \prg_map_break:Nn \seq_map_break: { } } 6145
6146 \cs_new:Npn \seq_map_break:n 6146
6147 { \prg_map_break:Nn \seq_map_break: } 6147
6148 \cs_new:Npn \seq_map_function:NN #1#2 6148
6149 { 6149
6150 \exp_after:wN \use_i_ii:nnn 6150
6151 \exp_after:wN \__seq_map_function:Nw 6151
6152 \exp_after:wN #2 6152
6153 #1 6153
6154 \prg_break: 6154
6155 \__seq_item:n { } \__seq_item:n { } \__seq_item:n { } \__seq_item:n { } 6155
6156 \prg_break_point: 6156
6157 \prg_break_point:Nn \seq_map_break: { } 6157
6158 } 6158
6159 \cs_new:Npn \__seq_map_function:Nw #1 6159
6160 #2 \__seq_item:n #3 6160
6161 #4 \__seq_item:n #5 6161
6162 #6 \__seq_item:n #7 6162
6163 #8 \__seq_item:n #9 6163
6164 { 6164
```

6165	#2 #1 {#3}	6165
6166	#4 #1 {#5}	6166
6167	#6 #1 {#7}	6167
6168	#8 #1 {#9}	6168
6169	__seq_map_function:Nw #1	6169
6170	}	6170
6171	\cs_generate_variant:Nn \seq_map_function:NN { c }	6171
6172	\cs_new_protected:Npn __seq_push_item_def:n	6172
6173	{	6173
6174	__seq_push_item_def:	6174
6175	\cs_gset:Npn __seq_item:n ##1	6175
6176	}	6176
6177	\cs_new_protected:Npn __seq_push_item_def:e	6177
6178	{	6178
6179	__seq_push_item_def:	6179
6180	\cs_gset:Npe __seq_item:n ##1	6180
6181	}	6181
6182	\cs_new_protected:Npn __seq_push_item_def:	6182
6183	{	6183
6184	\int_gincr:N \g__kernel_prg_map_int	6184
6185	\cs_gset_eq:cN { __seq_map_ \int_use:N \g__kernel_prg_map_int :w }	6185
6186	__seq_item:n	6186
6187	}	6187
6188	\cs_new_protected:Npn __seq_pop_item_def:	6188
6189	{	6189
6190	\cs_gset_eq:Nc __seq_item:n	6190
6191	{ __seq_map_ \int_use:N \g__kernel_prg_map_int :w }	6191
6192	\int_gdecr:N \g__kernel_prg_map_int	6192
6193	}	6193
6194	\cs_new_protected:Npn \seq_map_inline:Nn #1#2	6194
6195	{	6195
6196	__seq_push_item_def:n {#2}	6196
6197	#1	6197
6198	\prg_break_point:Nn \seq_map_break: { __seq_pop_item_def: }	6198
6199	}	6199
6200	\cs_generate_variant:Nn \seq_map_inline:Nn { c }	6200
6201	\cs_new:Npn \seq_map_tokens:Nn #1#2	6201
6202	{	6202
6203	\exp_last_unbraced:Nno	6203
6204	\use_i:nn { __seq_map_tokens:nw {#2} } #1	6204
6205	\prg_break:	6205
6206	__seq_item:n { } __seq_item:n { } __seq_item:n { } __seq_item:n { }	6206
6207	\prg_break_point:	6207
6208	\prg_break_point:Nn \seq_map_break: { }	6208
6209	}	6209
6210	\cs_generate_variant:Nn \seq_map_tokens:Nn { c }	6210

6211	\cs_new:Npn __seq_map_tokens:nw #1	6211
6212	#2 __seq_item:n #3	6212
6213	#4 __seq_item:n #5	6213
6214	#6 __seq_item:n #7	6214
6215	#8 __seq_item:n #9	6215
6216	{	6216
6217	#2 \use:n {#1} {#3}	6217
6218	#4 \use:n {#1} {#5}	6218
6219	#6 \use:n {#1} {#7}	6219
6220	#8 \use:n {#1} {#9}	6220
6221	__seq_map_tokens:nw {#1}	6221
6222	}	6222
6223	\cs_new_protected:Npn \seq_map_variable:NNn #1#2#3	6223
6224	{	6224
6225	__seq_push_item_def:e	6225
6226	{	6226
6227	\tl_set:Nn \exp_not:N #2 {##1}	6227
6228	\exp_not:n {#3}	6228
6229	}	6229
6230	#1	6230
6231	\prg_break_point:Nn \seq_map_break: { __seq_pop_item_def: }	6231
6232	}	6232
6233	\cs_generate_variant:Nn \seq_map_variable:NNn { Nc }	6233
6234	\cs_generate_variant:Nn \seq_map_variable:NNn { c , cc }	6234
6235	\cs_new_eq:NN __seq_sep: __kernel_int_sep:	6235
6236	\cs_new:Npn \seq_map_indexed_function:NN #1#2	6236
6237	{	6237
6238	__seq_map_indexed:NN #1#2	6238
6239	\prg_break_point:Nn \seq_map_break: { }	6239
6240	}	6240
6241	\cs_new_protected:Npn \seq_map_indexed_inline:Nn #1#2	6241
6242	{	6242
6243	\int_gincr:N \g__kernel_prg_map_int	6243
6244	\cs_gset_protected:cpn	6244
6245	{ __seq_map_ \int_use:N \g__kernel_prg_map_int :w } ##1##2 {#2}	6245
6246	\exp_args:NNc __seq_map_indexed:NN #1	6246
6247	{ __seq_map_ \int_use:N \g__kernel_prg_map_int :w }	6247
6248	\prg_break_point:Nn \seq_map_break:	6248
6249	{ \int_gdecr:N \g__kernel_prg_map_int }	6249
6250	}	6250
6251	\cs_new:Npn __seq_map_indexed:NN #1#2	6251
6252	{	6252
6253	\exp_after:wN __seq_map_indexed:Nw	6253
6254	\exp_after:wN #2	6254
6255	\int_value:w 1	6255
6256	\exp_after:wN \use_i:nn	6256

```

6257 \exp_after:wN \__seq_sep:
6258 #1
6259 \prg_break: \__seq_item:n { } \prg_break_point:
6260 }
6261 \cs_new:Npn \__seq_map_indexed:Nw #1#2 \__seq_sep: #3 \__seq_item:n #4
6262 {
6263 #3
6264 #1 {#2} {#4}
6265 \exp_after:wN \__seq_map_indexed:Nw
6266 \exp_after:wN #1
6267 \int_value:w \int_eval:w 1 + #2 \__seq_sep:
6268 }
6269 \cs_new:Npn \seq_map_pairwise_function:NNN #1#2#3
6270 { \exp_after:wN \__seq_map_pairwise_function:wNN #2 \s__seq_stop #1 #3 }
6271 \cs_new:Npn \__seq_map_pairwise_function:wNN \s__seq #1 \s__seq_stop #2#3
6272 {
6273 \exp_after:wN \__seq_map_pairwise_function:wNw #2 \s__seq_stop #3
6274 #1 { ? \prg_break: } { }
6275 \prg_break_point:
6276 \prg_break_point:Nn \seq_map_break: { }
6277 }
6278 \cs_new:Npn \__seq_map_pairwise_function:wNw \s__seq #1 \s__seq_stop #2
6279 {
6280 \__seq_map_pairwise_function:Nnnwnn #2
6281 #1 { ? \prg_break: } { }
6282 \s__seq_stop
6283 }
6284 \cs_new:Npn \__seq_map_pairwise_function:Nnnwnn #1#2#3#4 \s__seq_stop #5#6
6285 {
6286 \use_none:n #2
6287 \use_none:n #5
6288 #1 {#3} {#6}
6289 \__seq_map_pairwise_function:Nnnwnn #1 #4 \s__seq_stop
6290 }
6291 \cs_generate_variant:Nn \seq_map_pairwise_function:NNN { Nc , c , cc }
6292 \cs_new_protected:Npn \seq_set_map_e:NNn
6293 { \__seq_set_map_e:NNNn \__kernel_tl_set:Nx }
6294 \cs_new_protected:Npn \seq_gset_map_e:NNn
6295 { \__seq_set_map_e:NNNn \__kernel_tl_gset:Nx }
6296 \cs_new_protected:Npn \__seq_set_map_e:NNNn #1#2#3#4
6297 {
6298 \__seq_push_item_def:n { \exp_not:N \__seq_item:n {#4} }
6299 #1 #2 { #3 }
6300 \__seq_pop_item_def:
6301 }
6302 \cs_new_protected:Npn \seq_set_map:NNn

```

```

6303 { \__seq_set_map:NNNn \__kernel_tl_set:Nx }
6304 \cs_new_protected:Npn \seq_gset_map:NNn
6305 { \__seq_set_map:NNNn \__kernel_tl_gset:Nx }
6306 \cs_new_protected:Npn \__seq_set_map:NNNn #1#2#3#4
6307 {
6308     \__seq_push_item_def:n { \exp_not:n { \__seq_item:n {#4} } }
6309     #1 #2 { #3 }
6310     \__seq_pop_item_def:
6311 }
6312 \cs_new:Npn \seq_count:N #1
6313 {
6314     \int_eval:n
6315     {
6316         \exp_after:wN \use_i:nn
6317         \exp_after:wN \__seq_count:w
6318         #1
6319         \__seq_count_end:w \__seq_item:n 7
6320         \__seq_count_end:w \__seq_item:n 6
6321         \__seq_count_end:w \__seq_item:n 5
6322         \__seq_count_end:w \__seq_item:n 4
6323         \__seq_count_end:w \__seq_item:n 3
6324         \__seq_count_end:w \__seq_item:n 2
6325         \__seq_count_end:w \__seq_item:n 1
6326         \__seq_count_end:w \__seq_item:n 0
6327         \prg_break_point:
6328     }
6329 }
6330 \cs_new:Npn \__seq_count:w
6331     #1 \__seq_item:n #2 \__seq_item:n #3 \__seq_item:n #4 \__seq_item:n
6332     #5 \__seq_item:n #6 \__seq_item:n #7 \__seq_item:n #8 #9 \__seq_item:n
6333     { #9 8 + \__seq_count:w }
6334 \cs_new:Npn \__seq_count_end:w 8 + \__seq_count:w #1#2 \prg_break_point: {#1}
6335 \cs_generate_variant:Nn \seq_count:N { c }
6336 \cs_new:Npn \seq_use:Nnnn #1#2#3#4
6337 {
6338     \seq_if_exist:NTF #1
6339     {
6340         \int_case:nnF { \seq_count:N #1 }
6341         {
6342             { 0 } { }
6343             { 1 } { \exp_after:wN \__seq_use:NNnNnn #1 ? { } { } }
6344             { 2 } { \exp_after:wN \__seq_use:NNnNnn #1 {#2} }
6345         }
6346         {
6347             \exp_after:wN \__seq_use_setup:w #1 \__seq_item:n
6348             \s_seq_mark { \__seq_use:nwwwnwn {#3} }

```


6349	\s__seq_mark { __seq_use:nwwn {#4} }	6349
6350	\s__seq_stop { }	6350
6351	}	6351
6352	}	6352
6353	{	6353
6354	\msg_expandable_error:nnn	6354
6355	{ kernel } { bad-variable } {#1}	6355
6356	}	6356
6357	}	6357
6358	\cs_generate_variant:Nn \seq_use:Nnnn { c }	6358
6359	\cs_new:Npn __seq_use:NNnNnn #1#2#3#4#5#6 { \exp_not:n { #3 #6 #5 } }	6359
6360	\cs_new:Npn __seq_use_setup:w \s__seq { __seq_use:nwwwwnwn { } }	6360
6361	\cs_new:Npn __seq_use:nwwwwnwn	6361
6362	#1 __seq_item:n #2 __seq_item:n #3 __seq_item:n #4#5	6362
6363	\s__seq_mark #6#7 \s__seq_stop #8	6363
6364	{	6364
6365	#6 __seq_item:n {#3} __seq_item:n {#4} #5	6365
6366	\s__seq_mark {#6} #7 \s__seq_stop { #8 #1 #2 }	6366
6367	}	6367
6368	\cs_new:Npn __seq_use:nwwn #1 __seq_item:n #2 #3 \s__seq_stop #4	6368
6369	{ \exp_not:n { #4 #1 #2 } }	6369
6370	\cs_new:Npn \seq_use:Nn #1#2	6370
6371	{ \seq_use:Nnnn #1 {#2} {#2} {#2} }	6371
6372	\cs_generate_variant:Nn \seq_use:Nn { c }	6372
6373	\cs_new_eq:NN \seq_push:Nn \seq_put_left:Nn	6373
6374	\cs_generate_variant:Nn \seq_push:Nn { NV , Nv , Ne , c , cV , cv , ce }	6374
6375	\cs_generate_variant:Nn \seq_push:Nn { No , Nx , co , cx }	6375
6376	\cs_new_eq:NN \seq_gpush:Nn \seq_gput_left:Nn	6376
6377	\cs_generate_variant:Nn \seq_gpush:Nn { NV , Nv , Ne , c , cV , cv , ce }	6377
6378	\cs_generate_variant:Nn \seq_gpush:Nn { No , Nx , co , cx }	6378
6379	\cs_new_eq:NN \seq_get:NN \seq_get_left:NN	6379
6380	\cs_new_eq:NN \seq_get:cN \seq_get_left:cN	6380
6381	\cs_new_eq:NN \seq_pop:NN \seq_pop_left:NN	6381
6382	\cs_new_eq:NN \seq_pop:cN \seq_pop_left:cN	6382
6383	\cs_new_eq:NN \seq_gpop:NN \seq_gpop_left:NN	6383
6384	\cs_new_eq:NN \seq_gpop:cN \seq_gpop_left:cN	6384
6385	\prg_new_eq_conditional:NNn \seq_get:NN \seq_get_left:NN { T , F , TF }	6385
6386	\prg_new_eq_conditional:NNn \seq_get:cN \seq_get_left:cN { T , F , TF }	6386
6387	\prg_new_eq_conditional:NNn \seq_pop:NN \seq_pop_left:NN { T , F , TF }	6387
6388	\prg_new_eq_conditional:NNn \seq_pop:cN \seq_pop_left:cN { T , F , TF }	6388
6389	\prg_new_eq_conditional:NNn \seq_gpop:NN \seq_gpop_left:NN { T , F , TF }	6389
6390	\prg_new_eq_conditional:NNn \seq_gpop:cN \seq_gpop_left:cN { T , F , TF }	6390
6391	\cs_new_protected:Npn \seq_show:N { __seq_show:NN \msg_show:nneeee }	6391
6392	\cs_generate_variant:Nn \seq_show:N { c }	6392
6393	\cs_new_protected:Npn \seq_log:N { __seq_show:NN \msg_log:nneeee }	6393
6394	\cs_generate_variant:Nn \seq_log:N { c }	6394

```

6395 \cs_new_protected:Npn \__seq_show:NN #1#2
6396 {
6397     \__kernel_chk_tl_type:NnnT #2 { seq }
6398     {
6399         \s__seq
6400         \exp_after:wN \use_i:nn \exp_after:wN \__seq_show_validate:nn #2
6401         \q_recursion_tail \q_recursion_tail \q_recursion_stop
6402     }
6403     {
6404         #1 { seq } { show }
6405         { \token_to_str:N #2 }
6406         { \seq_map_function:NN #2 \msg_show_item:n }
6407         { } { }
6408     }
6409 }
6410 \cs_new:Npn \__seq_show_validate:nn #1#2
6411 {
6412     \quark_if_recursion_tail_stop:n {#2}
6413     \__seq_wrap_item:n {#2}
6414     \__seq_show_validate:nn
6415 }
6416 \seq_new:N \l_tmpa_seq
6417 \seq_new:N \l_tmpb_seq
6418 \seq_new:N \g_tmpa_seq
6419 \seq_new:N \g_tmpb_seq
6420 %% File: l3int.dtx
6421 \cs_new_eq:NN \int_value:w \tex_number:D
6422 \cs_new_eq:NN \__int_eval:w \tex_numexpr:D
6423 \cs_new_eq:NN \__int_eval_end: \tex_relax:D
6424 \cs_new_eq:NN \if_int_odd:w \tex_ifodd:D
6425 \cs_new_eq:NN \if_case:w \tex_ifcase:D
6426 \scan_new:N \s__int_mark
6427 \scan_new:N \s__int_stop
6428 \cs_new:Npn \__int_use_none_delimit_by_s_stop:w #1 \s__int_stop { }
6429 \quark_new:N \q__int_recursion_tail
6430 \quark_new:N \q__int_recursion_stop
6431 \__kernel_quark_new_test:N \__int_if_recursion_tail_stop_do:Nn
6432 \__kernel_quark_new_test:N \__int_if_recursion_tail_stop:N
6433 \cs_new:Npn \int_eval:n #1
6434 { \tex_the:D \__int_eval:w #1 \__int_eval_end: }
6435 \cs_new:Npn \int_eval:w { \tex_the:D \__int_eval:w }
6436 \cs_new_eq:NN \__int_sep: \__kernel_int_sep:
6437 \cs_new:Npn \int_sign:n #1
6438 {
6439     \int_value:w \exp_after:wN \__int_sign:Nw
6440     \int_value:w \__int_eval:w #1 \__int_eval_end: \__int_sep:

```

6441	\exp_stop_f:	6441
6442	}	6442
6443	\cs_new:Npn __int_sign:Nw #1#2 __int_sep:	6443
6444	{	6444
6445	\if_meaning:w 0 #1	6445
6446	0	6446
6447	\else:	6447
6448	\if_meaning:w - #1 - \fi: 1	6448
6449	\fi:	6449
6450	}	6450
6451	\cs_new:Npn \int_abs:n #1	6451
6452	{	6452
6453	\int_value:w \exp_after:wN __int_abs:N	6453
6454	\int_value:w __int_eval:w #1 __int_eval_end:	6454
6455	\exp_stop_f:	6455
6456	}	6456
6457	\cs_new:Npn __int_abs:N #1	6457
6458	{ \if_meaning:w - #1 \else: \exp_after:wN #1 \fi: }	6458
6459	\cs_new:Npn \int_max:nn #1#2	6459
6460	{	6460
6461	\int_value:w \exp_after:wN __int_maxmin:wwN	6461
6462	\int_value:w __int_eval:w #1 \exp_after:wN __int_sep:	6462
6463	\int_value:w __int_eval:w #2 __int_sep:	6463
6464	>	6464
6465	\exp_stop_f:	6465
6466	}	6466
6467	\cs_new:Npn \int_min:nn #1#2	6467
6468	{	6468
6469	\int_value:w \exp_after:wN __int_maxmin:wwN	6469
6470	\int_value:w __int_eval:w #1 \exp_after:wN __int_sep:	6470
6471	\int_value:w __int_eval:w #2 __int_sep:	6471
6472	<	6472
6473	\exp_stop_f:	6473
6474	}	6474
6475	\cs_new:Npn __int_maxmin:wwN #1 __int_sep: #2 __int_sep: #3	6475
6476	{	6476
6477	\if_int_compare:w #1 #3 #2 ~	6477
6478	#1	6478
6479	\else:	6479
6480	#2	6480
6481	\fi:	6481
6482	}	6482
6483	\cs_new:Npn \int_div_truncate:nn #1#2	6483
6484	{	6484
6485	\int_value:w __int_eval:w	6485
6486	\exp_after:wN __int_div_truncate:NwNw	6486

```
6487         \int_value:w \__int_eval:w #1 \exp_after:wN \__int_sep: 6487
6488         \int_value:w \__int_eval:w #2 \__int_sep: 6488
6489     \__int_eval_end: 6489
6490 } 6490
6491 \cs_new:Npn \__int_div_truncate:NwNw #1#2 \__int_sep: #3#4 \__int_sep: 6491
6492 { 6492
6493     \if_meaning:w 0 #1 6493
6494     0 6494
6495     \else: 6495
6496     ( 6496
6497         #1#2 6497
6498         \if_meaning:w - #1 + \else: - \fi: 6498
6499         ( \if_meaning:w - #3 - \fi: #3#4 - 1 ) / 2 6499
6500     ) 6500
6501     \fi: 6501
6502     / #3#4 6502
6503 } 6503
6504 \cs_new:Npn \int_div_round:nn #1#2 6504
6505 { \int_value:w \__int_eval:w ( #1 ) / ( #2 ) \__int_eval_end: } 6505
6506 \cs_new:Npn \int_mod:nn #1#2 6506
6507 { 6507
6508     \int_value:w \__int_eval:w \exp_after:wN \__int_mod:ww 6508
6509     \int_value:w \__int_eval:w #1 \exp_after:wN \__int_sep: 6509
6510     \int_value:w \__int_eval:w #2 \__int_sep: 6510
6511     \__int_eval_end: 6511
6512 } 6512
6513 \cs_new:Npn \__int_mod:ww #1 \__int_sep: #2 \__int_sep: 6513
6514 { #1 - ( \__int_div_truncate:NwNw #1 \__int_sep: #2 \__int_sep: ) * #2 } 6514
6515 \cs_new:Npn \__kernel_int_add:nnn #1#2#3 6515
6516 { 6516
6517     \int_value:w \__int_eval:w #1 6517
6518     \if_int_compare:w #2 < \c_zero_int \exp_after:wN \reverse_if:N \fi: 6518
6519     \if_int_compare:w #1 < \c_zero_int + #2 + #3 \else: + #3 + #2 \fi: 6519
6520     \__int_eval_end: 6520
6521 } 6521
6522 \cs_new_protected:Npn \int_new:N #1 6522
6523 { 6523
6524     \__kernel_chk_if_free_cs:N #1 6524
6525     \cs:w newcount \cs_end: #1 6525
6526 } 6526
6527 \cs_generate_variant:Nn \int_new:N { c } 6527
6528 \cs_new_protected:Npn \int_const:Nn #1#2 6528
6529 { \__int_const:eN { \int_eval:n {#2} } #1 } 6529
6530 \cs_generate_variant:Nn \int_const:Nn { c } 6530
6531 \cs_new_protected:Npn \__int_const:nN #1#2 6531
6532 { 6532
```

```

6533 \int_compare:nNnTF {#1} < \c_zero_int 6533
6534 { 6534
6535 \int_new:N #2 6535
6536 \tex_global:D 6536
6537 } 6537
6538 { 6538
6539 \int_compare:nNnTF {#1} > \c__int_max_constdef_int 6539
6540 { 6540
6541 \int_new:N #2 6541
6542 \tex_global:D 6542
6543 } 6543
6544 { 6544
6545 \__kernel_chk_if_free_cs:N #2 6545
6546 \tex_global:D \__int_constdef:Nw 6546
6547 } 6547
6548 } 6548
6549 #2 = \__int_eval:w #1 \__int_eval_end: 6549
6550 } 6550
6551 \cs_generate_variant:Nn \__int_const:nN { e } 6551
6552 \if_int_odd:w 0 6552
6553 \cs_if_exist:NT \tex luatexversion:D { 1 } 6553
6554 \cs_if_exist:NT \tex_omathchardef:D { 1 } 6554
6555 \cs_if_exist:NT \tex_XeTeXversion:D { 1 } ~ 6555
6556 \cs_if_exist:NTF \tex_omathchardef:D 6556
6557 { \cs_new_eq:NN \__int_constdef:Nw \tex_omathchardef:D } 6557
6558 { \cs_new_eq:NN \__int_constdef:Nw \tex_chardef:D } 6558
6559 \tex_global:D \__int_constdef:Nw \c__int_max_constdef_int 1114111 ~ 6559
6560 \else: 6560
6561 \cs_new_eq:NN \__int_constdef:Nw \tex_mathchardef:D 6561
6562 \tex_global:D \__int_constdef:Nw \c__int_max_constdef_int 32767 ~ 6562
6563 \fi: 6563
6564 \cs_new_protected:Npn \int_zero:N #1 { #1 = \c_zero_int } 6564
6565 \cs_new_protected:Npn \int_gzero:N #1 { \tex_global:D #1 = \c_zero_int } 6565
6566 \cs_generate_variant:Nn \int_zero:N { c } 6566
6567 \cs_generate_variant:Nn \int_gzero:N { c } 6567
6568 \cs_new_protected:Npn \int_zero_new:N #1 6568
6569 { \int_if_exist:NTF #1 { \int_zero:N #1 } { \int_new:N #1 } } 6569
6570 \cs_new_protected:Npn \int_gzero_new:N #1 6570
6571 { \int_if_exist:NTF #1 { \int_gzero:N #1 } { \int_new:N #1 } } 6571
6572 \cs_generate_variant:Nn \int_zero_new:N { c } 6572
6573 \cs_generate_variant:Nn \int_gzero_new:N { c } 6573
6574 \cs_new_protected:Npn \int_set_eq:NN #1#2 { #1 = #2 } 6574
6575 \cs_generate_variant:Nn \int_set_eq:NN { c , Nc , cc } 6575
6576 \cs_new_protected:Npn \int_gset_eq:NN #1#2 { \tex_global:D #1 = #2 } 6576
6577 \cs_generate_variant:Nn \int_gset_eq:NN { c , Nc , cc } 6577
6578 \prg_new_eq_conditional:NNn \int_if_exist:N \cs_if_exist:N 6578

```

```

6579 { TF , T , F , p }
6580 \prg_new_eq_conditional:NNn \int_if_exist:c \cs_if_exist:c
6581 { TF , T , F , p }
6582 \cs_new_protected:Npn \int_add:Nn #1#2
6583 { \tex_advance:D #1 \__int_eval:w #2 \__int_eval_end: }
6584 \cs_new_protected:Npn \int_sub:Nn #1#2
6585 { \tex_advance:D #1 - \__int_eval:w #2 \__int_eval_end: }
6586 \cs_new_protected:Npn \int_gadd:Nn #1#2
6587 { \tex_global:D \tex_advance:D #1 \__int_eval:w #2 \__int_eval_end: }
6588 \cs_new_protected:Npn \int_gsub:Nn #1#2
6589 { \tex_global:D \tex_advance:D #1 - \__int_eval:w #2 \__int_eval_end: }
6590 \cs_generate_variant:Nn \int_add:Nn { c }
6591 \cs_generate_variant:Nn \int_gadd:Nn { c }
6592 \cs_generate_variant:Nn \int_sub:Nn { c }
6593 \cs_generate_variant:Nn \int_gsub:Nn { c }
6594 \cs_new_protected:Npn \int_incr:N #1
6595 { \tex_advance:D #1 \c_one_int }
6596 \cs_new_protected:Npn \int_decr:N #1
6597 { \tex_advance:D #1 - \c_one_int }
6598 \cs_new_protected:Npn \int_gincr:N #1
6599 { \tex_global:D \tex_advance:D #1 \c_one_int }
6600 \cs_new_protected:Npn \int_gdecr:N #1
6601 { \tex_global:D \tex_advance:D #1 - \c_one_int }
6602 \cs_generate_variant:Nn \int_incr:N { c }
6603 \cs_generate_variant:Nn \int_decr:N { c }
6604 \cs_generate_variant:Nn \int_gincr:N { c }
6605 \cs_generate_variant:Nn \int_gdecr:N { c }
6606 \cs_new_protected:Npn \int_set:Nn #1#2
6607 { #1 = \__int_eval:w #2 \__int_eval_end: }
6608 \cs_new_protected:Npn \int_gset:Nn #1#2
6609 { \tex_global:D #1 = \__int_eval:w #2 \__int_eval_end: }
6610 \cs_generate_variant:Nn \int_set:Nn { NV , c , cV }
6611 \cs_generate_variant:Nn \int_gset:Nn { NV , c , cV }
6612 \cs_new_protected:Npn \int_set_regex_count:Nnn #1#2#3
6613 { \regex_count:nnN {#2} {#3} #1 }
6614 \cs_generate_variant:Nn \int_set_regex_count:Nnn { c }
6615 \cs_new_protected:Npn \int_gset_regex_count:Nnn #1#2#3
6616 {
6617   \group_begin:
6618     \int_set_eq:NN \l__int_internal_a_int #1
6619     \regex_count:nnN {#2} {#3} \l__int_internal_a_int
6620     \int_gset_eq:NN #1 \l__int_internal_a_int
6621   \group_end:
6622 }
6623 \cs_generate_variant:Nn \int_gset_regex_count:Nnn { c }
6624 \cs_new_protected:Npn \int_set_regex_count:NNn #1#2#3

```



```
6625 { \regex_count:NnN #2 {#3} #1 } 6625
6626 \cs_generate_variant:Nn \int_set_regex_count:Nnn { c } 6626
6627 \cs_new_protected:Npn \int_gset_regex_count:NNn #1#2#3 6627
6628 { 6628
6629 \group_begin: 6629
6630 \int_set_eq:NN \l__int_internal_a_int #1 6630
6631 \regex_count:NnN #2 {#3} \l__int_internal_a_int 6631
6632 \int_gset_eq:NN #1 \l__int_internal_a_int 6632
6633 \group_end: 6633
6634 } 6634
6635 \cs_generate_variant:Nn \int_gset_regex_count:NNn { c } 6635
6636 \cs_new_eq:NN \int_use:N \tex_the:D 6636
6637 \cs_new:Npn \int_use:c #1 { \tex_the:D \cs:w #1 \cs_end: } 6637
6638 \cs_new_protected:Npn \__int_compare_error: 6638
6639 { 6639
6640 \if_int_compare:w \c_zero_int \c_zero_int \fi: 6640
6641 = 6641
6642 \__int_compare_error: 6642
6643 } 6643
6644 \cs_new:Npn \__int_compare_error:Nw 6644
6645 #1#2 \s__int_stop 6645
6646 { 6646
6647 { } 6647
6648 \c_zero_int \fi: 6648
6649 \msg_expandable_error:nnn 6649
6650 { kernel } { unknown-comparison } {#1} 6650
6651 \prg_return_false: 6651
6652 } 6652
6653 \prg_new_conditional:Npnn \int_compare:n #1 { p , T , F , TF } 6653
6654 { 6654
6655 \exp_after:wN \__int_compare:w 6655
6656 \int_value:w \__int_eval:w #1 \__int_compare_error: 6656
6657 } 6657
6658 \cs_new:Npn \__int_compare:w #1 \__int_compare_error: 6658
6659 { 6659
6660 \exp_after:wN \if_false: \int_value:w 6660
6661 \__int_compare:Nw #1 e { = nd_ } \s__int_stop 6661
6662 } 6662
6663 \cs_new:Npn \__int_compare:Nw #1#2 \s__int_stop 6663
6664 { 6664
6665 \exp_after:wN \__int_compare:NNw 6665
6666 \__int_to_roman:w - 0 #2 \s__int_mark 6666
6667 #1#2 \s__int_stop 6667
6668 } 6668
6669 \cs_new:Npn \__int_compare:NNw #1#2#3 \s__int_mark 6669
6670 { 6670
```

```

6671 \__kernel_exp_not:w
6672 \use:c
6673 {
6674     __int_compare_ \token_to_str:N #1
6675     \if_meaning:w = #2 = \fi:
6676     :NNw
6677 }
6678 \__int_compare_error:Nw #1
6679 }
6680 \cs_new:cpn { __int_compare_end=:NNw } #1#2#3 e #4 \s__int_stop
6681 {
6682     {#3} \exp_stop_f:
6683     \prg_return_false: \else: \prg_return_true: \fi:
6684 }
6685 \cs_new:Npn \__int_compare:nnN #1#2#3
6686 {
6687     {#2} \exp_stop_f:
6688     \prg_return_false: \exp_after:wN \__int_use_none_delimit_by_s_stop:w
6689     \fi:
6690     #1 #2 #3 \exp_after:wN \__int_compare:Nw \int_value:w \__int_eval:w
6691 }
6692 \cs_new:cpn { __int_compare=:NNw } #1#2#3 =
6693 { \__int_compare:nnN { \reverse_if:N \if_int_compare:w } {#3} = }
6694 \cs_new:cpn { __int_compare<:NNw } #1#2#3 <
6695 { \__int_compare:nnN { \reverse_if:N \if_int_compare:w } {#3} < }
6696 \cs_new:cpn { __int_compare>:NNw } #1#2#3 >
6697 { \__int_compare:nnN { \reverse_if:N \if_int_compare:w } {#3} > }
6698 \cs_new:cpn { __int_compare==:NNw } #1#2#3 ==
6699 { \__int_compare:nnN { \reverse_if:N \if_int_compare:w } {#3} = }
6700 \cs_new:cpn { __int_compare!=:NNw } #1#2#3 !=
6701 { \__int_compare:nnN { \if_int_compare:w } {#3} = }
6702 \cs_new:cpn { __int_compare<=:NNw } #1#2#3 <=
6703 { \__int_compare:nnN { \if_int_compare:w } {#3} > }
6704 \cs_new:cpn { __int_compare>=:NNw } #1#2#3 >=
6705 { \__int_compare:nnN { \if_int_compare:w } {#3} < }
6706 \prg_new_conditional:Npnn \int_compare:nNn #1#2#3 { p , T , F , TF }
6707 {
6708     \if_int_compare:w \__int_eval:w #1 #2 \__int_eval:w #3 \__int_eval_end:
6709     \prg_return_true:
6710     \else:
6711     \prg_return_false:
6712     \fi:
6713 }
6714 \prg_new_conditional:Npnn \int_if_zero:n #1 { p , T , F , TF }
6715 {
6716     \if_int_compare:w \__int_eval:w #1 = \c_zero_int

```

```
6717 \prg_return_true: 6717
6718 \else: 6718
6719 \prg_return_false: 6719
6720 \fi: 6720
6721 } 6721
6722 \cs_new:Npn \int_case:nnTF #1 6722
6723 { 6723
6724 \exp:w 6724
6725 \exp_args:Nf \__int_case:nnTF { \int_eval:n {#1} } 6725
6726 } 6726
6727 \cs_new:Npn \int_case:nnT #1#2#3 6727
6728 { 6728
6729 \exp:w 6729
6730 \exp_args:Nf \__int_case:nnTF { \int_eval:n {#1} } {#2} {#3} { } 6730
6731 } 6731
6732 \cs_new:Npn \int_case:nnF #1#2 6732
6733 { 6733
6734 \exp:w 6734
6735 \exp_args:Nf \__int_case:nnTF { \int_eval:n {#1} } {#2} { } 6735
6736 } 6736
6737 \cs_new:Npn \int_case:nn #1#2 6737
6738 { 6738
6739 \exp:w 6739
6740 \exp_args:Nf \__int_case:nnTF { \int_eval:n {#1} } {#2} { } { } 6740
6741 } 6741
6742 \cs_new:Npn \__int_case:nnTF #1#2#3#4 6742
6743 { \__int_case:nw {#1} #2 {#1} { } \s__int_mark {#3} \s__int_mark {#4} \s__int_stop } 6743
6744 \cs_new:Npn \__int_case:nw #1#2#3 6744
6745 { 6745
6746 \int_compare:nNnTF {#1} = {#2} 6746
6747 { \__int_case_end:nw {#3} } 6747
6748 { \__int_case:nw {#1} } 6748
6749 } 6749
6750 \cs_new:Npn \__int_case_end:nw #1#2#3 \s__int_mark #4#5 \s__int_stop 6750
6751 { \exp_end: #1 #4 } 6751
6752 \prg_new_conditional:Npnn \int_if_odd:n #1 { p , T , F , TF} 6752
6753 { 6753
6754 \if_int_odd:w \__int_eval:w #1 \__int_eval_end: 6754
6755 \prg_return_true: 6755
6756 \else: 6756
6757 \prg_return_false: 6757
6758 \fi: 6758
6759 } 6759
6760 \prg_new_conditional:Npnn \int_if_even:n #1 { p , T , F , TF} 6760
6761 { 6761
6762 \reverse_if:N \if_int_odd:w \__int_eval:w #1 \__int_eval_end: 6762
```

6763	\prg_return_true:	6763
6764	\else:	6764
6765	\prg_return_false:	6765
6766	\fi:	6766
6767	}	6767
6768	\cs_new:Npn \int_while_do:nn #1#2	6768
6769	{	6769
6770	\int_compare:nT {#1}	6770
6771	{	6771
6772	#2	6772
6773	\int_while_do:nn {#1} {#2}	6773
6774	}	6774
6775	}	6775
6776	\cs_new:Npn \int_until_do:nn #1#2	6776
6777	{	6777
6778	\int_compare:nF {#1}	6778
6779	{	6779
6780	#2	6780
6781	\int_until_do:nn {#1} {#2}	6781
6782	}	6782
6783	}	6783
6784	\cs_new:Npn \int_do_while:nn #1#2	6784
6785	{	6785
6786	#2	6786
6787	\int_compare:nT {#1}	6787
6788	{ \int_do_while:nn {#1} {#2} }	6788
6789	}	6789
6790	\cs_new:Npn \int_do_until:nn #1#2	6790
6791	{	6791
6792	#2	6792
6793	\int_compare:nF {#1}	6793
6794	{ \int_do_until:nn {#1} {#2} }	6794
6795	}	6795
6796	\cs_new:Npn \int_while_do:nNnn #1#2#3#4	6796
6797	{	6797
6798	\int_compare:nNnT {#1} #2 {#3}	6798
6799	{	6799
6800	#4	6800
6801	\int_while_do:nNnn {#1} #2 {#3} {#4}	6801
6802	}	6802
6803	}	6803
6804	\cs_new:Npn \int_until_do:nNnn #1#2#3#4	6804
6805	{	6805
6806	\int_compare:nNnF {#1} #2 {#3}	6806
6807	{	6807
6808	#4	6808

6809	\int_until_do:nNnn {#1} #2 {#3} {#4}	6809
6810	}	6810
6811	}	6811
6812	\cs_new:Npn \int_do_while:nNnn #1#2#3#4	6812
6813	{	6813
6814	#4	6814
6815	\int_compare:nNnT {#1} #2 {#3}	6815
6816	{ \int_do_while:nNnn {#1} #2 {#3} {#4} }	6816
6817	}	6817
6818	\cs_new:Npn \int_do_until:nNnn #1#2#3#4	6818
6819	{	6819
6820	#4	6820
6821	\int_compare:nNnF {#1} #2 {#3}	6821
6822	{ \int_do_until:nNnn {#1} #2 {#3} {#4} }	6822
6823	}	6823
6824	\cs_new:Npn \int_step_function:nnnN #1#2#3	6824
6825	{	6825
6826	\exp_after:wN __int_step:w	6826
6827	\int_value:w __int_eval:w #1 \exp_after:wN __int_sep:	6827
6828	\int_value:w __int_eval:w #2 \exp_after:wN __int_sep:	6828
6829	\int_value:w __int_eval:w #3 __int_sep:	6829
6830	}	6830
6831	\cs_new:Npn __int_step:w #1 __int_sep: #2 __int_sep: #3 __int_sep: #4	6831
6832	{	6832
6833	\int_compare:nNnTF {#2} > \c_zero_int	6833
6834	{ __int_step:Nw > }	6834
6835	{	6835
6836	\if_meaning:w 0 #2	6836
6837	\exp_after:wN \use_ii:nn	6837
6838	\fi:	6838
6839	\use_none:n	6839
6840	{	6840
6841	\msg_expandable_error:nnn	6841
6842	{ kernel } { zero-step } {#4}	6842
6843	\prg_break:	6843
6844	}	6844
6845	__int_step:Nw <	6845
6846	}	6846
6847	#1 __int_sep: {#2} {#3} {#4}	6847
6848	\prg_break_point:	6848
6849	}	6849
6850	\cs_new:Npn __int_step:Nw #1#2 __int_sep: #3#4#5	6850
6851	{	6851
6852	\if_int_compare:w #2 #1 #4 \exp_stop_f:	6852
6853	\prg_break:n	6853
6854	\fi:	6854

6855	#5 {#2}	6855
6856	\exp_after:wN __int_step:Nw	6856
6857	\exp_after:wN #1	6857
6858	\int_value:w __int_eval:w #2 + #3 __int_sep: {#3} {#4} {#5}	6858
6859	}	6859
6860	\cs_new:Npn \int_step_function:nN	6860
6861	{ \int_step_function:nnnN \c_one_int \c_one_int }	6861
6862	\cs_new:Npn \int_step_function:nnN #1	6862
6863	{ \int_step_function:nnnN {#1} \c_one_int }	6863
6864	\cs_new_eq:NN \int_step_tokens:nn \int_step_function:nN	6864
6865	\cs_new_eq:NN \int_step_tokens:nnn \int_step_function:nnN	6865
6866	\cs_new_eq:NN \int_step_tokens:nnnn \int_step_function:nnnN	6866
6867	\cs_new_protected:Npn \int_step_inline:nn	6867
6868	{ \int_step_inline:nnnn { 1 } { 1 } }	6868
6869	\cs_new_protected:Npn \int_step_inline:nnn #1	6869
6870	{ \int_step_inline:nnnn {#1} { 1 } }	6870
6871	\cs_new_protected:Npn \int_step_inline:nnnn	6871
6872	{	6872
6873	\int_gincr:N \g__kernel_prg_map_int	6873
6874	\exp_args:NNc __int_step:NNnnnn	6874
6875	\cs_gset_protected:Npn	6875
6876	{ __int_map_ \int_use:N \g__kernel_prg_map_int :w }	6876
6877	}	6877
6878	\cs_new_protected:Npn \int_step_variable:nNn	6878
6879	{ \int_step_variable:nnnNn { 1 } { 1 } }	6879
6880	\cs_new_protected:Npn \int_step_variable:nnNn #1	6880
6881	{ \int_step_variable:nnnNn {#1} { 1 } }	6881
6882	\cs_new_protected:Npn \int_step_variable:nnnNn #1#2#3#4#5	6882
6883	{	6883
6884	\int_gincr:N \g__kernel_prg_map_int	6884
6885	\exp_args:NNc __int_step:NNnnnn	6885
6886	\cs_gset_protected:Npe	6886
6887	{ __int_map_ \int_use:N \g__kernel_prg_map_int :w }	6887
6888	{#1}{#2}{#3}	6888
6889	{	6889
6890	\tl_set:Nn \exp_not:N #4 {##1}	6890
6891	\exp_not:n {#5}	6891
6892	}	6892
6893	}	6893
6894	\cs_new_protected:Npn __int_step:NNnnnn #1#2#3#4#5#6	6894
6895	{	6895
6896	#1 #2 ##1 {#6}	6896
6897	\int_step_function:nnnN {#3} {#4} {#5} #2	6897
6898	\prg_break_point:Nn \scan_stop: { \int_gdecr:N \g__kernel_prg_map_int }	6898
6899	}	6899
6900	\cs_new_eq:NN \int_to_arabic:n \int_eval:n	6900


```
6901 \cs_generate_variant:Nn \int_to_arabic:n { v } 6901
6902 \cs_new:Npn \int_to_symbols:nnn #1#2#3 6902
6903 { 6903
6904   \int_compare:nNnTF {#1} > {#2} 6904
6905   { 6905
6906     \__int_to_symbols:ennn 6906
6907     { 6907
6908       \int_case:nn 6908
6909         { 1 + \int_mod:nn { #1 - 1 } {#2} } 6909
6910         {#3} 6910
6911       } 6911
6912       {#1} {#2} {#3} 6912
6913     } 6913
6914     { \int_case:nn {#1} {#3} } 6914
6915   } 6915
6916 \cs_new:Npn \__int_to_symbols:nnnn #1#2#3#4 6916
6917 { 6917
6918   \exp_args:Nf \int_to_symbols:nnn 6918
6919   { \int_div_truncate:nn { #2 - 1 } {#3} } {#3} {#4} 6919
6920   #1 6920
6921 } 6921
6922 \cs_generate_variant:Nn \__int_to_symbols:nnnn { e } 6922
6923 \cs_new:Npn \int_to_alph:n #1 6923
6924 { 6924
6925   \int_to_symbols:nnn {#1} { 26 } 6925
6926   { 6926
6927     { 1 } { a } 6927
6928     { 2 } { b } 6928
6929     { 3 } { c } 6929
6930     { 4 } { d } 6930
6931     { 5 } { e } 6931
6932     { 6 } { f } 6932
6933     { 7 } { g } 6933
6934     { 8 } { h } 6934
6935     { 9 } { i } 6935
6936     { 10 } { j } 6936
6937     { 11 } { k } 6937
6938     { 12 } { l } 6938
6939     { 13 } { m } 6939
6940     { 14 } { n } 6940
6941     { 15 } { o } 6941
6942     { 16 } { p } 6942
6943     { 17 } { q } 6943
6944     { 18 } { r } 6944
6945     { 19 } { s } 6945
6946     { 20 } { t } 6946
```

6947	{ 21 } { u }	6947
6948	{ 22 } { v }	6948
6949	{ 23 } { w }	6949
6950	{ 24 } { x }	6950
6951	{ 25 } { y }	6951
6952	{ 26 } { z }	6952
6953	}	6953
6954	}	6954
6955	\cs_new:Npn \int_to_Alph:n #1	6955
6956	{	6956
6957	\int_to_symbols:nnn {#1} { 26 }	6957
6958	{	6958
6959	{ 1 } { A }	6959
6960	{ 2 } { B }	6960
6961	{ 3 } { C }	6961
6962	{ 4 } { D }	6962
6963	{ 5 } { E }	6963
6964	{ 6 } { F }	6964
6965	{ 7 } { G }	6965
6966	{ 8 } { H }	6966
6967	{ 9 } { I }	6967
6968	{ 10 } { J }	6968
6969	{ 11 } { K }	6969
6970	{ 12 } { L }	6970
6971	{ 13 } { M }	6971
6972	{ 14 } { N }	6972
6973	{ 15 } { O }	6973
6974	{ 16 } { P }	6974
6975	{ 17 } { Q }	6975
6976	{ 18 } { R }	6976
6977	{ 19 } { S }	6977
6978	{ 20 } { T }	6978
6979	{ 21 } { U }	6979
6980	{ 22 } { V }	6980
6981	{ 23 } { W }	6981
6982	{ 24 } { X }	6982
6983	{ 25 } { Y }	6983
6984	{ 26 } { Z }	6984
6985	}	6985
6986	}	6986
6987	\cs_new:Npn \int_to_base:nn #1	6987
6988	{ \exp_args:Nf __int_to_base:nn { \int_eval:n {#1} } }	6988
6989	\cs_new:Npn \int_to_Base:nn #1	6989
6990	{ \exp_args:Nf __int_to_Base:nn { \int_eval:n {#1} } }	6990
6991	\cs_new:Npn __int_to_base:nn #1#2	6991
6992	{	6992

```
6993 \int_compare:nNnTF {#1} < 0 6993
6994 { \exp_args:No \__int_to_base:nnN { \use_none:n #1 } {#2} - } 6994
6995 { \__int_to_base:nnN {#1} {#2} \c_empty_tl } 6995
6996 } 6996
6997 \cs_new:Npn \__int_to_Base:nn #1#2 6997
6998 { 6998
6999 \int_compare:nNnTF {#1} < 0 6999
7000 { \exp_args:No \__int_to_Base:nnN { \use_none:n #1 } {#2} - } 7000
7001 { \__int_to_Base:nnN {#1} {#2} \c_empty_tl } 7001
7002 } 7002
7003 \cs_new:Npn \__int_to_base:nnN #1#2#3 7003
7004 { 7004
7005 \int_compare:nNnTF {#1} < {#2} 7005
7006 { \exp_last_unbraced:Nf #3 { \__int_to_letter:n {#1} } } 7006
7007 { 7007
7008 \exp_args:Nf \__int_to_base:nnnN 7008
7009 { \__int_to_letter:n { \int_mod:nn {#1} {#2} } } 7009
7010 {#1} 7010
7011 {#2} 7011
7012 #3 7012
7013 } 7013
7014 } 7014
7015 \cs_new:Npn \__int_to_base:nnnN #1#2#3#4 7015
7016 { 7016
7017 \exp_args:Nf \__int_to_base:nnN 7017
7018 { \int_div_truncate:nn {#2} {#3} } 7018
7019 {#3} 7019
7020 #4 7020
7021 #1 7021
7022 } 7022
7023 \cs_new:Npn \__int_to_Base:nnN #1#2#3 7023
7024 { 7024
7025 \int_compare:nNnTF {#1} < {#2} 7025
7026 { \exp_last_unbraced:Nf #3 { \__int_to_Letter:n {#1} } } 7026
7027 { 7027
7028 \exp_args:Nf \__int_to_Base:nnnN 7028
7029 { \__int_to_Letter:n { \int_mod:nn {#1} {#2} } } 7029
7030 {#1} 7030
7031 {#2} 7031
7032 #3 7032
7033 } 7033
7034 } 7034
7035 \cs_new:Npn \__int_to_Base:nnnN #1#2#3#4 7035
7036 { 7036
7037 \exp_args:Nf \__int_to_Base:nnN 7037
7038 { \int_div_truncate:nn {#2} {#3} } 7038
```

7039	{#3}	7039
7040	#4	7040
7041	#1	7041
7042	}	7042
7043	\cs_new:Npn __int_to_letter:n #1	7043
7044	{	7044
7045	\exp_after:wN \exp_after:wN	7045
7046	\if_case:w __int_eval:w #1 - 10 __int_eval_end:	7046
7047	a	7047
7048	\or: b	7048
7049	\or: c	7049
7050	\or: d	7050
7051	\or: e	7051
7052	\or: f	7052
7053	\or: g	7053
7054	\or: h	7054
7055	\or: i	7055
7056	\or: j	7056
7057	\or: k	7057
7058	\or: l	7058
7059	\or: m	7059
7060	\or: n	7060
7061	\or: o	7061
7062	\or: p	7062
7063	\or: q	7063
7064	\or: r	7064
7065	\or: s	7065
7066	\or: t	7066
7067	\or: u	7067
7068	\or: v	7068
7069	\or: w	7069
7070	\or: x	7070
7071	\or: y	7071
7072	\or: z	7072
7073	\else: \int_value:w __int_eval:w #1 \exp_after:wN __int_eval_end:	7073
7074	\fi:	7074
7075	}	7075
7076	\cs_new:Npn __int_to_Letter:n #1	7076
7077	{	7077
7078	\exp_after:wN \exp_after:wN	7078
7079	\if_case:w __int_eval:w #1 - 10 __int_eval_end:	7079
7080	A	7080
7081	\or: B	7081
7082	\or: C	7082
7083	\or: D	7083
7084	\or: E	7084

7085	\or: F	7085
7086	\or: G	7086
7087	\or: H	7087
7088	\or: I	7088
7089	\or: J	7089
7090	\or: K	7090
7091	\or: L	7091
7092	\or: M	7092
7093	\or: N	7093
7094	\or: O	7094
7095	\or: P	7095
7096	\or: Q	7096
7097	\or: R	7097
7098	\or: S	7098
7099	\or: T	7099
7100	\or: U	7100
7101	\or: V	7101
7102	\or: W	7102
7103	\or: X	7103
7104	\or: Y	7104
7105	\or: Z	7105
7106	\else: \int_value:w __int_eval:w #1 \exp_after:wN __int_eval_end:	7106
7107	\fi:	7107
7108	}	7108
7109	\cs_new:Npn \int_to_bin:n #1	7109
7110	{ \int_to_base:nn {#1} { 2 } }	7110
7111	\cs_new:Npn \int_to_hex:n #1	7111
7112	{ \int_to_base:nn {#1} { 16 } }	7112
7113	\cs_new:Npn \int_to_Hex:n #1	7113
7114	{ \int_to_Base:nn {#1} { 16 } }	7114
7115	\cs_new:Npn \int_to_oct:n #1	7115
7116	{ \int_to_base:nn {#1} { 8 } }	7116
7117	\cs_new:Npn \int_to_roman:n #1	7117
7118	{	7118
7119	\exp_after:wN __int_to_roman:N	7119
7120	__int_to_roman:w \int_eval:n {#1} Q	7120
7121	}	7121
7122	\cs_new:Npn __int_to_roman:N #1	7122
7123	{	7123
7124	\use:c { __int_to_roman_ #1 :w }	7124
7125	__int_to_roman:N	7125
7126	}	7126
7127	\cs_new:Npn \int_to_Roman:n #1	7127
7128	{	7128
7129	\exp_after:wN __int_to_Roman_aux:N	7129
7130	__int_to_roman:w \int_eval:n {#1} Q	7130

```

7131 }
7132 \cs_new:Npn \__int_to_Roman_aux:N #1
7133 {
7134     \use:c { __int_to_Roman_ #1 :w }
7135     \__int_to_Roman_aux:N
7136 }
7137 \cs_new:Npn \__int_to_roman_i:w { i }
7138 \cs_new:Npn \__int_to_roman_v:w { v }
7139 \cs_new:Npn \__int_to_roman_x:w { x }
7140 \cs_new:Npn \__int_to_roman_l:w { l }
7141 \cs_new:Npn \__int_to_roman_c:w { c }
7142 \cs_new:Npn \__int_to_roman_d:w { d }
7143 \cs_new:Npn \__int_to_roman_m:w { m }
7144 \cs_new:Npn \__int_to_roman_Q:w #1 { }
7145 \cs_new:Npn \__int_to_Roman_i:w { I }
7146 \cs_new:Npn \__int_to_Roman_v:w { V }
7147 \cs_new:Npn \__int_to_Roman_x:w { X }
7148 \cs_new:Npn \__int_to_Roman_l:w { L }
7149 \cs_new:Npn \__int_to_Roman_c:w { C }
7150 \cs_new:Npn \__int_to_Roman_d:w { D }
7151 \cs_new:Npn \__int_to_Roman_m:w { M }
7152 \cs_new:Npn \__int_to_Roman_Q:w #1 { }
7153 \cs_new:Npn \__int_pass_signs:wn #1
7154 {
7155     \if:w + \if:w - \exp_not:N #1 + \fi: \exp_not:N #1
7156     \exp_after:wN \__int_pass_signs:wn
7157     \else:
7158         \exp_after:wN \__int_pass_signs_end:wn
7159         \exp_after:wN #1
7160     \fi:
7161 }
7162 \cs_new:Npn \__int_pass_signs_end:wn #1 \s__int_stop #2 { #2 #1 }
7163 \cs_new:Npn \int_from_alph:n #1
7164 {
7165     \int_eval:n
7166     {
7167         \exp_after:wN \__int_pass_signs:wn \tl_to_str:n {#1}
7168         \s__int_stop { \__int_from_alph:nN { 0 } }
7169         \q__int_recursion_tail \q__int_recursion_stop
7170     }
7171 }
7172 \cs_new:Npn \__int_from_alph:nN #1#2
7173 {
7174     \__int_if_recursion_tail_stop_do:Nn #2 {#1}
7175     \exp_args:Nf \__int_from_alph:nN
7176     { \int_eval:n { #1 * 26 + \__int_from_alph:N #2 } }

```



```
7177 } 7177
7178 \cs_new:Npn \__int_from_alph:N #1 7178
7179 { `#1 - \int_compare:nNnTF { `#1 } < { 91 } { 64 } { 96 } } 7179
7180 \cs_new:Npn \int_from_base:nn #1#2 7180
7181 { 7181
7182 \int_eval:n 7182
7183 { 7183
7184 \exp_after:wN \__int_pass_signs:wn \tl_to_str:n {#1} 7184
7185 \s_int_stop { \__int_from_base:nnN { 0 } {#2} } 7185
7186 \q__int_recursion_tail \q__int_recursion_stop 7186
7187 } 7187
7188 } 7188
7189 \cs_new:Npn \__int_from_base:nnN #1#2#3 7189
7190 { 7190
7191 \__int_if_recursion_tail_stop_do:Nn #3 {#1} 7191
7192 \exp_args:Nf \__int_from_base:nnN 7192
7193 { \int_eval:n { #1 * #2 + \__int_from_base:N #3 } } 7193
7194 {#2} 7194
7195 } 7195
7196 \cs_new:Npn \__int_from_base:N #1 7196
7197 { 7197
7198 \int_compare:nNnTF { `#1 } < { 58 } 7198
7199 {#1} 7199
7200 { `#1 - \int_compare:nNnTF { `#1 } < { 91 } { 55 } { 87 } } 7200
7201 } 7201
7202 \cs_new:Npn \int_from_bin:n #1 7202
7203 { \int_from_base:nn {#1} { 2 } } 7203
7204 \cs_new:Npn \int_from_hex:n #1 7204
7205 { \int_from_base:nn {#1} { 16 } } 7205
7206 \cs_new:Npn \int_from_oct:n #1 7206
7207 { \int_from_base:nn {#1} { 8 } } 7207
7208 \int_const:cn { c__int_from_roman_i_int } { 1 } 7208
7209 \int_const:cn { c__int_from_roman_v_int } { 5 } 7209
7210 \int_const:cn { c__int_from_roman_x_int } { 10 } 7210
7211 \int_const:cn { c__int_from_roman_l_int } { 50 } 7211
7212 \int_const:cn { c__int_from_roman_c_int } { 100 } 7212
7213 \int_const:cn { c__int_from_roman_d_int } { 500 } 7213
7214 \int_const:cn { c__int_from_roman_m_int } { 1000 } 7214
7215 \int_const:cn { c__int_from_roman_I_int } { 1 } 7215
7216 \int_const:cn { c__int_from_roman_V_int } { 5 } 7216
7217 \int_const:cn { c__int_from_roman_X_int } { 10 } 7217
7218 \int_const:cn { c__int_from_roman_L_int } { 50 } 7218
7219 \int_const:cn { c__int_from_roman_C_int } { 100 } 7219
7220 \int_const:cn { c__int_from_roman_D_int } { 500 } 7220
7221 \int_const:cn { c__int_from_roman_M_int } { 1000 } 7221
7222 \cs_new:Npn \int_from_roman:n #1 7222
```

```

7223 {
7224     \int_eval:n
7225     {
7226         (
7227             0
7228             \exp_after:wN \__int_from_roman:NN \tl_to_str:n {#1}
7229             \q__int_recursion_tail \q__int_recursion_tail \q__int_recursion_stop
7230         )
7231     }
7232 }
7233 \cs_new:Npn \__int_from_roman:NN #1#2
7234 {
7235     \__int_if_recursion_tail_stop:N #1
7236     \int_if_exist:cF { c__int_from_roman_ #1 _int }
7237     { \__int_from_roman_error:w }
7238     \__int_if_recursion_tail_stop_do:Nn #2
7239     { + \use:c { c__int_from_roman_ #1 _int } }
7240     \int_if_exist:cF { c__int_from_roman_ #2 _int }
7241     { \__int_from_roman_error:w }
7242     \int_compare:nNnTF
7243     { \use:c { c__int_from_roman_ #1 _int } }
7244     <
7245     { \use:c { c__int_from_roman_ #2 _int } }
7246     {
7247         + \use:c { c__int_from_roman_ #2 _int }
7248         - \use:c { c__int_from_roman_ #1 _int }
7249         \__int_from_roman:NN
7250     }
7251     {
7252         + \use:c { c__int_from_roman_ #1 _int }
7253         \__int_from_roman:NN #2
7254     }
7255 }
7256 \cs_new:Npn \__int_from_roman_error:w #1 \q__int_recursion_stop #2
7257 { #2 * 0 - 1 }
7258 \cs_new_eq:NN \int_show:N \__kernel_register_show:N
7259 \cs_generate_variant:Nn \int_show:N { c }
7260 \cs_new_protected:Npn \int_show:n
7261 { \__kernel_msg_show_eval:Nn \int_eval:n }
7262 \cs_new_eq:NN \int_log:N \__kernel_register_log:N
7263 \cs_generate_variant:Nn \int_log:N { c }
7264 \cs_new_protected:Npn \int_log:n
7265 { \__kernel_msg_log_eval:Nn \int_eval:n }
7266 \int_const:Nn \c_one_int { 1 }
7267 \int_const:Nn \c_max_int { 2 147 483 647 }
7268 \int_const:Nn \c_max_char_int

```

```

7269 {
7270     \if_int_odd:w 0
7271         \cs_if_exist:NT \tex_luatexversion:D { 1 }
7272         \cs_if_exist:NT \tex_XeTeXversion:D { 1 } ~
7273         "10FFFF
7274     \else:
7275         "FF
7276     \fi:
7277 }
7278 \int_new:N \l_tmpa_int
7279 \int_new:N \l_tmpb_int
7280 \int_new:N \g_tmpa_int
7281 \int_new:N \g_tmpb_int
7282 \int_new:N \l__seq_internal_a_int
7283 \int_new:N \l__seq_internal_b_int
7284 %% File: l3flag.dtx
7285 \cs_new_eq:NN \__flag_sep: \__kernel_int_sep:
7286 \cs_new_protected:Npn \flag_new:N #1
7287 { \cs_new_protected:Npe #1 { \cs_to_str:N #1 } }
7288 \cs_generate_variant:Nn \flag_new:N { c }
7289 \flag_new:N \l_tmpa_flag
7290 \flag_new:N \l_tmpb_flag
7291 \cs_new_protected:Npn \flag_clear:N #1
7292 {
7293     \__flag_clear:wN 0 \__flag_sep: #1
7294     \prg_break_point:
7295 }
7296 \cs_generate_variant:Nn \flag_clear:N { c }
7297 \cs_new_protected:Npn \__flag_clear:wN #1 \__flag_sep: #2
7298 {
7299     \if_cs_exist:w #2 #1 \cs_end: \else:
7300         \prg_break:n
7301     \fi:
7302     \cs_set_eq:cN { #2 #1 } \tex_undefined:D
7303     \exp_after:wN \__flag_clear:wN
7304     \int_value:w \int_eval:w \c_one_int + #1 \__flag_sep: #2
7305 }
7306 \cs_new_protected:Npn \flag_clear_new:N #1
7307 { \flag_if_exist:NTF #1 { \flag_clear:N } { \flag_new:N } #1 }
7308 \cs_generate_variant:Nn \flag_clear_new:N { c }
7309 \cs_new_protected:Npn \flag_show:N { \__flag_show:NN \tl_show:n }
7310 \cs_generate_variant:Nn \flag_show:N { c }
7311 \cs_new_protected:Npn \flag_log:N { \__flag_show:NN \tl_log:n }
7312 \cs_generate_variant:Nn \flag_log:N { c }
7313 \cs_new_protected:Npn \__flag_show:NN #1#2
7314 {

```

7315	__kernel_chk_defined:NT #2	7315
7316	{ \exp_args:Ne #1 { \tl_to_str:n { #2 height } = \flag_height:N #2 } }	7316
7317	}	7317
7318	\prg_new_eq_conditional:NNn \flag_if_exist:N \cs_if_exist:N	7318
7319	{ TF , T , F , p }	7319
7320	\prg_new_eq_conditional:NNn \flag_if_exist:c \cs_if_exist:c	7320
7321	{ TF , T , F , p }	7321
7322	\prg_new_conditional:Npnn \flag_if_raised:N #1 { p , T , F , TF }	7322
7323	{	7323
7324	\if_cs_exist:w #1 0 \cs_end:	7324
7325	\prg_return_true:	7325
7326	\else:	7326
7327	\prg_return_false:	7327
7328	\fi:	7328
7329	}	7329
7330	\prg_generate_conditional_variant:Nnn \flag_if_raised:N	7330
7331	{ c } { p , T , F , TF }	7331
7332	\cs_new:Npn \flag_height:N #1 { __flag_height_loop:wN 0 __flag_sep: #1 }	7332
7333	\cs_new:Npn __flag_height_loop:wN #1 __flag_sep: #2	7333
7334	{	7334
7335	\if_cs_exist:w #2 #1 \cs_end: \else:	7335
7336	\exp_after:wN __flag_height_end:wN	7336
7337	\fi:	7337
7338	\exp_after:wN __flag_height_loop:wN	7338
7339	\int_value:w \int_eval:w \c_one_int + #1 __flag_sep: #2	7339
7340	}	7340
7341	\cs_new:Npn __flag_height_end:wN #1 + #2 __flag_sep: #3 {#2}	7341
7342	\cs_generate_variant:Nn \flag_height:N { c }	7342
7343	\cs_new:Npn \flag_raise:N #1	7343
7344	{ \exp_after:wN \use_none:n \cs:w #1 \flag_height:N #1 \cs_end: }	7344
7345	\cs_generate_variant:Nn \flag_raise:N { c }	7345
7346	\cs_new:Npn \flag_ensure_raised:N #1	7346
7347	{ \exp_after:wN \use_none:n \cs:w #1 0 \cs_end: }	7347
7348	\cs_generate_variant:Nn \flag_ensure_raised:N { c }	7348
7349	\cs_new_protected:Npn \flag_new:n #1 { \flag_new:c { l_#1_flag } }	7349
7350	\cs_new_protected:Npn \flag_clear:n #1 { \flag_clear:c { l_#1_flag } }	7350
7351	\cs_new_protected:Npn \flag_clear_new:n #1 { \flag_clear_new:c { l_#1_flag } }	7351
7352	\cs_new:Npn \flag_if_exist_p:n #1 { \flag_if_exist_p:c { l_#1_flag } }	7352
7353	\cs_new:Npn \flag_if_exist:nT #1 { \flag_if_exist:cT { l_#1_flag } }	7353
7354	\cs_new:Npn \flag_if_exist:nF #1 { \flag_if_exist:cF { l_#1_flag } }	7354
7355	\cs_new:Npn \flag_if_exist:nTF #1 { \flag_if_exist:cTF { l_#1_flag } }	7355
7356	\cs_new:Npn \flag_if_raised_p:n #1 { \flag_if_raised_p:c { l_#1_flag } }	7356
7357	\cs_new:Npn \flag_if_raised:nT #1 { \flag_if_raised:cT { l_#1_flag } }	7357
7358	\cs_new:Npn \flag_if_raised:nF #1 { \flag_if_raised:cF { l_#1_flag } }	7358
7359	\cs_new:Npn \flag_if_raised:nTF #1 { \flag_if_raised:cTF { l_#1_flag } }	7359
7360	\cs_new:Npn \flag_height:n #1 { \flag_height:c { l_#1_flag } }	7360

```
7361 \cs_new:Npn \flag_raise:n #1 { \flag_raise:c { l_#1_flag } } 7361
7362 \cs_new:Npn \flag_ensure_raised:n #1 { \flag_ensure_raised:c { l_#1_flag } } 7362
7363 \cs_new_protected:Npn \flag_show:n { \__flag_show:Nn \tl_show:n } 7363
7364 \cs_new_protected:Npn \flag_log:n { \__flag_show:Nn \tl_log:n } 7364
7365 \cs_new_protected:Npn \__flag_show:Nn #1#2 7365
7366 { 7366
7367   \exp_args:Nc \__kernel_chk_defined:NT { l_#2_flag } 7367
7368   { 7368
7369     \exp_args:Ne #1 7369
7370     { \tl_to_str:n { flag~#2~height } = \flag_height:n {#2} } 7370
7371   } 7371
7372 } 7372
7373 %% File: l3prg.dtx 7373
7374 \cs_new_eq:NN \if_predicate:w \tex_ifodd:D 7374
7375 \cs_new_protected:Npn \bool_new:N #1 { \cs_new_eq:NN #1 \c_false_bool } 7375
7376 \cs_generate_variant:Nn \bool_new:N { c } 7376
7377 \cs_new_protected:Npn \bool_const:Nn #1#2 7377
7378 { 7378
7379   \__kernel_chk_if_free_cs:N #1 7379
7380   \tex_global:D \tex_chardef:D #1 = \bool_if_p:n {#2} 7380
7381 } 7381
7382 \cs_generate_variant:Nn \bool_const:Nn { c } 7382
7383 \cs_new_protected:Npn \bool_set_true:N #1 7383
7384 { \cs_set_eq:NN #1 \c_true_bool } 7384
7385 \cs_new_protected:Npn \bool_set_false:N #1 7385
7386 { \cs_set_eq:NN #1 \c_false_bool } 7386
7387 \cs_new_protected:Npn \bool_gset_true:N #1 7387
7388 { \cs_gset_eq:NN #1 \c_true_bool } 7388
7389 \cs_new_protected:Npn \bool_gset_false:N #1 7389
7390 { \cs_gset_eq:NN #1 \c_false_bool } 7390
7391 \cs_generate_variant:Nn \bool_set_true:N { c } 7391
7392 \cs_generate_variant:Nn \bool_set_false:N { c } 7392
7393 \cs_generate_variant:Nn \bool_gset_true:N { c } 7393
7394 \cs_generate_variant:Nn \bool_gset_false:N { c } 7394
7395 \cs_new_eq:NN \bool_set_eq:NN \tl_set_eq:NN 7395
7396 \cs_new_eq:NN \bool_gset_eq:NN \tl_gset_eq:NN 7396
7397 \cs_generate_variant:Nn \bool_set_eq:NN { Nc, cN, cc } 7397
7398 \cs_generate_variant:Nn \bool_gset_eq:NN { Nc, cN, cc } 7398
7399 \cs_new_protected:Npn \bool_set:Nn #1#2 7399
7400 { 7400
7401   \exp_last_unbraced:NNNf 7401
7402   \tex_chardef:D #1 = { \bool_if_p:n {#2} } 7402
7403 } 7403
7404 \cs_new_protected:Npn \bool_gset:Nn #1#2 7404
7405 { 7405
7406   \exp_last_unbraced:NNNNf 7406
```

```
7407 \tex_global:D\tex_chardef:D #1 = { \bool_if_p:n {#2} } 7407
7408 } 7408
7409 \cs_generate_variant:Nn \bool_set:Nn { c } 7409
7410 \cs_generate_variant:Nn \bool_gset:Nn { c } 7410
7411 \cs_new_protected:Npn \bool_set_inverse:N #1 7411
7412 { \bool_if:NTF #1 { \bool_set_false:N } { \bool_set_true:N } #1 } 7412
7413 \cs_generate_variant:Nn \bool_set_inverse:N { c } 7413
7414 \cs_new_protected:Npn \bool_gset_inverse:N #1 7414
7415 { \bool_if:NTF #1 { \bool_gset_false:N } { \bool_gset_true:N } #1 } 7415
7416 \cs_generate_variant:Nn \bool_gset_inverse:N { c } 7416
7417 \quark_new:N \q__bool_recursion_tail 7417
7418 \quark_new:N \q__bool_recursion_stop 7418
7419 \cs_new:Npn \__bool_use_i_delimit_by_q_recursion_stop:nw 7419
7420 #1 #2 \q__bool_recursion_stop {#1} 7420
7421 \__kernel_quark_new_test:N \__bool_if_recursion_tail_stop_do:nn 7421
7422 \prg_new_conditional:Npnn \bool_if:N #1 { p , T , F , TF } 7422
7423 { 7423
7424 \if_bool:N #1 7424
7425 \prg_return_true: 7425
7426 \else: 7426
7427 \prg_return_false: 7427
7428 \fi: 7428
7429 } 7429
7430 \prg_generate_conditional_variant:Nnn \bool_if:N { c } { p , T , F , TF } 7430
7431 \cs_new:Npe \bool_to_str:N #1 7431
7432 { 7432
7433 \exp_not:N \bool_if:NTF #1 7433
7434 { \tl_to_str:n { true } } { \tl_to_str:n { false } } 7434
7435 } 7435
7436 \cs_generate_variant:Nn \bool_to_str:N { c } 7436
7437 \cs_new:Npe \bool_to_str:n #1 7437
7438 { 7438
7439 \exp_not:N \bool_if:nTF {#1} 7439
7440 { \tl_to_str:n { true } } { \tl_to_str:n { false } } 7440
7441 } 7441
7442 \cs_new_protected:Npn \bool_show:n 7442
7443 { \__kernel_msg_show_eval:Nn \bool_to_str:n } 7443
7444 \cs_new_protected:Npn \bool_log:n 7444
7445 { \__kernel_msg_log_eval:Nn \bool_to_str:n } 7445
7446 \cs_new_protected:Npn \bool_show:N { \__bool_show:NN \tl_show:n } 7446
7447 \cs_generate_variant:Nn \bool_show:N { c } 7447
7448 \cs_new_protected:Npn \bool_log:N { \__bool_show:NN \tl_log:n } 7448
7449 \cs_generate_variant:Nn \bool_log:N { c } 7449
7450 \cs_new_protected:Npn \__bool_show:NN #1#2 7450
7451 { 7451
7452 \__kernel_chk_defined:NT #2 7452
```



```

7453     {
7454         \token_case_meaning:NnF #2
7455         {
7456             \c_true_bool { \exp_args:Ne #1 { \token_to_str:N #2 = true } }
7457             \c_false_bool { \exp_args:Ne #1 { \token_to_str:N #2 = false } }
7458         }
7459         {
7460             \msg_error:nneee { kernel } { bad-type }
7461             { \token_to_str:N #2 } { \token_to_meaning:N #2 } { bool }
7462         }
7463     }
7464 }
7465 \bool_new:N \l_tmpa_bool
7466 \bool_new:N \l_tmpb_bool
7467 \bool_new:N \g_tmpa_bool
7468 \bool_new:N \g_tmpb_bool
7469 \prg_new_eq_conditional:NNn \bool_if_exist:N \cs_if_exist:N
7470 { TF , T , F , p }
7471 \prg_new_eq_conditional:NNn \bool_if_exist:c \cs_if_exist:c
7472 { TF , T , F , p }
7473 \prg_new_conditional:Npnn \bool_if:n #1 { T , F , TF }
7474 {
7475     \if_predicate:w \bool_if_p:n {#1}
7476     \prg_return_true:
7477     \else:
7478     \prg_return_false:
7479     \fi:
7480 }
7481 \cs_new:Npn \bool_if_p:n { \exp_args:Nf \__bool_if_p:n }
7482 \cs_new:Npn \__bool_if_p:n #1
7483 {
7484     \tl_if_empty:oT { \use_none:nn #1 . } { \__bool_if_p_aux:w }
7485     \group_align_safe_begin:
7486     \exp_after:wN
7487     \group_align_safe_end:
7488     \exp:w \exp_end_continue_f:w % (
7489     \__bool_get_next:NN \use_i:nnnn #1 )
7490 }
7491 \cs_new:Npn \__bool_if_p_aux:w #1 \use_i:nnnn #2#3
7492 { \bool_if:NTF #2 \c_true_bool \c_false_bool }
7493 \cs_new:Npn \__bool_get_next:NN #1#2
7494 {
7495     \use:c
7496     {
7497         __bool_
7498         \if_meaning:w !#2 ! \else: \if_meaning:w (#2 ( \else: p \fi: \fi:

```

```

7499         :Nw
7500     }
7501     #1 #2
7502 }
7503 \cs_new:cpn { __bool_!:Nw } #1#2
7504 {
7505     \exp_after:wN \__bool_get_next:NN
7506     #1 \use_ii:nnnn \use_i:nnnn \use_iii:nnnn \use_iv:nnnn
7507 }
7508 \cs_new:cpn { __bool_(:Nw } #1#2
7509 {
7510     \exp_after:wN \__bool_choose:NNN \exp_after:wN #1
7511     \int_value:w \__bool_get_next:NN \use_i:nnnn
7512 }
7513 \cs_new:cpn { __bool_p:Nw } #1
7514 { \exp_after:wN \__bool_choose:NNN \exp_after:wN #1 \int_value:w }
7515 \cs_new:Npn \__bool_choose:NNN #1#2#3
7516 {
7517     \use:c
7518     {
7519         __bool_ \token_to_str:N #3 _
7520         #1 #2 { \if_meaning:w 0 #2 1 \else: 0 \fi: } 2 0 :
7521     }
7522 }
7523 \cs_new:cpn { __bool_)_0: } { \c_false_bool }
7524 \cs_new:cpn { __bool_)_1: } { \c_true_bool }
7525 \cs_new:cpn { __bool_)_2: } { \c_true_bool }
7526 \cs_new:cpn { __bool_&_0: } & { \__bool_get_next:NN \use_iv:nnnn }
7527 \cs_new:cpn { __bool_&_1: } & { \__bool_get_next:NN \use_i:nnnn }
7528 \cs_new:cpn { __bool_&_2: } & { \__bool_get_next:NN \use_iii:nnnn }
7529 \cs_new:cpn { __bool_|_0: } | { \__bool_get_next:NN \use_i:nnnn }
7530 \cs_new:cpn { __bool_|_1: } | { \__bool_get_next:NN \use_iii:nnnn }
7531 \cs_new:cpn { __bool_|_2: } | { \__bool_get_next:NN \use_iii:nnnn }
7532 \cs_new:Npn \bool_lazy_all_p:n #1
7533 { \__bool_lazy_all:n #1 \q_bool_recursion_tail \q_bool_recursion_stop }
7534 \prg_new_conditional:Npnn \bool_lazy_all:n #1 { T , F , TF }
7535 {
7536     \if_predicate:w \bool_lazy_all_p:n {#1}
7537     \prg_return_true:
7538     \else:
7539     \prg_return_false:
7540     \fi:
7541 }
7542 \cs_new:Npn \__bool_lazy_all:n #1
7543 {
7544     \__bool_if_recursion_tail_stop_do:nn {#1} { \c_true_bool }

```

7545	\bool_if:nF {#1}	7545
7546	{ __bool_use_i_delimit_by_q_recursion_stop:nw { \c_false_bool } }	7546
7547	__bool_lazy_all:n	7547
7548	}	7548
7549	\prg_new_conditional:Npnn \bool_lazy_and:nn #1#2 { p , T , F , TF }	7549
7550	{	7550
7551	\if_predicate:w	7551
7552	\bool_if:nTF {#1} { \bool_if_p:n {#2} } { \c_false_bool }	7552
7553	\prg_return_true:	7553
7554	\else:	7554
7555	\prg_return_false:	7555
7556	\fi:	7556
7557	}	7557
7558	\cs_new:Npn \bool_lazy_any_p:n #1	7558
7559	{ __bool_lazy_any:n #1 \q_bool_recursion_tail \q_bool_recursion_stop }	7559
7560	\prg_new_conditional:Npnn \bool_lazy_any:n #1 { T , F , TF }	7560
7561	{	7561
7562	\if_predicate:w \bool_lazy_any_p:n {#1}	7562
7563	\prg_return_true:	7563
7564	\else:	7564
7565	\prg_return_false:	7565
7566	\fi:	7566
7567	}	7567
7568	\cs_new:Npn __bool_lazy_any:n #1	7568
7569	{	7569
7570	__bool_if_recursion_tail_stop_do:nn {#1} { \c_false_bool }	7570
7571	\bool_if:nT {#1}	7571
7572	{ __bool_use_i_delimit_by_q_recursion_stop:nw { \c_true_bool } }	7572
7573	__bool_lazy_any:n	7573
7574	}	7574
7575	\prg_new_conditional:Npnn \bool_lazy_or:nn #1#2 { p , T , F , TF }	7575
7576	{	7576
7577	\if_predicate:w	7577
7578	\bool_if:nTF {#1} { \c_true_bool } { \bool_if_p:n {#2} }	7578
7579	\prg_return_true:	7579
7580	\else:	7580
7581	\prg_return_false:	7581
7582	\fi:	7582
7583	}	7583
7584	\cs_new:Npn \bool_not_p:n #1 { \bool_if_p:n { ! (#1) } }	7584
7585	\prg_new_conditional:Npnn \bool_xor:nn #1#2 { p , T , F , TF }	7585
7586	{	7586
7587	\bool_if:nT {#1} \reverse_if:N	7587
7588	\if_predicate:w \bool_if_p:n {#2}	7588
7589	\prg_return_true:	7589
7590	\else:	7590

7591	\prg_return_false:	7591
7592	\fi:	7592
7593	}	7593
7594	\cs_new:Npn \bool_while_do:Nn #1#2	7594
7595	{ \bool_if:NT #1 { #2 \bool_while_do:Nn #1 {#2} } }	7595
7596	\cs_new:Npn \bool_until_do:Nn #1#2	7596
7597	{ \bool_if:NF #1 { #2 \bool_until_do:Nn #1 {#2} } }	7597
7598	\cs_generate_variant:Nn \bool_while_do:Nn { c }	7598
7599	\cs_generate_variant:Nn \bool_until_do:Nn { c }	7599
7600	\cs_new:Npn \bool_do_while:Nn #1#2	7600
7601	{ #2 \bool_if:NT #1 { \bool_do_while:Nn #1 {#2} } }	7601
7602	\cs_new:Npn \bool_do_until:Nn #1#2	7602
7603	{ #2 \bool_if:NF #1 { \bool_do_until:Nn #1 {#2} } }	7603
7604	\cs_generate_variant:Nn \bool_do_while:Nn { c }	7604
7605	\cs_generate_variant:Nn \bool_do_until:Nn { c }	7605
7606	\cs_new:Npn \bool_while_do:nn #1#2	7606
7607	{	7607
7608	\bool_if:nT {#1}	7608
7609	{	7609
7610	#2	7610
7611	\bool_while_do:nn {#1} {#2}	7611
7612	}	7612
7613	}	7613
7614	\cs_new:Npn \bool_do_while:nn #1#2	7614
7615	{	7615
7616	#2	7616
7617	\bool_if:nT {#1} { \bool_do_while:nn {#1} {#2} }	7617
7618	}	7618
7619	\cs_new:Npn \bool_until_do:nn #1#2	7619
7620	{	7620
7621	\bool_if:nF {#1}	7621
7622	{	7622
7623	#2	7623
7624	\bool_until_do:nn {#1} {#2}	7624
7625	}	7625
7626	}	7626
7627	\cs_new:Npn \bool_do_until:nn #1#2	7627
7628	{	7628
7629	#2	7629
7630	\bool_if:nF {#1} { \bool_do_until:nn {#1} {#2} }	7630
7631	}	7631
7632	\scan_new:N \s__bool_mark	7632
7633	\scan_new:N \s__bool_stop	7633
7634	\cs_new:Npn \bool_case:nTF	7634
7635	{ \exp:w __bool_case:nTF }	7635
7636	\cs_new:Npn \bool_case:nT #1#2	7636

[illegible]

```
7683 \cs_new:cpn { __prg_replicate_8:n } #1 7683
7684   { \cs_end: {#1#1#1#1#1#1#1#1#1#1} #1#1#1#1#1#1#1#1#1#1 } 7684
7685 \cs_new:cpn { __prg_replicate_9:n } #1 7685
7686   { \cs_end: {#1#1#1#1#1#1#1#1#1#1} #1#1#1#1#1#1#1#1#1#1 } 7686
7687 \cs_new:cpn { __prg_replicate_first_-:n } #1 7687
7688   { 7688
7689     \exp_end: 7689
7690     \msg_expandable_error:nn { prg } { negative-replication } 7690
7691   } 7691
7692 \cs_new:cpn { __prg_replicate_first_0:n } #1 { \exp_end: } 7692
7693 \cs_new:cpn { __prg_replicate_first_1:n } #1 { \exp_end: #1 } 7693
7694 \cs_new:cpn { __prg_replicate_first_2:n } #1 { \exp_end: #1#1 } 7694
7695 \cs_new:cpn { __prg_replicate_first_3:n } #1 { \exp_end: #1#1#1 } 7695
7696 \cs_new:cpn { __prg_replicate_first_4:n } #1 { \exp_end: #1#1#1#1 } 7696
7697 \cs_new:cpn { __prg_replicate_first_5:n } #1 { \exp_end: #1#1#1#1#1 } 7697
7698 \cs_new:cpn { __prg_replicate_first_6:n } #1 { \exp_end: #1#1#1#1#1#1 } 7698
7699 \cs_new:cpn { __prg_replicate_first_7:n } #1 { \exp_end: #1#1#1#1#1#1#1 } 7699
7700 \cs_new:cpn { __prg_replicate_first_8:n } #1 { \exp_end: #1#1#1#1#1#1#1#1 } 7700
7701 \cs_new:cpn { __prg_replicate_first_9:n } #1 7701
7702   { \exp_end: #1#1#1#1#1#1#1#1#1#1 } 7702
7703 \prg_new_conditional:Nppn \mode_if_vertical: { p , T , F , TF } 7703
7704   { \if_mode_vertical: \prg_return_true: \else: \prg_return_false: \fi: } 7704
7705 \prg_new_conditional:Nppn \mode_if_horizontal: { p , T , F , TF } 7705
7706   { \if_mode_horizontal: \prg_return_true: \else: \prg_return_false: \fi: } 7706
7707 \prg_new_conditional:Nppn \mode_if_inner: { p , T , F , TF } 7707
7708   { \if_mode_inner: \prg_return_true: \else: \prg_return_false: \fi: } 7708
7709 \prg_new_conditional:Nppn \mode_if_math: { p , T , F , TF } 7709
7710   { \if_mode_math: \prg_return_true: \else: \prg_return_false: \fi: } 7710
7711 \group_begin: 7711
7712 \tex_catcode:D \^^@ = 2 \exp_stop_f: 7712
7713 \cs_new:Npn \group_align_safe_begin: 7713
7714   { \exp:w \if_false: { \fi: ^^@ \exp_stop_f: } 7714
7715 \tex_catcode:D \^^@ = 1 \exp_stop_f: 7715
7716 \cs_new:Npn \group_align_safe_end: 7716
7717   { \exp:w ^^@ \if_false: } \fi: \exp_stop_f: } 7717
7718 \group_end: 7718
7719 \int_new:N \g__kernel_prg_map_int 7719
7720 % File: l3sys.dtx 7720
7721 \tl_new:N \l__sys_tmp_tl 7721
7722 \cs_new_protected:Npn \__sys_const:nn #1#2 7722
7723   { 7723
7724     \bool_if:nTF {#2} 7724
7725     { 7725
7726       \cs_new_eq:cN { #1 :T } \use:n 7726
7727       \cs_new_eq:cN { #1 :F } \use_none:n 7727
7728       \cs_new_eq:cN { #1 :TF } \use_i:nn 7728
```



```

7729 \cs_new_eq:cN { #1 _p: } \c_true_bool 7729
7730 } 7730
7731 { 7731
7732 \cs_new_eq:cN { #1 :T } \use_none:n 7732
7733 \cs_new_eq:cN { #1 :F } \use:n 7733
7734 \cs_new_eq:cN { #1 :TF } \use_ii:nn 7734
7735 \cs_new_eq:cN { #1 _p: } \c_false_bool 7735
7736 } 7736
7737 } 7737
7738 \str_const:Ne \c_sys_engine_str 7738
7739 { 7739
7740 \cs_if_exist:NT \tex luatexversion:D { luatex } 7740
7741 \cs_if_exist:NT \tex pdftexversion:D { pdftex } 7741
7742 \cs_if_exist:NT \tex kanjiskip:D 7742
7743 { 7743
7744 \cs_if_exist:NTF \tex enablecjktoken:D 7744
7745 { uptex } 7745
7746 { ptex } 7746
7747 } 7747
7748 \cs_if_exist:NT \tex XeTeXversion:D { xetex } 7748
7749 } 7749
7750 \tl_map_inline:nn { { luatex } { pdftex } { ptex } { uptex } { xetex } } 7750
7751 { 7751
7752 \__sys_const:nn { sys_if_engine_ #1 } 7752
7753 { \str_if_eq_p:Vn \c_sys_engine_str {#1} } 7753
7754 } 7754
7755 \__sys_const:nn 7755
7756 { sys_if_engine_opentype } 7756
7757 { \cs_if_exist_p:N \tex_Umathcode:D } 7757
7758 \group_begin: 7758
7759 \cs_set_eq:NN \lua_now:e \tex_directlua:D 7759
7760 \str_const:Ne \c_sys_engine_exec_str 7760
7761 { 7761
7762 \sys_if_engine_pdftex:T { pdf } 7762
7763 \sys_if_engine_xetex:T { xe } 7763
7764 \sys_if_engine_ptex:T { ep } 7764
7765 \sys_if_engine_uptex:T { eup } 7765
7766 \sys_if_engine_luatex:T 7766
7767 { 7767
7768 lua \lua_now:e 7768
7769 { 7769
7770 if (pcall(require, 'luaharfbuzz')) then ~ 7770
7771 tex.print("hb") ~ 7771
7772 end 7772
7773 } 7773
7774 } 7774

```

```
7775 tex 7775
7776 } 7776
7777 \group_end: 7777
7778 \str_const:Ne \c_sys_engine_format_str 7778
7779 { 7779
7780 \cs_if_exist:NTF \fmtname 7780
7781 { 7781
7782 \bool_lazy_or:nnTF 7782
7783 { \str_if_eq_p:Vn \fmtname { plain } } 7783
7784 { \str_if_eq_p:Vn \fmtname { LaTeX2e } } 7784
7785 { 7785
7786 \sys_if_engine_pdftex:T 7786
7787 { \int_compare:nNnT { \tex_pdfoutput:D } = { 1 } { pdf } } 7787
7788 \sys_if_engine_xetex:T { xe } 7788
7789 \sys_if_engine_ptex:T { p } 7789
7790 \sys_if_engine_uptex:T { up } 7790
7791 \sys_if_engine luatex:T 7791
7792 { 7792
7793 \int_compare:nNnT { \tex_pdfoutput:D } = { 0 } { dvi } 7793
7794 lua 7794
7795 } 7795
7796 \str_if_eq:VnTF \fmtname { LaTeX2e } 7796
7797 { latex } 7797
7798 { 7798
7799 \bool_lazy_and:nnT 7799
7800 { \sys_if_engine_pdftex_p: } 7800
7801 { \int_compare_p:nNn { \tex_pdfoutput:D } = { 0 } } 7801
7802 { e } 7802
7803 tex 7803
7804 } 7804
7805 } 7805
7806 { \fmtname } 7806
7807 } 7807
7808 { unknown } 7808
7809 } 7809
7810 \str_const:Ne \c_sys_engine_version_str 7810
7811 { 7811
7812 \str_case:on \c_sys_engine_str 7812
7813 { 7813
7814 { pdftex } 7814
7815 { 7815
7816 \int_div_truncate:nn { \tex_pdftexversion:D } { 100 } 7816
7817 . 7817
7818 \int_mod:nn { \tex_pdftexversion:D } { 100 } 7818
7819 . 7819
7820 \tex_pdftexrevision:D 7820
```

7821	}	7821
7822	{ ptex }	7822
7823	{	7823
7824	\cs_if_exist:NT \tex_ptexversion:D	7824
7825	{	7825
7826	p	7826
7827	\int_use:N \tex_ptexversion:D	7827
7828	.	7828
7829	\int_use:N \tex_ptexminorversion:D	7829
7830	\tex_ptexrevision:D	7830
7831	-	7831
7832	\int_use:N \tex_epTeXversion:D	7832
7833	}	7833
7834	}	7834
7835	{ luatex }	7835
7836	{	7836
7837	\int_div_truncate:nn { \tex_luatexversion:D } { 100 }	7837
7838	.	7838
7839	\int_mod:nn { \tex_luatexversion:D } { 100 }	7839
7840	.	7840
7841	\tex_luatexrevision:D	7841
7842	}	7842
7843	{ uptex }	7843
7844	{	7844
7845	\cs_if_exist:NT \tex_ptexversion:D	7845
7846	{	7846
7847	p	7847
7848	\int_use:N \tex_ptexversion:D	7848
7849	.	7849
7850	\int_use:N \tex_ptexminorversion:D	7850
7851	\tex_ptexrevision:D	7851
7852	-	7852
7853	u	7853
7854	\int_use:N \tex_uptexversion:D	7854
7855	\tex_uptexrevision:D	7855
7856	-	7856
7857	\int_use:N \tex_epTeXversion:D	7857
7858	}	7858
7859	}	7859
7860	{ xetex }	7860
7861	{	7861
7862	\int_use:N \tex_XeTeXversion:D	7862
7863	\tex_XeTeXrevision:D	7863
7864	}	7864
7865	}	7865
7866	}	7866

```

7867 \cs_new_protected:Npn \sys_load_backend:n #1
7868 {
7869     \sys_finalise:
7870     \str_if_exist:NTF \c_sys_backend_str
7871     {
7872         \str_if_eq:VnF \c_sys_backend_str {#1}
7873         { \msg_error:nn { sys } { backend-set } }
7874     }
7875     {
7876         \tl_if_blank:nF {#1}
7877         { \tl_gset:Nn \g__sys_backend_tl {#1} }
7878         \__sys_load_backend_check:N \g__sys_backend_tl
7879         \str_const:Ne \c_sys_backend_str { \g__sys_backend_tl }
7880         \__kernel_sys_configuration_load:n
7881         { l3backend- \c_sys_backend_str }
7882     }
7883 }
7884 \cs_new_protected:Npn \__sys_load_backend_check:N #1
7885 {
7886     \sys_if_engine_xetex:TF
7887     {
7888         \str_case:VnF #1
7889         {
7890             { dvisvgm } { }
7891             { xdvipdfmx } { \tl_gset:Nn #1 { xetex } }
7892             { xetex } { }
7893         }
7894         {
7895             \msg_error:nnee { sys } { wrong-backend }
7896             #1 { xetex }
7897             \tl_gset:Nn #1 { xetex }
7898         }
7899     }
7900     {
7901         \sys_if_output_pdf:TF
7902         {
7903             \str_if_eq:VnTF #1 { pdfmode }
7904             {
7905                 \sys_if_engine luatex:TF
7906                 { \tl_gset:Nn #1 { luatex } }
7907                 { \tl_gset:Nn #1 { pdftex } }
7908             }
7909             {
7910                 \bool_lazy_or:nnF
7911                 { \str_if_eq_p:Vn #1 { luatex } }
7912                 { \str_if_eq_p:Vn #1 { pdftex } }

```

```

7913         {
7914             \msg_error:nnee { sys } { wrong-backend }
7915             #1 { \sys_if_engine luatex:TF { luatex } { pdftex } }
7916             \sys_if_engine luatex:TF
7917                 { \tl_gset:Nn #1 { luatex } }
7918                 { \tl_gset:Nn #1 { pdftex } }
7919         }
7920     }
7921 }
7922 {
7923     \str_case:VnF #1
7924     {
7925         { dvipdfmx } { }
7926         { dvips } { }
7927         { dvisvgm } { }
7928     }
7929     {
7930         \msg_error:nnee { sys } { wrong-backend }
7931         #1 { dvips }
7932         \tl_gset:Nn #1 { dvips }
7933     }
7934 }
7935 }
7936 }
7937 \cs_new_protected:Npn \sys_ensure_backend:
7938 {
7939     \str_if_exist:NF \c_sys_backend_str
7940     { \sys_load_backend:n { } }
7941 }
7942 \bool_new:N \g__sys_debug_bool
7943 \cs_new_protected:Npn \sys_load_debug:
7944 {
7945     \bool_if:NF \g__sys_debug_bool
7946     { \__kernel_sys_configuration_load:n { l3debug } }
7947     \bool_gset_true:N \g__sys_debug_bool
7948 }
7949 \cs_if_exist:NT \@expl@finalise@setup@@
7950 {
7951     \tl_gput_right:Nn \@expl@finalise@setup@@
7952     {
7953         \tl_gput_right:Nn \@kernel@after@begindocument
7954         {
7955             \cs_gset_protected:Npn \sys_load_debug:
7956             { \msg_error:nn { sys } { load-debug-in-preamble } }
7957         }
7958     }

```

```

7959 }
7960 \tl_new:N \l__sys_internal_tl
7961 \tl_const:N \c__sys_marker_tl { : \token_to_str:N : }
7962 \cs_new_protected:Npn \sys_get_shell:nnN #1#2#3
7963 {
7964   \sys_get_shell:nnNF {#1} {#2} #3
7965   { \tl_set:Nn #3 { \q_no_value } }
7966 }
7967 \prg_new_protected_conditional:Npnn \sys_get_shell:nnN #1#2#3 { T , F , TF }
7968 {
7969   \sys_if_shell:TF
7970   { \exp_args:No \__sys_get:nnN { \tl_to_str:n {#1} } {#2} #3 }
7971   { \prg_return_false: }
7972 }
7973 \cs_new_protected:Npn \__sys_get:nnN #1#2#3
7974 {
7975   \tl_if_in:nnTF {#1} { " }
7976   {
7977     \msg_error:nne
7978     { kernel } { quote-in-shell } {#1}
7979     \prg_return_false:
7980   }
7981   {
7982     \group_begin:
7983     \if_false: { \fi:
7984       \int_set_eq:NN \tex_tracingnesting:D \c_zero_int
7985       \exp_args:No \tex_everyeof:D { \c__sys_marker_tl }
7986       #2 \scan_stop:
7987       \exp_after:wN \__sys_get_do:Nw
7988       \exp_after:wN #3
7989       \exp_after:wN \prg_do_nothing:
7990       \tex_input:D | "#1" \scan_stop:
7991       \if_false: } \fi:
7992       \prg_return_true:
7993     }
7994   }
7995   \exp_args:Nno \use:nn
7996   { \cs_new_protected:Npn \__sys_get_do:Nw #1#2 }
7997   { \c__sys_marker_tl }
7998   {
7999     \group_end:
8000     \tl_set:No #1 {#2}
8001   }
8002   \sys_if_engine luatex:F
8003   { \int_const:Nn \c__sys_shell_stream_int { 18 } }
8004   \sys_if_engine luatex:TF

```


8005	{	8005
8006	\cs_new_protected:Npn \sys_shell_now:n #1	8006
8007	{ __sys_shell_now:e { \exp_not:n {#1} } }	8007
8008	}	8008
8009	{	8009
8010	\cs_new_protected:Npn \sys_shell_now:n #1	8010
8011	{ \iow_now:Nn \c__sys_shell_stream_int {#1} }	8011
8012	}	8012
8013	\cs_generate_variant:Nn \sys_shell_now:n { e , x }	8013
8014	\sys_if_engine luatex:TF	8014
8015	{	8015
8016	\cs_new_protected:Npn \sys_shell_shipout:n #1	8016
8017	{ __sys_shell_shipout:e { \exp_not:n {#1} } }	8017
8018	}	8018
8019	{	8019
8020	\cs_new_protected:Npn \sys_shell_shipout:n #1	8020
8021	{ \iow_shipout:Nn \c__sys_shell_stream_int {#1} }	8021
8022	}	8022
8023	\cs_generate_variant:Nn \sys_shell_shipout:n { e , x }	8023
8024	\cs_new_protected:Npn __kernel_sys_everyjob:	8024
8025	{	8025
8026	\tl_use:N \g__sys_everyjob_tl	8026
8027	\tl_gclear:N \g__sys_everyjob_tl	8027
8028	}	8028
8029	\cs_new_protected:Npn __sys_everyjob:n #1	8029
8030	{ \tl_gput_right:Nn \g__sys_everyjob_tl {#1} }	8030
8031	\tl_new:N \g__sys_everyjob_tl	8031
8032	__sys_everyjob:n	8032
8033	{ \cs_new_eq:NN \c_sys_jobname_str \tex_jobname:D }	8033
8034	__sys_everyjob:n	8034
8035	{	8035
8036	\group_begin:	8036
8037	\cs_set:Npn __sys_tmp:w #1	8037
8038	{	8038
8039	\str_if_eq:eeTF { \cs_meaning:N #1 } { \token_to_str:N #1 }	8039
8040	{ #1 }	8040
8041	{	8041
8042	\cs_if_exist:NTF \tex_primitive:D	8042
8043	{	8043
8044	\bool_lazy_and:nnTF	8044
8045	{ \sys_if_engine_xetex_p: }	8045
8046	{	8046
8047	\int_compare_p:nNn	8047
8048	{ \exp_after:wN \use_none:n \tex_XeTeXrevision:D }	8048
8049	< { 99999 }	8049
8050	}	8050

```

8051         { 0 }
8052         { \tex_primitive:D #1 }
8053     }
8054     { 0 }
8055 }
8056 }
8057 \int_const:Nn \c_sys_minute_int
8058 { \int_mod:nn { \__sys_tmp:w \time } { 60 } }
8059 \int_const:Nn \c_sys_hour_int
8060 { \int_div_truncate:nn { \__sys_tmp:w \time } { 60 } }
8061 \int_const:Nn \c_sys_day_int { \__sys_tmp:w \day }
8062 \int_const:Nn \c_sys_month_int { \__sys_tmp:w \month }
8063 \int_const:Nn \c_sys_year_int { \__sys_tmp:w \year }
8064 \group_end:
8065 }
8066 \__sys_everyjob:n
8067 {
8068     \str_const:Ne \c_sys_timestamp_str
8069     {
8070         \cs_if_exist:NTF \tex_directlua:D
8071         { \tex_directlua:D { tex.print(pdf.getcreationdate()) } }
8072         { \tex_creationdate:D }
8073     }
8074 }
8075 \__sys_everyjob:n
8076 {
8077     \cs_new:Npn \sys_rand_seed: { \tex_the:D \tex_randomseed:D }
8078 }
8079 \__sys_everyjob:n
8080 {
8081     \cs_new_protected:Npn \sys_gset_rand_seed:n #1
8082     { \tex_setrandomseed:D \int_eval:n {#1} \exp_stop_f: }
8083 }
8084 \cs_new:Npe \sys_timer:
8085 {
8086     \sys_if_engine luatex:TF
8087     { \exp_not:N \__sys_elapsedtime: }
8088     { \exp_not:N \int_value:w \exp_not:N \tex_elapsedtime:D }
8089 }
8090 \__sys_everyjob:n
8091 {
8092     \int_const:Nn \c_sys_shell_escape_int
8093     {
8094         \sys_if_engine luatex:TF
8095         {
8096             \tex_directlua:D

```

```
8097         { tex.sprint(status.shell_escape~or~os.execute()) } 8097
8098     } 8098
8099     { \tex_shellescape:D } 8099
8100 } 8100
8101 } 8101
8102 \__sys_everyjob:n 8102
8103 { 8103
8104     \__sys_const:nn { sys_if_shell } 8104
8105     { \int_compare_p:nNn \c_sys_shell_escape_int > 0 } 8105
8106     \__sys_const:nn { sys_if_shell_unrestricted } 8106
8107     { \int_compare_p:nNn \c_sys_shell_escape_int = 1 } 8107
8108     \__sys_const:nn { sys_if_shell_restricted } 8108
8109     { \int_compare_p:nNn \c_sys_shell_escape_int = 2 } 8109
8110 } 8110
8111 \cs_new_protected:Npn \sys_get_query:nN #1#2 8111
8112 { \sys_get_query:nnnN {#1} { } { } #2 } 8112
8113 \cs_new_protected:Npn \sys_get_query:nnN #1#2#3 8113
8114 { \sys_get_query:nnnN {#1} { } {#2} #3 } 8114
8115 \cs_new_protected:Npn \sys_get_query:nnnN #1#2#3#4 8115
8116 { 8116
8117     \tl_clear:N #4 8117
8118     \__sys_get_query_auxi:neeN {#1} {#2} {#3} #4 8118
8119 } 8119
8120 \cs_new:Npn \__sys_get_query_auxi:nnnN #1#2#3#4 8120
8121 { 8121
8122     \__sys_get_query_auxii:neeN {#1} 8122
8123     { \tl_if_blank:nF {#2} { \tl_to_str:n { ~ #2 } } } 8123
8124     { 8124
8125         \tl_if_blank:nF {#3} 8125
8126         { 8126
8127             \c_space_tl 8127
8128             \sys_if_shell_restricted:F ' 8128
8129             \tl_to_str:n {#3} 8129
8130             \sys_if_shell_restricted:F ' 8130
8131         } 8131
8132     } 8132
8133     #4 8133
8134 } 8134
8135 \cs_generate_variant:Nn \__sys_get_query_auxi:nnnN { nee } 8135
8136 \cs_new_protected:Npn \__sys_get_query_auxii:nnnN #1#2#3#4 8136
8137 { 8137
8138     \sys_if_shell:T 8138
8139     { 8139
8140         \sys_get_shell:nnN 8140
8141         { l3sys-query~#1 #2 #3 } 8141
8142     } 8142
```

```

8143 \int_step_inline:nnn { 0 } { `A - 1 } 8143
8144 { \char_set_catcode_other:n {##1} } 8144
8145 \int_step_inline:nnn { `Z + 1 } { `a - 1 } 8145
8146 { \char_set_catcode_other:n {##1} } 8146
8147 \int_step_inline:nnn { `z + 1 } { 127 } 8147
8148 { \char_set_catcode_other:n {##1} } 8148
8149 \char_set_catcode_active:n { `\_ } 8149
8150 \tex_endlinechar:D 13 \scan_stop: 8150
8151 } 8151
8152 \l__sys_tmp_tl 8152
8153 \tl_if_empty:NF \l__sys_tmp_tl 8153
8154 { 8154
8155 \exp_after:wN \__sys_get_query:Nw \exp_after:wN #4 8155
8156 \l__sys_tmp_tl \q_stop 8156
8157 } 8157
8158 } 8158
8159 } 8159
8160 \cs_generate_variant:Nn \__sys_get_query_auxii:nnnN { nee } 8160
8161 \group_begin: 8161
8162 \tex_lccode:D `* = 13 \scan_stop: 8162
8163 \tex_lowercase:D 8163
8164 { 8164
8165 \group_end: 8165
8166 \cs_new_protected:Npn \__sys_get_query:Nw #1#2 * \q_stop 8166
8167 } 8167
8168 { \tl_set:Nn #1 {#2} } 8168
8169 \cs_new_protected:Npn \sys_split_query:nN #1#2 8169
8170 { \sys_split_query:nnnN {#1} { } { } #2 } 8170
8171 \cs_new_protected:Npn \sys_split_query:nnN #1#2#3 8171
8172 { \sys_split_query:nnnN {#1} { } {#2} #3 } 8172
8173 \group_begin: 8173
8174 \tex_lccode:D `* = 13 \scan_stop: 8174
8175 \tex_lowercase:D 8175
8176 { 8176
8177 \group_end: 8177
8178 \cs_new_protected:Npn \sys_split_query:nnnN #1#2#3#4 8178
8179 { 8179
8180 \seq_clear:N #4 8180
8181 \sys_get_query:nnnN {#1} {#2} {#3} \l__sys_tmp_tl 8181
8182 \tl_if_empty:NF \l__sys_tmp_tl 8182
8183 { \seq_set_split:NnV #4 * \l__sys_tmp_tl } 8183
8184 } 8184
8185 } 8185
8186 \__sys_everyjob:n 8186
8187 { \cs_gset_eq:NN \g_file_curr_name_str \tex_jobname:D } 8187
8188 \cs_new_protected:Npn \sys_finalise: 8188

```

```

8189 {
8190     \__kernel_sys_everyjob:
8191     \tl_use:N \g__sys_finalise_tl
8192     \tl_gclear:N \g__sys_finalise_tl
8193 }
8194 \cs_new_protected:Npn \__sys_finalise:n #1
8195 { \tl_gput_right:Nn \g__sys_finalise_tl {#1} }
8196 \tl_new:N \g__sys_finalise_tl
8197 \__sys_finalise:n
8198 {
8199     \str_const:Ne \c_sys_output_str
8200     {
8201         \int_compare:nNnTF
8202             { \cs_if_exist_use:NF \tex_pdfoutput:D { 0 } } > { 0 }
8203             { pdf }
8204             { dvi }
8205     }
8206     \__sys_const:nn { sys_if_output_dvi }
8207     { \str_if_eq_p:Vn \c_sys_output_str { dvi } }
8208     \__sys_const:nn { sys_if_output_pdf }
8209     { \str_if_eq_p:Vn \c_sys_output_str { pdf } }
8210 }
8211 \tl_new:N \g__sys_backend_tl
8212 \__sys_finalise:n
8213 {
8214     \__kernel_tl_gset:Nx \g__sys_backend_tl
8215     {
8216         \sys_if_engine_xetex:TF
8217         { xetex }
8218         {
8219             \sys_if_output_pdf:TF
8220             {
8221                 \sys_if_engine_pdftex:TF
8222                 { pdftex }
8223                 { luatex }
8224             }
8225             { dvips }
8226         }
8227     }
8228 }
8229 \__sys_finalise:n
8230 {
8231     \cs_if_exist:NT \@classoptionslist
8232     {
8233         \cs_if_eq:NNF \@classoptionslist \scan_stop:
8234         {

```

```

8235 \clist_map_inline:Nn \@classoptionslist
8236 {
8237     \str_case:nnT {#1}
8238     {
8239         { dvipdfmx }
8240         { \tl_gset:Nn \g__sys_backend_tl { dvipdfmx } }
8241         { dvips }
8242         { \tl_gset:Nn \g__sys_backend_tl { dvips } }
8243         { dvisvgm }
8244         { \tl_gset:Nn \g__sys_backend_tl { dvisvgm } }
8245         { pdftex }
8246         { \tl_gset:Nn \g__sys_backend_tl { pdfmode } }
8247         { xetex }
8248         { \tl_gset:Nn \g__sys_backend_tl { xdvipdfmx } }
8249     }
8250     { \clist_remove_all:Nn \@unusedoptionlist {#1} }
8251 }
8252 }
8253 }
8254 }
8255 %% File: l3clist.dtx
8256 \cs_new_eq:NN \c_empty_clist \c_empty_tl
8257 \tl_new:N \l__clist_internal_clist
8258 \scan_new:N \s__clist_mark
8259 \scan_new:N \s__clist_stop
8260 \cs_new:Npn \__clist_use_none_delimit_by_s_mark:w #1 \s__clist_mark { }
8261 \cs_new:Npn \__clist_use_none_delimit_by_s_stop:w #1 \s__clist_stop { }
8262 \cs_new:Npn \__clist_use_i_delimit_by_s_stop:nw #1 #2 \s__clist_stop {#1}
8263 \cs_new_protected:Npn \__clist_tmp:w { }
8264 \cs_new:Npn \__clist_trim_next:w #1 ,
8265 {
8266     \tl_if_empty:oTF { \use_none:nn #1 ? }
8267     { \__clist_trim_next:w \prg_do_nothing: }
8268     { \tl_trim_spaces_apply:oN {#1} \exp_end: }
8269 }
8270 \cs_new:Npn \__clist_sanitize:n #1
8271 {
8272     \exp_after:wN \__clist_sanitize:Nn \exp_after:wN \c_empty_tl
8273     \exp:w \__clist_trim_next:w \prg_do_nothing:
8274     #1 , \s__clist_stop \prg_break: , \prg_break_point:
8275 }
8276 \cs_new:Npn \__clist_sanitize:Nn #1#2
8277 {
8278     \__clist_use_none_delimit_by_s_stop:w #2 \s__clist_stop
8279     #1 \__clist_wrap_item:w #2 ,
8280     \exp_after:wN \__clist_sanitize:Nn \exp_after:wN ,

```



```
8281 \exp:w \__clist_trim_next:w \prg_do_nothing: 8281
8282 } 8282
8283 \prg_new_conditional:Npnn \__clist_if_wrap:n #1 { TF } 8283
8284 { 8284
8285 \tl_if_empty:oTF 8285
8286 { 8286
8287 \__clist_if_wrap:w 8287
8288 \s__clist_mark ? #1 ~ \s__clist_mark ? ~ #1 8288
8289 \s__clist_mark , ~ \s__clist_mark #1 , 8289
8290 } 8290
8291 { 8291
8292 \tl_if_head_is_group:nTF { #1 { } } 8292
8293 { 8293
8294 \tl_if_empty:nTF {#1} 8294
8295 { \prg_return_true: } 8295
8296 { 8296
8297 \tl_if_empty:oTF { \use_none:n #1} 8297
8298 { \prg_return_true: } 8298
8299 { \prg_return_false: } 8299
8300 } 8300
8301 } 8301
8302 { \prg_return_false: } 8302
8303 } 8303
8304 { \prg_return_true: } 8304
8305 } 8305
8306 \cs_new:Npn \__clist_if_wrap:w #1 \s__clist_mark ? ~ #2 ~ \s__clist_mark #3 , { } 8306
8307 \cs_new:Npn \__clist_wrap_item:w #1 , 8307
8308 { \__clist_if_wrap:nTF {#1} { \exp_not:n { {#1} } } { \exp_not:n {#1} } } 8308
8309 \cs_new_eq:NN \clist_new:N \tl_new:N 8309
8310 \cs_new_eq:NN \clist_new:c \tl_new:c 8310
8311 \cs_new_protected:Npn \clist_const:Nn #1#2 8311
8312 { \tl_const:Ne #1 { \__clist_sanitizе:n {#2} } } 8312
8313 \cs_generate_variant:Nn \clist_const:Nn { Ne , c , ce } 8313
8314 \cs_generate_variant:Nn \clist_const:Nn { Nx , cx } 8314
8315 \cs_new_eq:NN \clist_clear:N \tl_clear:N 8315
8316 \cs_new_eq:NN \clist_clear:c \tl_clear:c 8316
8317 \cs_new_eq:NN \clist_gclear:N \tl_gclear:N 8317
8318 \cs_new_eq:NN \clist_gclear:c \tl_gclear:c 8318
8319 \cs_new_eq:NN \clist_clear_new:N \tl_clear_new:N 8319
8320 \cs_new_eq:NN \clist_clear_new:c \tl_clear_new:c 8320
8321 \cs_new_eq:NN \clist_gclear_new:N \tl_gclear_new:N 8321
8322 \cs_new_eq:NN \clist_gclear_new:c \tl_gclear_new:c 8322
8323 \cs_new_eq:NN \clist_set_eq:NN \tl_set_eq:NN 8323
8324 \cs_new_eq:NN \clist_set_eq:Nc \tl_set_eq:Nc 8324
8325 \cs_new_eq:NN \clist_set_eq:cN \tl_set_eq:cN 8325
8326 \cs_new_eq:NN \clist_set_eq:cc \tl_set_eq:cc 8326
```

```
8327 \cs_new_eq:NN \clist_gset_eq:NN \tl_gset_eq:NN 8327
8328 \cs_new_eq:NN \clist_gset_eq:Nc \tl_gset_eq:Nc 8328
8329 \cs_new_eq:NN \clist_gset_eq:cN \tl_gset_eq:cN 8329
8330 \cs_new_eq:NN \clist_gset_eq:cc \tl_gset_eq:cc 8330
8331 \cs_new_protected:Npn \clist_set_from_seq:NN 8331
8332 { \__clist_set_from_seq:NNNN \clist_clear:N \__kernel_tl_set:Nx } 8332
8333 \cs_new_protected:Npn \clist_gset_from_seq:NN 8333
8334 { \__clist_set_from_seq:NNNN \clist_gclear:N \__kernel_tl_gset:Nx } 8334
8335 \cs_new_protected:Npn \__clist_set_from_seq:NNNN #1#2#3#4 8335
8336 { 8336
8337   \seq_if_empty:NTF #4 8337
8338   { #1 #3 } 8338
8339   { 8339
8340     #2 #3 8340
8341     { 8341
8342       \exp_after:wN \use_none:n \exp:w \exp_end_continue_f:w 8342
8343       \seq_map_function:NN #4 \__clist_set_from_seq:n 8343
8344     } 8344
8345   } 8345
8346 } 8346
8347 \cs_new:Npn \__clist_set_from_seq:n #1 8347
8348 { 8348
8349   , 8349
8350   \__clist_if_wrap:nTF {#1} 8350
8351   { \exp_not:n { {#1} } } 8351
8352   { \exp_not:n {#1} } 8352
8353 } 8353
8354 \cs_generate_variant:Nn \clist_set_from_seq:NN { Nc } 8354
8355 \cs_generate_variant:Nn \clist_set_from_seq:NN { c , cc } 8355
8356 \cs_generate_variant:Nn \clist_gset_from_seq:NN { Nc } 8356
8357 \cs_generate_variant:Nn \clist_gset_from_seq:NN { c , cc } 8357
8358 \cs_new_protected:Npn \clist_concat:NNN 8358
8359 { \__clist_concat:NNNN \__kernel_tl_set:Nx } 8359
8360 \cs_new_protected:Npn \clist_gconcat:NNN 8360
8361 { \__clist_concat:NNNN \__kernel_tl_gset:Nx } 8361
8362 \cs_new_protected:Npn \__clist_concat:NNNN #1#2#3#4 8362
8363 { 8363
8364   #1 #2 8364
8365   { 8365
8366     \exp_not:o #3 8366
8367     \clist_if_empty:NF #3 { \clist_if_empty:NF #4 { , } } 8367
8368     \exp_not:o #4 8368
8369   } 8369
8370 } 8370
8371 \cs_generate_variant:Nn \clist_concat:NNN { ccc } 8371
8372 \cs_generate_variant:Nn \clist_gconcat:NNN { ccc } 8372
```

```
8373 \prg_new_eq_conditional:NNn \clist_if_exist:N \cs_if_exist:N 8373
8374 { TF , T , F , p } 8374
8375 \prg_new_eq_conditional:NNn \clist_if_exist:c \cs_if_exist:c 8375
8376 { TF , T , F , p } 8376
8377 \cs_new_protected:Npn \clist_set:Nn #1#2 8377
8378 { \__kernel_tl_set:Nx #1 { \__clist_sanitiz:n {#2} } } 8378
8379 \cs_new_protected:Npn \clist_gset:Nn #1#2 8379
8380 { \__kernel_tl_gset:Nx #1 { \__clist_sanitiz:n {#2} } } 8380
8381 \cs_generate_variant:Nn \clist_set:Nn { NV , Ne , c , cV , ce } 8381
8382 \cs_generate_variant:Nn \clist_set:Nn { No , Nx , co , cx } 8382
8383 \cs_generate_variant:Nn \clist_gset:Nn { NV , Ne , c , cV , ce } 8383
8384 \cs_generate_variant:Nn \clist_gset:Nn { No , Nx , co , cx } 8384
8385 \cs_new_protected:Npn \clist_put_left:Nn 8385
8386 { \__clist_put_left:NNNn \clist_concat:NNN \clist_set:Nn } 8386
8387 \cs_new_protected:Npn \clist_gput_left:Nn 8387
8388 { \__clist_put_left:NNNn \clist_gconcat:NNN \clist_set:Nn } 8388
8389 \cs_new_protected:Npn \__clist_put_left:NNNn #1#2#3#4 8389
8390 { 8390
8391 #2 \l__clist_internal_clist {#4} 8391
8392 #1 #3 \l__clist_internal_clist #3 8392
8393 } 8393
8394 \cs_generate_variant:Nn \clist_put_left:Nn { NV , Nv , Ne , c , cV , cv , ce } 8394
8395 \cs_generate_variant:Nn \clist_put_left:Nn { No , Nx , co , cx } 8395
8396 \cs_generate_variant:Nn \clist_gput_left:Nn { NV , Nv , Ne , c , cV , cv , ce } 8396
8397 \cs_generate_variant:Nn \clist_gput_left:Nn { No , Nx , co , cx } 8397
8398 \cs_new_protected:Npn \clist_put_right:Nn 8398
8399 { \__clist_put_right:NNNn \clist_concat:NNN \clist_set:Nn } 8399
8400 \cs_new_protected:Npn \clist_gput_right:Nn 8400
8401 { \__clist_put_right:NNNn \clist_gconcat:NNN \clist_set:Nn } 8401
8402 \cs_new_protected:Npn \__clist_put_right:NNNn #1#2#3#4 8402
8403 { 8403
8404 #2 \l__clist_internal_clist {#4} 8404
8405 #1 #3 #3 \l__clist_internal_clist 8405
8406 } 8406
8407 \cs_generate_variant:Nn \clist_put_right:Nn 8407
8408 { NV , Nv , Ne , c , cV , cv , ce } 8408
8409 \cs_generate_variant:Nn \clist_put_right:Nn 8409
8410 { No , Nx , co , cx } 8410
8411 \cs_generate_variant:Nn \clist_gput_right:Nn 8411
8412 { NV , Nv , Ne , c , cV , cv , ce } 8412
8413 \cs_generate_variant:Nn \clist_gput_right:Nn 8413
8414 { No , Nx , co , cx } 8414
8415 \cs_new_protected:Npn \clist_get:NN #1#2 8415
8416 { 8416
8417 \if_meaning:w #1 \c_empty_clist 8417
8418 \tl_set:Nn #2 { \q_no_value } 8418
```

8419	\else:	8419
8420	\exp_after:wN __clist_get:wN #1 , \s__clist_stop #2	8420
8421	\fi:	8421
8422	}	8422
8423	\cs_new_protected:Npn __clist_get:wN #1 , #2 \s__clist_stop #3	8423
8424	{ \tl_set:Nn #3 {#1} }	8424
8425	\cs_generate_variant:Nn \clist_get:NN { c }	8425
8426	\cs_new_protected:Npn \clist_pop:NN	8426
8427	{ __clist_pop:NNN __kernel_tl_set:Nx }	8427
8428	\cs_new_protected:Npn \clist_gpop:NN	8428
8429	{ __clist_pop:NNN __kernel_tl_gset:Nx }	8429
8430	\cs_new_protected:Npn __clist_pop:NNN #1#2#3	8430
8431	{	8431
8432	\if_meaning:w #2 \c_empty_clist	8432
8433	\tl_set:Nn #3 { \q_no_value }	8433
8434	\else:	8434
8435	\exp_after:wN __clist_pop:wwNNN #2 , \s__clist_mark \s__clist_stop #1#2#3	8435
8436	\fi:	8436
8437	}	8437
8438	\cs_new_protected:Npn __clist_pop:wwNNN #1 , #2 \s__clist_stop #3#4#5	8438
8439	{	8439
8440	\tl_set:Nn #5 {#1}	8440
8441	#3 #4	8441
8442	{	8442
8443	__clist_pop:wN \prg_do_nothing:	8443
8444	#2 \exp_not:o	8444
8445	, \s__clist_mark \use_none:n	8445
8446	\s__clist_stop	8446
8447	}	8447
8448	}	8448
8449	\cs_new:Npn __clist_pop:wN #1 , \s__clist_mark #2 #3 \s__clist_stop { #2 {#1} }	8449
8450	\cs_generate_variant:Nn \clist_pop:NN { c }	8450
8451	\cs_generate_variant:Nn \clist_gpop:NN { c }	8451
8452	\prg_new_protected_conditional:Npnn \clist_get:NN #1#2 { T , F , TF }	8452
8453	{	8453
8454	\if_meaning:w #1 \c_empty_clist	8454
8455	\prg_return_false:	8455
8456	\else:	8456
8457	\exp_after:wN __clist_get:wN #1 , \s__clist_stop #2	8457
8458	\prg_return_true:	8458
8459	\fi:	8459
8460	}	8460
8461	\prg_generate_conditional_variant:Nnn \clist_get:NN { c } { T , F , TF }	8461
8462	\prg_new_protected_conditional:Npnn \clist_pop:NN #1#2 { T , F , TF }	8462
8463	{ __clist_pop_TF:NNN __kernel_tl_set:Nx #1 #2 }	8463
8464	\prg_new_protected_conditional:Npnn \clist_gpop:NN #1#2 { T , F , TF }	8464

```
8465 { \__clist_pop_TF:NNN \__kernel_tl_gset:Nx #1 #2 } 8465
8466 \cs_new_protected:Npn \__clist_pop_TF:NNN #1#2#3 8466
8467 { 8467
8468   \if_meaning:w #2 \c_empty_clist 8468
8469   \prg_return_false: 8469
8470   \else: 8470
8471     \exp_after:wN \__clist_pop:wwNNN #2 , \s__clist_mark \s__clist_stop #1#2#3 8471
8472     \prg_return_true: 8472
8473   \fi: 8473
8474 } 8474
8475 \prg_generate_conditional_variant:Nnn \clist_pop:NN { c } { T , F , TF } 8475
8476 \prg_generate_conditional_variant:Nnn \clist_gpop:NN { c } { T , F , TF } 8476
8477 \cs_new_eq:NN \clist_push:Nn \clist_put_left:Nn 8477
8478 \cs_generate_variant:Nn \clist_push:Nn { NV , No , Nx , c , cV , co , cx } 8478
8479 \cs_new_eq:NN \clist_gpush:Nn \clist_gput_left:Nn 8479
8480 \cs_generate_variant:Nn \clist_gpush:Nn { NV , No , Nx , c , cV , co , cx } 8480
8481 \clist_new:N \l__clist_internal_remove_clist 8481
8482 \seq_new:N \l__clist_internal_remove_seq 8482
8483 \cs_new_protected:Npn \clist_remove_duplicates:N 8483
8484 { \__clist_remove_duplicates:NN \clist_set_eq:NN } 8484
8485 \cs_new_protected:Npn \clist_gremove_duplicates:N 8485
8486 { \__clist_remove_duplicates:NN \clist_gset_eq:NN } 8486
8487 \cs_new_protected:Npn \__clist_remove_duplicates:NN #1#2 8487
8488 { 8488
8489   \clist_clear:N \l__clist_internal_remove_clist 8489
8490   \clist_map_inline:Nn #2 8490
8491   { 8491
8492     \clist_if_in:NnF \l__clist_internal_remove_clist {##1} 8492
8493     { 8493
8494       \tl_put_right:Ne \l__clist_internal_remove_clist 8494
8495       { 8495
8496         \clist_if_empty:NF \l__clist_internal_remove_clist { , } 8496
8497         \__clist_if_wrap:nTF {##1} { \exp_not:n { {##1} } } { \exp_not:n {##1} } 8497
8498       } 8498
8499     } 8499
8500   } 8500
8501   #1 #2 \l__clist_internal_remove_clist 8501
8502 } 8502
8503 \cs_generate_variant:Nn \clist_remove_duplicates:N { c } 8503
8504 \cs_generate_variant:Nn \clist_gremove_duplicates:N { c } 8504
8505 \cs_new_protected:Npn \clist_remove_all:Nn 8505
8506 { \__clist_remove_all:NNNn \clist_set_from_seq:NN \__kernel_tl_set:Nx } 8506
8507 \cs_new_protected:Npn \clist_gremove_all:Nn 8507
8508 { \__clist_remove_all:NNNn \clist_gset_from_seq:NN \__kernel_tl_gset:Nx } 8508
8509 \cs_new_protected:Npn \__clist_remove_all:NNNn #1#2#3#4 8509
8510 { 8510
```

```

8511 \__clist_if_wrap:nTF {#4}
8512 {
8513     \seq_set_from_clist:NN \l__clist_internal_remove_seq #3
8514     \seq_remove_all:Nn \l__clist_internal_remove_seq {#4}
8515     #1 #3 \l__clist_internal_remove_seq
8516 }
8517 {
8518     \cs_set:Npn \__clist_tmp:w ##1 , #4 ,
8519     {
8520         ##1
8521         , \s__clist_mark , \__clist_use_none_delimit_by_s_stop:w ,
8522         \__clist_remove_all:
8523     }
8524     #2 #3
8525     {
8526         \exp_after:wN \__clist_remove_all:
8527         #3 , \s__clist_mark , #4 , \s__clist_stop
8528     }
8529     \clist_if_empty:NF #3
8530     {
8531         #2 #3
8532         {
8533             \exp_args:No \exp_not:o
8534             { \exp_after:wN \use_none:n #3 }
8535         }
8536     }
8537 }
8538 }
8539 \cs_new:Npn \__clist_remove_all:
8540 { \exp_after:wN \__clist_remove_all:w \__clist_tmp:w , }
8541 \cs_new:Npn \__clist_remove_all:w #1 , \s__clist_mark , #2 , { \exp_not:n {#1} }
8542 \cs_generate_variant:Nn \clist_remove_all:Nn { c , NV , cV , Ne , ce }
8543 \cs_generate_variant:Nn \clist_gremove_all:Nn { c , NV , cV , Ne , ce }
8544 \cs_new_protected:Npn \clist_reverse:N #1
8545 { \__kernel_tl_set:Nx #1 { \exp_args:No \clist_reverse:n {#1} } }
8546 \cs_new_protected:Npn \clist_greverse:N #1
8547 { \__kernel_tl_gset:Nx #1 { \exp_args:No \clist_reverse:n {#1} } }
8548 \cs_generate_variant:Nn \clist_reverse:N { c }
8549 \cs_generate_variant:Nn \clist_greverse:N { c }
8550 \cs_new:Npn \clist_reverse:n #1
8551 {
8552     \__clist_reverse:wwNww ? #1 ,
8553     \s__clist_mark \__clist_reverse:wwNww ! ,
8554     \s__clist_mark \__clist_reverse_end:ww
8555     \s__clist_stop ? \s__clist_mark
8556 }

```


8557	\cs_new:Npn __clist_reverse:wwNww	8557
8558	#1 , #2 \s__clist_mark #3 #4 \s__clist_stop ? #5 \s__clist_mark	8558
8559	{ #3 ? #2 \s__clist_mark #3 #4 \s__clist_stop #1 , #5 \s__clist_mark }	8559
8560	\cs_new:Npn __clist_reverse_end:ww #1 ! #2 , \s__clist_mark	8560
8561	{ \exp_not:o { \use_none:n #2 } }	8561
8562	\prg_new_eq_conditional:NNn \clist_if_empty:N \tl_if_empty:N	8562
8563	{ p , T , F , TF }	8563
8564	\prg_new_eq_conditional:NNn \clist_if_empty:c \tl_if_empty:c	8564
8565	{ p , T , F , TF }	8565
8566	\prg_new_conditional:Npnn \clist_if_empty:n #1 { p , T , F , TF }	8566
8567	{	8567
8568	__clist_if_empty_n:w ? #1	8568
8569	, \s__clist_mark \prg_return_false:	8569
8570	, \s__clist_mark \prg_return_true:	8570
8571	\s__clist_stop	8571
8572	}	8572
8573	\cs_new:Npn __clist_if_empty_n:w #1 ,	8573
8574	{	8574
8575	\tl_if_empty:oTF { \use_none:nn #1 ? }	8575
8576	{ __clist_if_empty_n:w ? }	8576
8577	{ __clist_if_empty_n:wNw }	8577
8578	}	8578
8579	\cs_new:Npn __clist_if_empty_n:wNw #1 \s__clist_mark #2#3 \s__clist_stop {#2}	8579
8580	\prg_new_protected_conditional:Npnn \clist_if_in:Nn #1#2 { T , F , TF }	8580
8581	{	8581
8582	\exp_args:No __clist_if_in_return:nnN #1 {#2} #1	8582
8583	}	8583
8584	\prg_new_protected_conditional:Npnn \clist_if_in:nn #1#2 { T , F , TF }	8584
8585	{	8585
8586	\clist_set:Nn \l__clist_internal_clist {#1}	8586
8587	\exp_args:No __clist_if_in_return:nnN \l__clist_internal_clist {#2}	8587
8588	\l__clist_internal_clist	8588
8589	}	8589
8590	\cs_new_protected:Npn __clist_if_in_return:nnN #1#2#3	8590
8591	{	8591
8592	__clist_if_wrap:nTF {#2}	8592
8593	{	8593
8594	\cs_set:Npe __clist_tmp:w ##1	8594
8595	{	8595
8596	\exp_not:N \tl_if_eq:nnT {##1}	8596
8597	\exp_not:n	8597
8598	{	8598
8599	{#2}	8599
8600	{ \clist_map_break:n { \prg_return_true: \use_none:n } }	8600
8601	}	8601
8602	}	8602

```

8603         \clist_map_function:NN #3 \__clist_tmp:w      8603
8604         \prg_return_false:                             8604
8605     }                                                  8605
8606     {                                                  8606
8607         \cs_set:Npn \__clist_tmp:w ##1 ,#2, { }      8607
8608         \tl_if_empty:oTF                               8608
8609             { \__clist_tmp:w ,#1, {} {} ,#2, }      8609
8610             { \prg_return_false: } { \prg_return_true: } 8610
8611     }                                                  8611
8612 }                                                  8612
8613 \prg_generate_conditional_variant:Nnn \clist_if_in:Nn 8613
8614     { NV , No , c , cV , co } { T , F , TF }      8614
8615 \prg_generate_conditional_variant:Nnn \clist_if_in:nn 8615
8616     { nV , no } { T , F , TF }                    8616
8617 \cs_new:Npn \clist_map_function:NN #1#2             8617
8618     {                                                  8618
8619         \clist_if_empty:NF #1                       8619
8620         {                                             8620
8621             \exp_after:wN \__clist_map_function:Nw \exp_after:wN #2 #1 , 8621
8622             \s__clist_stop , \s__clist_stop , \s__clist_stop , \s__clist_stop , 8622
8623             \s__clist_stop , \s__clist_stop , \s__clist_stop , \s__clist_stop , 8623
8624             \prg_break_point:Nn \clist_map_break: { } 8624
8625         }                                             8625
8626     }                                                  8626
8627 \cs_new:Npn \__clist_map_function:Nw #1 #2, #3, #4, #5, #6, #7, #8, #9, 8627
8628     {                                                  8628
8629         \__clist_use_none_delimit_by_s_stop:w      8629
8630         #9 \__clist_map_function_end:w \s__clist_stop 8630
8631         #1 {#2} #1 {#3} #1 {#4} #1 {#5} #1 {#6} #1 {#7} #1 {#8} #1 {#9} 8631
8632         \__clist_map_function:Nw #1                 8632
8633     }                                                  8633
8634 \cs_new:Npn \__clist_map_function_end:w \s__clist_stop #1#2 8634
8635     {                                                  8635
8636         \__clist_use_none_delimit_by_s_stop:w #2 \clist_map_break: \s__clist_stop 8636
8637         #1 {#2}                                       8637
8638         \__clist_map_function_end:w \s__clist_stop 8638
8639     }                                                  8639
8640 \cs_generate_variant:Nn \clist_map_function:NN { c } 8640
8641 \cs_new:Npn \clist_map_function:nN #1#2             8641
8642     {                                                  8642
8643         \exp_after:wN \__clist_map_function_n:Nn \exp_after:wN #2      8643
8644         \exp:w \__clist_trim_next:w \prg_do_nothing: #1 ,              8644
8645         \s__clist_stop \clist_map_break: ,                            8645
8646         \prg_break_point:Nn \clist_map_break: { }                     8646
8647     }                                                  8647
8648 \cs_generate_variant:Nn \clist_map_function:nN { e } 8648

```

```
8649 \cs_new:Npn \__clist_map_function_n:Nn #1 #2 8649
8650 { 8650
8651     \__clist_use_none_delimit_by_s_stop:w #2 \s__clist_stop 8651
8652     \__clist_map_unbrace:wn #2 , #1 8652
8653     \exp_after:wN \__clist_map_function_n:Nn \exp_after:wN #1 8653
8654     \exp:w \__clist_trim_next:w \prg_do_nothing: 8654
8655 } 8655
8656 \cs_new:Npn \__clist_map_unbrace:wn #1, #2 { #2 {#1} } 8656
8657 \cs_new_protected:Npn \clist_map_inline:Nn #1#2 8657
8658 { 8658
8659     \clist_if_empty:NF #1 8659
8660     { 8660
8661         \int_gincr:N \g__kernel_prg_map_int 8661
8662         \cs_gset_protected:cpn 8662
8663         { __clist_map_ \int_use:N \g__kernel_prg_map_int :w } ##1 {#2} 8663
8664         \exp_last_unbraced:Nco \__clist_map_function:Nw 8664
8665         { __clist_map_ \int_use:N \g__kernel_prg_map_int :w } 8665
8666         #1 , 8666
8667         \s__clist_stop , \s__clist_stop , \s__clist_stop , \s__clist_stop , 8667
8668         \s__clist_stop , \s__clist_stop , \s__clist_stop , \s__clist_stop , 8668
8669         \prg_break_point:Nn \clist_map_break: 8669
8670         { \int_gdecr:N \g__kernel_prg_map_int } 8670
8671     } 8671
8672 } 8672
8673 \cs_new_protected:Npn \clist_map_inline:nn #1 8673
8674 { 8674
8675     \clist_set:Nn \l__clist_internal_clist {#1} 8675
8676     \clist_map_inline:Nn \l__clist_internal_clist 8676
8677 } 8677
8678 \cs_generate_variant:Nn \clist_map_inline:Nn { c } 8678
8679 \cs_new_protected:Npn \clist_map_variable:NNn #1#2#3 8679
8680 { \clist_map_tokens:Nn #1 { \__clist_map_variable:Nnn #2 {#3} } } 8680
8681 \cs_generate_variant:Nn \clist_map_variable:NNn { c } 8681
8682 \cs_new_protected:Npn \__clist_map_variable:Nnn #1#2#3 8682
8683 { \tl_set:Nn #1 {#3} #2 } 8683
8684 \cs_new_protected:Npn \clist_map_variable:nNn #1 8684
8685 { 8685
8686     \clist_set:Nn \l__clist_internal_clist {#1} 8686
8687     \clist_map_variable:NNn \l__clist_internal_clist 8687
8688 } 8688
8689 \cs_new:Npn \clist_map_tokens:Nn #1#2 8689
8690 { 8690
8691     \clist_if_empty:NF #1 8691
8692     { 8692
8693         \exp_last_unbraced:Nno \__clist_map_tokens:nw {#2} #1 , 8693
8694         \s__clist_stop , \s__clist_stop , \s__clist_stop , \s__clist_stop , 8694
```

```
8695 \s__clist_stop , \s__clist_stop , \s__clist_stop , \s__clist_stop , 8695
8696 \prg_break_point:Nn \clist_map_break: { } 8696
8697 } 8697
8698 } 8698
8699 \cs_new:Npn \__clist_map_tokens:nw #1 #2, #3, #4, #5, #6, #7, #8, #9, 8699
8700 { 8700
8701 \__clist_use_none_delimit_by_s_stop:w 8701
8702 #9 \__clist_map_tokens_end:w \s__clist_stop 8702
8703 \use:n {#1} {#2} \use:n {#1} {#3} \use:n {#1} {#4} \use:n {#1} {#5} 8703
8704 \use:n {#1} {#6} \use:n {#1} {#7} \use:n {#1} {#8} \use:n {#1} {#9} 8704
8705 \__clist_map_tokens:nw {#1} 8705
8706 } 8706
8707 \cs_new:Npn \__clist_map_tokens_end:w \s__clist_stop \use:n #1#2 8707
8708 { 8708
8709 \__clist_use_none_delimit_by_s_stop:w #2 \clist_map_break: \s__clist_stop 8709
8710 #1 {#2} 8710
8711 \__clist_map_tokens_end:w \s__clist_stop 8711
8712 } 8712
8713 \cs_generate_variant:Nn \clist_map_tokens:Nn { c } 8713
8714 \cs_new:Npn \clist_map_tokens:nn #1#2 8714
8715 { 8715
8716 \__clist_map_tokens_n:nw {#2} 8716
8717 \prg_do_nothing: #1 , \s__clist_stop \clist_map_break: , 8717
8718 \prg_break_point:Nn \clist_map_break: { } 8718
8719 } 8719
8720 \cs_new:Npn \__clist_map_tokens_n:nw #1#2 , 8720
8721 { 8721
8722 \tl_if_empty:oF { \use_none:nn #2 ? } 8722
8723 { 8723
8724 \__clist_use_none_delimit_by_s_stop:w #2 \s__clist_stop 8724
8725 \tl_trim_spaces_apply:oN {#2} \use_ii_i:nn 8725
8726 \__clist_map_unbrace:wn , {#1} 8726
8727 } 8727
8728 \__clist_map_tokens_n:nw {#1} \prg_do_nothing: 8728
8729 } 8729
8730 \cs_new:Npn \clist_map_break: 8730
8731 { \prg_map_break:Nn \clist_map_break: { } } 8731
8732 \cs_new:Npn \clist_map_break:n 8732
8733 { \prg_map_break:Nn \clist_map_break: } 8733
8734 \cs_new:Npn \clist_count:N #1 8734
8735 { 8735
8736 \int_eval:n 8736
8737 { 8737
8738 0 8738
8739 \clist_map_function:NN #1 \__clist_count:n 8739
8740 } 8740
```

8741	}	8741
8742	\cs_generate_variant:Nn \clist_count:N { c }	8742
8743	\cs_new:Npn __clist_count:n #1 { + 1 }	8743
8744	\cs_set_protected:Npn __clist_tmp:w #1	8744
8745	{	8745
8746	\cs_new:Npn \clist_count:n ##1	8746
8747	{	8747
8748	\int_eval:n	8748
8749	{	8749
8750	0	8750
8751	__clist_count:w #1	8751
8752	##1 , \s__clist_stop \prg_break: , \prg_break_point:	8752
8753	}	8753
8754	}	8754
8755	\cs_new:Npn __clist_count:w ##1 ,	8755
8756	{	8756
8757	__clist_use_none_delimit_by_s_stop:w ##1 \s__clist_stop	8757
8758	\tl_if_blank:nF {##1} { + 1 }	8758
8759	__clist_count:w #1	8759
8760	}	8760
8761	}	8761
8762	\exp_args:No __clist_tmp:w \c_space_tl	8762
8763	\cs_generate_variant:Nn \clist_count:n { e }	8763
8764	\cs_new:Npn \clist_use:Nnnn #1#2#3#4	8764
8765	{	8765
8766	\clist_if_exist:NTF #1	8766
8767	{	8767
8768	\int_case:nnF { \clist_count:N #1 }	8768
8769	{	8769
8770	{ 0 } { }	8770
8771	{ 1 } { \exp_after:wN __clist_use:wwn #1 , , { } }	8771
8772	{ 2 } { \exp_after:wN __clist_use:wwn #1 , {#2} }	8772
8773	}	8773
8774	{	8774
8775	\exp_after:wN __clist_use:nwwwwnwn	8775
8776	\exp_after:wN { \exp_after:wN } #1 ,	8776
8777	\s__clist_mark , { __clist_use:nwwwwnwn {#3} }	8777
8778	\s__clist_mark , { __clist_use:nwwn {#4} }	8778
8779	\s__clist_stop { }	8779
8780	}	8780
8781	}	8781
8782	{	8782
8783	\msg_expandable_error:nnn	8783
8784	{ kernel } { bad-variable } {#1}	8784
8785	}	8785
8786	}	8786

8787	\cs_generate_variant:Nn \clist_use:Nnnn { c }	8787
8788	\cs_new:Npn __clist_use:wwn #1 , #2 , #3 { \exp_not:n { #1 #3 #2 } }	8788
8789	\cs_new:Npn __clist_use:nwwwnwn	8789
8790	#1#2 , #3 , #4 , #5 \s__clist_mark , #6#7 \s__clist_stop #8	8790
8791	{ #6 {#3} , {#4} , #5 \s__clist_mark , {#6} #7 \s__clist_stop { #8 #1 #2 } }	8791
8792	\cs_new:Npn __clist_use:nwwn #1#2 , #3 \s__clist_stop #4	8792
8793	{ \exp_not:n { #4 #1 #2 } }	8793
8794	\cs_new:Npn \clist_use:Nn #1#2	8794
8795	{ \clist_use:Nnnn #1 {#2} {#2} {#2} }	8795
8796	\cs_generate_variant:Nn \clist_use:Nn { c }	8796
8797	\cs_new_eq:NN \clist_use:N \tl_use:N	8797
8798	\cs_generate_variant:Nn \clist_use:N { c }	8798
8799	\cs_new:Npn \clist_use:nnnn #1#2#3#4	8799
8800	{	8800
8801	__clist_use:Nw __clist_use_none_delimit_by_s_stop:w	8801
8802	__clist_use:Nw __clist_use_one:w	8802
8803	__clist_use:Nw __clist_use_end:w	8803
8804	__clist_use_more:w ;	8804
8805	{#2} {#3} {#4} ;	8805
8806	\prg_do_nothing: #1 , \s__clist_mark ,	8806
8807	\s__clist_stop	8807
8808	}	8808
8809	\cs_new:Npn __clist_use:Nw #1#2 ; #3 ; #4 ,	8809
8810	{	8810
8811	\tl_if_empty:oTF { \use_none:nn #4 ? }	8811
8812	{ __clist_use:Nw #1#2 ; }	8812
8813	{	8813
8814	__clist_use_none_delimit_by_s_mark:w #4 #1 \s__clist_mark	8814
8815	\tl_trim_spaces_apply:oN {#4} \use_ii_i:nn	8815
8816	__clist_map_unbrace:wn , { #2 ; }	8816
8817	}	8817
8818	#3 ; \prg_do_nothing:	8818
8819	}	8819
8820	\cs_new:Npn __clist_use_one:w \s__clist_mark #1 , #2#3#4 \s__clist_stop	8820
8821	{ \exp_not:n {#3} }	8821
8822	\cs_new:Npn __clist_use_end:w	8822
8823	\s__clist_mark #1 , #2#3#4#5#6 \s__clist_stop	8823
8824	{ \exp_not:n { #4 #5 #3 } }	8824
8825	\cs_new:Npn __clist_use_more:w ; #1#2#3#4#5#6 ;	8825
8826	{	8826
8827	\exp_not:n { #3 #5 }	8827
8828	__clist_use:Nw __clist_use_end:w __clist_use_more:w ;	8828
8829	{#1} {#2} {#6} {#5} {#6} ;	8829
8830	}	8830
8831	\cs_new:Npn \clist_use:nn #1#2 { \clist_use:nnnn {#1} {#2} {#2} {#2} }	8831
8832	\cs_new:Npn \clist_item:Nn #1#2	8832

```

8833 {
8834     \__clist_item:ffoN
8835     { \clist_count:N #1 }
8836     { \int_eval:n {#2} }
8837     #1
8838     \__clist_item_N_loop:nw
8839 }
8840 \cs_new:Npn \__clist_item:nnnN #1#2#3#4
8841 {
8842     \int_compare:nNnTF {#2} < 0
8843     {
8844         \int_compare:nNnTF {#2} < { - #1 }
8845         { \__clist_use_none_delimit_by_s_stop:w }
8846         { \exp_args:Nf #4 { \int_eval:n { #2 + 1 + #1 } } }
8847     }
8848     {
8849         \int_compare:nNnTF {#2} > {#1}
8850         { \__clist_use_none_delimit_by_s_stop:w }
8851         { #4 {#2} }
8852     }
8853     { } , #3 , \s__clist_stop
8854 }
8855 \cs_generate_variant:Nn \__clist_item:nnnN { ffo, ff }
8856 \cs_new:Npn \__clist_item_N_loop:nw #1 #2,
8857 {
8858     \int_compare:nNnTF {#1} = 0
8859     { \__clist_use_i_delimit_by_s_stop:nw { \exp_not:n {#2} } }
8860     { \exp_args:Nf \__clist_item_N_loop:nw { \int_eval:n { #1 - 1 } } }
8861 }
8862 \cs_generate_variant:Nn \clist_item:Nn { c }
8863 \cs_new:Npn \clist_item:nn #1#2
8864 {
8865     \__clist_item:ffnN
8866     { \clist_count:n {#1} }
8867     { \int_eval:n {#2} }
8868     {#1}
8869     \__clist_item_n:nw
8870 }
8871 \cs_generate_variant:Nn \clist_item:nn { e }
8872 \cs_new:Npn \__clist_item_n:nw #1
8873 { \__clist_item_n_loop:nw {#1} \prg_do_nothing: }
8874 \cs_new:Npn \__clist_item_n_loop:nw #1 #2,
8875 {
8876     \exp_args:No \tl_if_blank:nTF {#2}
8877     { \__clist_item_n_loop:nw {#1} \prg_do_nothing: }
8878     {

```



```

8879         \int_compare:nNnTF {#1} = 0
8880         { \exp_args:No \__clist_item_n_end:n {#2} }
8881         {
8882             \exp_args:Nf \__clist_item_n_loop:nw
8883             { \int_eval:n { #1 - 1 } }
8884             \prg_do_nothing:
8885         }
8886     }
8887 }
8888 \cs_new:Npn \__clist_item_n_end:n #1 #2 \s__clist_stop
8889 { \tl_trim_spaces_apply:nN {#1} \__clist_item_n_strip:n }
8890 \cs_new:Npn \__clist_item_n_strip:n #1 { \__clist_item_n_strip:w #1 , }
8891 \cs_new:Npn \__clist_item_n_strip:w #1 , { \exp_not:n {#1} }
8892 \cs_new:Npn \clist_rand_item:n #1
8893 { \exp_args:Nf \__clist_rand_item:nn { \clist_count:n {#1} } {#1} }
8894 \cs_new:Npn \__clist_rand_item:nn #1#2
8895 {
8896     \int_compare:nNnF {#1} = 0
8897     { \clist_item:nn {#2} { \int_rand:nn { 1 } {#1} } }
8898 }
8899 \cs_new:Npn \clist_rand_item:N #1
8900 {
8901     \clist_if_empty:NF #1
8902     { \clist_item:Nn #1 { \int_rand:nn { 1 } { \clist_count:N #1 } } }
8903 }
8904 \cs_generate_variant:Nn \clist_rand_item:N { c }
8905 \cs_new_protected:Npn \clist_show:N { \__clist_show:NN \msg_show:nneeee }
8906 \cs_generate_variant:Nn \clist_show:N { c }
8907 \cs_new_protected:Npn \clist_log:N { \__clist_show:NN \msg_log:nneeee }
8908 \cs_generate_variant:Nn \clist_log:N { c }
8909 \cs_new_protected:Npn \__clist_show:NN #1#2
8910 {
8911     \__kernel_chk_tl_type:NnnT #2 { clist } { \exp_not:o #2 }
8912     {
8913         \int_compare:nNnTF { \clist_count:N #2 }
8914         = { \exp_args:No \clist_count:n #2 }
8915         {
8916             #1 { clist } { show }
8917             { \token_to_str:N #2 }
8918             { \clist_map_function:NN #2 \msg_show_item:n }
8919             { } { }
8920         }
8921         {
8922             \msg_error:nnee { clist } { non-clist }
8923             { \token_to_str:N #2 } { \tl_to_str:N #2 }
8924         }

```

```
8925 }
8926 }
8927 \cs_new_protected:Npn \clist_show:n { \__clist_show:Nn \msg_show:nneeee }
8928 \cs_new_protected:Npn \clist_log:n { \__clist_show:Nn \msg_log:nneeee }
8929 \cs_new_protected:Npn \__clist_show:Nn #1#2
8930 {
8931     #1 { clist } { show }
8932     { } { \clist_map_function:nN {#2} \msg_show_item:n } { } { }
8933 }
8934 \clist_new:N \l_tmpa_clist
8935 \clist_new:N \l_tmpb_clist
8936 \clist_new:N \g_tmpa_clist
8937 \clist_new:N \g_tmpb_clist
8938 %% File: l3token.dtx
8939 \scan_new:N \s__char_stop
8940 \quark_new:N \q__char_no_value
8941 \__kernel_quark_new_conditional:Nn \__char_quark_if_no_value:N { TF }
8942 \cs_new_protected:Npn \char_set_catcode:nn #1#2
8943 { \tex_catcode:D \int_eval:n {#1} = \int_eval:n {#2} \exp_stop_f: }
8944 \cs_new:Npn \char_value_catcode:n #1
8945 { \tex_the:D \tex_catcode:D \int_eval:n {#1} \exp_stop_f: }
8946 \cs_new_protected:Npn \char_show_value_catcode:n #1
8947 { \exp_args:Nf \tl_show:n { \char_value_catcode:n {#1} } }
8948 \cs_new_protected:Npn \char_set_catcode_escape:N #1
8949 { \char_set_catcode:nn {`#1} { 0 } }
8950 \cs_new_protected:Npn \char_set_catcode_group_begin:N #1
8951 { \char_set_catcode:nn {`#1} { 1 } }
8952 \cs_new_protected:Npn \char_set_catcode_group_end:N #1
8953 { \char_set_catcode:nn {`#1} { 2 } }
8954 \cs_new_protected:Npn \char_set_catcode_math_toggle:N #1
8955 { \char_set_catcode:nn {`#1} { 3 } }
8956 \cs_new_protected:Npn \char_set_catcode_alignment:N #1
8957 { \char_set_catcode:nn {`#1} { 4 } }
8958 \cs_new_protected:Npn \char_set_catcode_end_line:N #1
8959 { \char_set_catcode:nn {`#1} { 5 } }
8960 \cs_new_protected:Npn \char_set_catcode_parameter:N #1
8961 { \char_set_catcode:nn {`#1} { 6 } }
8962 \cs_new_protected:Npn \char_set_catcode_math_superscript:N #1
8963 { \char_set_catcode:nn {`#1} { 7 } }
8964 \cs_new_protected:Npn \char_set_catcode_math_subscript:N #1
8965 { \char_set_catcode:nn {`#1} { 8 } }
8966 \cs_new_protected:Npn \char_set_catcode_ignore:N #1
8967 { \char_set_catcode:nn {`#1} { 9 } }
8968 \cs_new_protected:Npn \char_set_catcode_space:N #1
8969 { \char_set_catcode:nn {`#1} { 10 } }
8970 \cs_new_protected:Npn \char_set_catcode_letter:N #1
```

8971	{ \char_set_catcode:nn { `#1 } { 11 } }	8971
8972	\cs_new_protected:Npn \char_set_catcode_other:N #1	8972
8973	{ \char_set_catcode:nn { `#1 } { 12 } }	8973
8974	\cs_new_protected:Npn \char_set_catcode_active:N #1	8974
8975	{ \char_set_catcode:nn { `#1 } { 13 } }	8975
8976	\cs_new_protected:Npn \char_set_catcode_comment:N #1	8976
8977	{ \char_set_catcode:nn { `#1 } { 14 } }	8977
8978	\cs_new_protected:Npn \char_set_catcode_invalid:N #1	8978
8979	{ \char_set_catcode:nn { `#1 } { 15 } }	8979
8980	\cs_new_protected:Npn \char_set_catcode_escape:n #1	8980
8981	{ \char_set_catcode:nn {#1} { 0 } }	8981
8982	\cs_new_protected:Npn \char_set_catcode_group_begin:n #1	8982
8983	{ \char_set_catcode:nn {#1} { 1 } }	8983
8984	\cs_new_protected:Npn \char_set_catcode_group_end:n #1	8984
8985	{ \char_set_catcode:nn {#1} { 2 } }	8985
8986	\cs_new_protected:Npn \char_set_catcode_math_toggle:n #1	8986
8987	{ \char_set_catcode:nn {#1} { 3 } }	8987
8988	\cs_new_protected:Npn \char_set_catcode_alignment:n #1	8988
8989	{ \char_set_catcode:nn {#1} { 4 } }	8989
8990	\cs_new_protected:Npn \char_set_catcode_end_line:n #1	8990
8991	{ \char_set_catcode:nn {#1} { 5 } }	8991
8992	\cs_new_protected:Npn \char_set_catcode_parameter:n #1	8992
8993	{ \char_set_catcode:nn {#1} { 6 } }	8993
8994	\cs_new_protected:Npn \char_set_catcode_math_superscript:n #1	8994
8995	{ \char_set_catcode:nn {#1} { 7 } }	8995
8996	\cs_new_protected:Npn \char_set_catcode_math_subscript:n #1	8996
8997	{ \char_set_catcode:nn {#1} { 8 } }	8997
8998	\cs_new_protected:Npn \char_set_catcode_ignore:n #1	8998
8999	{ \char_set_catcode:nn {#1} { 9 } }	8999
9000	\cs_new_protected:Npn \char_set_catcode_space:n #1	9000
9001	{ \char_set_catcode:nn {#1} { 10 } }	9001
9002	\cs_new_protected:Npn \char_set_catcode_letter:n #1	9002
9003	{ \char_set_catcode:nn {#1} { 11 } }	9003
9004	\cs_new_protected:Npn \char_set_catcode_other:n #1	9004
9005	{ \char_set_catcode:nn {#1} { 12 } }	9005
9006	\cs_new_protected:Npn \char_set_catcode_active:n #1	9006
9007	{ \char_set_catcode:nn {#1} { 13 } }	9007
9008	\cs_new_protected:Npn \char_set_catcode_comment:n #1	9008
9009	{ \char_set_catcode:nn {#1} { 14 } }	9009
9010	\cs_new_protected:Npn \char_set_catcode_invalid:n #1	9010
9011	{ \char_set_catcode:nn {#1} { 15 } }	9011
9012	\cs_new_protected:Npn \char_set_mathcode:nn #1#2	9012
9013	{ \tex_mathcode:D \int_eval:n {#1} = \int_eval:n {#2} \exp_stop_f: }	9013
9014	\cs_new:Npn \char_value_mathcode:n #1	9014
9015	{ \tex_the:D \tex_mathcode:D \int_eval:n {#1} \exp_stop_f: }	9015
9016	\cs_new_protected:Npn \char_show_value_mathcode:n #1	9016

```
9017 { \exp_args:Nf \tl_show:n { \char_value_mathcode:n {#1} } } 9017
9018 \cs_new_protected:Npn \char_set_lccode:nn #1#2 9018
9019 { \tex_lccode:D \int_eval:n {#1} = \int_eval:n {#2} \exp_stop_f: } 9019
9020 \cs_new:Npn \char_value_lccode:n #1 9020
9021 { \tex_the:D \tex_lccode:D \int_eval:n {#1} \exp_stop_f: } 9021
9022 \cs_new_protected:Npn \char_show_value_lccode:n #1 9022
9023 { \exp_args:Nf \tl_show:n { \char_value_lccode:n {#1} } } 9023
9024 \cs_new_protected:Npn \char_set_uccode:nn #1#2 9024
9025 { \tex_uccode:D \int_eval:n {#1} = \int_eval:n {#2} \exp_stop_f: } 9025
9026 \cs_new:Npn \char_value_uccode:n #1 9026
9027 { \tex_the:D \tex_uccode:D \int_eval:n {#1} \exp_stop_f: } 9027
9028 \cs_new_protected:Npn \char_show_value_uccode:n #1 9028
9029 { \exp_args:Nf \tl_show:n { \char_value_uccode:n {#1} } } 9029
9030 \cs_new_protected:Npn \char_set_sfcode:nn #1#2 9030
9031 { \tex_sfcode:D \int_eval:n {#1} = \int_eval:n {#2} \exp_stop_f: } 9031
9032 \cs_new:Npn \char_value_sfcode:n #1 9032
9033 { \tex_the:D \tex_sfcode:D \int_eval:n {#1} \exp_stop_f: } 9033
9034 \cs_new_protected:Npn \char_show_value_sfcode:n #1 9034
9035 { \exp_args:Nf \tl_show:n { \char_value_sfcode:n {#1} } } 9035
9036 \seq_new:N \l_char_special_seq 9036
9037 \seq_set_split:Nnn \l_char_special_seq { } 9037
9038 { \_ \" \# \$ \% \& \\ \^ \_ \{ \} \~ } 9038
9039 \seq_new:N \l_char_active_seq 9039
9040 \seq_set_split:Nnn \l_char_active_seq { } 9040
9041 { \" \$ \% \^ \_ \~ } 9041
9042 \group_begin: 9042
9043 \char_set_catcode_active:N \^^@ 9043
9044 \cs_set_protected:Npn \__char_tmp:nN #1#2 9044
9045 { 9045
9046 \cs_new_protected:cpn { #1 :nN } ##1 9046
9047 { 9047
9048 \group_begin: 9048
9049 \char_set_lccode:nn { \^^@ } { ##1 } 9049
9050 \tex_lowercase:D { \group_end: #2 ^^@ } 9050
9051 } 9051
9052 \cs_new_protected:cpe { #1 :NN } ##1 9052
9053 { \exp_not:c { #1 : nN } { `##1 } } 9053
9054 } 9054
9055 \__char_tmp:nN { char_set_active_eq } \cs_set_eq:NN 9055
9056 \__char_tmp:nN { char_gset_active_eq } \cs_gset_eq:NN 9056
9057 \group_end: 9057
9058 \cs_generate_variant:Nn \char_set_active_eq:NN { Nc } 9058
9059 \cs_generate_variant:Nn \char_gset_active_eq:NN { Nc } 9059
9060 \cs_generate_variant:Nn \char_set_active_eq:nN { nc } 9060
9061 \cs_generate_variant:Nn \char_gset_active_eq:nN { nc } 9061
9062 \cs_new_eq:NN \__char_int_to_roman:w \tex_romannumeral:D 9062
```

```

9063 \cs_new_eq:NN \__char_sep: \__kernel_int_sep: 9063
9064 \cs_new:Npn \char_generate:nn #1#2 9064
9065 { 9065
9066     \exp:w \exp_after:wN \__char_generate_aux:w 9066
9067     \int_value:w \int_eval:n {#1} \exp_after:wN \__char_sep: 9067
9068     \int_value:w \int_eval:n {#2} \__char_sep: 9068
9069 } 9069
9070 \cs_new:Npn \__char_generate_aux:w #1 \__char_sep: #2 \__char_sep: 9070
9071 { 9071
9072     \if_int_odd:w 0 9072
9073         \if_int_compare:w #2 < 1 \exp_stop_f: 1 \fi: 9073
9074         \if_int_compare:w #2 = 5 \exp_stop_f: 1 \fi: 9074
9075         \if_int_compare:w #2 = 9 \exp_stop_f: 1 \fi: 9075
9076         \if_int_compare:w #2 > 13 \exp_stop_f: 1 \fi: \exp_stop_f: 9076
9077         \msg_expandable_error:nn { char } 9077
9078         { invalid-catcode } 9078
9079     \else: 9079
9080         \if_int_odd:w 0 9080
9081             \if_int_compare:w #1 < \c_zero_int 1 \fi: 9081
9082             \if_int_compare:w #1 > \c_max_char_int 1 \fi: \exp_stop_f: 9082
9083             \msg_expandable_error:nn { char } 9083
9084             { out-of-range } 9084
9085         \else: 9085
9086             \if_int_compare:w #2#1 = 100 \exp_stop_f: 9086
9087             \msg_expandable_error:nn { char } { null-space } 9087
9088             \else: 9088
9089                 \__char_generate_aux:nnw {#1} {#2} 9089
9090             \fi: 9090
9091             \fi: 9091
9092             \fi: 9092
9093             \exp_end: 9093
9094         } 9094
9095 \tl_new:N \l__char_tmp_tl 9095
9096 \group_begin: 9096
9097     \char_set_catcode_active:N ^^L 9097
9098     \cs_set:Npn ^^L { } 9098
9099     \if_cs_exist:N \tex_Ucharcat:D 9099
9100         \cs_new:Npn \__char_generate_aux:nnw #1#2#3 \exp_end: 9100
9101         { 9101
9102             #3 9102
9103             \exp_after:wN \exp_end: 9103
9104             \tex_Ucharcat:D #1 \exp_stop_f: #2 \exp_stop_f: 9104
9105         } 9105
9106     \else: 9106
9107         \char_set_catcode_active:n { 0 } 9107
9108         \tl_set:Nn \l__char_tmp_tl { \exp_not:N ^^@ \exp_not:N \or: } 9108

```

```

9109 \char_set_catcode_other:n { 0 } 9109
9110 \tl_put_right:Nn \l__char_tmp_tl { ^^@ \exp_not:N \or: } 9110
9111 \char_set_catcode_letter:n { 0 } 9111
9112 \tl_put_right:Nn \l__char_tmp_tl { ^^@ \exp_not:N \or: } 9112
9113 \tl_put_right:Nn \l__char_tmp_tl { \use:n { ~ } \exp_not:N \or: } 9113
9114 \tl_put_right:Nn \l__char_tmp_tl { \exp_not:N \or: } 9114
9115 \char_set_catcode_math_subscript:n { 0 } 9115
9116 \tl_put_right:Nn \l__char_tmp_tl { ^^@ \exp_not:N \or: } 9116
9117 \char_set_catcode_math_superscript:n { 0 } 9117
9118 \tl_put_right:Nn \l__char_tmp_tl { ^^@ \exp_not:N \or: } 9118
9119 \char_set_catcode_parameter:n { 0 } 9119
9120 \tl_put_right:Nn \l__char_tmp_tl { ^^@ \exp_not:N \or: } 9120
9121 \tl_put_right:Nn \l__char_tmp_tl { { \if_false: } \fi: \exp_not:N \or: } 9121
9122 \char_set_catcode_alignment:n { 0 } 9122
9123 \tl_put_right:Nn \l__char_tmp_tl { ^^@ \exp_not:N \or: } 9123
9124 \char_set_catcode_math_toggle:n { 0 } 9124
9125 \tl_put_right:Nn \l__char_tmp_tl { ^^@ \exp_not:N \or: } 9125
9126 \char_set_catcode_group_end:n { 0 } 9126
9127 \tl_put_right:Nn \l__char_tmp_tl { \if_false: { \fi: ^^@ \exp_not:N \or: } % } 9127
9128 \char_set_catcode_group_begin:n { 0 } % { 9128
9129 \tl_put_right:Nn \l__char_tmp_tl { ^^@ \exp_not:N \or: } } 9129
9130 \cs_set_protected:Npn \__char_tmp:n #1 9130
9131 { 9131
9132 \char_set_lccode:nn { 0 } {#1} 9132
9133 \char_set_lccode:nn { 32 } {#1} 9133
9134 \exp_args:Ne \tex_lowercase:D 9134
9135 { 9135
9136 \tl_const:Ne 9136
9137 \exp_not:c { c__char_ \__char_int_to_roman:w #1 _tl } 9137
9138 { \exp_not:o \l__char_tmp_tl } 9138
9139 } 9139
9140 } 9140
9141 \int_step_function:nnN { 0 } { 255 } \__char_tmp:n 9141
9142 \cs_new:Npn \__char_generate_aux:nnw #1#2#3 \exp_end: 9142
9143 { 9143
9144 #3 9144
9145 \if_false: { \fi: 9145
9146 \exp_after:wN \exp_after:wN \exp_after:wN \exp_end: 9146
9147 \exp_after:wN \exp_after:wN 9147
9148 \if_case:w \tex_numexpr:D 13 - #2 9148
9149 \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN 9149
9150 \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN \scan_stop: 9150
9151 \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN \exp_not:N 9151
9152 \cs:w c__char_ \__char_int_to_roman:w #1 _tl \cs_end: 9152
9153 } 9153
9154 \fi: 9154

```



```
9155     }
9156   \fi:
9157 \group_end:
9158 \group_begin:
9159   \char_set_catcode_active:N *
9160   \char_set_lccode:n { `* } { `\_}
9161   \tex_lowercase:D { \tl_const:Nn \c_catcode_active_space_tl { * } }
9162 \group_end:
9163 \tl_const:Ne \c_catcode_other_space_tl { \char_generate:nn { `\_ } { 12 } }
9164 \scan_new:N \s__token_mark
9165 \scan_new:N \s__token_stop
9166 \cs_new:Npn \token_to_catcode:N
9167   { \int_value:w \group_align_safe_begin: \_token_to_catcode:N }
9168 \cs_new:Npn \_token_to_catcode:N #1
9169   {
9170     \if_catcode:w \exp_not:N #1 \c_catcode_letter_token
9171       11
9172     \else:
9173       \if_catcode:w \exp_not:N #1 \c_catcode_other_token
9174         12
9175       \else:
9176         \if_catcode:w \exp_not:N #1 \c_math_toggle_token
9177           3
9178         \else:
9179           \if_catcode:w \exp_not:N #1 \c_alignment_token
9180             4
9181           \else:
9182             \if_catcode:w \exp_not:N #1 ##
9183               6
9184             \else:
9185               \if_catcode:w \exp_not:N #1 \c_math_superscript_token
9186                 7
9187             \else:
9188               \if_catcode:w \exp_not:N #1 \c_math_subscript_token
9189                 8
9190             \else:
9191               \if_catcode:w \exp_not:N #1 \c_group_begin_token
9192                 1
9193             \else:
9194               \if_catcode:w \exp_not:N #1 \c_group_end_token
9195                 2
9196             \else:
9197               \if_catcode:w \exp_not:N #1 \c_space_token
9198                 10
9199             \else:
9200               \token_if_cs:NTF #1 { 16 } { 13 }
```



```

9201         \fi:
9202         \fi:
9203         \fi:
9204         \fi:
9205         \fi:
9206         \fi:
9207         \fi:
9208         \fi:
9209         \fi:
9210         \fi:
9211         \group_align_safe_end:
9212         \exp_stop_f:
9213     }
9214 \group_begin:
9215     \__kernel_chk_if_free_cs:N \c_group_begin_token
9216     \tex_global:D \tex_let:D \c_group_begin_token {
9217     \__kernel_chk_if_free_cs:N \c_group_end_token
9218     \tex_global:D \tex_let:D \c_group_end_token }
9219     \char_set_catcode_math_toggle:N \*
9220     \cs_new_eq:NN \c_math_toggle_token *
9221     \char_set_catcode_alignment:N \*
9222     \cs_new_eq:NN \c_alignment_token *
9223     \cs_new_eq:NN \c_parameter_token #
9224     \cs_new_eq:NN \c_math_superscript_token ^
9225     \char_set_catcode_math_subscript:N \*
9226     \cs_new_eq:NN \c_math_subscript_token *
9227     \__kernel_chk_if_free_cs:N \c_space_token
9228     \use:n { \tex_global:D \tex_let:D \c_space_token = ~ } ~
9229     \cs_new_eq:NN \c_catcode_letter_token a
9230     \cs_new_eq:NN \c_catcode_other_token 1
9231 \group_end:
9232 \group_begin:
9233     \char_set_catcode_active:N \*
9234     \tl_const:Nn \c__token_active_tl { \exp_not:N * }
9235 \group_end:
9236 \prg_new_conditional:Npnn \token_if_group_begin:N #1 { p , T , F , TF }
9237 {
9238     \if_catcode:w \exp_not:N #1 \c_group_begin_token
9239     \prg_return_true: \else: \prg_return_false: \fi:
9240 }
9241 \prg_new_conditional:Npnn \token_if_group_end:N #1 { p , T , F , TF }
9242 {
9243     \if_catcode:w \exp_not:N #1 \c_group_end_token
9244     \prg_return_true: \else: \prg_return_false: \fi:
9245 }
9246 \prg_new_conditional:Npnn \token_if_math_toggle:N #1 { p , T , F , TF }

```

9247	{	9247
9248	\if_catcode:w \exp_not:N #1 \c_math_toggle_token	9248
9249	\prg_return_true: \else: \prg_return_false: \fi:	9249
9250	}	9250
9251	\prg_new_conditional:Npnn \token_if_alignment:N #1 { p , T , F , TF }	9251
9252	{	9252
9253	\if_catcode:w \exp_not:N #1 \c_alignment_token	9253
9254	\prg_return_true: \else: \prg_return_false: \fi:	9254
9255	}	9255
9256	\group_begin:	9256
9257	\cs_set_eq:NN \c_parameter_token \scan_stop:	9257
9258	\prg_new_conditional:Npnn \token_if_parameter:N #1 { p , T , F , TF }	9258
9259	{	9259
9260	\if_catcode:w \exp_not:N #1 \c_parameter_token	9260
9261	\prg_return_true: \else: \prg_return_false: \fi:	9261
9262	}	9262
9263	\group_end:	9263
9264	\prg_new_conditional:Npnn \token_if_math_superscript:N #1	9264
9265	{ p , T , F , TF }	9265
9266	{	9266
9267	\if_catcode:w \exp_not:N #1 \c_math_superscript_token	9267
9268	\prg_return_true: \else: \prg_return_false: \fi:	9268
9269	}	9269
9270	\prg_new_conditional:Npnn \token_if_math_subscript:N #1 { p , T , F , TF }	9270
9271	{	9271
9272	\if_catcode:w \exp_not:N #1 \c_math_subscript_token	9272
9273	\prg_return_true: \else: \prg_return_false: \fi:	9273
9274	}	9274
9275	\prg_new_conditional:Npnn \token_if_space:N #1 { p , T , F , TF }	9275
9276	{	9276
9277	\if_catcode:w \exp_not:N #1 \c_space_token	9277
9278	\prg_return_true: \else: \prg_return_false: \fi:	9278
9279	}	9279
9280	\prg_new_conditional:Npnn \token_if_letter:N #1 { p , T , F , TF }	9280
9281	{	9281
9282	\if_catcode:w \exp_not:N #1 \c_catcode_letter_token	9282
9283	\prg_return_true: \else: \prg_return_false: \fi:	9283
9284	}	9284
9285	\prg_new_conditional:Npnn \token_if_other:N #1 { p , T , F , TF }	9285
9286	{	9286
9287	\if_catcode:w \exp_not:N #1 \c_catcode_other_token	9287
9288	\prg_return_true: \else: \prg_return_false: \fi:	9288
9289	}	9289
9290	\prg_new_conditional:Npnn \token_if_active:N #1 { p , T , F , TF }	9290
9291	{	9291
9292	\if_catcode:w \exp_not:N #1 \c__token_active_tl	9292

```

9293 \prg_return_true: \else: \prg_return_false: \fi: 9293
9294 } 9294
9295 \prg_new_eq_conditional:NNn \token_if_eq_meaning:NN \cs_if_eq:NN 9295
9296 { p , T , F , TF } 9296
9297 \prg_new_conditional:Npnn \token_if_eq_catcode:NN #1#2 { p , T , F , TF } 9297
9298 { 9298
9299 \if_catcode:w \exp_not:N #1 \exp_not:N #2 9299
9300 \prg_return_true: \else: \prg_return_false: \fi: 9300
9301 } 9301
9302 \prg_new_conditional:Npnn \token_if_eq_charcode:NN #1#2 { p , T , F , TF } 9302
9303 { 9303
9304 \if_charcode:w \exp_not:N #1 \exp_not:N #2 9304
9305 \prg_return_true: \else: \prg_return_false: \fi: 9305
9306 } 9306
9307 \use:e 9307
9308 { 9308
9309 \prg_new_conditional:Npnn \exp_not:N \token_if_macro:N #1 9309
9310 { p , T , F , TF } 9310
9311 { 9311
9312 \exp_not:N \exp_after:wN \exp_not:N \__token_if_macro_p:w 9312
9313 \exp_not:N \token_to_meaning:N #1 \tl_to_str:n { ma : } 9313
9314 \s__token_stop 9314
9315 } 9315
9316 \cs_new:Npn \exp_not:N \__token_if_macro_p:w 9316
9317 #1 \tl_to_str:n { ma } #2 \c_colon_str #3 \s__token_stop 9317
9318 } 9318
9319 { 9319
9320 \str_if_eq:nnTF { #2 } { cro } 9320
9321 { \prg_return_true: } 9321
9322 { \prg_return_false: } 9322
9323 } 9323
9324 \prg_new_conditional:Npnn \token_if_cs:N #1 { p , T , F , TF } 9324
9325 { 9325
9326 \if_catcode:w \exp_not:N #1 \scan_stop: 9326
9327 \prg_return_true: \else: \prg_return_false: \fi: 9327
9328 } 9328
9329 \prg_new_conditional:Npnn \token_if_expandable:N #1 { p , T , F , TF } 9329
9330 { 9330
9331 \exp_after:wN \if_meaning:w \exp_not:N #1 #1 9331
9332 \prg_return_false: 9332
9333 \else: 9333
9334 \if_cs_exist:N #1 9334
9335 \prg_return_true: 9335
9336 \else: 9336
9337 \prg_return_false: 9337
9338 \fi: 9338

```

```

9339 \fi: 9339
9340 } 9340
9341 \group_begin: 9341
9342 \cs_set_protected:Npn \__token_tmp:w #1 9342
9343 { 9343
9344 \use:e 9344
9345 { 9345
9346 \cs_new:Npn \exp_not:c { __token_delimit_by_ #1 :w } 9346
9347 ##1 \tl_to_str:n {#1} ##2 \s_token_stop 9347
9348 { ##1 \tl_to_str:n {#1} } 9348
9349 } 9349
9350 } 9350
9351 \__token_tmp:w { char" } 9351
9352 \__token_tmp:w { count } 9352
9353 \__token_tmp:w { dimen } 9353
9354 \__token_tmp:w { ~ font } 9354
9355 \__token_tmp:w { macro } 9355
9356 \__token_tmp:w { muskip } 9356
9357 \__token_tmp:w { skip } 9357
9358 \__token_tmp:w { toks } 9358
9359 \group_end: 9359
9360 \group_begin: 9360
9361 \cs_set_protected:Npn \__token_tmp:w #1#2#3 9361
9362 { 9362
9363 \use:e 9363
9364 { 9364
9365 \prg_new_conditional:Npnn \exp_not:c { token_if_ #1 :N } ##1 9365
9366 { p , T , F , TF } 9366
9367 { 9367
9368 \cs_if_exist:cT { tex_ #2 :D } 9368
9369 { 9369
9370 \exp_not:N \if_meaning:w ##1 \exp_not:c { tex_ #2 :D } 9370
9371 \exp_not:N \prg_return_false: 9371
9372 \exp_not:N \else: 9372
9373 \exp_not:N \if_meaning:w ##1 \exp_not:c { tex_ #2 def:D } 9373
9374 \exp_not:N \prg_return_false: 9374
9375 \exp_not:N \else: 9375
9376 } 9376
9377 \exp_not:N \str_if_eq:eeTF 9377
9378 { 9378
9379 \exp_not:N \exp_after:wN 9379
9380 \exp_not:c { __token_delimit_by_ #2 :w } 9380
9381 \exp_not:N \token_to_meaning:N ##1 9381
9382 ? \tl_to_str:n {#2} \s_token_stop 9382
9383 } 9383
9384 { \exp_not:n {#3} } 9384

```

```

9385         { \exp_not:N \prg_return_true: }
9386         { \exp_not:N \prg_return_false: }
9387     \cs_if_exist:cT { tex_ #2 :D }
9388     {
9389         \exp_not:N \fi:
9390         \exp_not:N \fi:
9391     }
9392 }
9393 }
9394 }
9395 \__token_tmp:w { chardef } { char" } { \token_to_str:N \char" }
9396 \__token_tmp:w { mathchardef } { char" } { \token_to_str:N \mathchar" }
9397 \__token_tmp:w { long_macro } { macro } { \tl_to_str:n { \long } macro }
9398 \__token_tmp:w { protected_macro } { macro }
9399     { \tl_to_str:n { \protected } macro }
9400 \__token_tmp:w { protected_long_macro } { macro }
9401     { \token_to_str:N \protected \tl_to_str:n { \long } macro }
9402 \__token_tmp:w { font_selection } { ~ font } { select ~ font }
9403 \__token_tmp:w { dim_register } { dimen } { \token_to_str:N \dimen }
9404 \__token_tmp:w { int_register } { count } { \token_to_str:N \count }
9405 \__token_tmp:w { muskip_register } { muskip } { \token_to_str:N \muskip }
9406 \__token_tmp:w { skip_register } { skip } { \token_to_str:N \skip }
9407 \__token_tmp:w { toks_register } { toks } { \token_to_str:N \toks }
9408 \group_end:
9409 \sys_if_engine luatex:TF
9410 {
9411     \prg_new_conditional:Npnn \token_if_primitive:N #1 { p , T , F , TF }
9412     {
9413         \__token_if_primitive_lua:N #1
9414     }
9415 }
9416 {
9417     \tex_global:D \tex_chardef:D \c__token_A_int = `A ~ %
9418     \use:e
9419     {
9420         \prg_new_conditional:Npnn \exp_not:N \token_if_primitive:N #1
9421         { p , T , F , TF }
9422         {
9423             \exp_not:N \token_if_macro:NTF #1
9424             \exp_not:N \prg_return_false:
9425             {
9426                 \exp_not:N \exp_after:wN \exp_not:N \__token_if_primitive:NNw
9427                 \exp_not:N \token_to_meaning:N #1
9428                 \tl_to_str:n { : : : } \s__token_stop #1
9429             }
9430         }
9431     }

```

```
9431 \cs_new:Npn \exp_not:N \__token_if_primitive:NNw 9431
9432 #1#2 #3 \c_colon_str #4 \s__token_stop 9432
9433 { 9433
9434 \exp_not:N \tl_if_empty:oTF 9434
9435 { \exp_not:N \__token_if_primitive_space:w #3 ~ } 9435
9436 { 9436
9437 \exp_not:N \__token_if_primitive_loop:N #3 9437
9438 \c_colon_str \s__token_stop 9438
9439 } 9439
9440 { \exp_not:N \__token_if_primitive_nullfont:N } 9440
9441 } 9441
9442 } 9442
9443 \cs_new:Npn \__token_if_primitive_space:w #1 ~ { } 9443
9444 \cs_new:Npn \__token_if_primitive_nullfont:N #1 9444
9445 { 9445
9446 \if_meaning:w \tex_nullfont:D #1 9446
9447 \prg_return_true: 9447
9448 \else: 9448
9449 \prg_return_false: 9449
9450 \fi: 9450
9451 } 9451
9452 \cs_new:Npn \__token_if_primitive_loop:N #1 9452
9453 { 9453
9454 \if_int_compare:w `#1 < \c__token_A_int % 9454
9455 \exp_after:wN \__token_if_primitive:Nw 9455
9456 \exp_after:wN #1 9456
9457 \else: 9457
9458 \exp_after:wN \__token_if_primitive_loop:N 9458
9459 \fi: 9459
9460 } 9460
9461 \cs_new:Npn \__token_if_primitive:Nw #1 #2 \s__token_stop 9461
9462 { 9462
9463 \if:w : #1 9463
9464 \exp_after:wN \__token_if_primitive_undefined:N 9464
9465 \else: 9465
9466 \prg_return_false: 9466
9467 \exp_after:wN \use_none:n 9467
9468 \fi: 9468
9469 } 9469
9470 \cs_new:Npn \__token_if_primitive_undefined:N #1 9470
9471 { 9471
9472 \if_cs_exist:N #1 9472
9473 \prg_return_true: 9473
9474 \else: 9474
9475 \prg_return_false: 9475
9476 \fi: 9476
```

9477	}	9477
9478	}	9478
9479	\cs_new:Npn \token_case_catcode:Nn #1#2	9479
9480	{ \exp:w __token_case:NNnTF \token_if_eq_catcode:NNTF #1 {#2} { } { } }	9480
9481	\cs_new:Npn \token_case_catcode:NnT #1#2#3	9481
9482	{ \exp:w __token_case:NNnTF \token_if_eq_catcode:NNTF #1 {#2} {#3} { } }	9482
9483	\cs_new:Npn \token_case_catcode:NnF #1#2	9483
9484	{ \exp:w __token_case:NNnTF \token_if_eq_catcode:NNTF #1 {#2} { } }	9484
9485	\cs_new:Npn \token_case_catcode:NnTF	9485
9486	{ \exp:w __token_case:NNnTF \token_if_eq_catcode:NNTF }	9486
9487	\cs_new:Npn \token_case_charcode:Nn #1#2	9487
9488	{ \exp:w __token_case:NNnTF \token_if_eq_charcode:NNTF #1 {#2} { } { } }	9488
9489	\cs_new:Npn \token_case_charcode:NnT #1#2#3	9489
9490	{ \exp:w __token_case:NNnTF \token_if_eq_charcode:NNTF #1 {#2} {#3} { } }	9490
9491	\cs_new:Npn \token_case_charcode:NnF #1#2	9491
9492	{ \exp:w __token_case:NNnTF \token_if_eq_charcode:NNTF #1 {#2} { } }	9492
9493	\cs_new:Npn \token_case_charcode:NnTF	9493
9494	{ \exp:w __token_case:NNnTF \token_if_eq_charcode:NNTF }	9494
9495	\cs_new:Npn \token_case_meaning:Nn #1#2	9495
9496	{ \exp:w __token_case:NNnTF \token_if_eq_meaning:NNTF #1 {#2} { } { } }	9496
9497	\cs_new:Npn \token_case_meaning:NnT #1#2#3	9497
9498	{ \exp:w __token_case:NNnTF \token_if_eq_meaning:NNTF #1 {#2} {#3} { } }	9498
9499	\cs_new:Npn \token_case_meaning:NnF #1#2	9499
9500	{ \exp:w __token_case:NNnTF \token_if_eq_meaning:NNTF #1 {#2} { } }	9500
9501	\cs_new:Npn \token_case_meaning:NnTF	9501
9502	{ \exp:w __token_case:NNnTF \token_if_eq_meaning:NNTF }	9502
9503	\cs_new:Npn __token_case:NNnTF #1#2#3#4#5	9503
9504	{	9504
9505	__token_case:NNw #1 #2 #3 #2 { }	9505
9506	\s__token_mark {#4}	9506
9507	\s__token_mark {#5}	9507
9508	\s__token_stop	9508
9509	}	9509
9510	\cs_new:Npn __token_case:NNw #1#2#3#4	9510
9511	{	9511
9512	#1 #2 #3	9512
9513	{ __token_case_end:nw {#4} }	9513
9514	{ __token_case:NNw #1 #2 }	9514
9515	}	9515
9516	\cs_new:Npn __token_case_end:nw #1#2#3 \s__token_mark #4#5 \s__token_stop	9516
9517	{ \exp_end: #1 #4 }	9517
9518	\cs_new_eq:NN \l_peek_token ?	9518
9519	\cs_new_eq:NN \g_peek_token ?	9519
9520	\cs_new_eq:NN \l_peek_search_token ?	9520
9521	\tl_new:N \l_peek_search_tl	9521
9522	\cs_new:Npn __peek_true:w { }	9522


```

9523 \cs_new:Npn \__peek_true_aux:w { }
9524 \cs_new:Npn \__peek_false:w { }
9525 \cs_new:Npn \__peek_tmp:w { }
9526 \scan_new:N \s__peek_mark
9527 \scan_new:N \s__peek_stop
9528 \cs_new:Npn \__peek_use_none_delimit_by_s_stop:w #1 \s__peek_stop { }
9529 \cs_new_protected:Npn \peek_after:Nw
9530 { \tex_futurelet:D \l_peek_token }
9531 \cs_new_protected:Npn \peek_gafter:Nw
9532 { \tex_global:D \tex_futurelet:D \g_peek_token }
9533 \cs_new_protected:Npn \__peek_true_remove:w
9534 {
9535     \tex_afterassignment:D \__peek_true_aux:w
9536     \cs_set_eq:NN \__peek_tmp:w
9537 }
9538 \cs_new_protected:Npn \peek_remove_spaces:n #1
9539 {
9540     \cs_set:Npe \__peek_false:w { \exp_not:n {#1} }
9541     \group_align_safe_begin:
9542     \cs_set:Npn \__peek_true_aux:w { \peek_after:Nw \__peek_remove_spaces: }
9543     \__peek_true_aux:w
9544 }
9545 \cs_new_protected:Npn \__peek_remove_spaces:
9546 {
9547     \if_meaning:w \l_peek_token \c_space_token
9548     \exp_after:wN \__peek_true_remove:w
9549 \else:
9550     \group_align_safe_end:
9551     \exp_after:wN \__peek_false:w
9552 \fi:
9553 }
9554 \cs_new_protected:Npn \peek_remove_filler:n #1
9555 {
9556     \cs_set:Npn \__peek_true_aux:w { \__peek_remove_filler:w }
9557     \cs_set:Npe \__peek_false:w
9558     {
9559         \exp_not:N \group_align_safe_end:
9560         \exp_not:n {#1}
9561     }
9562     \group_align_safe_begin:
9563     \__peek_remove_filler:w
9564 }
9565 \cs_new_protected:Npn \__peek_remove_filler:w
9566 {
9567     \exp_after:wN \peek_after:Nw \exp_after:wN \__peek_remove_filler:
9568     \exp:w \exp_end_continue_f:w

```

```

9569 } 9569
9570 \cs_new_protected:Npn \__peek_remove_filler: 9570
9571 { 9571
9572 \if_catcode:w \exp_not:N \l_peek_token \c_space_token 9572
9573 \exp_after:wN \__peek_true_remove:w 9573
9574 \else: 9574
9575 \if_meaning:w \l_peek_token \scan_stop: 9575
9576 \exp_after:wN \exp_after:wN \exp_after:wN 9576
9577 \__peek_true_remove:w 9577
9578 \else: 9578
9579 \exp_after:wN \exp_after:wN \exp_after:wN 9579
9580 \__peek_remove_filler_expand:w 9580
9581 \fi: 9581
9582 \fi: 9582
9583 } 9583
9584 \cs_new_protected:Npn \__peek_remove_filler_expand:w 9584
9585 { 9585
9586 \exp_after:wN \if_meaning:w \exp_not:N \l_peek_token \l_peek_token 9586
9587 \exp_after:wN \__peek_false:w 9587
9588 \else: 9588
9589 \exp_after:wN \__peek_remove_filler:w 9589
9590 \fi: 9590
9591 } 9591
9592 \cs_new_protected:Npn \__peek_token_generic_aux:NNNTF #1#2#3#4#5 9592
9593 { 9593
9594 \group_align_safe_begin: 9594
9595 \cs_set_eq:NN \l__peek_search_token #3 9595
9596 \tl_set:Nn \l__peek_search_tl {#3} 9596
9597 \cs_set:Npe \__peek_true_aux:w 9597
9598 { 9598
9599 \exp_not:N \group_align_safe_end: 9599
9600 \exp_not:n {#4} 9600
9601 } 9601
9602 \cs_set_eq:NN \__peek_true:w #1 9602
9603 \cs_set:Npe \__peek_false:w 9603
9604 { 9604
9605 \exp_not:N \group_align_safe_end: 9605
9606 \exp_not:n {#5} 9606
9607 } 9607
9608 \peek_after:Nw #2 9608
9609 } 9609
9610 \cs_new_protected:Npn \__peek_token_generic:NNTF 9610
9611 { \__peek_token_generic_aux:NNNTF \__peek_true_aux:w } 9611
9612 \cs_new_protected:Npn \__peek_token_generic:NNT #1#2#3 9612
9613 { \__peek_token_generic:NNTF #1 #2 {#3} { } } 9613
9614 \cs_new_protected:Npn \__peek_token_generic:NNTF #1#2#3 9614

```

9615	{ __peek_token_generic:NNTF #1 #2 { } {#3} }	9615
9616	\cs_new_protected:Npn __peek_token_remove_generic:NNTF	9616
9617	{ __peek_token_generic_aux:NNNTF __peek_true_remove:w }	9617
9618	\cs_new_protected:Npn __peek_token_remove_generic:NNT #1#2#3	9618
9619	{ __peek_token_remove_generic:NNTF #1 #2 {#3} { } }	9619
9620	\cs_new_protected:Npn __peek_token_remove_generic:NNF #1#2#3	9620
9621	{ __peek_token_remove_generic:NNTF #1 #2 { } {#3} }	9621
9622	\cs_new:Npn __peek_execute_branches_meaning:	9622
9623	{	9623
9624	\if_meaning:w \l_peek_token \l__peek_search_token	9624
9625	\exp_after:wN __peek_true:w	9625
9626	\else:	9626
9627	\exp_after:wN __peek_false:w	9627
9628	\fi:	9628
9629	}	9629
9630	\cs_new:Npn __peek_execute_branches_catcode:	9630
9631	{ \if_catcode:w __peek_execute_branches_catcode_aux: }	9631
9632	\cs_new:Npn __peek_execute_branches_charcode:	9632
9633	{ \if_charcode:w __peek_execute_branches_catcode_aux: }	9633
9634	\cs_new:Npn __peek_execute_branches_catcode_aux:	9634
9635	{	9635
9636	\if_catcode:w \exp_not:N \l_peek_token \scan_stop:	9636
9637	\exp_after:wN \exp_after:wN	9637
9638	\exp_after:wN __peek_execute_branches_catcode_auxii:N	9638
9639	\exp_after:wN \exp_not:N	9639
9640	\else:	9640
9641	\exp_after:wN __peek_execute_branches_catcode_auxiii:	9641
9642	\fi:	9642
9643	}	9643
9644	\cs_new:Npn __peek_execute_branches_catcode_auxii:N #1	9644
9645	{	9645
9646	\exp_not:N #1	9646
9647	\exp_after:wN \exp_not:N \l__peek_search_tl	9647
9648	\exp_after:wN __peek_true:w	9648
9649	\else:	9649
9650	\exp_after:wN __peek_false:w	9650
9651	\fi:	9651
9652	#1	9652
9653	}	9653
9654	\cs_new:Npn __peek_execute_branches_catcode_auxiii:	9654
9655	{	9655
9656	\exp_not:N \l_peek_token	9656
9657	\exp_after:wN \exp_not:N \l__peek_search_tl	9657
9658	\exp_after:wN __peek_true:w	9658
9659	\else:	9659
9660	\exp_after:wN __peek_false:w	9660

```

9661 \fi:
9662 }
9663 \tl_map_inline:nn { { catcode } { charcode } { meaning } }
9664 {
9665   \tl_map_inline:nn { { } { _remove } }
9666   {
9667     \tl_map_inline:nn { { TF } { T } { F } }
9668     {
9669       \cs_new_protected:cpe { peek_ #1 ##1 :N ####1 }
9670       {
9671         \exp_not:c { __peek_token ##1 _generic:NN ####1 }
9672         \exp_not:c { __peek_execute_branches_ #1 : }
9673       }
9674     }
9675   }
9676 }
9677 \group_begin:
9678   \cs_set_protected:Npn \__peek_tmp:w #1 \s__peek_stop
9679   {
9680     \cs_new_protected:Npn \__peek_execute_branches_N_type:
9681     {
9682       \if_int_odd:w
9683         \if_catcode:w \exp_not:N \l_peek_token { \c_zero_int \fi:
9684         \if_catcode:w \exp_not:N \l_peek_token } \c_zero_int \fi:
9685         \if_meaning:w \l_peek_token \c_space_token \c_zero_int \fi:
9686         \c_one_int
9687       \exp_after:wN \__peek_N_type:w
9688       \token_to_meaning:N \l_peek_token
9689       \s__peek_mark \__peek_N_type_aux:nnw
9690       #1 \s__peek_mark \__peek_use_none_delimit_by_s_stop:w
9691       \s__peek_stop
9692       \exp_after:wN \__peek_true:w
9693     \else:
9694       \exp_after:wN \__peek_false:w
9695     \fi:
9696   }
9697   \cs_new_protected:Npn \__peek_N_type:w ##1 #1 ##2 \s__peek_mark ##3
9698   { ##3 {##1} {##2} }
9699 }
9700 \exp_after:wN \__peek_tmp:w \tl_to_str:n { outer } \s__peek_stop
9701 \group_end:
9702 \cs_new_protected:Npn \__peek_N_type_aux:nnw #1 #2 #3 \fi:
9703 {
9704   \fi:
9705   \tl_if_in:noTF {#1} { \tl_to_str:n {ma} }
9706   { \__peek_true:w }

```

```
9707 { \tl_if_empty:nTF {#2} { \__peek_true:w } { \__peek_false:w } } 9707
9708 } 9708
9709 \cs_new_protected:Npn \peek_N_type:TF 9709
9710 { 9710
9711 \__peek_token_generic:NNTF 9711
9712 \__peek_execute_branches_N_type: \scan_stop: 9712
9713 } 9713
9714 \cs_new_protected:Npn \peek_N_type:T 9714
9715 { \__peek_token_generic:NNT \__peek_execute_branches_N_type: \scan_stop: } 9715
9716 \cs_new_protected:Npn \peek_N_type:F 9716
9717 { \__peek_token_generic:NNTF \__peek_execute_branches_N_type: \scan_stop: } 9717
9718 %% File: l3prop.dtx 9718
9719 \cs_new_eq:NN \__prop_tmp:w ? 9719
9720 \tl_new:N \l__prop_internal_tl 9720
9721 \scan_new:N \s__prop_mark 9721
9722 \scan_new:N \s__prop_stop 9722
9723 \quark_new:N \q__prop_recursion_tail 9723
9724 \quark_new:N \q__prop_recursion_stop 9724
9725 \__kernel_quark_new_test:N \__prop_if_recursion_tail_stop:n 9725
9726 \cs_generate_variant:Nn \__prop_if_recursion_tail_stop:n { o } 9726
9727 \scan_new:N \s__prop 9727
9728 \cs_new_protected:Npn \__prop_chk:w { \__prop_chk_loop:nw { } } 9728
9729 \cs_new_protected:Npn \__prop_chk_loop:nw #1 9729
9730 { 9730
9731 \peek_meaning:NNTF \__prop_pair:wn 9731
9732 { \__prop_chk_get:nw {#1} } 9732
9733 { \msg_error:nne { prop } { misused } {#1} } 9733
9734 } 9734
9735 \cs_new_protected:Npn \__prop_chk_get:nw #1 \__prop_pair:wn #2 \s__prop #3 9735
9736 { \__prop_chk_loop:nw { #1 , ~ {#2} = { \tl_to_str:n {#3} } } } 9736
9737 \cs_new:Npn \__prop_pair:wn #1 \s__prop #2 { } 9737
9738 \cs_new_protected:Npn \__prop_flatten:w #1 \s__prop #2#3 9738
9739 { \use:e { \__prop_flatten_aux:N #3 } } 9739
9740 \cs_new:Npn \__prop_flatten:N #1 9740
9741 { \exp_after:wN \__prop_flatten_aux:w #1 } 9741
9742 \cs_new:Npn \__prop_flatten_aux:w #1 \s__prop #2 { \__prop_flatten_aux:N } 9742
9743 \cs_new:Npn \__prop_flatten_aux:N #1 9743
9744 { 9744
9745 \s__prop \__prop_chk:w 9745
9746 \exp_after:wN \__prop_flatten_loop:w #1 9746
9747 \use_none:nnnn \__prop_pair:wn \s__prop { } 9747
9748 } 9748
9749 \cs_new:Npn \__prop_flatten_loop:w #1#2#3 \__prop_pair:wn #4 \s__prop #5 9749
9750 { 9750
9751 #3 9751
9752 \exp_not:n { \__prop_pair:wn #4 \s__prop {#5} } 9752
```

```

9753 \exp_after:wN \__prop_flatten_loop:w 9753
9754 } 9754
9755 \int_new:N \g__prop_prefix_int 9755
9756 \int_const:Nn \c__prop_basis_int { \c_max_char_int - `!\ } 9756
9757 \cs_new_protected:Npn \__prop_next_prefix: 9757
9758 { 9758
9759 \tl_set:Ne \l__prop_internal_tl 9759
9760 { \__prop_to_prefix:n { \g__prop_prefix_int } } 9760
9761 \int_gincr:N \g__prop_prefix_int 9761
9762 } 9762
9763 \cs_new:Npn \__prop_to_prefix:n #1 9763
9764 { 9764
9765 \int_compare:nNnTF {#1} > \c__prop_basis_int 9765
9766 { 9766
9767 \exp_args:Nf \__prop_to_prefix:n 9767
9768 { \int_div_truncate:nn {#1} \c__prop_basis_int } 9768
9769 \exp_args:Nf \__prop_to_prefix:n 9769
9770 { \int_mod:nn {#1} \c__prop_basis_int } 9770
9771 } 9771
9772 { \char_generate:nn { `!\ + #1 } { 12 } } 9772
9773 } 9773
9774 \cs_new:Npn \__prop_if_flat:NTF #1 9774
9775 { 9775
9776 \prop_if_exist:NT #1 9776
9777 \exp_after:wN \__prop_if_flat_aux:w #1 9777
9778 \s__prop_mark \use_ii:nn 9778
9779 \__prop_flatten:w \s__prop_mark \use_i:nn \s__prop_stop 9779
9780 } 9780
9781 \cs_new:Npn \__prop_if_flat_aux:w 9781
9782 #1 \__prop_flatten:w #2 \s__prop_mark #3 #4 \s__prop_stop {#3} 9782
9783 \tl_const:Nn \c_empty_prop { \s__prop \__prop_chk:w } 9783
9784 \cs_new_protected:Npn \prop_new:N #1 9784
9785 { 9785
9786 \__kernel_chk_if_free_cs:N #1 9786
9787 \cs_gset_eq:NN #1 \c_empty_prop 9787
9788 } 9788
9789 \cs_generate_variant:Nn \prop_new:N { c } 9789
9790 \cs_new_protected:Npn \prop_new_linked:N #1 9790
9791 { 9791
9792 \__kernel_chk_if_free_cs:N #1 9792
9793 \__prop_new_linked:N #1 9793
9794 } 9794
9795 \cs_new_protected:Npn \__prop_new_linked:N #1 9795
9796 { 9796
9797 \__prop_next_prefix: 9797
9798 \cs_gset_nopar:Npe #1 9798

```



```
9799 { 9799
9800 \__prop_flatten:w 9800
9801 \exp_not:c { __prop ~ \l__prop_internal_tl } 9801
9802 \s__prop { \l__prop_internal_tl } 9802
9803 \exp_not:c { __prop ~ \l__prop_internal_tl } 9803
9804 } 9804
9805 \cs_gset_nopar:cpe { __prop ~ \l__prop_internal_tl } 9805
9806 { 9806
9807 \exp_not:N \use_none:n 9807
9808 \exp_not:N #1 9808
9809 } 9809
9810 } 9810
9811 \cs_generate_variant:Nn \prop_new_linked:N { c } 9811
9812 \cs_new_protected:Npn \prop_clear:N 9812
9813 { \__prop_clear:NNN \cs_set_eq:NN \cs_set_nopar:Npe } 9813
9814 \cs_generate_variant:Nn \prop_clear:N { c } 9814
9815 \cs_new_protected:Npn \prop_gclear:N 9815
9816 { \__prop_clear:NNN \cs_gset_eq:NN \cs_gset_nopar:Npe } 9816
9817 \cs_generate_variant:Nn \prop_gclear:N { c } 9817
9818 \cs_new_protected:Npn \__prop_clear:NNN #1#2#3 9818
9819 { 9819
9820 \__prop_if_flat:NTF #3 9820
9821 { #1 #3 \c_empty_prop } 9821
9822 { \exp_after:wN \__prop_clear:wNNN #3 #1 #2 #3 } 9822
9823 } 9823
9824 \cs_new_protected:Npn \__prop_clear:wNNN 9824
9825 \__prop_flatten:w #1 \s__prop #2#3#4#5#6 9825
9826 { 9826
9827 \__prop_clear_entries:NN #4 #3 9827
9828 #5 #6 { \exp_not:n { \__prop_flatten:w #1 \s__prop {#2} #1 } } 9828
9829 #5 #1 { \exp_not:n { \use_none:n #6 } } 9829
9830 } 9830
9831 \cs_new_protected:Npn \__prop_clear_entries:NN #1#2 9831
9832 { 9832
9833 \exp_after:wN \__prop_clear_loop:Nw \exp_after:wN #1 #2 9833
9834 \use_none:nnnn \__prop_pair:wn \s__prop { } 9834
9835 } 9835
9836 \cs_new_protected:Npn \__prop_clear_loop:Nw 9836
9837 #1#2#3#4 \__prop_pair:wn #5 \s__prop #6 9837
9838 { 9838
9839 #1 #3 \tex_undefined:D 9839
9840 #4 9840
9841 \exp_after:wN \__prop_clear_loop:Nw 9841
9842 \exp_after:wN #1 9842
9843 } 9843
9844 \cs_new_protected:Npn \prop_clear_new:N #1 9844
```

```
9845 { \prop_if_exist:NTF #1 { \prop_clear:N #1 } { \prop_new:N #1 } } 9845
9846 \cs_generate_variant:Nn \prop_clear_new:N { c } 9846
9847 \cs_new_protected:Npn \prop_gclear_new:N #1 9847
9848 { \prop_if_exist:NTF #1 { \prop_gclear:N #1 } { \prop_new:N #1 } } 9848
9849 \cs_generate_variant:Nn \prop_gclear_new:N { c } 9849
9850 \cs_new_protected:Npn \prop_clear_new_linked:N #1 9850
9851 { \prop_if_exist:NTF #1 { \prop_clear:N #1 } { \prop_new_linked:N #1 } } 9851
9852 \cs_generate_variant:Nn \prop_clear_new_linked:N { c } 9852
9853 \cs_new_protected:Npn \prop_gclear_new_linked:N #1 9853
9854 { \prop_if_exist:NTF #1 { \prop_gclear:N #1 } { \prop_new_linked:N #1 } } 9854
9855 \cs_generate_variant:Nn \prop_gclear_new_linked:N { c } 9855
9856 \cs_new_protected:Npn \prop_set_eq:NN 9856
9857 { \__prop_set_eq:NNNN \cs_set_eq:NN \cs_set_nopar:Npe } 9857
9858 \cs_generate_variant:Nn \prop_set_eq:NN { Nc , cN , cc } 9858
9859 \cs_new_protected:Npn \prop_gset_eq:NN 9859
9860 { \__prop_set_eq:NNNN \cs_gset_eq:NN \cs_gset_nopar:Npe } 9860
9861 \cs_generate_variant:Nn \prop_gset_eq:NN { Nc , cN , cc } 9861
9862 \cs_new_protected:Npn \__prop_set_eq:NNNN #1#2#3#4 9862
9863 { 9863
9864 \cs_if_eq:NNF #3#4 9864
9865 { 9865
9866 \__prop_if_flat:NTF #3 9866
9867 { 9867
9868 \__prop_if_flat:NTF #4 9868
9869 { #1 #3 #4 } 9869
9870 { #2 #3 { \__prop_flatten:N #4 } } 9870
9871 } 9871
9872 { \exp_after:wN \__prop_set_eq:wNNNN #3 #1#2#3#4 } 9872
9873 } 9873
9874 } 9874
9875 \cs_new_protected:Npn \__prop_set_eq:wNNNN 9875
9876 \__prop_flatten:w #1 \s__prop #2#3#4#5#6#7 9876
9877 { 9877
9878 \__prop_clear_entries:NN #4 #3 9878
9879 \exp_args:Nf \__prop_set_eq:nNnNN {#7} #1 {#2} #5 #6 9879
9880 } 9880
9881 \cs_new_protected:Npn \__prop_set_eq:nNnNN #1#2#3#4#5 9881
9882 { 9882
9883 \use_i:nnn 9883
9884 { 9884
9885 \__prop_set_eq_loop:NNnw #5 #4 {#3} 9885
9886 \__prop_flatten:w #2 \s__prop {#3} 9886
9887 } 9887
9888 #1 9888
9889 \use_none:n \__prop_pair:wn ? \s__prop 9889
9890 } 9890
```

```

9891 \cs_new_protected:Npn \__prop_set_eq_loop:NNnw 9891
9892   #1#2#3#4 \s__prop #5#6 \__prop_pair:wn #7 \s__prop 9892
9893 { 9893
9894   \tl_set:Ne \l__prop_internal_tl { \exp_not:c { __prop ~ #3 #6 ~ #7 } } 9894
9895   #2 #1 { \exp_not:n { #4 \s__prop {#5} } \exp_not:o \l__prop_internal_tl } 9895
9896   \use_none:n #6 \__prop_set_eq_end:w 9896
9897   \exp_after:wN \__prop_set_eq_loop:NNnw \l__prop_internal_tl #2 {#3} 9897
9898   \use_none:n #1 \__prop_pair:wn #7 \s__prop 9898
9899 } 9899
9900 \cs_new_protected:Npn \__prop_set_eq_end:w 9900
9901   \exp_after:wN \__prop_set_eq_loop:NNnw #1#2#3 9901
9902   \use_none:n #4#5 \s__prop 9902
9903 { 9903
9904   \exp_after:wN #2 \l__prop_internal_tl { \exp_not:n { \use_none:n #4 } } 9904
9905 } 9905
9906 \cs_new_protected:Npn \prop_make_flat:N #1 9906
9907 { 9907
9908   \int_compare:nNnTF { \tex_currentgrouplevel:D } = 0 9908
9909   { 9909
9910     \__prop_if_flat:NTF #1 { } 9910
9911     { \exp_args:NNf \__prop_make_flat:Nn #1 {#1} } 9911
9912   } 9912
9913   { 9913
9914     \msg_error:nnee { prop } { inner-make } 9914
9915     { \token_to_str:N \prop_make_flat:N } { \token_to_str:N #1 } 9915
9916   } 9916
9917 } 9917
9918 \cs_generate_variant:Nn \prop_make_flat:N { c } 9918
9919 \cs_new_protected:Npn \__prop_make_flat:Nn #1#2 9919
9920 { 9920
9921   \exp_after:wN \__prop_clear:wNNN #1 \cs_set_eq:NN \cs_set_nopar:Npe #1 9921
9922   \cs_set_nopar:Npe #1 { \exp_not:n {#2} } 9922
9923 } 9923
9924 \cs_new_protected:Npn \prop_make_linked:N #1 9924
9925 { 9925
9926   \int_compare:nNnTF { \tex_currentgrouplevel:D } = 0 9926
9927   { 9927
9928     \__prop_if_flat:NTF #1 9928
9929     { \exp_args:NNo \__prop_make_linked:Nn #1 {#1} } { } 9929
9930   } 9930
9931   { 9931
9932     \msg_error:nnee { prop } { inner-make } 9932
9933     { \token_to_str:N \prop_make_linked:N } { \token_to_str:N #1 } 9933
9934   } 9934
9935 } 9935
9936 \cs_generate_variant:Nn \prop_make_linked:N { c } 9936

```

```
9937 \cs_new_protected:Npn \__prop_make_linked:Nn #1#2
9938 {
9939     \__prop_new_linked:N #1
9940     \tl_set:Nn \l__prop_internal_tl {#2}
9941     \exp_after:wN \__prop_set_eq:wNNNN #1
9942     \cs_set_eq:NN \cs_set_nopar:Npe #1 \l__prop_internal_tl
9943 }
9944 \prop_new:N \l_tmpa_prop
9945 \prop_new:N \l_tmpb_prop
9946 \prop_new:N \g_tmpa_prop
9947 \prop_new:N \g_tmpb_prop
9948 \cs_new_protected:Npn \prop_concat:NNN
9949 { \__prop_concat:NNNNN \cs_set_eq:NN \cs_set_nopar:Npe }
9950 \cs_generate_variant:Nn \prop_concat:NNN { ccc }
9951 \cs_new_protected:Npn \prop_gconcat:NNN
9952 { \__prop_concat:NNNNN \cs_gset_eq:NN \cs_gset_nopar:Npe }
9953 \cs_generate_variant:Nn \prop_gconcat:NNN { ccc }
9954 \cs_new_protected:Npn \__prop_concat:NNNNN #1#2#3#4#5
9955 {
9956     \cs_if_eq:NNTF #3 #5
9957     { \__prop_concat:nNNN \use_none:nnn #2 #3 #4 }
9958     {
9959         \__prop_set_eq:NNNN #1 #2 #3 #4
9960         \__prop_concat:nNNN { } #2 #3 #5
9961     }
9962 }
9963 \cs_new_protected:Npn \__prop_concat:nNNN #1#2#3#4
9964 {
9965     \cs_gset_eq:NN \__prop_tmp:w \__prop_pair:wn
9966     \cs_gset_protected:Npn \__prop_pair:wn ##1 \s__prop
9967     { \__prop_put:nNNnn {#1} #2 #3 {##1} }
9968     \exp_last_unbraced:Nf \use_none:nn #4
9969     \cs_gset_eq:NN \__prop_pair:wn \__prop_tmp:w
9970 }
9971 \cs_new_protected:Npn \prop_put_from_keyval:Nn #1
9972 { \__prop_from_keyval:nn { \__prop_put:nNNnn { } \cs_set_nopar:Npe #1 } }
9973 \cs_generate_variant:Nn \prop_put_from_keyval:Nn { c }
9974 \cs_new_protected:Npn \prop_gput_from_keyval:Nn #1
9975 { \__prop_from_keyval:nn { \__prop_put:nNNnn { } \cs_gset_nopar:Npe #1 } }
9976 \cs_generate_variant:Nn \prop_gput_from_keyval:Nn { c }
9977 \cs_new_protected:Npn \__prop_from_keyval:nn
9978 {
9979     \bool_if:NTF \l__kernel_keyval_allow_blank_keys_bool
9980     { \__prop_from_keyval:Nnn \c_true_bool }
9981     { \__prop_from_keyval:Nnn \c_false_bool }
9982 }
```

```
9983 \cs_new_protected:Npn \__prop_from_keyval:Nnn #1#2#3 9983
9984 { 9984
9985     \bool_set_eq:NN \l__kernel_keyval_allow_blank_keys_bool \c_true_bool 9985
9986     \keyval_parse:nnn \__prop_missing_eq:n {#2} {#3} 9986
9987     \bool_set_eq:NN \l__kernel_keyval_allow_blank_keys_bool #1 9987
9988 } 9988
9989 \cs_new_protected:Npn \__prop_missing_eq:n 9989
9990 { \msg_error:nnn { prop } { prop-keyval } } 9990
9991 \cs_new_protected:Npn \prop_set_from_keyval:Nn #1 9991
9992 { 9992
9993     \__prop_clear:NNN \cs_set_eq:NN \cs_set_nopar:Npe #1 9993
9994     \prop_put_from_keyval:Nn #1 9994
9995 } 9995
9996 \cs_generate_variant:Nn \prop_set_from_keyval:Nn { c } 9996
9997 \cs_new_protected:Npn \prop_gset_from_keyval:Nn #1 9997
9998 { 9998
9999     \__prop_clear:NNN \cs_gset_eq:NN \cs_gset_nopar:Npe #1 9999
10000     \prop_gput_from_keyval:Nn #1 10000
10001 } 10001
10002 \cs_generate_variant:Nn \prop_gset_from_keyval:Nn { c } 10002
10003 \cs_new_protected:Npn \prop_const_from_keyval:Nn #1 10003
10004 { 10004
10005     \prop_new:N #1 10005
10006     \__prop_from_keyval:nn { \__prop_put:nNNnn { } \cs_gset_nopar:Npe #1 } 10006
10007 } 10007
10008 \cs_generate_variant:Nn \prop_const_from_keyval:Nn { c } 10008
10009 \cs_new_protected:Npn \prop_const_linked_from_keyval:Nn #1 10009
10010 { 10010
10011     \prop_new_linked:N #1 10011
10012     \__prop_from_keyval:nn { \__prop_put:nNNnn { } \cs_gset_nopar:Npe #1 } 10012
10013 } 10013
10014 \cs_generate_variant:Nn \prop_const_linked_from_keyval:Nn { c } 10014
10015 \cs_new_protected:Npn \__prop_split:NnTFn #1#2 10015
10016 { 10016
10017     \exp_after:wN \__prop_split_aux:nNTFn 10017
10018     \exp_after:wN { \tl_to_str:n {#2} } #1 10018
10019 } 10019
10020 \cs_new_protected:Npn \__prop_split_aux:nNTFn #1#2#3 10020
10021 { 10021
10022     \cs_set:Npn \__prop_split_flat:w \__prop_split_linked:w ##1 10022
10023     \__prop_pair:wn #1 \s__prop ##2 ##3 \s__prop_mark ##4 ##5 \s__prop_stop 10023
10024     { ##4 {#3} } 10024
10025     \exp_after:wN \__prop_split_test:wn #2 \s__prop_mark \use_i:nnn 10025
10026     \__prop_pair:wn #1 \s__prop { ? \fi: \__prop_split_wrong:Nw #2 } 10026
10027     \s__prop_mark \use_ii:nnn 10027
10028     \s__prop_stop 10028
```

```
10029 }
10030 \cs_new:Npn \__prop_split_flat:w { }
10031 \cs_new_protected:Npn \__prop_split_test:wn #1 \s__prop #2
10032 {
10033     \if_meaning:w \__prop_chk:w #2 \exp_after:wN \__prop_split_flat:w \fi:
10034     \__prop_split_linked:w
10035 }
10036 \cs_new_protected:Npn \__prop_split_linked:w #1 \s__prop_stop #2#3 {#3}
10037 \cs_new_protected:Npn \__prop_split_wrong:Nw #1#2 \s__prop_stop #3#4
10038 {
10039     \prop_show:N #1
10040     \cs_gset_eq:NN #1 \c_empty_prop
10041     #3
10042 }
10043 \cs_new_protected:Npn \prop_get:NnN #1#2#3
10044 {
10045     \__prop_get:NnnTF #1 {#2}
10046     { \tl_set:Nn #3 { } } { \tl_set:Nn #3 { \q_no_value } }
10047 }
10048 \cs_generate_variant:Nn \prop_get:NnN { NV , Nv , Ne , c , cV , cv , ce }
10049 \cs_generate_variant:Nn \prop_get:NnN { No , Nx , co , cx }
10050 \cs_generate_variant:Nn \prop_get:NnN { cnc }
10051 \prg_new_protected_conditional:Npnn \prop_get:NnN #1#2#3 { T , F , TF }
10052 {
10053     \__prop_get:NnnTF #1 {#2}
10054     { \tl_set:Nn #3 { } } \prg_return_true: \prg_return_false:
10055 }
10056 \prg_generate_conditional_variant:Nnn \prop_get:NnN
10057 { NV , Nv , Ne , c , cV , cv , ce } { T , F , TF }
10058 \prg_generate_conditional_variant:Nnn \prop_get:NnN
10059 { No , Nx , co , cx } { T , F , TF }
10060 \prg_generate_conditional_variant:Nnn \prop_get:NnN
10061 { cnc } { T , F , TF }
10062 \cs_new_protected:Npn \__prop_get:NnnTF #1#2#3#4#5
10063 {
10064     \__prop_split:NnTFn #1 {#2}
10065     { #3 {##2} #4 }
10066     {#5}
10067     { \exp_after:wN \__prop_get_linked:w #1 {#2} {#3} {#4} {#5} }
10068 }
10069 \cs_new:Npn \__prop_get_linked:w
10070     \__prop_flatten:w #1 \s__prop #2#3#4#5#6#7
10071 {
10072     \if_cs_exist:w __prop ~ #2 ~ \tl_to_str:n {#4} \cs_end:
10073     \exp_after:wN \exp_after:wN \exp_after:wN \__prop_get_linked_aux:w
10074     \cs:w __prop ~ #2 ~ \tl_to_str:n {#4} \exp_after:wN \cs_end:
```


10075	\else:	10075
10076	\exp_after:wN __prop_get_linked_aux:w	10076
10077	\fi:	10077
10078	\s__prop_mark {#5} {#6}	10078
10079	\s__prop { } \s__prop_mark \use_none:n {#7}	10079
10080	\s__prop_stop	10080
10081	}	10081
10082	\cs_new:Npn __prop_get_linked_aux:w	10082
10083	#1 \s__prop #2 #3 \s__prop_mark #4 #5 #6 \s__prop_stop { #4 {#2} #5 }	10083
10084	\cs_new:Npn \prop_item:Nn #1#2	10084
10085	{	10085
10086	__prop_if_flat:NTF #1	10086
10087	{	10087
10088	\exp_args:NNo \prop_map_tokens:Nn #1	10088
10089	{	10089
10090	\exp_after:wN __prop_item:nnn	10090
10091	\exp_after:wN { \tl_to_str:n {#2} }	10091
10092	}	10092
10093	}	10093
10094	{ \exp_after:wN __prop_get_linked:w #1 {#2} \exp_not:n { } { } }	10094
10095	}	10095
10096	\cs_new:Npn __prop_item:nnn #1#2#3	10096
10097	{	10097
10098	\str_if_eq:eeT {#1} {#2}	10098
10099	{ \prop_map_break:n { \exp_not:n {#3} } }	10099
10100	}	10100
10101	\cs_generate_variant:Nn \prop_item:Nn { NV , No , Ne , c , cV , co , ce }	10101
10102	\cs_new_protected:Npn __prop_pop:NnNnTF #1#2#3#4#5#6#7	10102
10103	{	10103
10104	__prop_split:NnTFn #1 {#2}	10104
10105	{	10105
10106	#4 #1 { \exp_not:n { \s__prop __prop_chk:w ##1 ##3 } }	10106
10107	#5 {##2}	10107
10108	#6	10108
10109	}	10109
10110	{#7}	10110
10111	{	10111
10112	\exp_after:wN __prop_pop_linked:wnNnTF #1 {#2}	10112
10113	#3 #4 {#5} {#6} {#7}	10113
10114	}	10114
10115	}	10115
10116	\cs_new_protected:Npn __prop_pop_linked:wnNnTF	10116
10117	__prop_flatten:w #1 \s__prop #2#3#4#5#6#7	10117
10118	{	10118
10119	\if_cs_exist:w __prop ~ #2 ~ \tl_to_str:n {#4} \cs_end:	10119
10120	\exp_after:wN __prop_pop_linked:NNNn	10120

10121	\cs:w __prop ~ #2 ~ \tl_to_str:n {#4} \cs_end:	10121
10122	#5 #6 {#7}	10122
10123	\else:	10123
10124	\exp_after:wN \use_iii:nnn	10124
10125	\fi:	10125
10126	\use_i:nn	10126
10127	}	10127
10128	\cs_new_protected:Npn __prop_pop_linked:NNNn #1#2#3#4	10128
10129	{	10129
10130	\if_meaning:w \scan_stop: #1	10130
10131	\exp_after:wN \exp_after:wN \exp_after:wN \use_iii:nnn	10131
10132	\else:	10132
10133	\exp_after:wN __prop_pop_linked:w #1 #1 #2 #3 {#4}	10133
10134	\fi:	10134
10135	}	10135
10136	\cs_new_protected:Npn __prop_pop_linked:w	10136
10137	\use_none:n #1#2 \s__prop #3#4#5#6#7#8	10137
10138	{	10138
10139	#6 #5 \tex_undefined:D	10139
10140	#7 #1	10140
10141	{	10141
10142	\exp_after:wN __prop_pop_linked_prev:w #1	10142
10143	\exp_not:N #4	10143
10144	}	10144
10145	#7 #4	10145
10146	{	10146
10147	\exp_not:n { \use_none:n #1 }	10147
10148	\exp_not:f { \exp_after:wN __prop_pop_linked_next:w #4 }	10148
10149	}	10149
10150	#8 {#3}	10150
10151	}	10151
10152	\cs_new:Npn __prop_pop_linked_prev:w #1 \s__prop #2#3	10152
10153	{ \exp_not:n { #1 \s__prop {#2} } }	10153
10154	\cs_new:Npn __prop_pop_linked_next:w \use_none:n #1 { \exp_stop_f: }	10154
10155	\cs_new_protected:Npn \prop_remove:Nn #1#2	10155
10156	{	10156
10157	__prop_pop:NnNNnTF #1 {#2}	10157
10158	\cs_set_eq:NN \cs_set_nopar:Npe	10158
10159	\use_none:n { } { }	10159
10160	}	10160
10161	\cs_new_protected:Npn \prop_gremove:Nn #1#2	10161
10162	{	10162
10163	__prop_pop:NnNNnTF #1 {#2}	10163
10164	\cs_gset_eq:NN \cs_gset_nopar:Npe	10164
10165	\use_none:n { } { }	10165
10166	}	10166

10167	\cs_generate_variant:Nn \prop_remove:Nn { NV , Ne , c , cV , ce }	10167
10168	\cs_generate_variant:Nn \prop_gremove:Nn { NV , Ne , c , cV , ce }	10168
10169	\cs_new_protected:Npn \prop_pop:NnN #1#2#3	10169
10170	{	10170
10171	__prop_pop:NnNNnTF #1 {#2}	10171
10172	\cs_set_eq:NN \cs_set_nopar:Npe	10172
10173	{ \tl_set:Nn #3 } { } { \tl_set:Nn #3 { \q_no_value } }	10173
10174	}	10174
10175	\cs_new_protected:Npn \prop_gpop:NnN #1#2#3	10175
10176	{	10176
10177	__prop_pop:NnNNnTF #1 {#2}	10177
10178	\cs_gset_eq:NN \cs_gset_nopar:Npe	10178
10179	{ \tl_set:Nn #3 } { } { \tl_set:Nn #3 { \q_no_value } }	10179
10180	}	10180
10181	\cs_generate_variant:Nn \prop_pop:NnN { NV , No }	10181
10182	\cs_generate_variant:Nn \prop_pop:NnN { c , cV , co }	10182
10183	\cs_generate_variant:Nn \prop_gpop:NnN { NV , No }	10183
10184	\cs_generate_variant:Nn \prop_gpop:NnN { c , cV , co }	10184
10185	\prg_new_protected_conditional:Npnn \prop_pop:NnN #1#2#3 { T , F , TF }	10185
10186	{	10186
10187	__prop_pop:NnNNnTF #1 {#2}	10187
10188	\cs_set_eq:NN \cs_set_nopar:Npe	10188
10189	{ \tl_set:Nn #3 } \prg_return_true: \prg_return_false:	10189
10190	}	10190
10191	\prg_new_protected_conditional:Npnn \prop_gpop:NnN #1#2#3 { T , F , TF }	10191
10192	{	10192
10193	__prop_pop:NnNNnTF #1 {#2}	10193
10194	\cs_gset_eq:NN \cs_gset_nopar:Npe	10194
10195	{ \tl_set:Nn #3 } \prg_return_true: \prg_return_false:	10195
10196	}	10196
10197	\prg_generate_conditional_variant:Nnn \prop_pop:NnN	10197
10198	{ NV , No , c , cV , co } { T , F , TF }	10198
10199	\prg_generate_conditional_variant:Nnn \prop_gpop:NnN	10199
10200	{ NV , No , c , cV , co } { T , F , TF }	10200
10201	\cs_new_protected:Npn \prop_put:Nnn	10201
10202	{ __prop_put:nNNnn { } \cs_set_nopar:Npe }	10202
10203	\cs_new_protected:Npn \prop_gput:Nnn	10203
10204	{ __prop_put:nNNnn { } \cs_gset_nopar:Npe }	10204
10205	\cs_new_protected:Npn \prop_put_if_not_in:Nnn	10205
10206	{ __prop_put:nNNnn \use_none:nnn \cs_set_nopar:Npe }	10206
10207	\cs_new_protected:Npn \prop_gput_if_not_in:Nnn	10207
10208	{ __prop_put:nNNnn \use_none:nnn \cs_gset_nopar:Npe }	10208
10209	\cs_generate_variant:Nn \prop_put:Nnn	10209
10210	{	10210
10211	NnV , Nnv , Nne , NV , NVV , NVv , NVe ,	10211
10212	Nv , NvV , Nvv , Nve , Ne , NeV , Nev , Nee	10212

```
10213 }
10214 \cs_generate_variant:Nn \prop_put:Nnn
10215 { Nno , No , Noo , Nnx , NVx , NxV , Nxx }
10216 \cs_generate_variant:Nn \prop_put:Nnn
10217 {
10218     c , cnV , cnv , cne , cV , cVV , cVv , cVe ,
10219     cv , cvV , cvv , cve , ce , ceV , cev , cee
10220 }
10221 \cs_generate_variant:Nn \prop_put:Nnn
10222 { cno , co , coo , cnx , cVx , cxV , cxx }
10223 \cs_generate_variant:Nn \prop_gput:Nnn
10224 {
10225     NnV , Nnv , Nne , NV , NVV , NVv , NVe ,
10226     Nv , NvV , Nvv , Nve , Ne , NeV , Nev , Nee
10227 }
10228 \cs_generate_variant:Nn \prop_gput:Nnn
10229 { Nno , No , Noo , Nnx , NVx , NxV , Nxx }
10230 \cs_generate_variant:Nn \prop_gput:Nnn
10231 {
10232     c , cnV , cnv , cne , cV , cVV , cVv , cVe ,
10233     cv , cvV , cvv , cve , ce , ceV , cev , cee
10234 }
10235 \cs_generate_variant:Nn \prop_gput:Nnn
10236 { cno , co , coo , cnx , cVx , cxV , cxx }
10237 \cs_generate_variant:Nn \prop_put_if_not_in:Nnn
10238 {
10239     NnV , Nnv , Nne , NV , NVV , NVv , NVe ,
10240     Nv , NvV , Nvv , Nve , Ne , NeV , Nev , Nee ,
10241     c , cnV , cnv , cne , cV , cVV , cVv , cVe ,
10242     cv , cvV , cvv , cve , ce , ceV , cev , cee
10243 }
10244 \cs_generate_variant:Nn \prop_gput_if_not_in:Nnn
10245 {
10246     NnV , Nnv , Nne , NV , NVV , NVv , NVe ,
10247     Nv , NvV , Nvv , Nve , Ne , NeV , Nev , Nee ,
10248     c , cnV , cnv , cne , cV , cVV , cVv , cVe ,
10249     cv , cvV , cvv , cve , ce , ceV , cev , cee
10250 }
10251 \cs_new_protected:Npn \__prop_put:nNNnn #1#2#3#4#5
10252 {
10253     \tl_set:Nn \l__prop_internal_tl
10254     {
10255         \exp_not:N \__prop_pair:wn \tl_to_str:n {#4}
10256         \s__prop { \exp_not:n {#5} }
10257     }
10258     \__prop_split:NnTFn #3 {#4}
```

10259	{	10259
10260	#1 #2 #3	10260
10261	{	10261
10262	\s__prop __prop_chk:w \exp_not:n {##1}	10262
10263	\l__prop_internal_tl \exp_not:n {##3}	10263
10264	}	10264
10265	}	10265
10266	{ #2 #3 { \exp_not:o {#3} \l__prop_internal_tl } }	10266
10267	{ \exp_after:wN __prop_put_linked:wnnN #3 {#4} {#1} #2 }	10267
10268	}	10268
10269	\cs_new_protected:Npn __prop_put_linked:wnnN	10269
10270	__prop_flatten:w #1 \s__prop #2#3#4	10270
10271	{	10271
10272	\exp_after:wN __prop_put_linked:NNnN	10272
10273	\cs:w __prop ~ #2 ~ \tl_to_str:n {#4} \cs_end:	10273
10274	#1	10274
10275	}	10275
10276	\cs_new_protected:Npn __prop_put_linked:NNnN #1#2#3#4	10276
10277	{	10277
10278	\if_meaning:w \scan_stop: #1	10278
10279	\exp_after:wN __prop_put_linked_new:w #2 #1 #2 #4	10279
10280	\else:	10280
10281	\exp_after:wN __prop_put_linked_old:w #1 { #3 #4 #1 }	10281
10282	\fi:	10282
10283	}	10283
10284	\cs_new_protected:Npn __prop_put_linked_new:w	10284
10285	\use_none:n #1#2#3#4	10285
10286	{	10286
10287	#4 #1	10287
10288	{	10288
10289	\exp_after:wN __prop_pop_linked_prev:w #1	10289
10290	\exp_not:N #2	10290
10291	}	10291
10292	#4 #2	10292
10293	{	10293
10294	\exp_not:n { \use_none:n #1 }	10294
10295	\l__prop_internal_tl	10295
10296	\exp_not:N #3	10296
10297	}	10297
10298	#4 #3 { \exp_not:n { \use_none:n #2 } }	10298
10299	}	10299
10300	\cs_new_protected:Npn __prop_put_linked_old:w	10300
10301	\use_none:n #1#2 \s__prop #3#4#5	10301
10302	{	10302
10303	#5	10303
10304	{	10304

```
10305         \exp_not:n { \use_none:n #1 }
10306         \l__prop_internal_tl
10307         \exp_not:N #4
10308     }
10309 }
10310 \prg_new_eq_conditional:NNn \prop_if_exist:N \cs_if_exist:N
10311 { TF , T , F , p }
10312 \prg_new_eq_conditional:NNn \prop_if_exist:c \cs_if_exist:c
10313 { TF , T , F , p }
10314 \prg_new_conditional:Npnn \prop_if_empty:N #1 { p , T , F , TF }
10315 {
10316     \if_meaning:w #1 \c_empty_prop
10317     \prg_return_true:
10318 }else:
10319     \exp_after:wN \__prop_if_empty_return:w #1
10320     \__prop_flatten:w 2 \s__prop 34 \s__prop_stop
10321     \fi:
10322 }
10323 \cs_new:Npn \__prop_if_empty_return:w
10324 #1 \__prop_flatten:w #2 \s__prop #3#4#5 \s__prop_stop
10325 {
10326     \if_meaning:w #2 #4
10327     \prg_return_true:
10328 }else:
10329     \prg_return_false:
10330     \fi:
10331 }
10332 \prg_generate_conditional_variant:Nnn \prop_if_empty:N
10333 { c } { p , T , F , TF }
10334 \prg_new_conditional:Npnn \prop_if_in:Nn #1#2 { p , T , F , TF }
10335 {
10336     \__prop_if_flat:NTF #1
10337     {
10338         \exp_after:wN \prop_map_tokens:Nn \exp_after:wN #1
10339         {
10340             \exp_after:wN \__prop_if_in_flat:nnn
10341             \exp_after:wN { \tl_to_str:n {#2} }
10342         }
10343         \prg_return_false:
10344     }
10345     {
10346         \exp_after:wN \__prop_get_linked:w #1 {#2}
10347         \use_none:n \prg_return_true: \prg_return_false:
10348     }
10349 }
10350 \cs_new:Npn \__prop_if_in_flat:nnn #1#2#3
```



```
10351 {
10352     \str_if_eq:eeT {#1} {#2}
10353     { \prop_map_break:n { \use_i:nn \prg_return_true: } }
10354 }
10355 \prg_generate_conditional_variant:Nnn \prop_if_in:Nn
10356 { NV , Ne , No , c , cV , ce , co } { p , T , F , TF }
10357 \cs_new:Npn \prop_map_function:NN #1#2
10358 {
10359     \exp_last_unbraced:Nnf
10360     \use_i:nnn { \__prop_map_function:Nw #2 } #1
10361     \__prop_pair:wn \fi: \prop_map_break: \s__prop { }
10362     \__prop_pair:wn \fi: \prop_map_break: \s__prop { }
10363     \__prop_pair:wn \fi: \prop_map_break: \s__prop { }
10364     \__prop_pair:wn \fi: \prop_map_break: \s__prop { }
10365     \prg_break_point:Nn \prop_map_break: { }
10366 }
10367 \cs_new:Npn \__prop_map_function:Nw #1
10368     \__prop_pair:wn #2 \s__prop #3
10369     \__prop_pair:wn #4 \s__prop #5
10370     \__prop_pair:wn #6 \s__prop #7
10371     \__prop_pair:wn #8 \s__prop #9
10372 {
10373     \if_false: #2 \fi: #1 {#2} {#3}
10374     \if_false: #4 \fi: #1 {#4} {#5}
10375     \if_false: #6 \fi: #1 {#6} {#7}
10376     \if_false: #8 \fi: #1 {#8} {#9}
10377     \__prop_map_function:Nw #1
10378 }
10379 \cs_generate_variant:Nn \prop_map_function:NN { Nc , c , cc }
10380 \cs_new_protected:Npn \prop_map_inline:Nn #1#2
10381 {
10382     \cs_gset_eq:cN
10383     { \__prop_map_ \int_use:N \g__kernel_prg_map_int :wn } \__prop_pair:wn
10384     \int_gincr:N \g__kernel_prg_map_int
10385     \cs_gset_protected:Npn \__prop_pair:wn ##1 \s__prop ##2 {#2}
10386     \exp_last_unbraced:Nf \use_none:nn #1
10387     \prg_break_point:Nn \prop_map_break:
10388     {
10389         \int_gdecr:N \g__kernel_prg_map_int
10390         \cs_gset_eq:Nc \__prop_pair:wn
10391         { \__prop_map_ \int_use:N \g__kernel_prg_map_int :wn }
10392     }
10393 }
10394 \cs_generate_variant:Nn \prop_map_inline:Nn { c }
10395 \cs_new:Npn \prop_map_tokens:Nn #1#2
10396 {
```

```

10397 \exp_last_unbraced:Nnf
10398 \use_i:nnn { \__prop_map_tokens:nw {#2} } #1
10399 \__prop_pair:wn \fi: \prop_map_break: \s__prop { }
10400 \__prop_pair:wn \fi: \prop_map_break: \s__prop { }
10401 \__prop_pair:wn \fi: \prop_map_break: \s__prop { }
10402 \__prop_pair:wn \fi: \prop_map_break: \s__prop { }
10403 \prg_break_point:Nn \prop_map_break: { }
10404 }
10405 \cs_new:Npn \__prop_map_tokens:nw #1
10406 \__prop_pair:wn #2 \s__prop #3
10407 \__prop_pair:wn #4 \s__prop #5
10408 \__prop_pair:wn #6 \s__prop #7
10409 \__prop_pair:wn #8 \s__prop #9
10410 {
10411 \if_false: #2 \fi: \use:n {#1} {#2} {#3}
10412 \if_false: #4 \fi: \use:n {#1} {#4} {#5}
10413 \if_false: #6 \fi: \use:n {#1} {#6} {#7}
10414 \if_false: #8 \fi: \use:n {#1} {#8} {#9}
10415 \__prop_map_tokens:nw {#1}
10416 }
10417 \cs_generate_variant:Nn \prop_map_tokens:Nn { c }
10418 \cs_new:Npn \prop_map_break:
10419 { \prg_map_break:Nn \prop_map_break: { } }
10420 \cs_new:Npn \prop_map_break:n
10421 { \prg_map_break:Nn \prop_map_break: }
10422 \cs_new:Npn \prop_count:N #1
10423 {
10424 \int_eval:n
10425 {
10426 0
10427 \prop_map_function:NN #1 \__prop_count:nn
10428 }
10429 }
10430 \cs_new:Npn \__prop_count:nn #1#2 { + 1 }
10431 \cs_generate_variant:Nn \prop_count:N { c }
10432 \cs_new:Npn \prop_to_keyval:N #1
10433 {
10434 \__kernel_exp_not:w
10435 \prop_if_empty:NTF #1
10436 { {} }
10437 {
10438 \exp_after:wN \exp_after:wN \exp_after:wN
10439 {
10440 \tex_expanded:D
10441 {
10442 \exp_not:N \use_none:n

```

```
10443 \prop_map_function:NN #1 \__prop_to_keyval:nn 10443
10444 } 10444
10445 } 10445
10446 } 10446
10447 } 10447
10448 \cs_new:Npn \__prop_to_keyval:nn #1#2 10448
10449 { , ~ {#1} =~ { \__kernel_exp_not:w {#2} } } 10449
10450 \cs_new_protected:Npn \prop_show:N { \__prop_show:NN \msg_show:nneeee } 10450
10451 \cs_generate_variant:Nn \prop_show:N { c } 10451
10452 \cs_new_protected:Npn \prop_log:N { \__prop_show:NN \msg_log:nneeee } 10452
10453 \cs_generate_variant:Nn \prop_log:N { c } 10453
10454 \cs_new_protected:Npn \__prop_show:NN #1#2 10454
10455 { 10455
10456 \__kernel_chk_tl_type:NnnT #2 { prop } 10456
10457 { 10457
10458 \__prop_if_flat:NTF #2 10458
10459 { 10459
10460 \s__prop \__prop_chk:w 10460
10461 \exp_after:wN \__prop_show_flat:w #2 10461
10462 \s__prop { } 10462
10463 \__prop_pair:wn \q__prop_recursion_tail \s__prop { } 10463
10464 \q__prop_recursion_stop 10464
10465 } 10465
10466 { \exp_after:wN \__prop_show_linked:w #2 \s__prop ! ? \s__prop_stop } 10466
10467 } 10467
10468 { 10468
10469 \__prop_if_flat:NTF #2 10469
10470 { \__prop_show_finally:NNn #1 #2 { flat } } 10470
10471 { 10471
10472 \tl_set:Nn \l__prop_internal_tl { #1 #2 } 10472
10473 \exp_after:wN \__prop_show_prepare:w #2 #2 10473
10474 } 10474
10475 } 10475
10476 } 10476
10477 \cs_new:Npn \__prop_show_flat:w #1 \__prop_pair:wn #2 \s__prop #3 10477
10478 { 10478
10479 \__prop_if_recursion_tail_stop:n {#2} 10479
10480 \exp_not:N \__prop_pair:wn \tl_to_str:n {#2} \s__prop \exp_not:n { {#3} } 10480
10481 \__prop_show_flat:w 10481
10482 } 10482
10483 \cs_new:Npn \__prop_show_linked:w #1 \s__prop #2#3#4 \s__prop_stop 10483
10484 { 10484
10485 \exp_not:N \__prop_flatten:w 10485
10486 \exp_not:c { __prop ~ \tl_to_str:n {#2} } 10486
10487 \s__prop { \tl_to_str:n {#2} } 10487
10488 \exp_not:n {#3} 10488
```

```
10489 } 10489
10490 \cs_new_protected:Npn \__prop_show_finally:NNn #1#2#3 10490
10491 { 10491
10492     #1 { prop } { show } 10492
10493     { \token_to_str:N #2 } 10493
10494     { \prop_map_function:NN #2 \msg_show_item:nn } 10494
10495     {#3} { } 10495
10496 } 10496
10497 \cs_new_protected:Npn \__prop_show_prepare:w 10497
10498     \__prop_flatten:w #1 \s__prop #2#3#4 10498
10499 { 10499
10500     \use:e 10500
10501     { 10501
10502         \cs_set_nopar:Npn \exp_not:N \__prop_tmp:w 10502
10503         ##1 \token_to_str:N #1 ##2 \s__prop_mark ##3 \s__prop_stop 10503
10504         { 10504
10505             \exp_not:N \tl_if_empty:nTF {##1} 10505
10506             { 10506
10507                 \exp_not:N \tl_if_head_is_space:nTF {##2} 10507
10508                 { \exp_not:N \exp_args:Nf \__prop_show_loop_key:wNNN } 10508
10509                 { \exp_not:N \tl_if_empty:nTF } 10509
10510                 {##2} 10510
10511             } 10511
10512             { \exp_not:N \use_ii:nn } 10512
10513             \__prop_show_end:NNN 10513
10514             \__prop_show_bad_name:NNN 10514
10515         } 10515
10516     } 10516
10517     \exp_last_unbraced:NNNo \__prop_show_loop:NNw #1 #4 #4 10517
10518 } 10518
10519 \cs_new_protected:Npn \__prop_show_loop:NNw #1#2 #3 \s__prop #4#5 10519
10520 { 10520
10521     \exp_last_two_unbraced:Noo \__prop_tmp:w 10521
10522     { \token_to_str:N #5 \s__prop_mark } 10522
10523     { \token_to_str:N #1 \s__prop_mark \s__prop_stop } 10523
10524     #1 #2 #5 10524
10525 } 10525
10526 \cs_new_protected:Npn \__prop_show_bad_name:NNN #1#2#3 10526
10527 { 10527
10528     \msg_error:nneeee { prop } { bad-link } 10528
10529     { \tl_tail:N \l__prop_internal_tl } 10529
10530     { \token_to_str:N #2 } 10530
10531     { \token_to_str:N #3 } 10531
10532     { \token_to_str:N #1 } 10532
10533 } 10533
10534 \cs_new_protected:Npn \__prop_show_end:NNN #1#2#3 10534
```

```

10535 {
10536     \__kernel_chk_tl_type:NnnT #3
10537     { \tl_tail:N \l__prop_internal_tl prop~entry }
10538     { \exp_not:n { \use_none:n #2 } }
10539     {
10540         \exp_after:wN \__prop_show_finally:NNn
10541         \l__prop_internal_tl { linked }
10542     }
10543 }
10544 \cs_new_protected:Npn \__prop_show_loop_key:wNNN #1#2#3#4#5#6
10545 {
10546     \__kernel_chk_tl_type:NnnT #6
10547     { \tl_tail:N \l__prop_internal_tl prop~entry }
10548     {
10549         \exp_not:n { \use_none:n #5 }
10550         \exp_after:wN \__prop_show_flat:w #6 \s__prop { }
10551         \__prop_pair:wn \q__prop_recursion_tail \s__prop { }
10552         \q__prop_recursion_stop
10553         \tl_item:Nn #6 { -1 }
10554     }
10555     { \exp_last_unbraced:NNNo \__prop_show_loop:NNw #4 #6 #6 }
10556 }
10557 %% File: l3msg.dtx
10558 \tl_new:N \l__msg_internal_tl
10559 \str_new:N \l__msg_name_str
10560 \str_new:N \l__msg_text_str
10561 \scan_new:N \s__msg_mark
10562 \scan_new:N \s__msg_stop
10563 \cs_new:Npn \__msg_use_none_delimit_by_s_stop:w #1 \s__msg_stop { }
10564 \tl_const:Nn \c__msg_text_prefix_tl { msg~text~>~ }
10565 \tl_const:Nn \c__msg_more_text_prefix_tl { msg~extra~text~>~ }
10566 \prg_new_conditional:Npnn \msg_if_exist:nn #1#2 { p , T , F , TF }
10567 {
10568     \cs_if_exist:cTF { \c__msg_text_prefix_tl #1 / #2 }
10569     { \prg_return_true: } { \prg_return_false: }
10570 }
10571 \cs_new_protected:Npn \__msg_chk_free:nn #1#2
10572 {
10573     \msg_if_exist:nnT {#1} {#2}
10574     {
10575         \msg_error:nnnn { msg } { already-defined }
10576         {#1} {#2}
10577     }
10578 }
10579 \cs_new_protected:Npn \msg_new:nnnn #1#2#3#4
10580 {

```

10581	_msg_chk_free:nn {#1} {#2}	10581
10582	\cs_gset:cpn { \c__msg_text_prefix_tl #1 / #2 }	10582
10583	##1##2##3##4 {#3}	10583
10584	\cs_gset:cpn { \c__msg_more_text_prefix_tl #1 / #2 }	10584
10585	##1##2##3##4 {#4}	10585
10586	}	10586
10587	\cs_generate_variant:Nn \msg_new:nnnn { nnee , nxxx }	10587
10588	\cs_new_protected:Npn \msg_new:nnn #1#2#3	10588
10589	{ \msg_new:nnnn {#1} {#2} {#3} { } }	10589
10590	\cs_generate_variant:Nn \msg_new:nnn { nne , nnx }	10590
10591	\cs_new_protected:Npn \msg_set:nnnn #1#2#3#4	10591
10592	{	10592
10593	\cs_set:cpn { \c__msg_text_prefix_tl #1 / #2 }	10593
10594	##1##2##3##4 {#3}	10594
10595	\cs_set:cpn { \c__msg_more_text_prefix_tl #1 / #2 }	10595
10596	##1##2##3##4 {#4}	10596
10597	}	10597
10598	\cs_new_protected:Npn \msg_set:nnn #1#2#3	10598
10599	{ \msg_set:nnnn {#1} {#2} {#3} { } }	10599
10600	\tl_const:Nn \c__msg_coding_error_text_tl	10600
10601	{	10601
10602	This~is~a~coding~error.	10602
10603	\\ \\	10603
10604	}	10604
10605	\tl_const:Nn \c__msg_continue_text_tl	10605
10606	{ Type~<return>~to~continue }	10606
10607	\tl_const:Nn \c__msg_critical_text_tl	10607
10608	{ Reading~the~current~file~'\g_file_curr_name_str'~will~stop. }	10608
10609	\tl_const:Nn \c__msg_fatal_text_tl	10609
10610	{ This~is~a~fatal~error:~LaTeX~will~abort. }	10610
10611	\tl_const:Nn \c__msg_help_text_tl	10611
10612	{ For~immediate~help~type~H~<return> }	10612
10613	\tl_const:Nn \c__msg_no_info_text_tl	10613
10614	{	10614
10615	LaTeX~does~not~know~anything~more~about~this~error,~sorry.	10615
10616	\c__msg_return_text_tl	10616
10617	}	10617
10618	\tl_const:Nn \c__msg_on_line_text_tl { on~line }	10618
10619	\tl_const:Nn \c__msg_return_text_tl	10619
10620	{	10620
10621	\\ \\	10621
10622	Try~typing~<return>~to~proceed.	10622
10623	\\	10623
10624	If~that~doesn't~work,~type~X~<return>~to~quit.	10624
10625	}	10625
10626	\tl_const:Nn \c__msg_trouble_text_tl	10626


```
10627 { 10627
10628 \\ \\ 10628
10629 More~errors~will~almost~certainly~follow: \\ 10629
10630 the~LaTeX~run~should~be~aborted. 10630
10631 } 10631
10632 \cs_new:Npn \msg_line_number: { \int_use:N \tex_inputlineno:D } 10632
10633 \cs_gset:Npn \msg_line_context: 10633
10634 { 10634
10635 \c__msg_on_line_text_tl 10635
10636 \c_space_tl 10636
10637 \msg_line_number: 10637
10638 } 10638
10639 \cs_new_protected:Npn \__msg_interrupt:NnnnN #1#2#3#4#5 10639
10640 { 10640
10641 \str_set:Ne \l__msg_text_str { #1 {#2} } 10641
10642 \str_set:Ne \l__msg_name_str { \msg_module_name:n {#2} } 10642
10643 \cs_if_eq:cNTF 10643
10644 { \c__msg_more_text_prefix_tl #2 / #3 } 10644
10645 \__msg_no_more_text:nnnn 10645
10646 { 10646
10647 \__msg_interrupt_wrap:nnn 10647
10648 { \use:c { \c__msg_text_prefix_tl #2 / #3 } #4 } 10648
10649 { \c__msg_continue_text_tl } 10649
10650 { 10650
10651 \c__msg_no_info_text_tl 10651
10652 \tl_if_empty:NF #5 10652
10653 { \\ \\ #5 } 10653
10654 } 10654
10655 } 10655
10656 { 10656
10657 \__msg_interrupt_wrap:nnn 10657
10658 { \use:c { \c__msg_text_prefix_tl #2 / #3 } #4 } 10658
10659 { \c__msg_help_text_tl } 10659
10660 { 10660
10661 \use:c { \c__msg_more_text_prefix_tl #2 / #3 } #4 10661
10662 \tl_if_empty:NF #5 10662
10663 { \\ \\ #5 } 10663
10664 } 10664
10665 } 10665
10666 } 10666
10667 \cs_new:Npn \__msg_no_more_text:nnnn #1#2#3#4 { } 10667
10668 \cs_new_protected:Npn \__msg_interrupt_wrap:nnn #1#2#3 10668
10669 { 10669
10670 \iow_wrap:nnnN { \\ #3 } { } { } \__msg_interrupt_more_text:n 10670
10671 \group_begin: 10671
10672 \int_sub:Nn \l_iow_line_count_int { 2 } 10672
```

```

10673 \iow_wrap:nenN { \l__msg_text_str : ~ #1 } 10673
10674 { 10674
10675 ( \l__msg_name_str ) 10675
10676 \prg_replicate:nn 10676
10677 { 10677
10678 \str_count:N \l__msg_text_str 10678
10679 - \str_count:N \l__msg_name_str 10679
10680 + 2 10680
10681 } 10681
10682 { ~ } 10682
10683 } 10683
10684 { } \__msg_interrupt_text:n 10684
10685 \iow_wrap:nnnN { \l__msg_internal_tl \\ \\ #2 } { } { } 10685
10686 \__msg_interrupt:n 10686
10687 } 10687
10688 \cs_new_protected:Npn \__msg_interrupt_text:n #1 10688
10689 { 10689
10690 \group_end: 10690
10691 \tl_set:Nn \l__msg_internal_tl {#1} 10691
10692 } 10692
10693 \cs_new_protected:Npn \__msg_interrupt_more_text:n #1 10693
10694 { \exp_args:Ne \tex_errhelp:D { #1 \iow_newline: } } 10694
10695 \group_begin: 10695
10696 \char_set_lccode:nn { 38 } { 32 } % & 10696
10697 \char_set_lccode:nn { 46 } { 32 } % . 10697
10698 \char_set_lccode:nn { 123 } { 32 } % { 10698
10699 \char_set_lccode:nn { 125 } { 32 } % } 10699
10700 \char_set_catcode_active:N \& 10700
10701 \tex_lowercase:D 10701
10702 { 10702
10703 \group_end: 10703
10704 \cs_new_protected:Npn \__msg_interrupt:n #1 10704
10705 { 10705
10706 \iow_term:n { } 10706
10707 \__kernel_iow_with:Nnn \tex_newlinechar:D { ``^^J } 10707
10708 { 10708
10709 \__kernel_iow_with:Nnn \tex_errorcontextlines:D { -1 } 10709
10710 { 10710
10711 \group_begin: 10711
10712 \cs_set_protected:Npn & 10712
10713 { 10713
10714 \tex_errmessage:D 10714
10715 { 10715
10716 #1 10716
10717 \use_none:n 10717
10718 { ..... } 10718

```

```

10719         }
10720     }
10721     \exp_after:wN
10722     \group_end:
10723     &
10724 }
10725 }
10726 }
10727 }
10728 \int_gset:Nn \tex_errorcontextlines:D { -1 }
10729 \cs_new:Npn \msg_fatal_text:n #1
10730 {
10731     Fatal ~
10732     \msg_error_text:n {#1}
10733 }
10734 \cs_new:Npn \msg_critical_text:n #1
10735 {
10736     Critical ~
10737     \msg_error_text:n {#1}
10738 }
10739 \cs_new:Npn \msg_error_text:n #1
10740 { \__msg_text:nn {#1} { Error } }
10741 \cs_new:Npn \msg_warning_text:n #1
10742 { \__msg_text:nn {#1} { Warning } }
10743 \cs_new:Npn \msg_info_text:n #1
10744 { \__msg_text:nn {#1} { Info } }
10745 \cs_new:Npn \__msg_text:nn #1#2
10746 {
10747     \exp_args:Nf \__msg_text:n { \msg_module_type:n {#1} }
10748     \exp_args:Nf \__msg_text:n { \msg_module_name:n {#1} }
10749     #2
10750 }
10751 \cs_new:Npn \__msg_text:n #1
10752 {
10753     \tl_if_blank:nF {#1}
10754     { #1 ~ }
10755 }
10756 \prop_new:N \g_msg_module_name_prop
10757 \prop_new:N \g_msg_module_type_prop
10758 \prop_gput:Nnn \g_msg_module_type_prop { LaTeX } { }
10759 \cs_new:Npn \msg_module_type:n #1
10760 {
10761     \prop_if_in:NnTF \g_msg_module_type_prop {#1}
10762     { \prop_item:Nn \g_msg_module_type_prop {#1} }
10763     { Package }
10764 }

```

```
10765 \cs_new:Npn \msg_module_name:n #1 10765
10766 { 10766
10767     \prop_if_in:NnTF \g_msg_module_name_prop {#1} 10767
10768     { \prop_item:Nn \g_msg_module_name_prop {#1} } 10768
10769     {#1} 10769
10770 } 10770
10771 \cs_new:Npn \msg_see_documentation_text:n #1 10771
10772 { 10772
10773     See~the~ \msg_module_name:n {#1} ~ 10773
10774     documentation~for~further~information. 10774
10775 } 10775
10776 \group_begin: 10776
10777     \cs_set_protected:Npn \__msg_class_new:nn #1#2 10777
10778     { 10778
10779         \prop_new:c { l__msg_redirect_ #1 _prop } 10779
10780         \cs_new_protected:cpn { __msg_ #1 _code:nnnnnn } 10780
10781         ##1##2##3##4##5##6 {#2} 10781
10782         \cs_new_protected:cpn { msg_ #1 :nnnnnn } ##1##2##3##4##5##6 10782
10783         { 10783
10784             \use:e 10784
10785             { 10785
10786                 \exp_not:n { \__msg_use:nnnnnnn {#1} {##1} {##2} } 10786
10787                 { \tl_to_str:n {##3} } { \tl_to_str:n {##4} } 10787
10788                 { \tl_to_str:n {##5} } { \tl_to_str:n {##6} } 10788
10789             } 10789
10790         } 10790
10791         \cs_new_protected:cpe { msg_ #1 :nnnnn } ##1##2##3##4##5 10791
10792         { \exp_not:c { msg_ #1 :nnnnnn } {##1} {##2} {##3} {##4} {##5} { } } 10792
10793         \cs_new_protected:cpe { msg_ #1 :nnnn } ##1##2##3##4 10793
10794         { \exp_not:c { msg_ #1 :nnnnnn } {##1} {##2} {##3} {##4} { } { } } 10794
10795         \cs_new_protected:cpe { msg_ #1 :nnn } ##1##2##3 10795
10796         { \exp_not:c { msg_ #1 :nnnnnn } {##1} {##2} {##3} { } { } { } } 10796
10797         \cs_new_protected:cpe { msg_ #1 :nn } ##1##2 10797
10798         { \exp_not:c { msg_ #1 :nnnnnn } {##1} {##2} { } { } { } { } } 10798
10799         \cs_generate_variant:cn { msg_ #1 :nnn } 10799
10800         { nnV , nne , nnx } 10800
10801         \cs_generate_variant:cn { msg_ #1 :nnnn } 10801
10802         { nnVV , nnVn , nnnV , nnne , nnnx , nnee , nnxx } 10802
10803         \cs_generate_variant:cn { msg_ #1 :nnnnn } 10803
10804         { nnnee , nnnxx , nneee , nnxxx } 10804
10805         \cs_generate_variant:cn { msg_ #1 :nnnnnn } { nneeee , nnxxxx } 10805
10806     } 10806
10807     \__msg_class_new:nn { fatal } 10807
10808     { 10808
10809         \__msg_interrupt:NnnnN 10809
10810         \msg_fatal_text:n {#1} {#2} 10810
```

10811	{ {#3} {#4} {#5} {#6} }	10811
10812	\c__msg_fatal_text_tl	10812
10813	_msg_fatal_exit:	10813
10814	}	10814
10815	\cs_new_protected:Npn _msg_fatal_exit:	10815
10816	{	10816
10817	\tex_batchmode:D	10817
10818	\tex_read:D -1 to \l__msg_internal_tl	10818
10819	}	10819
10820	_msg_class_new:nn { critical }	10820
10821	{	10821
10822	_msg_interrupt:NnnnN	10822
10823	\msg_critical_text:n {#1} {#2}	10823
10824	{ {#3} {#4} {#5} {#6} }	10824
10825	\c__msg_critical_text_tl	10825
10826	\tex_endinput:D	10826
10827	}	10827
10828	\cs_undefine:N \msg_error:nnee	10828
10829	\cs_undefine:N \msg_error:nne	10829
10830	\cs_undefine:N \msg_error:nn	10830
10831	_msg_class_new:nn { error }	10831
10832	{	10832
10833	_msg_interrupt:NnnnN	10833
10834	\msg_error_text:n {#1} {#2}	10834
10835	{ {#3} {#4} {#5} {#6} }	10835
10836	\c_empty_tl	10836
10837	}	10837
10838	\cs_new_protected:Npn _msg_info_aux:NNnnnnnn #1#2#3#4#5#6#7#8	10838
10839	{	10839
10840	\str_set:Ne \l__msg_text_str { #2 {#3} }	10840
10841	\str_set:Ne \l__msg_name_str { \msg_module_name:n {#3} }	10841
10842	#1 { }	10842
10843	\iow_wrap:nenN	10843
10844	{	10844
10845	\l__msg_text_str : ~	10845
10846	\use:c { \c__msg_text_prefix_tl #3 / #4 } {#5} {#6} {#7} {#8}	10846
10847	}	10847
10848	{	10848
10849	(\l__msg_name_str)	10849
10850	\prg_replicate:nn	10850
10851	{	10851
10852	\str_count:N \l__msg_text_str	10852
10853	- \str_count:N \l__msg_name_str	10853
10854	}	10854
10855	{ ~ }	10855
10856	}	10856

```
10857     { } #1
10858     #1 { }
10859 }
10860 \__msg_class_new:nn { warning }
10861 {
10862     \__msg_info_aux:NNnnnnnn \iow_term:n \msg_warning_text:n
10863     {#1} {#2} {#3} {#4} {#5} {#6}
10864 }
10865 \__msg_class_new:nn { note }
10866 {
10867     \__msg_info_aux:NNnnnnnn \iow_term:n \msg_info_text:n
10868     {#1} {#2} {#3} {#4} {#5} {#6}
10869 }
10870 \__msg_class_new:nn { info }
10871 {
10872     \__msg_info_aux:NNnnnnnn \iow_log:n \msg_info_text:n
10873     {#1} {#2} {#3} {#4} {#5} {#6}
10874 }
10875 \__msg_class_new:nn { log }
10876 {
10877     \iow_wrap:nnnN
10878     { \use:c { \c__msg_text_prefix_tl #1 / #2 } {#3} {#4} {#5} {#6} }
10879     { } { } \iow_log:n
10880 }
10881 \__msg_class_new:nn { term }
10882 {
10883     \iow_wrap:nnnN
10884     { \use:c { \c__msg_text_prefix_tl #1 / #2 } {#3} {#4} {#5} {#6} }
10885     { } { } \iow_term:n
10886 }
10887 \__msg_class_new:nn { none } { }
10888 \__msg_class_new:nn { show }
10889 {
10890     \iow_wrap:nnnN
10891     { \use:c { \c__msg_text_prefix_tl #1 / #2 } {#3} {#4} {#5} {#6} }
10892     { } { } \__msg_show:n
10893 }
10894 \cs_new_protected:Npn \__msg_show:n #1
10895 {
10896     \tl_if_in:nnTF { ^^J #1 } { ^^J > ~ }
10897     {
10898         \tl_if_in:nnTF { #1 \s__msg_mark } { . \s__msg_mark }
10899         { \__msg_show_dot:w } { \__msg_show:w }
10900         ^^J #1 \s__msg_stop
10901     }
10902     { \__msg_show:nn { ? #1 } { } }
```



```

10903 }
10904 \cs_new:Npn \__msg_show_dot:w #1 ^^J > ~ #2 . \s__msg_stop
10905 { \__msg_show:nn {#1} {#2} }
10906 \cs_new:Npn \__msg_show:w #1 ^^J > ~ #2 \s__msg_stop
10907 { \__msg_show:nn {#1} {#2} }
10908 \cs_new_protected:Npn \__msg_show:nn #1#2
10909 {
10910   \tl_if_empty:nF {#1}
10911   { \exp_args:No \iow_term:n { \use_none:n #1 } }
10912   \tl_set:Nn \l__msg_internal_tl {#2}
10913   \__kernel_iow_with:Nnn \tex_newlinechar:D { 10 }
10914   {
10915     \__kernel_iow_with:Nnn \tex_errorcontextlines:D { -1 }
10916     {
10917       \tex_showtokens:D \exp_after:wN \exp_after:wN \exp_after:wN
10918       { \exp_after:wN \l__msg_internal_tl }
10919     }
10920   }
10921 }
10922 \group_end:
10923 \cs_new:Npe \msg_show_item:n #1
10924 { ^^J > ~ \c_space_tl \exp_not:N \tl_to_str:n { {#1} } }
10925 \cs_new:Npe \msg_show_item_unbraced:n #1
10926 { ^^J > ~ \c_space_tl \exp_not:N \tl_to_str:n {#1} }
10927 \cs_new:Npe \msg_show_item:nn #1#2
10928 {
10929   ^^J > \use:nn { ~ } { ~ }
10930   \exp_not:N \tl_to_str:n { {#1} }
10931   \use:nn { ~ } { ~ } => \use:nn { ~ } { ~ }
10932   \exp_not:N \tl_to_str:n { {#2} }
10933 }
10934 \cs_new:Npe \msg_show_item_unbraced:nn #1#2
10935 {
10936   ^^J > \use:nn { ~ } { ~ }
10937   \exp_not:N \tl_to_str:n {#1}
10938   \use:nn { ~ } { ~ } => \use:nn { ~ } { ~ }
10939   \exp_not:N \tl_to_str:n {#2}
10940 }
10941 \cs_new:Npn \__msg_class_chk_exist:nT #1
10942 {
10943   \cs_if_free:cTF { __msg_ #1 _code:nnnnnn }
10944   { \msg_error:nnn { msg } { class-unknown } {#1} }
10945 }
10946 \tl_new:N \l__msg_class_tl
10947 \tl_new:N \l__msg_current_class_tl
10948 \prop_new:N \l__msg_redirect_prop

```

```
10949 \seq_new:N \l__msg_hierarchy_seq 10949
10950 \seq_new:N \l__msg_class_loop_seq 10950
10951 \cs_new_protected:Npn \__msg_use:nnnnnnn #1#2#3#4#5#6#7 10951
10952 { 10952
10953     \cs_if_exist_use:N \conditionally@traceoff 10953
10954     \msg_if_exist:nnTF {#2} {#3} 10954
10955     { 10955
10956         \__msg_class_chk_exist:nT {#1} 10956
10957         { 10957
10958             \tl_set:Nn \l__msg_current_class_tl {#1} 10958
10959             \cs_set_protected:Npe \__msg_use_code: 10959
10960             { 10960
10961                 \exp_not:n 10961
10962                 { 10962
10963                     \use:c { __msg_ \l__msg_class_tl _code:nnnnnn } 10963
10964                     {#2} {#3} {#4} {#5} {#6} {#7} 10964
10965                 } 10965
10966             } 10966
10967             \__msg_use_redirect_name:n { #2 / #3 } 10967
10968         } 10968
10969     } 10969
10970     { \msg_error:nnnn { msg } { unknown } {#2} {#3} } 10970
10971     \cs_if_exist_use:N \conditionally@tracelon 10971
10972 } 10972
10973 \cs_new_protected:Npn \__msg_use_code: { } 10973
10974 \cs_new_protected:Npn \__msg_use_redirect_name:n #1 10974
10975 { 10975
10976     \prop_get:NnNTF \l__msg_redirect_prop { / #1 } \l__msg_class_tl 10976
10977     { \__msg_use_code: } 10977
10978     { 10978
10979         \seq_clear:N \l__msg_hierarchy_seq 10979
10980         \__msg_use_hierarchy:nwWN { } 10980
10981         #1 \s__msg_mark \__msg_use_hierarchy:nwWN 10981
10982         / \s__msg_mark \__msg_use_none_delimit_by_s_stop:w 10982
10983         \s__msg_stop 10983
10984         \__msg_use_redirect_module:n { } 10984
10985     } 10985
10986 } 10986
10987 \cs_new_protected:Npn \__msg_use_hierarchy:nwWN #1#2 / #3 \s__msg_mark #4 10987
10988 { 10988
10989     \seq_put_left:Nn \l__msg_hierarchy_seq {#1} 10989
10990     #4 { #1 / #2 } #3 \s__msg_mark #4 10990
10991 } 10991
10992 \cs_new_protected:Npn \__msg_use_redirect_module:n #1 10992
10993 { 10993
10994     \seq_map_inline:Nn \l__msg_hierarchy_seq 10994
```

```
10995 { 10995
10996 \prop_get:cnNTF { l__msg_redirect_ \l__msg_current_class_tl _prop } 10996
10997 {##1} \l__msg_class_tl 10997
10998 { 10998
10999 \seq_map_break:n 10999
11000 { 11000
11001 \tl_if_eq:NNTF \l__msg_current_class_tl \l__msg_class_tl 11001
11002 { \__msg_use_code: } 11002
11003 { 11003
11004 \tl_set_eq:NN \l__msg_current_class_tl \l__msg_class_tl 11004
11005 \__msg_use_redirect_module:n {##1} 11005
11006 } 11006
11007 } 11007
11008 } 11008
11009 { 11009
11010 \str_if_eq:nnT {##1} {#1} 11010
11011 { 11011
11012 \tl_set_eq:NN \l__msg_class_tl \l__msg_current_class_tl 11012
11013 \seq_map_break:n { \__msg_use_code: } 11013
11014 } 11014
11015 } 11015
11016 } 11016
11017 } 11017
11018 \cs_new_protected:Npn \msg_redirect_name:nnn #1#2#3 11018
11019 { 11019
11020 \tl_if_empty:nTF {#3} 11020
11021 { \prop_remove:Nn \l__msg_redirect_prop { / #1 / #2 } } 11021
11022 { 11022
11023 \__msg_class_chk_exist:nT {#3} 11023
11024 { \prop_put:Nnn \l__msg_redirect_prop { / #1 / #2 } {#3} } 11024
11025 } 11025
11026 } 11026
11027 \cs_new_protected:Npn \msg_redirect_class:nn 11027
11028 { \__msg_redirect:nnn { } } 11028
11029 \cs_new_protected:Npn \msg_redirect_module:nnn #1 11029
11030 { \__msg_redirect:nnn { / #1 } } 11030
11031 \cs_new_protected:Npn \__msg_redirect:nnn #1#2#3 11031
11032 { 11032
11033 \__msg_class_chk_exist:nT {#2} 11033
11034 { 11034
11035 \tl_if_empty:nTF {#3} 11035
11036 { \prop_remove:cn { l__msg_redirect_ #2 _prop } {#1} } 11036
11037 { 11037
11038 \__msg_class_chk_exist:nT {#3} 11038
11039 { 11039
11040 \prop_put:cnn { l__msg_redirect_ #2 _prop } {#1} {#3} 11040
```

```

11041         \tl_set:Nn \l__msg_current_class_tl {#2} 11041
11042         \seq_clear:N \l__msg_class_loop_seq 11042
11043         \__msg_redirect_loop_chk:nnn {#2} {#3} {#1} 11043
11044     } 11044
11045 } 11045
11046 } 11046
11047 } 11047
11048 \cs_new_protected:Npn \__msg_redirect_loop_chk:nnn #1#2#3 11048
11049 { 11049
11050     \seq_put_right:Nn \l__msg_class_loop_seq {#1} 11050
11051     \prop_get:cnNT { l__msg_redirect_ #1 _prop } {#3} \l__msg_class_tl 11051
11052     { 11052
11053         \str_if_eq:VnF \l__msg_class_tl {#1} 11053
11054         { 11054
11055             \tl_if_eq:NNTF \l__msg_class_tl \l__msg_current_class_tl 11055
11056             { 11056
11057                 \prop_put:cnn { l__msg_redirect_ #2 _prop } {#3} {#2} 11057
11058                 \msg_warning:nneeee 11058
11059                 { msg } { redirect-loop } 11059
11060                 { \seq_item:Nn \l__msg_class_loop_seq { 1 } } 11060
11061                 { \seq_item:Nn \l__msg_class_loop_seq { 2 } } 11061
11062                 {#3} 11062
11063                 { 11063
11064                     \seq_map_function:NN \l__msg_class_loop_seq 11064
11065                     \__msg_redirect_loop_list:n 11065
11066                     { \seq_item:Nn \l__msg_class_loop_seq { 1 } } 11066
11067                 } 11067
11068             } 11068
11069             { \__msg_redirect_loop_chk:onn \l__msg_class_tl {#2} {#3} } 11069
11070         } 11070
11071     } 11071
11072 } 11072
11073 \cs_generate_variant:Nn \__msg_redirect_loop_chk:nnn { o } 11073
11074 \cs_new:Npn \__msg_redirect_loop_list:n #1 { {#1} ~ => ~ } 11074
11075 \cs_new_protected:Npn \__kernel_msg_show_eval:Nn #1#2 11075
11076     { \exp_args:Nf \__msg_show_eval:nnN { #1 {#2} } {#2} \tl_show:n } 11076
11077 \cs_new_protected:Npn \__kernel_msg_log_eval:Nn #1#2 11077
11078     { \exp_args:Nf \__msg_show_eval:nnN { #1 {#2} } {#2} \tl_log:n } 11078
11079 \cs_new_protected:Npn \__msg_show_eval:nnN #1#2#3 { #3 { #2 = #1 } } 11079
11080 \cs_new_protected:Npn \__kernel_msg_new:nnnn #1 11080
11081     { \msg_new:nnnn { LaTeX / #1 } } 11081
11082 \cs_new_protected:Npn \__kernel_msg_new:nnn #1 11082
11083     { \msg_new:nnn { LaTeX / #1 } } 11083
11084 \cs_new_protected:Npn \__kernel_msg_info:nnxx #1 11084
11085     { \msg_info:nnee { LaTeX / #1 } } 11085
11086 \cs_new_protected:Npn \__kernel_msg_warning:nnx #1 11086

```

```

11087 { \msg_warning:nne { LaTeX / #1 } } 11087
11088 \cs_new_protected:Npn \__kernel_msg_warning:nnxx #1 11088
11089 { \msg_warning:nnee { LaTeX / #1 } } 11089
11090 \cs_new_protected:Npn \__kernel_msg_error:nnx #1 11090
11091 { \msg_error:nne { LaTeX / #1 } } 11091
11092 \cs_new_protected:Npn \__kernel_msg_error:nnxx #1 11092
11093 { \msg_error:nnee { LaTeX / #1 } } 11093
11094 \cs_new_protected:Npn \__kernel_msg_error:nnxxx #1 11094
11095 { \msg_error:nneee { LaTeX / #1 } } 11095
11096 \cs_new:Npn \__kernel_msg_expandable_error:nnn #1 11096
11097 { \msg_expandable_error:nnn { LaTeX / #1 } } 11097
11098 \cs_new:Npn \__kernel_msg_expandable_error:nnf #1 11098
11099 { \msg_expandable_error:nnf { LaTeX / #1 } } 11099
11100 \cs_new:Npn \__kernel_msg_expandable_error:nnff #1 11100
11101 { \msg_expandable_error:nnff { LaTeX / #1 } } 11101
11102 \msg_new:nnnn { msg } { already-defined } 11102
11103 { Message~'#2'~for~module~'#1'~already~defined. } 11103
11104 { 11104
11105 \c_msg_coding_error_text_tl 11105
11106 LaTeX~was~asked~to~define~a~new~message~called~'#2'\\ 11106
11107 by~the~module~'#1':~this~message~already~exists. 11107
11108 \c_msg_return_text_tl 11108
11109 } 11109
11110 \msg_new:nnnn { msg } { unknown } 11110
11111 { Unknown~message~'#2'~for~module~'#1'. } 11111
11112 { 11112
11113 \c_msg_coding_error_text_tl 11113
11114 LaTeX~was~asked~to~display~a~message~called~'#2'\\ 11114
11115 by~the~module~'#1':~this~message~does~not~exist. 11115
11116 \c_msg_return_text_tl 11116
11117 } 11117
11118 \msg_new:nnnn { msg } { class-unknown } 11118
11119 { Unknown~message~class~'#1'. } 11119
11120 { 11120
11121 LaTeX~has~been~asked~to~redirect~messages~to~a~class~'#1':\\ 11121
11122 this~was~never~defined. 11122
11123 \c_msg_return_text_tl 11123
11124 } 11124
11125 \msg_new:nnnn { msg } { redirect-loop } 11125
11126 { 11126
11127 Message~redirection~loop~caused~by~ {#1} ~=>~ {#2} 11127
11128 \tl_if_empty:nF {#3} { ~for~module~' \use_none:n #3 ' } . 11128
11129 } 11129
11130 { 11130
11131 Adding~the~message~redirection~ {#1} ~=>~ {#2} 11131
11132 \tl_if_empty:nF {#3} { ~for~the~module~' \use_none:n #3 ' } ~ 11132

```

```

11133         created~an~infinite~loop~\ \ \
11134         \iow_indent:n { #4 \ \ \ \ }
11135     }
11136 \msg_new:nnnn { kernel } { bad-number-of-arguments }
11137 { Function~'#1'~cannot~be~defined~with~#2~arguments. }
11138 {
11139     \c__msg_coding_error_text_tl
11140     LaTeX~has~been~asked~to~define~a~function~'#1'~with~
11141     #2~arguments.~
11142     TeX~allows~between~0~and~9~arguments~for~a~single~function.
11143 }
11144 \msg_new:nnnn { kernel } { command-already-defined }
11145 { Control~sequence~#1~already~defined. }
11146 {
11147     \c__msg_coding_error_text_tl
11148     LaTeX~has~been~asked~to~create~a~new~control~sequence~'#1'~
11149     but~this~name~has~already~been~used~elsewhere. \ \ \
11150     The~current~meaning~is:\ \
11151     \ \ #2
11152 }
11153 \msg_new:nnnn { kernel } { command-not-defined }
11154 { Control~sequence~#1~undefined. }
11155 {
11156     \c__msg_coding_error_text_tl
11157     LaTeX~has~been~asked~to~use~a~control~sequence~'#1':\ \
11158     this~has~not~been~defined~yet.
11159 }
11160 \msg_new:nnnn { kernel } { empty-search-pattern }
11161 { Empty~search~pattern. }
11162 {
11163     \c__msg_coding_error_text_tl
11164     LaTeX~has~been~asked~to~replace~an~empty~pattern~by~'#1':~that~
11165     would~lead~to~an~infinite~loop!
11166 }
11167 \msg_new:nnnn { kernel } { non-base-function }
11168 { Function~'#1'~is~not~a~base~function }
11169 {
11170     \c__msg_coding_error_text_tl
11171     Functions~defined~through~\iow_char:N\cs_new:Nn~must~have~
11172     a~signature~consisting~of~only~normal~arguments~'N'~and~'n'.~
11173     The~signature~'#2'~of~'#1'~contains~other~arguments~'#3'.~
11174     To~define~variants~use~\iow_char:N\cs_generate_variant:Nn~
11175     and~to~define~other~functions~use~\iow_char:N\cs_new:Npn.
11176 }
11177 \msg_new:nnnn { kernel } { missing-colon }
11178 { Function~'#1'~contains~no~':'. }

```

```

11179 { 11179
11180 \c__msg_coding_error_text_tl 11180
11181 Code-level~functions~must~contain~':'~to~separate~the~ 11181
11182 argument~specification~from~the~function~name.~This~is~ 11182
11183 needed~when~defining~conditionals~or~variants,~or~when~building~a~ 11183
11184 parameter~text~from~the~number~of~arguments~of~the~function. 11184
11185 } 11185
11186 \msg_new:nnnn { kernel } { overflow } 11186
11187 { Integers~larger~than~2^{30}-1~cannot~be~stored~in~arrays. } 11187
11188 { 11188
11189 An~attempt~was~made~to~store~#3~ 11189
11190 \tl_if_empty:nF {#2} { at~position~#2~ } in~the~array~'#1'.~ 11190
11191 The~largest~allowed~value~#4~will~be~used~instead. 11191
11192 } 11192
11193 \msg_new:nnnn { kernel } { out-of-bounds } 11193
11194 { Access~to~an~entry~beyond~an~array's~bounds. } 11194
11195 { 11195
11196 An~attempt~was~made~to~access~or~store~data~at~position~#2~of~the~ 11196
11197 array~'#1',~but~this~array~has~entries~at~positions~from~1~to~#3. 11197
11198 } 11198
11199 \msg_new:nnnn { kernel } { protected-predicate } 11199
11200 { Predicate~'#1'~must~be~expandable. } 11200
11201 { 11201
11202 \c__msg_coding_error_text_tl 11202
11203 LaTeX~has~been~asked~to~define~'#1'~as~a~protected~predicate.~ 11203
11204 Only~expandable~tests~can~have~a~predicate~version. 11204
11205 } 11205
11206 \msg_new:nnn { kernel } { randint-backward-range } 11206
11207 { Wrong~order~of~bounds~in~\iow_char:N\\int_rand:nn{#1}{#2}. } 11207
11208 \msg_new:nnnn { kernel } { conditional-base-undefined } 11208
11209 { Undefined~conditional~base~function~'#1'. } 11209
11210 { 11210
11211 \c__msg_coding_error_text_tl 11211
11212 LaTeX~has~been~asked~to~define~a~variant~of~the~conditional~'#1',~ 11212
11213 but~the~latter~is~not~defined. 11213
11214 } 11214
11215 \msg_new:nnnn { kernel } { conditional-form-unknown } 11215
11216 { Conditional~form~'#1'~for~function~'#2'~unknown. } 11216
11217 { 11217
11218 \c__msg_coding_error_text_tl 11218
11219 LaTeX~has~been~asked~to~define~the~conditional~form~'#1'~of~ 11219
11220 the~function~'#2',~but~only~'TF',~'T',~'F',~and~'p'~forms~exist. 11220
11221 } 11221
11222 \msg_new:nnnn { kernel } { variant-too-long } 11222
11223 { Variant~form~'#1'~longer~than~base~signature~of~'#2'. } 11223
11224 { 11224

```



```

11225 \c__msg_coding_error_text_tl 11225
11226 LaTeX~has~been~asked~to~create~a~variant~of~the~function~'#2'~ 11226
11227 with~a~signature~starting~with~'#1',~but~that~is~longer~than~ 11227
11228 the~signature~(part~after~the~colon)~of~'#2'. 11228
11229 } 11229
11230 \msg_new:nnnn { kernel } { invalid-variant } 11230
11231 { Variant~form~'#1'~invalid~for~base~form~'#2'. } 11231
11232 { 11232
11233 \c__msg_coding_error_text_tl 11233
11234 LaTeX~has~been~asked~to~create~a~variant~of~the~function~'#2'~ 11234
11235 with~a~signature~starting~with~'#1',~but~cannot~change~an~argument~ 11235
11236 from~type~'#3'~to~type~'#4'. 11236
11237 } 11237
11238 \msg_new:nnnn { kernel } { invalid-exp-args } 11238
11239 { Invalid~variant~specifier~'#1'~in~'#2'. } 11239
11240 { 11240
11241 \c__msg_coding_error_text_tl 11241
11242 LaTeX~has~been~asked~to~create~an~\iow_char:N\\exp_args:N...~ 11242
11243 function~with~signature~'N#2'~but~'#1'~is~not~a~valid~argument~ 11243
11244 specifier. 11244
11245 } 11245
11246 \msg_new:nnn { kernel } { deprecated-variant } 11246
11247 { 11247
11248 Variant~form~'#1'~deprecated~for~base~form~'#2'.~ 11248
11249 One~should~not~change~an~argument~from~type~'#3'~to~type~'#4' 11249
11250 \str_case:nnF {#3} 11250
11251 { 11251
11252 { n } { :~use~a~'\token_if_eq_charcode:NNTF #4 c v V'~variant? } 11252
11253 { N } { :~base~form~only~accepts~a~single~token~argument. } 11253
11254 {#4} { :~base~form~is~already~a~variant. } 11254
11255 } { . } 11255
11256 } 11256
11257 \msg_new:nnn { char } { active } 11257
11258 { Cannot~generate~active~chars. } 11258
11259 \msg_new:nnn { char } { invalid-catcode } 11259
11260 { Invalid~catcode~for~char~generation. } 11260
11261 \msg_new:nnn { char } { null-space } 11261
11262 { Cannot~generate~null~char~as~a~space. } 11262
11263 \msg_new:nnn { char } { out-of-range } 11263
11264 { Charcode~requested~out~of~engine~range. } 11264
11265 \msg_new:nnn { dim } { zero-unit } 11265
11266 { Zero~unit~in~conversion. } 11266
11267 \msg_new:nnnn { kernel } { quote-in-shell } 11267
11268 { Quotes~in~shell~command~'#1'. } 11268
11269 { Shell~commands~cannot~contain~quotes~("). } 11269
11270 \msg_new:nnnn { keys } { no-property } 11270

```

```

11271 { No~property~given~in~definition~of~key~'#1'. } 11271
11272 { 11272
11273 \c__msg_coding_error_text_tl 11273
11274 Inside~\keys_define:nn each~key~name~ 11274
11275 needs~a~property: \ \ \ 11275
11276 \iow_indent:n { #1 .<property> } \ \ \ 11276
11277 LaTeX~did~not~find~a~'.'~to~indicate~the~start~of~a~property. 11277
11278 } 11278
11279 \msg_new:nnnn { keys } { property-boolean-values-only } 11279
11280 { The~property~'#1'~accepts~boolean~values~only. } 11280
11281 { 11281
11282 \c__msg_coding_error_text_tl 11282
11283 The~property~'#1'~only~accepts~the~values~'true'~and~'false'. 11283
11284 } 11284
11285 \msg_new:nnnn { keys } { property-requires-value } 11285
11286 { The~property~'#1'~requires~a~value. } 11286
11287 { 11287
11288 \c__msg_coding_error_text_tl 11288
11289 LaTeX~was~asked~to~set~property~'#1'~for~key~'#2'. \ \ 11289
11290 No~value~was~given~for~the~property,~and~one~is~required. 11290
11291 } 11291
11292 \msg_new:nnnn { keys } { property-unknown } 11292
11293 { The~key~property~'#1'~is~unknown. } 11293
11294 { 11294
11295 \c__msg_coding_error_text_tl 11295
11296 LaTeX~has~been~asked~to~set~the~property~'#1'~for~key~'#2':~ 11296
11297 this~property~is~not~defined. 11297
11298 } 11298
11299 \msg_new:nnnn { quark } { invalid-function } 11299
11300 { Quark~test~function~'#1'~is~invalid. } 11300
11301 { 11301
11302 \c__msg_coding_error_text_tl 11302
11303 LaTeX~has~been~asked~to~create~quark~test~function~'#1'~ 11303
11304 \tl_if_empty:nTF {#2} 11304
11305 { but~that~name~ } 11305
11306 { with~signature~'#2',~but~that~signature~ } 11306
11307 is~not~valid. 11307
11308 } 11308
11309 \__kernel_msg_new:nnn { quark } { invalid } 11309
11310 { Invalid~quark~variable~'#1'. } 11310
11311 \msg_new:nnnn { scanmark } { already-defined } 11311
11312 { Scan~mark~'#1'~already~defined. } 11312
11313 { 11313
11314 \c__msg_coding_error_text_tl 11314
11315 LaTeX~has~been~asked~to~create~a~new~scan~mark~'#1'~ 11315
11316 but~this~name~has~already~been~used~for~a~scan~mark. 11316

```

```

11317 }
11318 \msg_new:nnnn { seq } { item-too-large }
11319 { Sequence~'#1'~does~not~have~an~item~#3 }
11320 {
11321   An~attempt~was~made~to~push~or~pop~the~item~at~position~#3~
11322   of~'#1',~but~this~
11323   \int_compare:nTF { #3 = 0 }
11324     { position~does~not~exist. }
11325     { sequence~only~has~#2~item \int_compare:nF { #2 = 1 } {s}. }
11326 }
11327 \msg_new:nnnn { seq } { shuffle-too-large }
11328 { The~sequence~#1~is~too~long~to~be~shuffled~by~TeX. }
11329 {
11330   TeX~has~ \int_eval:n { \c_max_register_int + 1 } ~
11331   toks~registers:~this~only~allows~to~shuffle~up~to~
11332   \int_use:N \c_max_register_int \_items.~
11333   The~list~will~not~be~shuffled.
11334 }
11335 \msg_new:nnnn { kernel } { variable-not-defined }
11336 { Variable~#1~undefined. }
11337 {
11338   \c__msg_coding_error_text_tl
11339   LaTeX~has~been~asked~to~show~a~variable~#1,~but~this~has~not~
11340   been~defined~yet.
11341 }
11342 \msg_new:nnnn { kernel } { bad-type }
11343 { Variable~'#1'~is~not~a~valid~#3. }
11344 {
11345   \c__msg_coding_error_text_tl
11346   The~variable~'#1'~with~\tl_if_empty:nTF {#4} {meaning} {value}\\\\
11347   \iow_indent:n {#2}\\\\
11348   should~be~a~#3~variable,~but~
11349   \tl_if_empty:nTF {#4}
11350     { it~is~not \str_if_eq:nnF {#3} { bool } { ~a~short~macro } . }
11351     {
11352       it~does~not~have~the~correct~
11353       \str_if_eq:nnTF {#2} {#4}
11354       { category~codes. }
11355       { internal~structure:\\\\\iow_indent:n {#4} }
11356     }
11357 }
11358 \msg_new:nnnn { prop } { bad-link }
11359 { Variable~'#1'~is~not~a~valid~(linked)~prop. }
11360 {
11361   \c__msg_coding_error_text_tl
11362   The~variable~'#1'~has~an~incorrect~internal~structure.~

```

```

11363     Its~internal~entry~'#2'~points~to~'#3',~whose~name~is~not~of~the~
11364     form~'#4~<key>' .
11365 }
11366 \msg_new:nnnn { clist } { non-clist }
11367 { Variable~'#1'~is~not~a~valid~clist. }
11368 {
11369     \c__msg_coding_error_text_tl
11370     The~variable~'#1'~with~value\\\\
11371     \iow_indent:n {#2}\\\\
11372     should~be~a~clist~variable,~but~it~includes~empty~or~blank~items~
11373     without~braces.
11374 }
11375 \msg_new:nnnn { prop } { misused }
11376 { A~property~list~was~misused. }
11377 {
11378     \c__msg_coding_error_text_tl
11379     A~property~list~variable~was~used~without~an~accessor~function.~
11380     It~
11381     \tl_if_empty:nTF {#1}
11382     { is~empty. }
11383     { contains~the~key-value~pairs \use_none:n #1 . }
11384 }
11385 \msg_new:nnnn { prop } { inner-make }
11386 { '#1'~ cannot~ be~ used~ in~ a~ group. }
11387 {
11388     \c__msg_coding_error_text_tl
11389     The~ command~ '#1'~ was~ applied~ to~ the~ property~ list~
11390     variable~ '#2', but~ the~ storage~ type~ can~ only~ be~ changed~
11391     at~ the~ outermost~ group~ level.
11392 }
11393 \msg_new:nnn { kernel } { bad-exp-end-f }
11394 { Misused~\exp_end_continue_f:w or~:nw }
11395 \msg_new:nnn { kernel } { bad-variable }
11396 { Erroneous~variable~#1 used! }
11397 \msg_new:nnn { seq } { misused }
11398 { A~sequence~was~misused. }
11399 \msg_new:nnn { prg } { negative-replication }
11400 { Negative~argument~for~\iow_char:N\prg_replicate:nn. }
11401 \msg_new:nnn { prop } { prop-keyval }
11402 { Missing~'='~in~'#1'~(in~'..._keyval:Nn') }
11403 \msg_new:nnn { kernel } { unknown-comparison }
11404 { Relation~'#1'~not~among~=,<,>,==,!=,<=,>=. }
11405 \msg_new:nnn { kernel } { zero-step }
11406 { Zero~step~size~for~function~#1. }
11407 \msg_new:nnn { clist } { show }
11408 {

```

```

11409 The~comma~list~\tl_if_empty:nF {#1} { #1 ~ } 11409
11410 \tl_if_empty:nTF {#2} 11410
11411 { is~empty \>~ . } 11411
11412 { contains~the~items~(without~outer~braces): #2 . } 11412
11413 } 11413
11414 \msg_new:nnn { intarray } { show } 11414
11415 { The~integer~array~#1~contains~#2~items: \> #3 . } 11415
11416 \msg_new:nnn { prop } { show } 11416
11417 { 11417
11418 The~ \str_if_eq:nnF {#3} { flat } { #3~ } 11418
11419 property~list~#1~ 11419
11420 \tl_if_empty:nTF {#2} 11420
11421 { is~empty \>~ . } 11421
11422 { contains~the~pairs~(without~outer~braces): #2 . } 11422
11423 } 11423
11424 \msg_new:nnn { seq } { show } 11424
11425 { 11425
11426 The~sequence~#1~ 11426
11427 \tl_if_empty:nTF {#2} 11427
11428 { is~empty \>~ . } 11428
11429 { contains~the~items~(without~outer~braces): #2 . } 11429
11430 } 11430
11431 \msg_new:nnn { kernel } { show-streams } 11431
11432 { 11432
11433 \tl_if_empty:nTF {#2} { No~ } { The~following~ } 11433
11434 \str_case:nn {#1} 11434
11435 { 11435
11436 { ior } { input ~ } 11436
11437 { iow } { output ~ } 11437
11438 } 11438
11439 streams~are~ 11439
11440 \tl_if_empty:nTF {#2} { open } { in~use: #2 . } 11440
11441 } 11441
11442 \msg_new:nnnn { sys } { backend-set } 11442
11443 { Backend~configuration~already~set. } 11443
11444 { 11444
11445 Run~time~backend~selection~may~only~be~carried~out~once~during~a~run.~ 11445
11446 This~second~attempt~to~set~them~will~be~ignored. 11446
11447 } 11447
11448 \msg_new:nnnn { sys } { load-debug-in-preamble } 11448
11449 { Load~debug~support~in~the~preamble. } 11449
11450 { 11450
11451 Debugging~requires~support~loaded~in~the~preamble: \> 11451
11452 Use~\sys_load_debug:~before~\begin{document}. 11452
11453 } 11453
11454 \msg_new:nnnn { sys } { wrong-backend } 11454

```

```
11455 { Backend~request~inconsistent~with~engine::~using~'#2'~backend. } 11455
11456 { 11456
11457 You~have~requested~backend~'#1',~but~this~is~not~suitable~for~use~with~the~ 11457
11458 active~engine.~LaTeX~will~use~the~'#2'~backend~instead. 11458
11459 } 11459
11460 \cs_set_protected:Npn \__msg_tmp:w #1 11460
11461 { 11461
11462 \cs_new:Npn #1 ? { } 11462
11463 \cs_new:Npn \__msg_expandable_error:nn ##1##2 11463
11464 { 11464
11465 \exp_after:wN \exp_after:wN 11465
11466 \exp_after:wN \__msg_use_none_delimit_by_s_stop:w 11466
11467 \use:n { #1 ~ ! ~ ##2 : ~ ##1 } \s__msg_stop 11467
11468 } 11468
11469 } 11469
11470 \exp_args:Nc \__msg_tmp:w { ??? } 11470
11471 \exp_args_generate:n { oooo } 11471
11472 \cs_new:Npn \msg_expandable_error:nnnnnn #1#2#3#4#5#6 11472
11473 { 11473
11474 \exp_args:Nee \__msg_expandable_error:nn 11474
11475 { 11475
11476 \exp_args:Nc \exp_args:Noooo 11476
11477 { \c__msg_text_prefix_tl #1 / #2 } 11477
11478 { \tl_to_str:n {#3} } 11478
11479 { \tl_to_str:n {#4} } 11479
11480 { \tl_to_str:n {#5} } 11480
11481 { \tl_to_str:n {#6} } 11481
11482 } 11482
11483 { \msg_error_text:n {#1} } 11483
11484 } 11484
11485 \cs_new:Npn \msg_expandable_error:nnnnn #1#2#3#4#5 11485
11486 { \msg_expandable_error:nnnnnn {#1} {#2} {#3} {#4} {#5} { } } 11486
11487 \cs_new:Npn \msg_expandable_error:nnnn #1#2#3#4 11487
11488 { \msg_expandable_error:nnnnnn {#1} {#2} {#3} {#4} { } { } } 11488
11489 \cs_new:Npn \msg_expandable_error:nnn #1#2#3 11489
11490 { \msg_expandable_error:nnnnnn {#1} {#2} {#3} { } { } { } } 11490
11491 \cs_new:Npn \msg_expandable_error:nn #1#2 11491
11492 { \msg_expandable_error:nnnnnn {#1} {#2} { } { } { } { } } 11492
11493 \cs_generate_variant:Nn \msg_expandable_error:nnnnnn { nnffff } 11493
11494 \cs_generate_variant:Nn \msg_expandable_error:nnnnn { nnfff } 11494
11495 \cs_generate_variant:Nn \msg_expandable_error:nnnn { nnff } 11495
11496 \cs_generate_variant:Nn \msg_expandable_error:nnn { nnf } 11496
11497 \prop_gput:Nnn \g_msg_module_name_prop { kernel } { LaTeX } 11497
11498 \prop_gput:Nnn \g_msg_module_type_prop { kernel } { } 11498
11499 \clist_map_inline:nn 11499
11500 { 11500
```

```

11501 char , clist , coffin , debug , deprecation , dim, msg , 11501
11502 quark , prg , prop , scanmark , seq , sys 11502
11503 } 11503
11504 { 11504
11505 \prop_gput:Nnn \g_msg_module_name_prop {#1} { LaTeX } 11505
11506 \prop_gput:Nnn \g_msg_module_type_prop {#1} { } 11506
11507 } 11507
11508 \prop_gput:Nnn \g_msg_module_name_prop { LaTeX / cmd } { LaTeX } 11508
11509 \prop_gput:Nnn \g_msg_module_type_prop { LaTeX / cmd } { } 11509
11510 \prop_gput:Nnn \g_msg_module_name_prop { LaTeX / ltcmd } { LaTeX } 11510
11511 \prop_gput:Nnn \g_msg_module_type_prop { LaTeX / ltcmd } { } 11511
11512 %% File: l3file.dtx 11512
11513 \tl_new:N \l__ior_internal_tl 11513
11514 \int_const:Nn \c__ior_term_ior { 16 } 11514
11515 \seq_new:N \g__ior_streams_seq 11515
11516 \tl_new:N \l__ior_stream_tl 11516
11517 \prop_new:N \g__ior_streams_prop 11517
11518 \int_step_inline:nnn 11518
11519 { 0 } 11519
11520 { 11520
11521 \cs_if_exist:NTF \contextversion 11521
11522 { \tex_count:D 38 ~ } 11522
11523 { 11523
11524 \tex_count:D 16 ~ % 11524
11525 \cs_if_exist:NT \loccount { - 1 } 11525
11526 } 11526
11527 } 11527
11528 { 11528
11529 \prop_gput:Nnn \g__ior_streams_prop {#1} { Reserved~by~format } 11529
11530 } 11530
11531 \cs_new_protected:Npn \ior_new:N #1 { \cs_new_eq:NN #1 \c__ior_term_ior } 11531
11532 \cs_generate_variant:Nn \ior_new:N { c } 11532
11533 \ior_new:N \g_tmpa_ior 11533
11534 \ior_new:N \g_tmpb_ior 11534
11535 \cs_new_protected:Npn \ior_open:Nn #1#2 11535
11536 { \ior_open:NnF #1 {#2} { \__kernel_file_missing:n {#2} } } 11536
11537 \cs_generate_variant:Nn \ior_open:Nn { c } 11537
11538 \tl_new:N \l__ior_file_name_tl 11538
11539 \prg_new_protected_conditional:Npnn \ior_open:Nn #1#2 { T , F , TF } 11539
11540 { 11540
11541 \file_get_full_name:nNTF {#2} \l__ior_file_name_tl 11541
11542 { 11542
11543 \__kernel_ior_open:No #1 \l__ior_file_name_tl 11543
11544 \prg_return_true: 11544
11545 } 11545
11546 { \prg_return_false: } 11546

```



```

11547 } 11547
11548 \prg_generate_conditional_variant:Nnn \ior_open:Nn { c } { T , F , TF } 11548
11549 \exp_args:NNf \cs_new_protected:Npn \__ior_new:N 11549
11550 { \exp_args:NNc \exp_after:wN \exp_stop_f: { newread } } 11550
11551 \cs_if_exist:NT \contextversion 11551
11552 { 11552
11553 \cs_new_eq:NN \__ior_new_aux:N \__ior_new:N 11553
11554 \cs_gset_protected:Npn \__ior_new:N #1 11554
11555 { 11555
11556 \cs_undefine:N #1 11556
11557 \__ior_new_aux:N #1 11557
11558 } 11558
11559 } 11559
11560 \cs_new_protected:Npn \__kernel_ior_open:Nn #1#2 11560
11561 { 11561
11562 \ior_close:N #1 11562
11563 \seq_gpop:NNTF \g__ior_streams_seq \l__ior_stream_tl 11563
11564 { \__ior_open_stream:Nn #1 {#2} } 11564
11565 { 11565
11566 \__ior_new:N #1 11566
11567 \__kernel_tl_set:Nx \l__ior_stream_tl { \int_eval:n {#1} } 11567
11568 \__ior_open_stream:Nn #1 {#2} 11568
11569 } 11569
11570 } 11570
11571 \cs_generate_variant:Nn \__kernel_ior_open:Nn { No } 11571
11572 \cs_new_protected:Npe \__ior_open_stream:Nn #1#2 11572
11573 { 11573
11574 \tex_global:D \tex_chardef:D #1 = \exp_not:N \l__ior_stream_tl \scan_stop: 11574
11575 \prop_gput:Nvn \exp_not:N \g__ior_streams_prop #1 {#2} 11575
11576 \tex_openin:D #1 11576
11577 \sys_if_engine luatex:TF 11577
11578 { {#2} } 11578
11579 { \exp_not:N \__kernel_file_name_quote:n {#2} \scan_stop: } 11579
11580 } 11580
11581 \cs_new_protected:Npn \ior_shell_open:Nn #1#2 11581
11582 { 11582
11583 \sys_if_shell:TF 11583
11584 { \__ior_shell_open:oN { \tl_to_str:n {#2} } #1 } 11584
11585 { \msg_error:nn { kernel } { pipe-failed } } 11585
11586 } 11586
11587 \cs_new_protected:Npn \__ior_shell_open:nN #1#2 11587
11588 { 11588
11589 \tl_if_in:nnTF {#1} { " } 11589
11590 { 11590
11591 \msg_error:nne 11591
11592 { kernel } { quote-in-shell } {#1} 11592

```

```

11593 } 11593
11594 { \_kernel_ior_open:Nn #2 { |#1 } } 11594
11595 } 11595
11596 \cs_generate_variant:Nn \_ior_shell_open:nN { o } 11596
11597 \msg_new:nnnn { kernel } { pipe-failed } 11597
11598 { Cannot~run~piped~system~commands. } 11598
11599 { 11599
11600 LaTeX~tried~to~call~a~system~process~but~this~was~not~possible.\ 11600
11601 Try~the~"--shell-escape"~(or~"--enable-pipes")~option. 11601
11602 } 11602
11603 \cs_new_protected:Npn \ior_close:N #1 11603
11604 { 11604
11605 \int_compare:nT { -1 < #1 < \c__ior_term_ior } 11605
11606 { 11606
11607 \tex_closein:D #1 11607
11608 \prop_gremove:NV \g__ior_streams_prop #1 11608
11609 \seq_if_in:NVF \g__ior_streams_seq #1 11609
11610 { \seq_gpush:NV \g__ior_streams_seq #1 } 11610
11611 \cs_gset_eq:NN #1 \c__ior_term_ior 11611
11612 } 11612
11613 } 11613
11614 \cs_generate_variant:Nn \ior_close:N { c } 11614
11615 \cs_new_protected:Npn \ior_show:N { \_ior_show:NN \tl_show:n } 11615
11616 \cs_generate_variant:Nn \ior_show:N { c } 11616
11617 \cs_new_protected:Npn \ior_log:N { \_ior_show:NN \tl_log:n } 11617
11618 \cs_generate_variant:Nn \ior_log:N { c } 11618
11619 \cs_new_protected:Npn \_ior_show:NN #1#2 11619
11620 { 11620
11621 \_kernel_chk_defined:NT #2 11621
11622 { 11622
11623 \prop_get:NVNTF \g__ior_streams_prop #2 \l__ior_internal_tl 11623
11624 { 11624
11625 \exp_args:Ne #1 11625
11626 { \token_to_str:N #2 ~ open: ~ \l__ior_internal_tl } 11626
11627 } 11627
11628 { \exp_args:Ne #1 { \token_to_str:N #2 ~ closed } } 11628
11629 } 11629
11630 } 11630
11631 \cs_new_protected:Npn \ior_show_list: { \_ior_list:N \msg_show:nneeee } 11631
11632 \cs_new_protected:Npn \ior_log_list: { \_ior_list:N \msg_log:nneeee } 11632
11633 \cs_new_protected:Npn \_ior_list:N #1 11633
11634 { 11634
11635 #1 { kernel } { show-streams } 11635
11636 { ior } 11636
11637 { 11637
11638 \prop_map_function:NN \g__ior_streams_prop 11638

```

```
11639         \msg_show_item_unbraced:nn
11640     }
11641     { } { }
11642 }
11643 \cs_new_eq:NN \if_eof:w \tex_ifeof:D
11644 \prg_new_conditional:Npnn \ior_if_eof:N #1 { p , T , F , TF }
11645 {
11646     \if_int_compare:w -1 < #1
11647         \if_int_compare:w #1 < \c__ior_term_ior
11648             \if_eof:w #1
11649                 \prg_return_true:
11650             \else:
11651                 \prg_return_false:
11652             \fi:
11653         \else:
11654             \prg_return_true:
11655         \fi:
11656     \else:
11657         \prg_return_true:
11658     \fi:
11659 }
11660 \cs_new_protected:Npn \ior_get:NN #1#2
11661 { \ior_get:NMF #1 #2 { \tl_set:Nn #2 { \q_no_value } } }
11662 \cs_new_protected:Npn \__ior_get:NN #1#2
11663 { \tex_read:D #1 to #2 }
11664 \prg_new_protected_conditional:Npnn \ior_get:NN #1#2 { T , F , TF }
11665 {
11666     \ior_if_eof:NTF #1
11667     { \prg_return_false: }
11668     {
11669         \__ior_get:NN #1 #2
11670         \prg_return_true:
11671     }
11672 }
11673 \cs_new_protected:Npn \ior_str_get:NN #1#2
11674 { \ior_str_get:NMF #1 #2 { \tl_set:Nn #2 { \q_no_value } } }
11675 \cs_new_protected:Npn \__ior_str_get:NN #1#2
11676 {
11677     \exp_args:Nno \use:n
11678     {
11679         \int_set:Nn \tex_endlinechar:D { -1 }
11680         \tex_readline:D #1 to #2
11681         \int_set:Nn \tex_endlinechar:D
11682     } { \int_use:N \tex_endlinechar:D }
11683 }
11684 \prg_new_protected_conditional:Npnn \ior_str_get:NN #1#2 { T , F , TF }
```

```

11685 {
11686     \ior_if_eof:NTF #1
11687     { \prg_return_false: }
11688     {
11689         \__ior_str_get:NN #1 #2
11690         \prg_return_true:
11691     }
11692 }
11693 \int_const:Nn \c__ior_term_noprompt_ior { -1 }
11694 \cs_new_protected:Npn \ior_get_term:nN #1#2
11695 { \__ior_get_term:NnN \__ior_get:NN {#1} #2 }
11696 \cs_new_protected:Npn \ior_str_get_term:nN #1#2
11697 { \__ior_get_term:NnN \__ior_str_get:NN {#1} #2 }
11698 \cs_new_protected:Npn \__ior_get_term:NnN #1#2#3
11699 {
11700     \group_begin:
11701     \tex_escapechar:D = -1 \scan_stop:
11702     \tl_if_blank:nTF {#2}
11703     { \exp_args:NNc #1 \c__ior_term_noprompt_ior }
11704     { \exp_args:NNc #1 \c__ior_term_ior }
11705     {#2}
11706     \exp_args:NNNv \group_end:
11707     \tl_set:Nn #3 {#2}
11708 }
11709 \cs_new:Npn \ior_map_break:
11710 { \prg_map_break:Nn \ior_map_break: { } }
11711 \cs_new:Npn \ior_map_break:n
11712 { \prg_map_break:Nn \ior_map_break: }
11713 \cs_new_protected:Npn \ior_map_inline:Nn
11714 { \__ior_map_inline:NNn \__ior_get:NN }
11715 \cs_new_protected:Npn \ior_str_map_inline:Nn
11716 { \__ior_map_inline:NNn \__ior_str_get:NN }
11717 \cs_new_protected:Npn \__ior_map_inline:NNn
11718 {
11719     \int_gincr:N \g__kernel_pr_g_map_int
11720     \exp_args:Nc \__ior_map_inline:NNNn
11721     { __ior_map_ \int_use:N \g__kernel_pr_g_map_int :n }
11722 }
11723 \cs_new_protected:Npn \__ior_map_inline:NNNn #1#2#3#4
11724 {
11725     \cs_gset_protected:Npn #1 ##1 {#4}
11726     \ior_if_eof:NF #3 { \__ior_map_inline_loop:NNN #1#2#3 }
11727     \prg_break_point:Nn \ior_map_break:
11728     { \int_gdecr:N \g__kernel_pr_g_map_int }
11729 }
11730 \cs_new_protected:Npn \__ior_map_inline_loop:NNN #1#2#3

```

```

11731 {
11732     #2 #3 \l__ior_internal_tl
11733     \if_eof:w #3
11734         \exp_after:wN \ior_map_break:
11735         \fi:
11736         \exp_args:No #1 \l__ior_internal_tl
11737         \__ior_map_inline_loop:NNN #1#2#3
11738     }
11739     \cs_new_protected:Npn \ior_map_variable:NNn
11740     { \__ior_map_variable:NNNn \ior_get:NN }
11741     \cs_new_protected:Npn \ior_str_map_variable:NNn
11742     { \__ior_map_variable:NNNn \ior_str_get:NN }
11743     \cs_new_protected:Npn \__ior_map_variable:NNNn #1#2#3#4
11744     {
11745         \ior_if_eof:NF #2 { \__ior_map_variable_loop:NNNn #1#2#3 {#4} }
11746         \prg_break_point:Nn \ior_map_break: { }
11747     }
11748     \cs_new_protected:Npn \__ior_map_variable_loop:NNNn #1#2#3#4
11749     {
11750         #1 #2 #3
11751         \if_eof:w #2
11752             \exp_after:wN \ior_map_break:
11753             \fi:
11754             #4
11755             \__ior_map_variable_loop:NNNn #1#2#3 {#4}
11756     }
11757     \tl_new:N \l__iow_internal_tl
11758     \int_const:Nn \c_log_iow { -1 }
11759     \int_const:Nn \c_term_iow
11760     {
11761         \bool_lazy_and:nnTF
11762             { \sys_if_engine luatex_p: }
11763             { \int_compare_p:nNn \tex luatexversion:D > { 80 } }
11764             { 128 }
11765             { 16 }
11766     }
11767     \seq_new:N \g__iow_streams_seq
11768     \tl_new:N \l__iow_stream_tl
11769     \prop_new:N \g__iow_streams_prop
11770     \int_step_inline:nnn
11771     { 0 }
11772     {
11773         \cs_if_exist:NTF \contextversion
11774             { \tex_count:D 39 ~ }
11775             {
11776                 \tex_count:D 17 ~

```

```
11777 \cs_if_exist:NT \loccount { - 1 } 11777
11778 } 11778
11779 } 11779
11780 { 11780
11781 \prop_gput:Nnn \g__iow_streams_prop {#1} { Reserved~by~format } 11781
11782 } 11782
11783 \scan_new:N \s__iow_mark 11783
11784 \scan_new:N \s__iow_stop 11784
11785 \cs_new:Npn \__iow_use_i_delimit_by_s_stop:nw #1 #2 \s__iow_stop {#1} 11785
11786 \quark_new:N \q__iow_nil 11786
11787 \cs_new_protected:Npn \iow_new:N #1 { \cs_new_eq:NN #1 \c_term_iow } 11787
11788 \cs_generate_variant:Nn \iow_new:N { c } 11788
11789 \iow_new:N \g_tmpa_iow 11789
11790 \iow_new:N \g_tmpb_iow 11790
11791 \exp_args:NNf \cs_new_protected:Npn \__iow_new:N 11791
11792 { \exp_args:NNc \exp_after:wN \exp_stop_f: { newwrite } } 11792
11793 \cs_if_exist:NT \contextversion 11793
11794 { 11794
11795 \cs_new_eq:NN \__iow_new_aux:N \__iow_new:N 11795
11796 \cs_gset_protected:Npn \__iow_new:N #1 11796
11797 { 11797
11798 \cs_undefine:N #1 11798
11799 \__iow_new_aux:N #1 11799
11800 } 11800
11801 } 11801
11802 \tl_new:N \l__iow_file_name_tl 11802
11803 \cs_new_protected:Npn \iow_open:Nn #1#2 11803
11804 { 11804
11805 \__kernel_tl_set:Nx \l__iow_file_name_tl 11805
11806 { \__kernel_file_name_sanitiz:n {#2} } 11806
11807 \__kernel_iow_open:No #1 \l__iow_file_name_tl 11807
11808 } 11808
11809 \cs_generate_variant:Nn \iow_open:Nn { NV , c , cV } 11809
11810 \cs_new_protected:Npn \__kernel_iow_open:Nn #1#2 11810
11811 { 11811
11812 \iow_close:N #1 11812
11813 \seq_gpop:NNTF \g__iow_streams_seq \l__iow_stream_tl 11813
11814 { \__iow_open_stream:Nn #1 {#2} } 11814
11815 { 11815
11816 \__iow_new:N #1 11816
11817 \__kernel_tl_set:Nx \l__iow_stream_tl { \int_eval:n {#1} } 11817
11818 \__iow_open_stream:Nn #1 {#2} 11818
11819 } 11819
11820 } 11820
11821 \cs_generate_variant:Nn \__kernel_iow_open:Nn { No } 11821
11822 \cs_new_protected:Npn \__iow_open_stream:Nn #1#2 11822
```

```

11823 {
11824     \tex_global:D \tex_chardef:D #1 = \l__iow_stream_tl \scan_stop:
11825     \prop_gput:NVn \g__iow_streams_prop #1 {#2}
11826     \tex_immediate:D \tex_openout:D
11827         #1 \__kernel_file_name_quote:n {#2} \scan_stop:
11828 }
11829 \cs_generate_variant:Nn \__iow_open_stream:Nn { NV }
11830 \cs_new_protected:Npn \iow_shell_open:Nn #1#2
11831 {
11832     \sys_if_shell:TF
11833         { \__iow_shell_open:oN { \tl_to_str:n {#2} } #1 }
11834         { \msg_error:nn { kernel } { pipe-failed } }
11835     }
11836 \cs_new_protected:Npn \__iow_shell_open:nN #1#2
11837 {
11838     \tl_if_in:nnTF {#1} { " }
11839     {
11840         \msg_error:nne
11841             { kernel } { quote-in-shell } {#1}
11842     }
11843     { \__kernel_iow_open:Nn #2 { |#1 } }
11844 }
11845 \cs_generate_variant:Nn \__iow_shell_open:nN { o }
11846 \cs_new_protected:Npn \iow_close:N #1
11847 {
11848     \int_compare:nT { \c_log_iow < #1 < \c_term_iow }
11849     {
11850         \tex_immediate:D \tex_closeout:D #1
11851         \prop_gremove:NV \g__iow_streams_prop #1
11852         \seq_if_in:NVF \g__iow_streams_seq #1
11853             { \seq_gpush:NV \g__iow_streams_seq #1 }
11854         \cs_gset_eq:NN #1 \c_term_iow
11855     }
11856 }
11857 \cs_generate_variant:Nn \iow_close:N { c }
11858 \cs_new_protected:Npn \iow_show:N { \__iow_show:NN \tl_show:n }
11859 \cs_generate_variant:Nn \iow_show:N { c }
11860 \cs_new_protected:Npn \iow_log:N { \__iow_show:NN \tl_log:n }
11861 \cs_generate_variant:Nn \iow_log:N { c }
11862 \cs_new_protected:Npn \__iow_show:NN #1#2
11863 {
11864     \__kernel_chk_defined:NT #2
11865     {
11866         \prop_get:NVNTF \g__iow_streams_prop #2 \l__iow_internal_tl
11867         {
11868             \exp_args:Ne #1

```



```

11869         { \token_to_str:N #2 ~ open: ~ \l_iow_internal_tl }
11870     }
11871     { \exp_args:Ne #1 { \token_to_str:N #2 ~ closed } }
11872 }
11873 }
11874 \cs_new_protected:Npn \iow_show_list: { \__iow_list:N \msg_show:nneeee }
11875 \cs_new_protected:Npn \iow_log_list: { \__iow_list:N \msg_log:nneeee }
11876 \cs_new_protected:Npn \__iow_list:N #1
11877 {
11878     #1 { kernel } { show-streams }
11879     { iow }
11880     {
11881         \prop_map_function:NN \g__iow_streams_prop
11882         \msg_show_item_unbraced:nn
11883     }
11884     { } { }
11885 }
11886 \cs_new_protected:Npn \iow_shipout_e:Nn #1#2
11887 { \tex_write:D #1 {#2} }
11888 \cs_generate_variant:Nn \iow_shipout_e:Nn { Ne , c, ce }
11889 \cs_new_protected:Npn \iow_shipout:Nn #1#2
11890 { \tex_write:D #1 { \exp_not:n {#2} } }
11891 \cs_generate_variant:Nn \iow_shipout:Nn { Ne , c, ce }
11892 \cs_generate_variant:Nn \iow_shipout:Nn { Nx , cx }
11893 \cs_new_protected:Npn \__kernel_iow_with:Nnn #1#2
11894 {
11895     \int_compare:nNnTF {#1} = {#2}
11896     { \use:n }
11897     { \__iow_with:oNnn { \int_use:N #1 } #1 {#2} }
11898 }
11899 \cs_new_protected:Npn \__iow_with:nNnn #1#2#3#4
11900 {
11901     \int_set:Nn #2 {#3}
11902     #4
11903     \int_set:Nn #2 {#1}
11904 }
11905 \cs_generate_variant:Nn \__iow_with:nNnn { o }
11906 \cs_new_protected:Npn \iow_now:Nn #1#2
11907 {
11908     \__kernel_iow_with:Nnn \tex_newlinechar:D { \^^J }
11909     { \tex_immediate:D \tex_write:D #1 { \exp_not:n {#2} } }
11910 }
11911 \cs_generate_variant:Nn \iow_now:Nn { NV , Ne , c , cV , ce }
11912 \cs_generate_variant:Nn \iow_now:Nn { Nx , cx }
11913 \cs_new_protected:Npn \iow_log:n { \iow_now:Nn \c_log_iow }
11914 \cs_set_protected:Npn \iow_log:e { \iow_now:Ne \c_log_iow }

```

```

11915 \cs_generate_variant:Nn \iow_log:n { x } 11915
11916 \cs_new_protected:Npn \iow_term:n { \iow_now:Nn \c_term_iow } 11916
11917 \cs_set_protected:Npn \iow_term:e { \iow_now:Ne \c_term_iow } 11917
11918 \cs_generate_variant:Nn \iow_term:n { x } 11918
11919 \cs_new:Npn \iow_newline: { ^^J } 11919
11920 \cs_new_eq:NN \iow_char:N \cs_to_str:N 11920
11921 \int_new:N \l_iow_line_count_int 11921
11922 \int_set:Nn \l_iow_line_count_int { 78 } 11922
11923 \tl_new:N \l__iow_newline_tl 11923
11924 \int_new:N \l__iow_line_target_int 11924
11925 \tl_new:N \l__iow_one_indent_tl 11925
11926 \int_new:N \l__iow_one_indent_int 11926
11927 \cs_new:Npn \__iow_unindent:w { } 11927
11928 \cs_new_protected:Npn \__iow_set_indent:n #1 11928
11929 { 11929
11930     \__kernel_tl_set:Nx \l__iow_one_indent_tl 11930
11931     { \exp_args:No \__kernel_str_to_other_fast:n { \tl_to_str:n {#1} } } 11931
11932     \int_set:Nn \l__iow_one_indent_int 11932
11933     { \str_count:N \l__iow_one_indent_tl } 11933
11934     \exp_last_unbraced:NNo 11934
11935     \cs_set:Npn \__iow_unindent:w \l__iow_one_indent_tl { } 11935
11936 } 11936
11937 \exp_args:Ne \__iow_set_indent:n { \prg_replicate:nn { 4 } { ~ } } 11937
11938 \tl_new:N \l__iow_indent_tl 11938
11939 \int_new:N \l__iow_indent_int 11939
11940 \tl_new:N \l__iow_line_tl 11940
11941 \tl_new:N \l__iow_line_part_tl 11941
11942 \bool_new:N \l__iow_line_break_bool 11942
11943 \tl_new:N \l__iow_wrap_tl 11943
11944 \group_begin: 11944
11945     \int_set:Nn \tex_escapechar:D { -1 } 11945
11946     \tl_const:Ne \c__iow_wrap_marker_tl 11946
11947     { \tl_to_str:n { \^^I \^^O \^^W \^^_ \^^W \^^R \^^A \^^P } } 11947
11948 \group_end: 11948
11949 \tl_map_inline:nn 11949
11950 { { end } { newline } { allow_break } { indent } { unindent } } 11950
11951 { 11951
11952     \tl_const:ce { c__iow_wrap_ #1 _marker_tl } 11952
11953     { 11953
11954         \c__iow_wrap_marker_tl 11954
11955         #1 11955
11956         \c_catcode_other_space_tl 11956
11957     } 11957
11958 } 11958
11959 \cs_new_protected:Npn \iow_wrap_allow_break: 11959
11960 { 11960

```

```

11961 \msg_error:nnnn { kernel } { iow-indent } 11961
11962 { \iow_wrap:nnnN } { \iow_wrap_allow_break: } 11962
11963 } 11963
11964 \cs_new:Npe \__iow_wrap_allow_break: { \c__iow_wrap_allow_break_marker_tl } 11964
11965 \cs_new:Npn \__iow_wrap_allow_break_error: 11965
11966 { 11966
11967 \msg_expandable_error:nnnn { kernel } { iow-indent } 11967
11968 { \iow_wrap:nnnN } { \iow_wrap_allow_break: } 11968
11969 } 11969
11970 \cs_new_protected:Npn \iow_indent:n #1 11970
11971 { 11971
11972 \msg_error:nnnnn { kernel } { iow-indent } 11972
11973 { \iow_wrap:nnnN } { \iow_indent:n } {#1} 11973
11974 #1 11974
11975 } 11975
11976 \cs_new:Npe \__iow_indent:n #1 11976
11977 { 11977
11978 \c__iow_wrap_indent_marker_tl 11978
11979 #1 11979
11980 \c__iow_wrap_unindent_marker_tl 11980
11981 } 11981
11982 \cs_new:Npn \__iow_indent_error:n #1 11982
11983 { 11983
11984 \msg_expandable_error:nnnnn { kernel } { iow-indent } 11984
11985 { \iow_wrap:nnnN } { \iow_indent:n } {#1} 11985
11986 #1 11986
11987 } 11987
11988 \cs_new_protected:Npn \iow_wrap:nnnN #1#2#3#4 11988
11989 { 11989
11990 \group_begin: 11990
11991 \cs_if_exist_use:N \conditionally@traceoff 11991
11992 \int_set:Nn \tex_escapechar:D { -1 } 11992
11993 \cs_set:Npe \{ { \token_to_str:N \{ } 11993
11994 \cs_set:Npe \# { \token_to_str:N \# } 11994
11995 \cs_set:Npe \} { \token_to_str:N \} } 11995
11996 \cs_set:Npe \% { \token_to_str:N \% } 11996
11997 \cs_set:Npe \~ { \token_to_str:N \~ } 11997
11998 \int_set:Nn \tex_escapechar:D { 92 } 11998
11999 \cs_set_eq:NN \ \ \iow_newline: 11999
12000 \cs_set_eq:NN \_ \c_catcode_other_space_tl 12000
12001 \cs_set_eq:NN \iow_wrap_allow_break: \__iow_wrap_allow_break: 12001
12002 \cs_set_eq:NN \iow_indent:n \__iow_indent:n 12002
12003 #3 12003
12004 \cs_set_eq:NN \protect \token_to_str:N 12004
12005 \__kernel_tl_set:Nx \l__iow_wrap_tl {#1} 12005
12006 \cs_set_eq:NN \iow_wrap_allow_break: \__iow_wrap_allow_break_error: 12006

```

12007	\cs_set_eq:NN \iow_indent:n __iow_indent_error:n	12007
12008	__kernel_tl_set:Nx \l__iow_newline_tl { \iow_newline: #2 }	12008
12009	__kernel_tl_set:Nx \l__iow_newline_tl { \tl_to_str:N \l__iow_newline_tl }	12009
12010	\int_set:Nn \l__iow_line_target_int	12010
12011	{ \l_iow_line_count_int - \str_count:N \l__iow_newline_tl + 1 }	12011
12012	\int_compare:nNnT { \l__iow_line_target_int } < 0	12012
12013	{	12013
12014	\tl_set:Nn \l__iow_newline_tl { \iow_newline: }	12014
12015	\int_set:Nn \l__iow_line_target_int	12015
12016	{ \l_iow_line_count_int + 1 }	12016
12017	}	12017
12018	__iow_wrap_do:	12018
12019	\exp_args:NNf \group_end:	12019
12020	#4 { \tl_to_str:N \l__iow_wrap_tl }	12020
12021	}	12021
12022	\cs_generate_variant:Nn \iow_wrap:nnnN { ne }	12022
12023	\cs_new_protected:Npn __iow_wrap_do:	12023
12024	{	12024
12025	__kernel_tl_set:Nx \l__iow_wrap_tl	12025
12026	{	12026
12027	\exp_args:No __kernel_str_to_other_fast:n \l__iow_wrap_tl	12027
12028	\c__iow_wrap_end_marker_tl	12028
12029	}	12029
12030	__kernel_tl_set:Nx \l__iow_wrap_tl	12030
12031	{	12031
12032	\exp_after:wN __iow_wrap_fix_newline:w \l__iow_wrap_tl	12032
12033	^^J \q__iow_nil ^^J \s__iow_stop	12033
12034	}	12034
12035	\exp_after:wN __iow_wrap_start:w \l__iow_wrap_tl	12035
12036	}	12036
12037	\cs_new:Npn __iow_wrap_fix_newline:w #1 ^^J #2 ^^J	12037
12038	{	12038
12039	#1	12039
12040	\if_meaning:w \q__iow_nil #2	12040
12041	__iow_use_i_delimit_by_s_stop:nw	12041
12042	\fi:	12042
12043	\c__iow_wrap_newline_marker_tl	12043
12044	__iow_wrap_fix_newline:w #2 ^^J	12044
12045	}	12045
12046	\cs_new_protected:Npn __iow_wrap_start:w	12046
12047	{	12047
12048	\bool_set_false:N \l__iow_line_break_bool	12048
12049	\tl_clear:N \l__iow_line_tl	12049
12050	\tl_clear:N \l__iow_line_part_tl	12050
12051	\tl_set:Nn \l__iow_wrap_tl { ~ \use_none:n }	12051
12052	\int_zero:N \l__iow_indent_int	12052

12053	\tl_clear:N \l__iow_indent_tl	12053
12054	__iow_wrap_chunk:nw { \l_iow_line_count_int }	12054
12055	}	12055
12056	\cs_new_eq:NN __iow_sep: __kernel_int_sep:	12056
12057	\cs_set_protected:Npn __iow_tmp:w #1#2	12057
12058	{	12058
12059	\cs_new_protected:Npn __iow_wrap_chunk:nw ##1##2 #2	12059
12060	{	12060
12061	\tl_if_empty:nTF {##2}	12061
12062	{	12062
12063	\tl_clear:N \l__iow_line_part_tl	12063
12064	__iow_wrap_next:nw {##1}	12064
12065	}	12065
12066	{	12066
12067	\tl_if_empty:NTF \l__iow_line_tl	12067
12068	{	12068
12069	__iow_wrap_line:nw	12069
12070	{ \l__iow_indent_tl }	12070
12071	##1 - \l__iow_indent_int __iow_sep:	12071
12072	}	12072
12073	{ __iow_wrap_line:nw { } ##1 __iow_sep: }	12073
12074	##2 #1	12074
12075	__iow_wrap_end_chunk:w 7 6 5 4 3 2 1 0 \s__iow_stop	12075
12076	}	12076
12077	}	12077
12078	\cs_new_protected:Npn __iow_wrap_next:nw ##1##2 #1	12078
12079	{ \use:c { __iow_wrap_##2:n } {##1} }	12079
12080	}	12080
12081	\exp_args:NVV __iow_tmp:w \c_catcode_other_space_tl \c__iow_wrap_marker_tl	12081
12082	\cs_new_protected:Npn __iow_wrap_line:nw #1	12082
12083	{	12083
12084	\tex_edef:D \l__iow_line_part_tl { \if_false: } \fi:	12084
12085	#1	12085
12086	\exp_after:wN __iow_wrap_line_loop:w	12086
12087	\int_value:w \int_eval:w	12087
12088	}	12088
12089	\cs_new:Npn __iow_wrap_line_loop:w #1 __iow_sep: #2#3#4#5#6#7#8#9	12089
12090	{	12090
12091	\if_int_compare:w #1 < 8 \exp_stop_f:	12091
12092	__iow_wrap_line_aux:Nw #1	12092
12093	\fi:	12093
12094	#2 #3 #4 #5 #6 #7 #8 #9	12094
12095	\exp_after:wN __iow_wrap_line_loop:w	12095
12096	\int_value:w \int_eval:w #1 - 8 __iow_sep:	12096
12097	}	12097
12098	\cs_new:Npn __iow_wrap_line_aux:Nw #1#2#3 \exp_after:wN #4 __iow_sep:	12098

```

12099 {
12100 #2
12101 \exp_after:wN \__iow_wrap_line_end:NnnnnnnnnN
12102 \exp_after:wN #1
12103 \exp:w \exp_end_continue_f:w
12104 \exp_after:wN \exp_after:wN
12105 \if_case:w #1 \exp_stop_f:
12106 \prg_do_nothing:
12107 \or: \use_none:n
12108 \or: \use_none:nn
12109 \or: \use_none:nnn
12110 \or: \use_none:nnnn
12111 \or: \use_none:nnnnn
12112 \or: \use_none:nnnnnn
12113 \or: \__iow_wrap_line_seven:nnnnnnnn
12114 \fi:
12115 { } { } { } { } { } { } { } { } #3
12116 }
12117 \cs_new:Npn \__iow_wrap_line_seven:nnnnnnnn #1#2#3#4#5#6#7 { \exp_stop_f: }
12118 \cs_new:Npn \__iow_wrap_line_end:NnnnnnnnnN #1#2#3#4#5#6#7#8#9
12119 {
12120 #2 #3 #4 #5 #6 #7 #8
12121 \use_none:nnnnnn \int_eval:w 8 - \__iow_sep: #9
12122 \token_if_eq_charcode:NNTF \c_space_token #9
12123 { \__iow_wrap_line_end:nw { } }
12124 { \if_false: { \fi: } \__iow_wrap_break:w #9 }
12125 }
12126 \cs_new:Npn \__iow_wrap_line_end:nw #1
12127 {
12128 \if_false: { \fi: }
12129 \__iow_wrap_store_do:n {#1}
12130 \__iow_wrap_next_line:w
12131 }
12132 \cs_new:Npn \__iow_wrap_end_chunk:w
12133 #1 \int_eval:w #2 - #3 \__iow_sep: #4#5 \s__iow_stop
12134 {
12135 \if_false: { \fi: }
12136 \exp_args:Nf \__iow_wrap_next:nw { \int_eval:n { #2 - #4 } }
12137 }
12138 \cs_set_protected:Npn \__iow_tmp:w #1
12139 {
12140 \cs_new:Npn \__iow_wrap_break:w
12141 {
12142 \tex_edef:D \l__iow_line_part_tl
12143 { \if_false: } \fi:
12144 \exp_after:wN \__iow_wrap_break_first:w

```

```

12145         \l__iow_line_part_tl                                12145
12146         #1                                                    12146
12147         { ? \__iow_wrap_break_end:w }                        12147
12148         \s__iow_mark                                          12148
12149     }                                                         12149
12150 \cs_new:Npn \__iow_wrap_break_first:w ##1 #1 ##2            12150
12151 {                                                            12151
12152     \use_none:nn ##2 \__iow_wrap_break_none:w              12152
12153     \__iow_wrap_break_loop:w ##1 #1 ##2                    12153
12154 }                                                            12154
12155 \cs_new:Npn \__iow_wrap_break_none:w ##1##2 #1 ##3 \s__iow_mark ##4 #1 12155
12156 {                                                            12156
12157     \tl_if_empty:NTF \l__iow_line_tl                        12157
12158     { ##2 ##4 \__iow_wrap_line_end:nw { } }                 12158
12159     { \__iow_wrap_line_end:nw { \__iow_wrap_trim:N } ##2 ##4 #1 } 12159
12160 }                                                            12160
12161 \cs_new:Npn \__iow_wrap_break_loop:w ##1 #1 ##2 #1 ##3     12161
12162 {                                                            12162
12163     \use_none:n ##3                                          12163
12164     ##1 #1                                                    12164
12165     \__iow_wrap_break_loop:w ##2 #1 ##3                    12165
12166 }                                                            12166
12167 \cs_new:Npn \__iow_wrap_break_end:w ##1 #1 ##2 ##3 #1 ##4 \s__iow_mark 12167
12168 { ##1 \__iow_wrap_line_end:nw { } ##3 }                    12168
12169 }                                                            12169
12170 \exp_args:NV \__iow_tmp:w \c_catcode_other_space_tl         12170
12171 \cs_new_protected:Npn \__iow_wrap_next_line:w #1#2 \s__iow_stop 12171
12172 {                                                            12172
12173     \tl_clear:N \l__iow_line_tl                              12173
12174     \token_if_eq_meaning:NNTF #1 \__iow_wrap_end_chunk:w    12174
12175     {                                                         12175
12176         \tl_clear:N \l__iow_line_part_tl                    12176
12177         \bool_set_true:N \l__iow_line_break_bool            12177
12178         \__iow_wrap_next:nw { \l__iow_line_target_int }      12178
12179     }                                                         12179
12180     {                                                         12180
12181         \__iow_wrap_line:nw                                  12181
12182         { \l__iow_indent_tl }                                 12182
12183         \l__iow_line_target_int - \l__iow_indent_int \__iow_sep: 12183
12184         #1 #2 \s__iow_stop                                    12184
12185     }                                                         12185
12186 }                                                            12186
12187 \cs_new_protected:Npn \__iow_wrap_allow_break:n #1           12187
12188 {                                                            12188
12189     \__kernel_tl_set:Nx \l__iow_line_tl                     12189
12190     { \l__iow_line_tl \__iow_wrap_trim:N \l__iow_line_part_tl } 12190

```



```
12191 \bool_set_false:N \l__iow_line_break_bool 12191
12192 \tl_if_empty:NTF \l__iow_line_part_tl 12192
12193 { \__iow_wrap_chunk:nw {#1} } 12193
12194 { \exp_args:Nf \__iow_wrap_chunk:nw { \int_eval:n { #1 + 1 } } } 12194
12195 } 12195
12196 \cs_new_protected:Npn \__iow_wrap_indent:n #1 12196
12197 { 12197
12198 \tl_put_right:Ne \l__iow_line_tl { \l__iow_line_part_tl } 12198
12199 \bool_set_false:N \l__iow_line_break_bool 12199
12200 \int_add:Nn \l__iow_indent_int { \l__iow_one_indent_int } 12200
12201 \tl_put_right:No \l__iow_indent_tl { \l__iow_one_indent_tl } 12201
12202 \__iow_wrap_chunk:nw {#1} 12202
12203 } 12203
12204 \cs_new_protected:Npn \__iow_wrap_unindent:n #1 12204
12205 { 12205
12206 \tl_put_right:Ne \l__iow_line_tl { \l__iow_line_part_tl } 12206
12207 \bool_set_false:N \l__iow_line_break_bool 12207
12208 \int_sub:Nn \l__iow_indent_int { \l__iow_one_indent_int } 12208
12209 \__kernel_tl_set:Nx \l__iow_indent_tl 12209
12210 { \exp_after:wN \__iow_unindent:w \l__iow_indent_tl } 12210
12211 \__iow_wrap_chunk:nw {#1} 12211
12212 } 12212
12213 \cs_new_protected:Npn \__iow_wrap_newline:n #1 12213
12214 { 12214
12215 \bool_if:NF \l__iow_line_break_bool 12215
12216 { \__iow_wrap_store_do:n { \__iow_wrap_trim:N } } 12216
12217 \bool_set_false:N \l__iow_line_break_bool 12217
12218 \__iow_wrap_chunk:nw { \l__iow_line_target_int } 12218
12219 } 12219
12220 \cs_new_protected:Npn \__iow_wrap_end:n #1 12220
12221 { 12221
12222 \bool_if:NF \l__iow_line_break_bool 12222
12223 { \__iow_wrap_store_do:n { \__iow_wrap_trim:N } } 12223
12224 \bool_set_false:N \l__iow_line_break_bool 12224
12225 } 12225
12226 \cs_new_protected:Npn \__iow_wrap_store_do:n #1 12226
12227 { 12227
12228 \__kernel_tl_set:Nx \l__iow_line_tl 12228
12229 { \l__iow_line_tl \l__iow_line_part_tl } 12229
12230 \__kernel_tl_set:Nx \l__iow_wrap_tl 12230
12231 { 12231
12232 \l__iow_wrap_tl 12232
12233 \l__iow_newline_tl 12233
12234 #1 \l__iow_line_tl 12234
12235 } 12235
12236 \tl_clear:N \l__iow_line_tl 12236
```

```

12237 } 12237
12238 \cs_set_protected:Npn \__iow_tmp:w #1 12238
12239 { 12239
12240 \cs_new:Npn \__iow_wrap_trim:N ##1 12240
12241 { \exp_after:wN \__iow_wrap_trim:w ##1 \s__iow_mark #1 \s__iow_mark \s__iow_stop } 12241
12242 \cs_new:Npn \__iow_wrap_trim:w ##1 #1 \s__iow_mark 12242
12243 { \__iow_wrap_trim_aux:w ##1 \s__iow_mark } 12243
12244 \cs_new:Npn \__iow_wrap_trim_aux:w ##1 \s__iow_mark ##2 \s__iow_stop {##1} 12244
12245 } 12245
12246 \exp_args:NV \__iow_tmp:w \c_catcode_other_space_tl 12246
12247 \tl_new:N \l__file_internal_tl 12247
12248 \str_new:N \g_file_curr_dir_str 12248
12249 \str_new:N \g_file_curr_ext_str 12249
12250 \str_new:N \g_file_curr_name_str 12250
12251 \seq_new:N \g__file_stack_seq 12251
12252 \group_begin: 12252
12253 \cs_set_protected:Npn \__file_tmp:w #1#2#3 12253
12254 { 12254
12255 \tl_if_blank:nTF {#1} 12255
12256 { 12256
12257 \cs_set:Npn \__file_tmp:w ##1 " ##2 " ##3 \s__file_stop 12257
12258 { { } {##2} { } } 12258
12259 \seq_gput_right:Ne \g__file_stack_seq 12259
12260 { 12260
12261 \exp_after:wN \__file_tmp:w \tex_jobname:D 12261
12262 " \tex_jobname:D " \s__file_stop 12262
12263 } 12263
12264 } 12264
12265 { 12265
12266 \seq_gput_right:Nn \g__file_stack_seq { { } {#1} {#2} } 12266
12267 \__file_tmp:w 12267
12268 } 12268
12269 } 12269
12270 \cs_if_exist:NT \@currnamestack 12270
12271 { 12271
12272 \tl_if_empty:NF \@currnamestack 12272
12273 { \exp_after:wN \__file_tmp:w \@currnamestack } 12273
12274 } 12274
12275 \group_end: 12275
12276 \seq_new:N \g__file_record_seq 12276
12277 \tl_new:N \l__file_base_name_tl 12277
12278 \tl_new:N \l__file_full_name_tl 12278
12279 \str_new:N \l__file_dir_str 12279
12280 \str_new:N \l__file_ext_str 12280
12281 \str_new:N \l__file_name_str 12281
12282 \seq_new:N \l_file_search_path_seq 12282

```

```
12283 \seq_new:N \l__file_tmp_seq 12283
12284 \scan_new:N \s__file_stop 12284
12285 \quark_new:N \q__file_nil 12285
12286 \__kernel_quark_new_conditional:Nn \__file_quark_if_nil:n { TF } 12286
12287 \quark_new:N \q__file_recursion_tail 12287
12288 \quark_new:N \q__file_recursion_stop 12288
12289 \__kernel_quark_new_test:N \__file_if_recursion_tail_stop:N 12289
12290 \__kernel_quark_new_test:N \__file_if_recursion_tail_stop_do:nn 12290
12291 \cs_new:Npn \__kernel_file_name_sanitiz:n #1 12291
12292 { 12292
12293     \exp_args:Ne \__file_name_trim_spaces:n 12293
12294     { 12294
12295         \exp_args:Ne \__file_name_strip_quotes:n 12295
12296         { \__file_name_expand:n {#1} } 12296
12297     } 12297
12298 } 12298
12299 \cs_new:Npn \__file_name_expand:n #1 12299
12300 { 12300
12301     \exp_after:wN \__file_name_expand_cleanup:Nw 12301
12302     \cs:w __file_name = #1 \cs_end: 12302
12303     \__file_name_expand_end: 12303
12304 } 12304
12305 \cs_new:Npn \__file_name_expand_cleanup:Nw #1 #2 \__file_name_expand_end: 12305
12306 { 12306
12307     \tl_if_empty:nF {#2} 12307
12308     { \__file_name_expand_error:Nw #2 \__file_name_expand_end: } 12308
12309     \exp_after:wN \__file_name_expand_cleanup:w \token_to_str:N #1 12309
12310 } 12310
12311 \exp_last_unbraced:NNNN 12311
12312 \cs_new:Npn \__file_name_expand_cleanup:w #1 \tl_to_str:n { __file_name = } { } 12312
12313 \cs_new:Npn \__file_name_expand_end: 12313
12314 { 12314
12315     \msg_expandable_error:nn 12315
12316     { kernel } { filename-missing-endcsname } 12316
12317     \cs_end: \__file_name_expand_end: 12317
12318 } 12318
12319 \cs_new:Npn \__file_name_expand_error:Nw #1 #2 \__file_name_expand_end: 12319
12320 { \__file_name_expand_error_aux:Nw #1 #2 \cs_end: \__file_name_expand_end: } 12320
12321 \cs_new:Npn \__file_name_expand_error_aux:Nw #1 #2 \cs_end: #3 12321
12322     \__file_name_expand_end: 12322
12323 { 12323
12324     \msg_expandable_error:nnff 12324
12325     { kernel } { filename-chars-lost } 12325
12326     { \token_to_str:N #1 } { \exp_stop_f: #2 } 12326
12327 } 12327
12328 \cs_new:Npn \__file_name_strip_quotes:n #1 12328
```

```
12329 { 12329
12330 \__file_name_strip_quotes:nw { 0 } 12330
12331 #1 " \q__file_recursion_tail " \q__file_recursion_stop {#1} 12331
12332 } 12332
12333 \cs_new:Npn \__file_name_strip_quotes:nw #1#2 " 12333
12334 { 12334
12335 \if_meaning:w \q__file_recursion_tail #2 12335
12336 \__file_name_strip_quotes_end:wnwn 12336
12337 \fi: 12337
12338 #2 12338
12339 \__file_name_strip_quotes:nw { #1 + 1 } 12339
12340 } 12340
12341 \cs_new:Npn \__file_name_strip_quotes_end:wnwn \fi: #1 12341
12342 \__file_name_strip_quotes:nw #2 \q__file_recursion_stop #3 12342
12343 { 12343
12344 \fi: 12344
12345 \int_if_odd:nT {#2} 12345
12346 { 12346
12347 \msg_expandable_error:nnn 12347
12348 { kernel } { unbalanced-quote-in-filename } {#3} 12348
12349 } 12349
12350 } 12350
12351 \cs_new:Npn \__file_name_trim_spaces:n #1 12351
12352 { \__file_name_trim_spaces:nw {#1} #1 . \q__file_nil . \s__file_stop } 12352
12353 \cs_new:Npn \__file_name_trim_spaces:nw #1#2 . #3 . #4 \s__file_stop 12353
12354 { 12354
12355 \__file_quark_if_nil:nTF {#3} 12355
12356 { 12356
12357 \tl_trim_spaces_apply:nN { #1 \s__file_stop } 12357
12358 \__file_name_trim_spaces_aux:n 12358
12359 } 12359
12360 { \tl_trim_spaces:n {#1} } 12360
12361 } 12361
12362 \cs_new:Npn \__file_name_trim_spaces_aux:n #1 12362
12363 { \__file_name_trim_spaces_aux:w #1 } 12363
12364 \cs_new:Npn \__file_name_trim_spaces_aux:w #1 \s__file_stop {#1} 12364
12365 \cs_new:Npn \__kernel_file_name_quote:n #1 12365
12366 { \__file_name_quote:nw {#1} #1 ~ \q__file_nil \s__file_stop } 12366
12367 \cs_new:Npn \__file_name_quote:nw #1 #2 ~ #3 \s__file_stop 12367
12368 { 12368
12369 \__file_quark_if_nil:nTF {#3} 12369
12370 { #1 } 12370
12371 { "#1" } 12371
12372 } 12372
12373 \tl_const:Ne \c__file_marker_tl { : \token_to_str:N : } 12373
12374 \cs_new_protected:Npn \file_get:nnN #1#2#3 12374
```

```

12375 {
12376     \file_get:nnNF {#1} {#2} #3
12377     { \tl_set:Nn #3 { \q_no_value } }
12378 }
12379 \cs_generate_variant:Nn \file_get:nnN { V }
12380 \prg_new_protected_conditional:Npnn \file_get:nnN #1#2#3 { T , F , TF }
12381 {
12382     \file_get_full_name:nNTF {#1} \l__file_full_name_tl
12383     {
12384         \exp_args:NV \__file_get_aux:nnN
12385             \l__file_full_name_tl
12386             {#2} #3
12387         \prg_return_true:
12388     }
12389     { \prg_return_false: }
12390 }
12391 \prg_generate_conditional_variant:Nnn \file_get:nnN { V } { T , F , TF }
12392 \cs_new_protected:Npe \__file_get_aux:nnN #1#2#3
12393 {
12394     \exp_not:N \if_false: { \exp_not:N \fi:
12395     \group_begin:
12396         \int_set_eq:NN \tex_tracingnesting:D \c_zero_int
12397         \exp_not:N \exp_args:No \tex_everyeof:D
12398         { \exp_not:N \c__file_marker_tl }
12399         #2 \scan_stop:
12400         \exp_not:N \exp_after:wN \exp_not:N \__file_get_do:Nw
12401         \exp_not:N \exp_after:wN #3
12402         \exp_not:N \exp_after:wN \exp_not:N \prg_do_nothing:
12403         \exp_not:N \tex_input:D
12404         \sys_if_engine luatex:TF
12405         { {#1} }
12406         { \exp_not:N \__kernel_file_name_quote:n {#1} \scan_stop: }
12407     \exp_not:N \if_false: } \exp_not:N \fi:
12408 }
12409 \exp_args:Nno \use:nn
12410 { \cs_new_protected:Npn \__file_get_do:Nw #1#2 }
12411 { \c__file_marker_tl }
12412 {
12413     \group_end:
12414     \tl_set:No #1 {#2}
12415 }
12416 \cs_new_eq:NN \__file_size:n \tex_filesize:D
12417 \cs_new:Npn \file_full_name:n #1
12418 {
12419     \exp_args:Ne \__file_full_name:n
12420     { \__kernel_file_name_sanitize:n {#1} }

```

```
12421 } 12421
12422 \cs_generate_variant:Nn \file_full_name:n { V } 12422
12423 \cs_new:Npn \__file_full_name:n #1 12423
12424 { 12424
12425     \tl_if_blank:nF {#1} 12425
12426     { \exp_args:Nne \__file_full_name_auxii:nn {#1} { \__file_full_name_aux:n {#1} } } 12426
12427 } 12427
12428 \cs_new:Npn \__file_full_name_aux:n #1 12428
12429 { 12429
12430     \if_cs_exist:w __file_seen_ \tl_to_str:n {#1} : \cs_end: 12430
12431     -1 12431
12432     \else: 12432
12433         \exp_args:Ne \__file_full_name_auxi:nn { \__file_size:n {#1} } {#1} 12433
12434         \fi: 12434
12435     } 12435
12436 \cs_new:Npn \__file_full_name_auxi:nn #1#2 12436
12437 { 12437
12438     \if:w \scan_stop: #1 \scan_stop: 12438
12439     \else: 12439
12440         \exp_after:wN \use_none:n 12440
12441         \cs:w __file_seen_ \tl_to_str:n {#2} : \cs_end: 12441
12442         #1 12442
12443         \fi: 12443
12444     } 12444
12445 \cs_new:Npn \__file_full_name_auxii:nn #1 #2 12445
12446 { 12446
12447     \tl_if_blank:nTF {#2} 12447
12448     { 12448
12449         \seq_map_tokens:Nn \l_file_search_path_seq 12449
12450         { \__file_full_name_aux:Nnn \seq_map_break:n {#1} } 12450
12451         \cs_if_exist:NT \input@path 12451
12452         { 12452
12453             \tl_map_tokens:Nn \input@path 12453
12454             { \__file_full_name_aux:Nnn \tl_map_break:n {#1} } 12454
12455         } 12455
12456         \__file_name_end: 12456
12457     } 12457
12458     { \__file_ext_check:nn {#1} {#2} } 12458
12459 } 12459
12460 \cs_new:Npn \__file_full_name_aux:Nnn #1#2#3 12460
12461 { 12461
12462     \exp_args:Ne \__file_full_name_aux:nN 12462
12463     { \__file_full_name_slash:n {#3} #2 } 12463
12464     #1 12464
12465 } 12465
12466 \cs_new:Npn \__file_full_name_slash:n #1 12466
```

```
12467 { 12467
12468 \_file_full_name_slash:nw {#1} #1 \q_nil / \q_nil / \q_nil \q_stop 12468
12469 } 12469
12470 \cs_new:Npn \_file_full_name_slash:nw #1#2 / \q_nil / #3 \q_stop 12470
12471 { 12471
12472 \quark_if_nil:nTF {#3} 12472
12473 { #1 / } 12473
12474 { #2 / } 12474
12475 } 12475
12476 \cs_new:Npn \_file_full_name_aux:nN #1 12476
12477 { \exp_args:Nne \_file_full_name_aux:nnN {#1} { \_file_full_name_aux:n {#1} } } 12477
12478 \cs_new:Npn \_file_full_name_aux:nnN #1 #2 #3 12478
12479 { 12479
12480 \tl_if_blank:nF {#2} 12480
12481 { 12481
12482 #3 12482
12483 { 12483
12484 \_file_ext_check:nn {#1} {#2} 12484
12485 \_file_name_cleanup:w 12485
12486 } 12486
12487 } 12487
12488 } 12488
12489 \cs_new:Npn \_file_name_cleanup:w #1 \_file_name_end: { } 12489
12490 \cs_new:Npn \_file_name_end: { } 12490
12491 \cs_new:Npn \_file_ext_check:nn #1 #2 12491
12492 { \_file_ext_check:nnw {#2} { / } #1 / \q_file_nil / \s_file_stop } 12492
12493 \cs_new:Npn \_file_ext_check:nnw #1 #2 #3 / #4 / #5 \s_file_stop 12493
12494 { 12494
12495 \_file_quark_if_nil:nTF {#4} 12495
12496 { 12496
12497 \exp_args:No \_file_ext_check:nnnw 12497
12498 { \use_none:n #2 } {#1} {#3} #3 . \q_file_nil . \s_file_stop 12498
12499 } 12499
12500 { \_file_ext_check:nnw {#1} { #2 #3 / } #4 / #5 \s_file_stop } 12500
12501 } 12501
12502 \cs_new:Npe \_file_ext_check:nnnw #1#2#3#4 . #5 . #6 \s_file_stop 12502
12503 { 12503
12504 \exp_not:N \_file_quark_if_nil:nTF {#5} 12504
12505 { 12505
12506 \exp_not:N \_file_ext_check:nnn 12506
12507 { #1 #3 \tl_to_str:n { .tex } } { #1 #3 } {#2} 12507
12508 } 12508
12509 { #1 #3 } 12509
12510 } 12510
12511 \cs_new:Npn \_file_ext_check:nnn #1 12511
12512 { \exp_args:Nne \_file_ext_check:nnnn {#1} { \_file_full_name_aux:n {#1} } } 12512
```


12513	\cs_new:Npn __file_ext_check:nnnn #1#2#3#4	12513
12514	{	12514
12515	\tl_if_blank:nTF {#2}	12515
12516	{#3}	12516
12517	{	12517
12518	\bool_lazy_or:nnTF	12518
12519	{ \int_compare_p:nNn {#4} = {#2} }	12519
12520	{ \int_compare_p:nNn {#2} = { -1 } }	12520
12521	{#1}	12521
12522	{#3}	12522
12523	}	12523
12524	}	12524
12525	\cs_new_protected:Npn \file_forget:n #1	12525
12526	{ \cs_undefine:c { __file_seen_ \file_full_name:n {#1} : } }	12526
12527	\cs_new_protected:Npn \file_get_full_name:nN #1#2	12527
12528	{	12528
12529	\file_get_full_name:nNF {#1} #2	12529
12530	{ \tl_set:Nn #2 { \q_no_value } }	12530
12531	}	12531
12532	\cs_generate_variant:Nn \file_get_full_name:nN { V }	12532
12533	\prg_new_protected_conditional:Npnn \file_get_full_name:nN #1#2 { T , F , TF }	12533
12534	{	12534
12535	__kernel_tl_set:Nx #2	12535
12536	{ \file_full_name:n {#1} }	12536
12537	\tl_if_empty:NTF #2	12537
12538	{ \prg_return_false: }	12538
12539	{ \prg_return_true: }	12539
12540	}	12540
12541	\prg_generate_conditional_variant:Nnn \file_get_full_name:nN	12541
12542	{ V } { T , F , TF }	12542
12543	\ior_new:N \g__file_internal_ior	12543
12544	\cs_new:Npn \file_size:n #1	12544
12545	{ __file_details:nn {#1} { size } }	12545
12546	\cs_generate_variant:Nn \file_size:n { V }	12546
12547	\cs_new:Npn \file_timestamp:n #1	12547
12548	{ __file_details:nn {#1} { moddate } }	12548
12549	\cs_generate_variant:Nn \file_timestamp:n { V }	12549
12550	\cs_new:Npn __file_details:nn #1#2	12550
12551	{	12551
12552	\exp_args:Ne __file_details_aux:nn	12552
12553	{ \file_full_name:n {#1} } {#2}	12553
12554	}	12554
12555	\cs_new:Npn __file_details_aux:nn #1#2	12555
12556	{	12556
12557	\tl_if_blank:nF {#1}	12557
12558	{ \use:c { tex_file #2 :D } {#1} }	12558

12559	}	12559
12560	\cs_new:Npn \file_mdfive_hash:n #1	12560
12561	{ \exp_args:Ne __file_mdfive_hash:n { \file_full_name:n {#1} } }	12561
12562	\cs_generate_variant:Nn \file_mdfive_hash:n { V }	12562
12563	\cs_new:Npn __file_mdfive_hash:n #1	12563
12564	{ \tex_mdffivesum:D file {#1} }	12564
12565	\cs_new:Npn \file_hex_dump:nnn #1#2#3	12565
12566	{	12566
12567	\exp_args:Neee __file_hex_dump_auxi:nnn	12567
12568	{ \file_full_name:n {#1} }	12568
12569	{ \int_eval:n {#2} }	12569
12570	{ \int_eval:n {#3} }	12570
12571	}	12571
12572	\cs_generate_variant:Nn \file_hex_dump:nnn { V }	12572
12573	\cs_new:Npn __file_hex_dump_auxi:nnn #1#2#3	12573
12574	{	12574
12575	\bool_lazy_any:nF	12575
12576	{	12576
12577	{ \tl_if_blank_p:n {#1} }	12577
12578	{ \int_compare_p:nNn {#2} = 0 }	12578
12579	{ \int_compare_p:nNn {#3} = 0 }	12579
12580	}	12580
12581	{	12581
12582	\exp_args:Ne __file_hex_dump_auxii:nnnn	12582
12583	{ __file_details_aux:nn {#1} { size } }	12583
12584	{#1} {#2} {#3}	12584
12585	}	12585
12586	}	12586
12587	\cs_new:Npn __file_hex_dump_auxii:nnnn #1#2#3#4	12587
12588	{	12588
12589	\int_compare:nNnTF {#3} > 0	12589
12590	{ __file_hex_dump_auxiii:nnnn {#3} }	12590
12591	{	12591
12592	\exp_args:Ne __file_hex_dump_auxiii:nnnn	12592
12593	{ \int_eval:n { #1 + #3 } }	12593
12594	}	12594
12595	{#1} {#2} {#4}	12595
12596	}	12596
12597	\cs_new:Npn __file_hex_dump_auxiii:nnnn #1#2#3#4	12597
12598	{	12598
12599	\int_compare:nNnTF {#4} > 0	12599
12600	{ __file_hex_dump_auxiv:nnn {#4} }	12600
12601	{	12601
12602	\exp_args:Ne __file_hex_dump_auxiv:nnn	12602
12603	{ \int_eval:n { #2 + #4 } }	12603
12604	}	12604

12605	{#1} {#3}	12605
12606	}	12606
12607	\cs_new:Npn __file_hex_dump_auxiv:nnn #1#2#3	12607
12608	{	12608
12609	\tex_dump:D	12609
12610	offset ~ \int_eval:n { #2 - 1 } ~	12610
12611	length ~ \int_eval:n { #1 - #2 + 1 }	12611
12612	{#3}	12612
12613	}	12613
12614	\cs_new:Npn \file_hex_dump:n #1	12614
12615	{ \exp_args:Ne __file_hex_dump:n { \file_full_name:n {#1} } }	12615
12616	\cs_generate_variant:Nn \file_hex_dump:n { V }	12616
12617	\sys_if_engine luatex:TF	12617
12618	{	12618
12619	\cs_new:Npn __file_hex_dump:n #1	12619
12620	{	12620
12621	\tl_if_blank:nF {#1}	12621
12622	{ \tex_dump:D whole {#1} {#1} }	12622
12623	}	12623
12624	}	12624
12625	{	12625
12626	\cs_new:Npn __file_hex_dump:n #1	12626
12627	{	12627
12628	\tl_if_blank:nF {#1}	12628
12629	{ \tex_dump:D length \tex_filesize:D {#1} {#1} }	12629
12630	}	12630
12631	}	12631
12632	\cs_new_protected:Npn \file_get_hex_dump:nN #1#2	12632
12633	{ \file_get_hex_dump:nNF {#1} #2 { \tl_set:Nn #2 { \q_no_value } } }	12633
12634	\cs_generate_variant:Nn \file_get_hex_dump:nN { V }	12634
12635	\cs_new_protected:Npn \file_get_md5_hash:nN #1#2	12635
12636	{ \file_get_md5_hash:nNF {#1} #2 { \tl_set:Nn #2 { \q_no_value } } }	12636
12637	\cs_generate_variant:Nn \file_get_md5_hash:nN { V }	12637
12638	\cs_new_protected:Npn \file_get_size:nN #1#2	12638
12639	{ \file_get_size:nNF {#1} #2 { \tl_set:Nn #2 { \q_no_value } } }	12639
12640	\cs_generate_variant:Nn \file_get_size:nN { V }	12640
12641	\cs_new_protected:Npn \file_get_timestamp:nN #1#2	12641
12642	{ \file_get_timestamp:nNF {#1} #2 { \tl_set:Nn #2 { \q_no_value } } }	12642
12643	\cs_generate_variant:Nn \file_get_timestamp:nN { V }	12643
12644	\prg_new_protected_conditional:Npnn \file_get_hex_dump:nN #1#2 { T , F , TF }	12644
12645	{ __file_get_details:nnN {#1} { hex_dump } #2 }	12645
12646	\prg_generate_conditional_variant:Nnn \file_get_hex_dump:nN	12646
12647	{ V } { T , F , TF }	12647
12648	\prg_new_protected_conditional:Npnn \file_get_md5_hash:nN #1#2 { T , F , TF }	12648
12649	{ __file_get_details:nnN {#1} { md5_hash } #2 }	12649
12650	\prg_generate_conditional_variant:Nnn \file_get_md5_hash:nN	12650

12651	{ V } { T , F , TF }	12651
12652	\prg_new_protected_conditional:Npnn \file_get_size:nN #1#2 { T , F , TF }	12652
12653	{ __file_get_details:nnN {#1} { size } #2 }	12653
12654	\prg_generate_conditional_variant:Nnn \file_get_size:nN	12654
12655	{ V } { T , F , TF }	12655
12656	\prg_new_protected_conditional:Npnn \file_get_timestamp:nN #1#2 { T , F , TF }	12656
12657	{ __file_get_details:nnN {#1} { timestamp } #2 }	12657
12658	\prg_generate_conditional_variant:Nnn \file_get_timestamp:nN	12658
12659	{ V } { T , F , TF }	12659
12660	\cs_new_protected:Npn __file_get_details:nnN #1#2#3	12660
12661	{	12661
12662	__kernel_tl_set:Nx #3	12662
12663	{ \use:c { file_ #2 :n } {#1} }	12663
12664	\tl_if_empty:NTF #3	12664
12665	{ \prg_return_false: }	12665
12666	{ \prg_return_true: }	12666
12667	}	12667
12668	\cs_new_protected:Npn \file_get_hex_dump:nnnN #1#2#3#4	12668
12669	{	12669
12670	\file_get_hex_dump:nnnNF {#1} {#2} {#3} #4	12670
12671	{ \tl_set:Nn #4 { \q_no_value } }	12671
12672	}	12672
12673	\cs_generate_variant:Nn \file_get_hex_dump:nnnN { V }	12673
12674	\prg_new_protected_conditional:Npnn \file_get_hex_dump:nnnN #1#2#3#4	12674
12675	{ T , F , TF }	12675
12676	{	12676
12677	__kernel_tl_set:Nx #4	12677
12678	{ \file_hex_dump:nnn {#1} {#2} {#3} }	12678
12679	\tl_if_empty:NTF #4	12679
12680	{ \prg_return_false: }	12680
12681	{ \prg_return_true: }	12681
12682	}	12682
12683	\prg_generate_conditional_variant:Nnn \file_get_hex_dump:nnnN	12683
12684	{ V } { T , F , TF }	12684
12685	\cs_new_eq:NN __file_str_cmp:nn \tex_strcmp:D	12685
12686	\prg_new_conditional:Npnn \file_compare_timestamp:nNn #1#2#3	12686
12687	{ p , T , F , TF }	12687
12688	{	12688
12689	\exp_args:Nee __file_compare_timestamp:nnN	12689
12690	{ \file_full_name:n {#1} }	12690
12691	{ \file_full_name:n {#3} }	12691
12692	#2	12692
12693	}	12693
12694	\prg_generate_conditional_variant:Nnn \file_compare_timestamp:nNn	12694
12695	{ nNV , V , VNV } { p , T , F , TF }	12695
12696	\cs_new:Npn __file_compare_timestamp:nnN #1#2#3	12696

```
12697 { 12697
12698 \tl_if_blank:nTF {#1} 12698
12699 { 12699
12700 \if_charcode:w #3 < 12700
12701 \prg_return_true: 12701
12702 \else: 12702
12703 \prg_return_false: 12703
12704 \fi: 12704
12705 } 12705
12706 { 12706
12707 \tl_if_blank:nTF {#2} 12707
12708 { 12708
12709 \if_charcode:w #3 > 12709
12710 \prg_return_true: 12710
12711 \else: 12711
12712 \prg_return_false: 12712
12713 \fi: 12713
12714 } 12714
12715 { 12715
12716 \if_int_compare:w 12716
12717 \__file_str_cmp:nn 12717
12718 { \__file_timestamp:n {#1} } 12718
12719 { \__file_timestamp:n {#2} } 12719
12720 #3 \c_zero_int 12720
12721 \prg_return_true: 12721
12722 \else: 12722
12723 \prg_return_false: 12723
12724 \fi: 12724
12725 } 12725
12726 } 12726
12727 } 12727
12728 \cs_new_eq:NN \__file_timestamp:n \tex_filemoddate:D 12728
12729 \prg_new_conditional:Npnn \file_if_exist:n #1 { p , T , F , TF } 12729
12730 { 12730
12731 \tl_if_blank:eTF { \file_full_name:n {#1} } 12731
12732 { \prg_return_false: } 12732
12733 { \prg_return_true: } 12733
12734 } 12734
12735 \prg_generate_conditional_variant:Nnn \file_if_exist:n { V } { p , T , F , TF } 12735
12736 \cs_new_protected:Npn \file_if_exist_input:n #1 12736
12737 { 12737
12738 \file_get_full_name:nNT {#1} \l__file_full_name_tl 12738
12739 { \__file_input:V \l__file_full_name_tl } 12739
12740 } 12740
12741 \cs_generate_variant:Nn \file_if_exist_input:n { V } 12741
12742 \cs_new_protected:Npn \file_if_exist_input:nF #1#2 12742
```

12743	{	12743
12744	\file_get_full_name:nNTF {#1} \l__file_full_name_tl	12744
12745	{ __file_input:V \l__file_full_name_tl }	12745
12746	{#2}	12746
12747	}	12747
12748	\cs_generate_variant:Nn \file_if_exist_input:nF { V }	12748
12749	\cs_new_protected:Npn \file_input_stop: { \tex_endinput:D }	12749
12750	\cs_new_protected:Npn __kernel_file_missing:n #1	12750
12751	{	12751
12752	\msg_error:nne { kernel } { file-not-found }	12752
12753	{ __kernel_file_name_sanitiz:n {#1} }	12753
12754	}	12754
12755	\cs_new_protected:Npn \file_input:n #1	12755
12756	{	12756
12757	\file_get_full_name:nNTF {#1} \l__file_full_name_tl	12757
12758	{ __file_input:V \l__file_full_name_tl }	12758
12759	{ __kernel_file_missing:n {#1} }	12759
12760	}	12760
12761	\cs_generate_variant:Nn \file_input:n { V }	12761
12762	\cs_new_protected:Npe __file_input:n #1	12762
12763	{	12763
12764	\exp_not:N \clist_if_exist:NTF \exp_not:N \@filelist	12764
12765	{ \exp_not:N \@addtofilelist {#1} }	12765
12766	{ \seq_gput_right:Nn \exp_not:N \g__file_record_seq {#1} }	12766
12767	\exp_not:N __file_input_push:n {#1}	12767
12768	\exp_not:N \tex_input:D	12768
12769	\sys_if_engine luatex:TF	12769
12770	{ {#1} }	12770
12771	{ \exp_not:N __kernel_file_name_quote:n {#1} \scan_stop: }	12771
12772	\exp_not:N __file_input_pop:	12772
12773	}	12773
12774	\cs_generate_variant:Nn __file_input:n { V }	12774
12775	\cs_new_protected:Npn __file_input_push:n #1	12775
12776	{	12776
12777	\seq_gpush:Ne \g__file_stack_seq	12777
12778	{	12778
12779	{ \g_file_curr_dir_str }	12779
12780	{ \g_file_curr_name_str }	12780
12781	{ \g_file_curr_ext_str }	12781
12782	}	12782
12783	\file_parse_full_name:nNNN {#1}	12783
12784	\l__file_dir_str \l__file_name_str \l__file_ext_str	12784
12785	\str_gset_eq:NN \g_file_curr_dir_str \l__file_dir_str	12785
12786	\str_gset_eq:NN \g_file_curr_name_str \l__file_name_str	12786
12787	\str_gset_eq:NN \g_file_curr_ext_str \l__file_ext_str	12787
12788	}	12788

```
12789 \cs_new_eq:NN \__kernel_file_input_push:n \__file_input_push:n 12789
12790 \cs_new_protected:Npn \__file_input_pop: 12790
12791 { 12791
12792     \seq_gpop:NN \g__file_stack_seq \l__file_internal_tl 12792
12793     \exp_after:wN \__file_input_pop:nnn \l__file_internal_tl 12793
12794 } 12794
12795 \cs_new_eq:NN \__kernel_file_input_pop: \__file_input_pop: 12795
12796 \cs_new_protected:Npn \__file_input_pop:nnn #1#2#3 12796
12797 { 12797
12798     \str_gset:Nn \g_file_curr_dir_str {#1} 12798
12799     \str_gset:Nn \g_file_curr_name_str {#2} 12799
12800     \str_gset:Nn \g_file_curr_ext_str {#3} 12800
12801 } 12801
12802 \cs_new:Npn \file_input_raw:n #1 12802
12803 { \exp_args:Ne \__file_input_raw:nn { \file_full_name:n {#1} } {#1} } 12803
12804 \cs_generate_variant:Nn \file_input_raw:n { V } 12804
12805 \cs_new:Npe \__file_input_raw:nn #1#2 12805
12806 { 12806
12807     \exp_not:N \tl_if_blank:nTF {#1} 12807
12808     { 12808
12809         \exp_not:N \exp_args:Nnne \exp_not:N \msg_expandable_error:nnn 12809
12810         { kernel } { file-not-found } 12810
12811         { \exp_not:N \__kernel_file_name_sanitiz:n {#2} } 12811
12812     } 12812
12813     { 12813
12814         \exp_not:N \tex_input:D 12814
12815         \sys_if_engine luatex:TF 12815
12816         { {#1} } 12816
12817         { \exp_not:N \__kernel_file_name_quote:n {#1} \scan_stop: } 12817
12818     } 12818
12819 } 12819
12820 \exp_args_generate:n { nne } 12820
12821 \cs_new:Npn \file_parse_full_name:n #1 12821
12822 { 12822
12823     \file_parse_full_name_apply:nN {#1} 12823
12824     \prg_do_nothing: 12824
12825 } 12825
12826 \cs_generate_variant:Nn \file_parse_full_name:n { V } 12826
12827 \cs_new:Npn \file_parse_full_name_apply:nN #1 12827
12828 { 12828
12829     \exp_args:Ne \__file_parse_full_name_auxi:nN 12829
12830     { \__kernel_file_name_sanitiz:n {#1} } 12830
12831 } 12831
12832 \cs_generate_variant:Nn \file_parse_full_name_apply:nN { V } 12832
12833 \cs_new:Npn \__file_parse_full_name_auxi:nN #1 12833
12834 { 12834
```



```
12835     \__file_parse_full_name_area:nw { } #1
12836     / \s__file_stop
12837 }
12838 \cs_new:Npn \__file_parse_full_name_area:nw #1 #2 / #3 \s__file_stop
12839 {
12840     \tl_if_empty:nTF {#3}
12841     { \__file_parse_full_name_base:nw { } #2 . \s__file_stop {#1} }
12842     { \__file_parse_full_name_area:nw { #1 / #2 } #3 \s__file_stop }
12843 }
12844 \cs_new:Npn \__file_parse_full_name_base:nw #1 #2 . #3 \s__file_stop
12845 {
12846     \tl_if_empty:nTF {#3}
12847     {
12848         \tl_if_empty:nTF {#1}
12849         {
12850             \tl_if_empty:nTF {#2}
12851             { \__file_parse_full_name_tidy:nnnN { } { } }
12852             { \__file_parse_full_name_tidy:nnnN { .#2 } { } }
12853         }
12854         { \__file_parse_full_name_tidy:nnnN {#1} { .#2 } }
12855     }
12856     { \__file_parse_full_name_base:nw { #1 . #2 } #3 \s__file_stop }
12857 }
12858 \cs_new:Npn \__file_parse_full_name_tidy:nnnN #1 #2 #3 #4
12859 {
12860     \exp_args:Nee #4
12861     {
12862         \str_if_eq:nnF {#3} { / } { \use_none:n }
12863         #3 \prg_do_nothing:
12864     }
12865     { \use_none:n #1 \prg_do_nothing: }
12866     {#2}
12867 }
12868 \cs_new_protected:Npn \file_parse_full_name:nNNN #1 #2 #3 #4
12869 {
12870     \file_parse_full_name_apply:nN {#1}
12871     \__file_full_name_assign:nnnNNN #2 #3 #4
12872 }
12873 \cs_new_protected:Npn \__file_full_name_assign:nnnNNN #1 #2 #3 #4 #5 #6
12874 {
12875     \str_set:Nn #4 {#1}
12876     \str_set:Nn #5 {#2}
12877     \str_set:Nn #6 {#3}
12878 }
12879 \cs_generate_variant:Nn \file_parse_full_name:nNNN { V }
12880 \cs_new_protected:Npn \file_show_list: { \__file_list:N \msg_show:nneeee }
```

12881	\cs_new_protected:Npn \file_log_list: { __file_list:N \msg_log:nneeee }	12881
12882	\cs_new_protected:Npn __file_list:N #1	12882
12883	{	12883
12884	\seq_clear:N \l__file_tmp_seq	12884
12885	\clist_if_exist:NT \@filelist	12885
12886	{	12886
12887	\exp_args:NNe \seq_set_from_clist:Nn \l__file_tmp_seq	12887
12888	{ \tl_to_str:N \@filelist }	12888
12889	}	12889
12890	\seq_concat:NNN \l__file_tmp_seq \l__file_tmp_seq \g__file_record_seq	12890
12891	\seq_remove_duplicates:N \l__file_tmp_seq	12891
12892	#1 { kernel } { file-list }	12892
12893	{ \seq_map_function:NN \l__file_tmp_seq __file_list_aux:n }	12893
12894	{ } { } { }	12894
12895	}	12895
12896	\cs_new:Npn __file_list_aux:n #1 { \iow_newline: #1 }	12896
12897	\cs_if_exist:NT \@filelist	12897
12898	{	12898
12899	\AtBeginDocument	12899
12900	{	12900
12901	\exp_args:NNe \seq_set_from_clist:Nn \l__file_tmp_seq	12901
12902	{ \tl_to_str:N \@filelist }	12902
12903	\seq_gconcat:NNN	12903
12904	\g__file_record_seq	12904
12905	\g__file_record_seq	12905
12906	\l__file_tmp_seq	12906
12907	}	12907
12908	}	12908
12909	\cs_new_protected:Npn \GetIdInfo	12909
12910	{	12910
12911	\tl_clear_new:N \ExplFileDescription	12911
12912	\tl_clear_new:N \ExplFileDate	12912
12913	\tl_clear_new:N \ExplFileName	12913
12914	\tl_clear_new:N \ExplFileExtension	12914
12915	\tl_clear_new:N \ExplFileVersion	12915
12916	\group_begin:	12916
12917	\char_set_catcode_space:n { 32 }	12917
12918	\exp_after:wN	12918
12919	\group_end:	12919
12920	__file_id_info_auxi:w	12920
12921	}	12921
12922	\cs_new_protected:Npn __file_id_info_auxi:w \$ #1 \$ #2	12922
12923	{	12923
12924	\tl_set:Nn \ExplFileDescription {#2}	12924
12925	\str_if_eq:nnTF {#1} { Id }	12925
12926	{	12926

12927	\tl_set:Nn \ExplFileDate { 0000/00/00 }	12927
12928	\tl_set:Nn \ExplFileName { [unknown] }	12928
12929	\tl_set:Nn \ExplFileExtension { [unknown~extension] }	12929
12930	\tl_set:Nn \ExplFileVersion {-1}	12930
12931	}	12931
12932	{ __file_id_info_auxii:w #1 ~ \s__file_stop }	12932
12933	}	12933
12934	\cs_new_protected:Npn __file_id_info_auxii:w	12934
12935	#1 ~ #2.#3 ~ #4 ~ #5 ~ #6 \s__file_stop	12935
12936	{	12936
12937	\tl_set:Nn \ExplFileName {#2}	12937
12938	\tl_set:Nn \ExplFileExtension {#3}	12938
12939	\tl_set:Nn \ExplFileVersion {#4}	12939
12940	\str_if_eq:nnTF {#4} {-1}	12940
12941	{ \tl_set:Nn \ExplFileDate { 0000/00/00 } }	12941
12942	{ __file_id_info_auxiii:w #5 - 0 - 0 - \s__file_stop }	12942
12943	}	12943
12944	\cs_new_protected:Npn __file_id_info_auxiii:w #1 - #2 - #3 - #4 \s__file_stop	12944
12945	{ \tl_set:Nn \ExplFileDate { #1/#2/#3 } }	12945
12946	\cs_new_protected:Npn __kernel_dependency_version_check:Nn #1	12946
12947	{ \exp_args:NV __kernel_dependency_version_check:nn #1 }	12947
12948	\cs_new_protected:Npn __kernel_dependency_version_check:nn #1	12948
12949	{	12949
12950	\cs_if_exist:NTF \c__kernel_expl_date_tl	12950
12951	{	12951
12952	\exp_args:NV __file_kernel_dependency_compare:nnn	12952
12953	\c__kernel_expl_date_tl {#1}	12953
12954	}	12954
12955	{ __file_kernel_dependency_compare:nnn { 0000-00-00 } {#1} }	12955
12956	}	12956
12957	\cs_new_protected:Npn __file_kernel_dependency_compare:nnn #1 #2 #3	12957
12958	{	12958
12959	\int_compare:nNnT	12959
12960	{ __file_parse_version:w #1 \s__file_stop } <	12960
12961	{ __file_parse_version:w #2 \s__file_stop }	12961
12962	{ __file_mismatched_dependency_error:nn {#2} {#3} }	12962
12963	}	12963
12964	\cs_new:Npn __file_parse_version:w #1 - #2 - #3 \s__file_stop {#1#2#3}	12964
12965	\cs_new_protected:Npn __file_mismatched_dependency_error:nn #1 #2	12965
12966	{	12966
12967	\exp_args:NNe \ior_shell_open:Nn \g__file_internal_ior	12967
12968	{	12968
12969	kpsewhich ~ --all ~	12969
12970	--engine = \c_sys_engine_exec_str	12970
12971	\c_space_tl \c_sys_engine_format_str	12971
12972	\bool_lazy_and:nnT	12972

```

12973         { \tl_if_exist_p:N \development@branch@name } 12973
12974         { ! \tl_if_empty_p:N \development@branch@name } 12974
12975         { -dev } .fmt 12975
12976     } 12976
12977     \seq_clear:N \l__file_tmp_seq 12977
12978     \ior_map_inline:Nn \g__file_internal_ior 12978
12979         { \seq_put_right:Nn \l__file_tmp_seq {##1} } 12979
12980     \ior_close:N \g__file_internal_ior 12980
12981     \msg_error:nnnn { kernel } { mismatched-support-file } 12981
12982         {#1} {#2} 12982
12983     \tex_endinput:D 12983
12984 } 12984
12985 \msg_new:nnnn { kernel } { mismatched-support-file } 12985
12986 { 12986
12987     Mismatched~LaTeX~support~files~detected. \\\ 12987
12988     Loading~'#2'~aborted! 12988
12989     \tl_if_exist:NT \c__kernel_expl_date_tl 12989
12990     { 12990
12991         \\\ \\\ 12991
12992         The~L3~programming~layer~in~the~LaTeX~format \\\ 12992
12993         is~dated~\c__kernel_expl_date_tl,~but~in~your~TeX~ 12993
12994         tree~the~files~require \\\ at~least~#1. 12994
12995     } 12995
12996 } 12996
12997 { 12997
12998     \int_compare:nNnTF { \seq_count:N \l__file_tmp_seq } > 1 12998
12999     { 12999
13000         The~cause~seems~to~be~an~old~format~file~in~the~user~tree. \\\ 13000
13001         LaTeX~found~these~files: 13001
13002         \seq_map_tokens:Nn \l__file_tmp_seq { \\\~~~\use:n } \\\ 13002
13003         Try~deleting~the~file~in~the~user~tree~then~run~LaTeX~again. 13003
13004     } 13004
13005     { 13005
13006         The~most~likely~causes~are: 13006
13007         \\\~~~A~recent~format~generation~failed; 13007
13008         \\\~~~A~stray~format~file~in~the~user~tree~which~needs~ 13008
13009         to~be~removed~or~rebuilt; 13009
13010         \\\~~~You~are~running~a~manually~installed~version~of~#2 \\\ 13010
13011         \_\_\_ which~is~incompatible~with~the~version~in~LaTeX. \\\ 13011
13012     } 13012
13013     \\\ 13013
13014     LaTeX~will~abort~loading~the~incompatible~support~files~ 13014
13015     but~this~may~lead~to \\\ later~errors.~Please~ensure~that~ 13015
13016     your~LaTeX~format~is~correctly~regenerated. 13016
13017 } 13017
13018 \msg_new:nnnn { kernel } { file-not-found } 13018

```

```

13019 { File~'#1'~not~found. } 13019
13020 { 13020
13021 The~requested~file~could~not~be~found~in~the~current~directory,~ 13021
13022 in~the~TeX~search~path~or~in~the~LaTeX~search~path. 13022
13023 } 13023
13024 \msg_new:nnn { kernel } { file-list } 13024
13025 { 13025
13026 >~File~List~< 13026
13027 #1 \\\ 13027
13028 ..... 13028
13029 } 13029
13030 \msg_new:nnnn { kernel } { filename-chars-lost } 13030
13031 { #1~invalid~in~file~name.~Lost:~#2. } 13031
13032 { 13032
13033 There~was~an~invalid~token~in~the~file~name~that~caused~ 13033
13034 the~characters~following~it~to~be~lost. 13034
13035 } 13035
13036 \msg_new:nnnn { kernel } { filename-missing-endcsname } 13036
13037 { Missing~\iow_char:N\\endcsname~inserted~in~filename. } 13037
13038 { 13038
13039 The~file~name~had~more~\iow_char:N\\csname~commands~than~ 13039
13040 \iow_char:N\\endcsname~ones.~LaTeX~will~add~the~missing~ 13040
13041 \iow_char:N\\endcsname~and~try~to~continue~as~best~as~it~can. 13041
13042 } 13042
13043 \msg_new:nnnn { kernel } { unbalanced-quote-in-filename } 13043
13044 { Unbalanced~quotes~in~file~name~'#1'. } 13044
13045 { 13045
13046 File~names~must~contain~balanced~numbers~of~quotes~("). 13046
13047 } 13047
13048 \msg_new:nnnn { kernel } { iow-indent } 13048
13049 { Only~#1~allows~#2 } 13049
13050 { 13050
13051 The~command~#2~can~only~be~used~in~messages~ 13051
13052 which~will~be~wrapped~using~#1. 13052
13053 \tl_if_empty:nF {#3} { ~ It~was~called~with~argument~'#3'. } 13053
13054 } 13054
13055 \sys_if_engine luatex:TF 13055
13056 { 13056
13057 \str_const:Ne \c_sys_platform_str 13057
13058 { \tex_directlua:D { tex.print(os.type) } } 13058
13059 } 13059
13060 { 13060
13061 \file_if_exist:nTF { nul: } 13061
13062 { 13062
13063 \file_if_exist:nF { /dev/null } 13063
13064 { \str_const:Nn \c_sys_platform_str { windows } } 13064

```

```

13065     }
13066     {
13067         \file_if_exist:nT { /dev/null }
13068         { \str_const:Nn \c_sys_platform_str { unix } }
13069     }
13070 }
13071 \cs_if_exist:NF \c_sys_platform_str
13072 { \str_const:Nn \c_sys_platform_str { unknown } }
13073 \clist_map_inline:nn { unix , windows }
13074 {
13075     \__sys_const:nn { sys_if_platform_ #1 }
13076     { \str_if_eq_p:Vn \c_sys_platform_str { #1 } }
13077 }
13078 %% File: l3skip.dtx
13079 \cs_new_eq:NN \if_dim:w \tex_ifdim:D
13080 \cs_new_eq:NN \__dim_eval:w \tex_dimexpr:D
13081 \cs_new_eq:NN \__dim_eval_end: \tex_relax:D
13082 \scan_new:N \s__dim_mark
13083 \scan_new:N \s__dim_stop
13084 \cs_new:Npn \__dim_use_none_delimit_by_s_stop:w #1 \s__dim_stop { }
13085 \cs_new_protected:Npn \dim_new:N #1
13086 {
13087     \__kernel_chk_if_free_cs:N #1
13088     \cs:w newdimen \cs_end: #1
13089 }
13090 \cs_generate_variant:Nn \dim_new:N { c }
13091 \cs_new_protected:Npn \dim_const:Nn #1#2
13092 {
13093     \dim_new:N #1
13094     \tex_global:D #1 = \dim_eval:n {#2} \scan_stop:
13095 }
13096 \cs_generate_variant:Nn \dim_const:Nn { c }
13097 \cs_new_protected:Npn \dim_zero:N #1 { #1 = \c_zero_skip }
13098 \cs_new_protected:Npn \dim_gzero:N #1
13099 { \tex_global:D #1 = \c_zero_skip }
13100 \cs_generate_variant:Nn \dim_zero:N { c }
13101 \cs_generate_variant:Nn \dim_gzero:N { c }
13102 \cs_new_protected:Npn \dim_zero_new:N #1
13103 { \dim_if_exist:NTF #1 { \dim_zero:N #1 } { \dim_new:N #1 } }
13104 \cs_new_protected:Npn \dim_gzero_new:N #1
13105 { \dim_if_exist:NTF #1 { \dim_gzero:N #1 } { \dim_new:N #1 } }
13106 \cs_generate_variant:Nn \dim_zero_new:N { c }
13107 \cs_generate_variant:Nn \dim_gzero_new:N { c }
13108 \prg_new_eq_conditional:NNn \dim_if_exist:N \cs_if_exist:N
13109 { TF , T , F , p }
13110 \prg_new_eq_conditional:NNn \dim_if_exist:c \cs_if_exist:c

```

```

13111 { TF , T , F , p }
13112 \cs_new_protected:Npn \dim_set:Nn #1#2
13113 { #1 = \__dim_eval:w #2 \__dim_eval_end: \scan_stop: }
13114 \cs_new_protected:Npn \dim_gset:Nn #1#2
13115 { \tex_global:D #1 = \__dim_eval:w #2 \__dim_eval_end: \scan_stop: }
13116 \cs_generate_variant:Nn \dim_set:Nn { NV , c , cV }
13117 \cs_generate_variant:Nn \dim_gset:Nn { NV , c , cV }
13118 \cs_new_protected:Npn \dim_set_eq:NN #1#2
13119 { #1 = #2 \scan_stop: }
13120 \cs_generate_variant:Nn \dim_set_eq:NN { c , Nc , cc }
13121 \cs_new_protected:Npn \dim_gset_eq:NN #1#2
13122 { \tex_global:D #1 = #2 \scan_stop: }
13123 \cs_generate_variant:Nn \dim_gset_eq:NN { c , Nc , cc }
13124 \cs_new_protected:Npn \dim_add:Nn #1#2
13125 { \tex_advance:D #1 \__dim_eval:w #2 \__dim_eval_end: \scan_stop: }
13126 \cs_new_protected:Npn \dim_gadd:Nn #1#2
13127 {
13128     \tex_global:D \tex_advance:D #1
13129     \__dim_eval:w #2 \__dim_eval_end: \scan_stop:
13130 }
13131 \cs_generate_variant:Nn \dim_add:Nn { c }
13132 \cs_generate_variant:Nn \dim_gadd:Nn { c }
13133 \cs_new_protected:Npn \dim_sub:Nn #1#2
13134 { \tex_advance:D #1 - \__dim_eval:w #2 \__dim_eval_end: \scan_stop: }
13135 \cs_new_protected:Npn \dim_gsub:Nn #1#2
13136 {
13137     \tex_global:D \tex_advance:D #1
13138     -\__dim_eval:w #2 \__dim_eval_end: \scan_stop:
13139 }
13140 \cs_generate_variant:Nn \dim_sub:Nn { c }
13141 \cs_generate_variant:Nn \dim_gsub:Nn { c }
13142 \cs_new_eq:NN \__dim_sep: \__kernel_int_sep:
13143 \cs_new:Npn \dim_abs:n #1
13144 {
13145     \exp_after:wN \__dim_abs:N
13146     \dim_use:N \__dim_eval:w #1 \__dim_eval_end:
13147 }
13148 \cs_new:Npn \__dim_abs:N #1
13149 { \if_meaning:w - #1 \else: \exp_after:wN #1 \fi: }
13150 \cs_new:Npn \dim_max:nn #1#2
13151 {
13152     \dim_use:N \__dim_eval:w \exp_after:wN \__dim_maxmin:wwN
13153     \dim_use:N \__dim_eval:w #1 \exp_after:wN \__dim_sep:
13154     \dim_use:N \__dim_eval:w #2 \__dim_sep:
13155     >
13156     \__dim_eval_end:

```



```

13157 }
13158 \cs_new:Npn \dim_min:nn #1#2
13159 {
13160   \dim_use:N \__dim_eval:w \exp_after:wN \__dim_maxmin:wwN
13161   \dim_use:N \__dim_eval:w #1 \exp_after:wN \__dim_sep:
13162   \dim_use:N \__dim_eval:w #2 \__dim_sep:
13163   <
13164   \__dim_eval_end:
13165 }
13166 \cs_new:Npn \__dim_maxmin:wwN #1 \__dim_sep: #2 \__dim_sep: #3
13167 {
13168   \if_dim:w #1 #3 #2 ~
13169     #1
13170   \else:
13171     #2
13172   \fi:
13173 }
13174 \cs_new:Npn \dim_ratio:nn #1#2
13175 { \__dim_ratio:n {#1} / \__dim_ratio:n {#2} }
13176 \cs_new:Npn \__dim_ratio:n #1
13177 { \int_value:w \__dim_eval:w (#1) \__dim_eval_end: }
13178 \prg_new_conditional:Npnn \dim_compare:nNn #1#2#3 { p , T , F , TF }
13179 {
13180   \if_dim:w \__dim_eval:w #1 #2 \__dim_eval:w #3 \__dim_eval_end:
13181   \prg_return_true: \else: \prg_return_false: \fi:
13182 }
13183 \prg_new_conditional:Npnn \dim_compare:n #1 { p , T , F , TF }
13184 {
13185   \exp_after:wN \__dim_compare:w
13186   \dim_use:N \__dim_eval:w #1 \__dim_compare_error:
13187 }
13188 \cs_new:Npn \__dim_compare:w #1 \__dim_compare_error:
13189 {
13190   \exp_after:wN \if_false: \exp:w \exp_end_continue_f:w
13191   \__dim_compare:wNN #1 ? { = \__dim_compare_end:w \else: } \s__dim_stop
13192 }
13193 \exp_args:Nno \use:nn
13194 { \cs_new:Npn \__dim_compare:wNN #1 } { \tl_to_str:n {pt} #2#3 }
13195 {
13196   \if_meaning:w = #3
13197     \use:c { __dim_compare_#2:w }
13198   \fi:
13199   #1 pt \exp_stop_f:
13200   \prg_return_false:
13201   \exp_after:wN \__dim_use_none_delimit_by_s_stop:w
13202   \fi:

```

13203	\reverse_if:N \if_dim:w #1 pt #2	13203
13204	\exp_after:wN __dim_compare:wNN	13204
13205	\dim_use:N __dim_eval:w #3	13205
13206	}	13206
13207	\cs_new:cpn { __dim_compare_ ! :w }	13207
13208	#1 \reverse_if:N #2 ! #3 = { #1 #2 = #3 }	13208
13209	\cs_new:cpn { __dim_compare_ = :w }	13209
13210	#1 __dim_eval:w = { #1 __dim_eval:w }	13210
13211	\cs_new:cpn { __dim_compare_ < :w }	13211
13212	#1 \reverse_if:N #2 < #3 = { #1 #2 > #3 }	13212
13213	\cs_new:cpn { __dim_compare_ > :w }	13213
13214	#1 \reverse_if:N #2 > #3 = { #1 #2 < #3 }	13214
13215	\cs_new:Npn __dim_compare_end:w #1 \prg_return_false: #2 \s__dim_stop	13215
13216	{ #1 \prg_return_false: \else: \prg_return_true: \fi: }	13216
13217	\cs_new_protected:Npn __dim_compare_error:	13217
13218	{	13218
13219	\if_int_compare:w \c_zero_int \c_zero_int \fi:	13219
13220	=	13220
13221	__dim_compare_error:	13221
13222	}	13222
13223	\cs_new:Npn \dim_case:nnTF #1	13223
13224	{	13224
13225	\exp:w	13225
13226	\exp_args:Nf __dim_case:nnTF { \dim_eval:n {#1} }	13226
13227	}	13227
13228	\cs_new:Npn \dim_case:nnT #1#2#3	13228
13229	{	13229
13230	\exp:w	13230
13231	\exp_args:Nf __dim_case:nnTF { \dim_eval:n {#1} } {#2} {#3} { }	13231
13232	}	13232
13233	\cs_new:Npn \dim_case:nnF #1#2	13233
13234	{	13234
13235	\exp:w	13235
13236	\exp_args:Nf __dim_case:nnTF { \dim_eval:n {#1} } {#2} { }	13236
13237	}	13237
13238	\cs_new:Npn \dim_case:nn #1#2	13238
13239	{	13239
13240	\exp:w	13240
13241	\exp_args:Nf __dim_case:nnTF { \dim_eval:n {#1} } {#2} { } { }	13241
13242	}	13242
13243	\cs_new:Npn __dim_case:nnTF #1#2#3#4	13243
13244	{ __dim_case:nw {#1} #2 {#1} { } \s__dim_mark {#3} \s__dim_mark {#4} \s__dim_stop }	13244
13245	\cs_new:Npn __dim_case:nw #1#2#3	13245
13246	{	13246
13247	\dim_compare:nNnTF {#1} = {#2}	13247
13248	{ __dim_case_end:nw {#3} }	13248

13249	{ _dim_case:nw {#1} }	13249
13250	}	13250
13251	\cs_new:Npn _dim_case_end:nw #1#2#3 \s__dim_mark #4#5 \s__dim_stop	13251
13252	{ \exp_end: #1 #4 }	13252
13253	\cs_new:Npn \dim_while_do:nn #1#2	13253
13254	{	13254
13255	\dim_compare:nT {#1}	13255
13256	{	13256
13257	#2	13257
13258	\dim_while_do:nn {#1} {#2}	13258
13259	}	13259
13260	}	13260
13261	\cs_new:Npn \dim_until_do:nn #1#2	13261
13262	{	13262
13263	\dim_compare:nF {#1}	13263
13264	{	13264
13265	#2	13265
13266	\dim_until_do:nn {#1} {#2}	13266
13267	}	13267
13268	}	13268
13269	\cs_new:Npn \dim_do_while:nn #1#2	13269
13270	{	13270
13271	#2	13271
13272	\dim_compare:nT {#1}	13272
13273	{ \dim_do_while:nn {#1} {#2} }	13273
13274	}	13274
13275	\cs_new:Npn \dim_do_until:nn #1#2	13275
13276	{	13276
13277	#2	13277
13278	\dim_compare:nF {#1}	13278
13279	{ \dim_do_until:nn {#1} {#2} }	13279
13280	}	13280
13281	\cs_new:Npn \dim_while_do:nNnn #1#2#3#4	13281
13282	{	13282
13283	\dim_compare:nNnT {#1} #2 {#3}	13283
13284	{	13284
13285	#4	13285
13286	\dim_while_do:nNnn {#1} #2 {#3} {#4}	13286
13287	}	13287
13288	}	13288
13289	\cs_new:Npn \dim_until_do:nNnn #1#2#3#4	13289
13290	{	13290
13291	\dim_compare:nNnF {#1} #2 {#3}	13291
13292	{	13292
13293	#4	13293
13294	\dim_until_do:nNnn {#1} #2 {#3} {#4}	13294

```

13295 }
13296 }
13297 \cs_new:Npn \dim_do_while:nNnn #1#2#3#4
13298 {
13299     #4
13300     \dim_compare:nNnT {#1} #2 {#3}
13301     { \dim_do_while:nNnn {#1} #2 {#3} {#4} }
13302 }
13303 \cs_new:Npn \dim_do_until:nNnn #1#2#3#4
13304 {
13305     #4
13306     \dim_compare:nNnF {#1} #2 {#3}
13307     { \dim_do_until:nNnn {#1} #2 {#3} {#4} }
13308 }
13309 \cs_new:Npn \dim_step_function:nnnN #1#2#3
13310 {
13311     \exp_after:wN \__dim_step:wwwN
13312     \tex_the:D \__dim_eval:w #1 \exp_after:wN \__dim_sep:
13313     \tex_the:D \__dim_eval:w #2 \exp_after:wN \__dim_sep:
13314     \tex_the:D \__dim_eval:w #3 \__dim_sep:
13315 }
13316 \cs_new:Npn \__dim_step:wwwN #1 \__dim_sep: #2 \__dim_sep: #3 \__dim_sep: #4
13317 {
13318     \dim_compare:nNnTF {#2} > \c_zero_dim
13319     { \__dim_step:NnnnN > }
13320     {
13321         \dim_compare:nNnTF {#2} = \c_zero_dim
13322         {
13323             \msg_expandable_error:nnn { kernel } { zero-step } {#4}
13324             \use_none:nnnn
13325         }
13326         { \__dim_step:NnnnN < }
13327     }
13328     {#1} {#2} {#3} #4
13329 }
13330 \cs_new:Npn \__dim_step:NnnnN #1#2#3#4#5
13331 {
13332     \dim_compare:nNnF {#2} #1 {#4}
13333     {
13334         #5 {#2}
13335         \exp_args:NNf \__dim_step:NnnnN
13336         #1 { \dim_eval:n { #2 + #3 } } {#3} {#4} #5
13337     }
13338 }
13339 \cs_new_protected:Npn \dim_step_inline:nnnn
13340 {

```

```

13341 \int_gincr:N \g__kernel_prg_map_int 13341
13342 \exp_args:NNc \__dim_step:NNnnnn 13342
13343 \cs_gset_protected:Npn 13343
13344 { \__dim_map_ \int_use:N \g__kernel_prg_map_int :w } 13344
13345 } 13345
13346 \cs_new_protected:Npn \dim_step_variable:nnnNn #1#2#3#4#5 13346
13347 { 13347
13348 \int_gincr:N \g__kernel_prg_map_int 13348
13349 \exp_args:NNc \__dim_step:NNnnnn 13349
13350 \cs_gset_protected:Npe 13350
13351 { \__dim_map_ \int_use:N \g__kernel_prg_map_int :w } 13351
13352 {#1}{#2}{#3} 13352
13353 { 13353
13354 \tl_set:Nn \exp_not:N #4 {##1} 13354
13355 \exp_not:n {#5} 13355
13356 } 13356
13357 } 13357
13358 \cs_new_protected:Npn \__dim_step:NNnnnn #1#2#3#4#5#6 13358
13359 { 13359
13360 #1 #2 ##1 {#6} 13360
13361 \dim_step_function:nnnN {#3} {#4} {#5} #2 13361
13362 \prg_break_point:Nn \scan_stop: { \int_gdecr:N \g__kernel_prg_map_int } 13362
13363 } 13363
13364 \cs_new:Npn \dim_eval:n #1 13364
13365 { \dim_use:N \__dim_eval:w #1 \__dim_eval_end: } 13365
13366 \cs_new:Npn \dim_sign:n #1 13366
13367 { 13367
13368 \int_value:w \exp_after:wN \__dim_sign:Nw 13368
13369 \dim_use:N \__dim_eval:w #1 \__dim_eval_end: \__dim_sep: 13369
13370 \exp_stop_f: 13370
13371 } 13371
13372 \cs_new:Npn \__dim_sign:Nw #1#2 \__dim_sep: 13372
13373 { 13373
13374 \if_dim:w #1#2 > \c_zero_dim 13374
13375 1 13375
13376 \else: 13376
13377 \if_meaning:w - #1 13377
13378 -1 13378
13379 \else: 13379
13380 0 13380
13381 \fi: 13381
13382 \fi: 13382
13383 } 13383
13384 \cs_new_eq:NN \dim_use:N \tex_the:D 13384
13385 \cs_new:Npn \dim_use:c #1 { \tex_the:D \cs:w #1 \cs_end: } 13385
13386 \cs_new:Npn \dim_to_decimal:n #1 13386

```

```

13387 {
13388     \exp_after:wN
13389     \__dim_to_decimal:w \dim_use:N \__dim_eval:w #1 \__dim_eval_end:
13390 }
13391 \use:e
13392 {
13393     \cs_new:Npn \exp_not:N \__dim_to_decimal:w
13394         #1 . #2 \tl_to_str:n { pt }
13395 }
13396 {
13397     \int_compare:nNnTF {#2} > \c_zero_int
13398         { #1 . #2 }
13399         { #1 }
13400 }
13401 \cs_new:Npn \dim_to_decimal_in_sp:n #1
13402 { \int_value:w \__dim_eval:w #1 \__dim_eval_end: }
13403 \group_begin:
13404     \cs_set_protected:Npn \__dim_tmp:w #1#2
13405     {
13406         \cs_new:cpn { dim_to_decimal_in_ #1 :n } ##1
13407         {
13408             \exp_after:wN \__dim_to_decimal_aux:w
13409             \int_value:w \__dim_eval:w ##1 \__dim_eval_end: \__dim_sep: #2 \__dim_sep:
13410         }
13411     }
13412     \__dim_tmp:w { in } { 50 / 7227 } % delta = 7227/100
13413     \__dim_tmp:w { pc } { 1 / 24 } % delta = 12/1
13414     \__dim_tmp:w { cm } { 127 / 7227 } % delta = 7227/254
13415     \__dim_tmp:w { mm } { 1270 / 7227 } % delta = 7227/2540
13416     \__dim_tmp:w { bp } { 400 / 803 } % delta = 7227/7200
13417     \__dim_tmp:w { dd } { 1157 / 2476 } % delta = 1238/1157
13418     \__dim_tmp:w { cc } { 1157 / 29712 } % delta = 14856/1157
13419 \group_end:
13420 \cs_new:Npn \__dim_to_decimal_aux:w #1#2 \__dim_sep: #3 \__dim_sep:
13421 {
13422     \dim_to_decimal:n
13423     {
13424         \int_eval:n
13425         { ( 2 * #1#2 \if:w #1 - - \else: + \fi: 1 ) * #3 }
13426         sp
13427     }
13428 }
13429 \cs_new:Npn \dim_to_decimal_in_unit:nn #1#2
13430 {
13431     \exp_after:wN \__dim_chk_unit:w
13432     \int_value:w \__dim_eval:w #2 \__dim_eval_end: \__dim_sep: {#1}

```

```
13433 } 13433
13434 \cs_new:Npn \__dim_chk_unit:w #1#2 \__dim_sep: #3 13434
13435 { 13435
13436 \token_if_eq_charcode:NNTF #1 0 13436
13437 { \msg_expandable_error:nn { dim } { zero-unit } } 13437
13438 { 13438
13439 \exp_after:wN \__dim_branch_unit:w 13439
13440 \int_value:w \if:w #1 - - \fi: \__dim_eval:w #3 \exp_after:wN \__dim_sep: 13440
13441 \int_value:w \if:w #1 - - \fi: #1#2 \__dim_sep: 13441
13442 } 13442
13443 } 13443
13444 \cs_new:Npn \__dim_branch_unit:w #1 \__dim_sep: #2 \__dim_sep: 13444
13445 { 13445
13446 \int_compare:nNnTF {#2} > { 65536 } 13446
13447 { \__dim_to_decimal_aux:w #1 \__dim_sep: 32768 / #2 \__dim_sep: } 13447
13448 { 13448
13449 \int_compare:nNnTF {#2} = { 65536 } 13449
13450 { \dim_to_decimal:n { #1sp } } 13450
13451 { \__dim_get_quotient:w #1 \__dim_sep: #2 \__dim_sep: } 13451
13452 } 13452
13453 } 13453
13454 \cs_new:Npn \__dim_get_quotient:w #1#2 \__dim_sep: #3 \__dim_sep: 13454
13455 { 13455
13456 \token_if_eq_charcode:NNTF #1 0 13456
13457 { 0 } 13457
13458 { 13458
13459 \token_if_eq_charcode:NNTF #1 - 13459
13460 { 13460
13461 \exp_after:wN \exp_after:wN \exp_after:wN \__dim_get_remainder:w 13461
13462 \int_eval:n { ( 2 * #2 - #3 ) / ( 2 * #3 ) } \__dim_sep: 13462
13463 #2 \__dim_sep: #3 \__dim_sep: - \__dim_sep: 13463
13464 } 13464
13465 { 13465
13466 \exp_after:wN \exp_after:wN \exp_after:wN \__dim_get_remainder:w 13466
13467 \int_eval:n { ( 2 * #1#2 - #3 ) / ( 2 * #3 ) } \__dim_sep: 13467
13468 #1#2 \__dim_sep: #3 \__dim_sep: \__dim_sep: 13468
13469 } 13469
13470 } 13470
13471 } 13471
13472 \cs_new:Npn \__dim_get_remainder:w #1 \__dim_sep: #2 \__dim_sep: #3 \__dim_sep: 13472
13473 { 13473
13474 \exp_after:wN \exp_after:wN \exp_after:wN \__dim_convert_remainder:w 13474
13475 \int_eval:n { #2 - #1 * #3 } \__dim_sep: 13475
13476 #3 \__dim_sep: #1 \__dim_sep: 13476
13477 } 13477
13478 \cs_new:Npn \__dim_convert_remainder:w #1 \__dim_sep: #2 \__dim_sep: 13478
```



```

13479 {
13480     \exp_after:wN \exp_after:wN \exp_after:wN \_dim_test_candidate:w
13481     \int_eval:n { #1 * 65536 / #2 } \_dim_sep:
13482     #1 \_dim_sep: #2 \_dim_sep:
13483 }
13484 \cs_new:Npn \_dim_test_candidate:w #1 \_dim_sep: #2 \_dim_sep: #3 \_dim_sep:
13485 {
13486     \dim_compare:nNnTF { #2sp } =
13487     { \dim_to_decimal:n { #1sp } \_dim_eval:w #3sp \_dim_eval_end: }
13488     { \_dim_parse_decimal:w #1 \_dim_sep: }
13489     {
13490         \_dim_parse_decimal:w \int_eval:n { #1 + 1 } \_dim_sep:
13491     }
13492 }
13493 \cs_new:Npn \_dim_parse_decimal:w #1 \_dim_sep: #2 \_dim_sep: #3 \_dim_sep:
13494 {
13495     \exp_after:wN \_dim_parse_decimal_aux:w
13496     \int_value:w #3 \int_eval:w #2 + \dim_to_decimal:n { #1sp } \_dim_sep:
13497 }
13498 \cs_new:Npn \_dim_parse_decimal_aux:w #1 \_dim_sep: {#1}
13499 \cs_new_eq:NN \dim_show:N \__kernel_register_show:N
13500 \cs_generate_variant:Nn \dim_show:N { c }
13501 \cs_new_protected:Npn \dim_show:n
13502 { \__kernel_msg_show_eval:Nn \dim_eval:n }
13503 \cs_new_eq:NN \dim_log:N \__kernel_register_log:N
13504 \cs_new_eq:NN \dim_log:c \__kernel_register_log:c
13505 \cs_new_protected:Npn \dim_log:n
13506 { \__kernel_msg_log_eval:Nn \dim_eval:n }
13507 \dim_const:Nn \c_zero_dim { 0 pt }
13508 \dim_const:Nn \c_max_dim { 16383.99999 pt }
13509 \dim_new:N \l_tmpa_dim
13510 \dim_new:N \l_tmpb_dim
13511 \dim_new:N \g_tmpa_dim
13512 \dim_new:N \g_tmpb_dim
13513 \scan_new:N \s__skip_stop
13514 \cs_new_protected:Npn \skip_new:N #1
13515 {
13516     \__kernel_chk_if_free_cs:N #1
13517     \cs:w newskip \cs_end: #1
13518 }
13519 \cs_generate_variant:Nn \skip_new:N { c }
13520 \cs_new_protected:Npn \skip_const:Nn #1#2
13521 {
13522     \skip_new:N #1
13523     \tex_global:D #1 = \skip_eval:n {#2} \scan_stop:
13524 }

```

```

13525 \cs_generate_variant:Nn \skip_const:Nn { c }
13526 \cs_new_eq:NN \skip_zero:N \dim_zero:N
13527 \cs_new_eq:NN \skip_gzero:N \dim_gzero:N
13528 \cs_generate_variant:Nn \skip_zero:N { c }
13529 \cs_generate_variant:Nn \skip_gzero:N { c }
13530 \cs_new_protected:Npn \skip_zero_new:N #1
13531 { \skip_if_exist:NTF #1 { \skip_zero:N #1 } { \skip_new:N #1 } }
13532 \cs_new_protected:Npn \skip_gzero_new:N #1
13533 { \skip_if_exist:NTF #1 { \skip_gzero:N #1 } { \skip_new:N #1 } }
13534 \cs_generate_variant:Nn \skip_zero_new:N { c }
13535 \cs_generate_variant:Nn \skip_gzero_new:N { c }
13536 \prg_new_eq_conditional:NNn \skip_if_exist:N \cs_if_exist:N
13537 { TF , T , F , p }
13538 \prg_new_eq_conditional:NNn \skip_if_exist:c \cs_if_exist:c
13539 { TF , T , F , p }
13540 \cs_new_protected:Npn \skip_set:Nn #1#2
13541 { #1 = \tex_glueexpr:D #2 \scan_stop: }
13542 \cs_new_protected:Npn \skip_gset:Nn #1#2
13543 { \tex_global:D #1 = \tex_glueexpr:D #2 \scan_stop: }
13544 \cs_generate_variant:Nn \skip_set:Nn { NV , c , cV }
13545 \cs_generate_variant:Nn \skip_gset:Nn { NV , c , cV }
13546 \cs_new_protected:Npn \skip_set_eq:NN #1#2 { #1 = #2 }
13547 \cs_generate_variant:Nn \skip_set_eq:NN { c , Nc , cc }
13548 \cs_new_protected:Npn \skip_gset_eq:NN #1#2 { \tex_global:D #1 = #2 }
13549 \cs_generate_variant:Nn \skip_gset_eq:NN { c , Nc , cc }
13550 \cs_new_protected:Npn \skip_add:Nn #1#2
13551 { \tex_advance:D #1 \tex_glueexpr:D #2 \scan_stop: }
13552 \cs_new_protected:Npn \skip_gadd:Nn #1#2
13553 { \tex_global:D \tex_advance:D #1 \tex_glueexpr:D #2 \scan_stop: }
13554 \cs_generate_variant:Nn \skip_add:Nn { c }
13555 \cs_generate_variant:Nn \skip_gadd:Nn { c }
13556 \cs_new_protected:Npn \skip_sub:Nn #1#2
13557 { \tex_advance:D #1 - \tex_glueexpr:D #2 \scan_stop: }
13558 \cs_new_protected:Npn \skip_gsub:Nn #1#2
13559 { \tex_global:D \tex_advance:D #1 - \tex_glueexpr:D #2 \scan_stop: }
13560 \cs_generate_variant:Nn \skip_sub:Nn { c }
13561 \cs_generate_variant:Nn \skip_gsub:Nn { c }
13562 \prg_new_conditional:Npnn \skip_if_eq:nn #1#2 { p , T , F , TF }
13563 {
13564     \str_if_eq:eeTF { \skip_eval:n {#1} } { \skip_eval:n {#2} }
13565     { \prg_return_true: }
13566     { \prg_return_false: }
13567 }
13568 \cs_new_eq:NN \__skip_sep: \__kernel_int_sep:
13569 \cs_set_protected:Npn \__skip_tmp:w #1
13570 {

```

13571	\prg_new_conditional:Npn \skip_if_finite:n ##1 { p , T , F , TF }	13571
13572	{	13572
13573	\exp_after:wN __skip_if_finite:wwNw	13573
13574	\skip_use:N \tex_glueexpr:D ##1 __skip_sep: \prg_return_false:	13574
13575	#1 __skip_sep: \prg_return_true: \s__skip_stop	13575
13576	}	13576
13577	\cs_new:Npn __skip_if_finite:wwNw ##1 #1 ##2 __skip_sep: ##3 ##4 \s__skip_stop	13577 ✓
13578	{##3}	13578
13579	\exp_args:No __skip_tmp:w { \tl_to_str:n { fil } }	13579
13580	\cs_new:Npn \skip_eval:n #1	13580
13581	{ \skip_use:N \tex_glueexpr:D #1 \scan_stop: }	13581
13582	\cs_new_eq:NN \skip_use:N \dim_use:N	13582
13583	\cs_new_eq:NN \skip_use:c \dim_use:c	13583
13584	\cs_new_eq:NN \skip_horizontal:N \tex_hskip:D	13584
13585	\cs_new:Npn \skip_horizontal:n #1	13585
13586	{ \skip_horizontal:N \tex_glueexpr:D #1 \scan_stop: }	13586
13587	\cs_new_eq:NN \skip_vertical:N \tex_vskip:D	13587
13588	\cs_new:Npn \skip_vertical:n #1	13588
13589	{ \skip_vertical:N \tex_glueexpr:D #1 \scan_stop: }	13589
13590	\cs_generate_variant:Nn \skip_horizontal:N { c }	13590
13591	\cs_generate_variant:Nn \skip_vertical:N { c }	13591
13592	\cs_new_eq:NN \skip_show:N __kernel_register_show:N	13592
13593	\cs_generate_variant:Nn \skip_show:N { c }	13593
13594	\cs_new_protected:Npn \skip_show:n	13594
13595	{ __kernel_msg_show_eval:Nn \skip_eval:n }	13595
13596	\cs_new_eq:NN \skip_log:N __kernel_register_log:N	13596
13597	\cs_new_eq:NN \skip_log:c __kernel_register_log:c	13597
13598	\cs_new_protected:Npn \skip_log:n	13598
13599	{ __kernel_msg_log_eval:Nn \skip_eval:n }	13599
13600	\skip_const:Nn \c_zero_skip { \c_zero_dim }	13600
13601	\skip_const:Nn \c_max_skip { \c_max_dim }	13601
13602	\skip_new:N \l_tmpa_skip	13602
13603	\skip_new:N \l_tmpb_skip	13603
13604	\skip_new:N \g_tmpa_skip	13604
13605	\skip_new:N \g_tmpb_skip	13605
13606	\cs_new_protected:Npn \muskip_new:N #1	13606
13607	{	13607
13608	__kernel_chk_if_free_cs:N #1	13608
13609	\cs:w newmuskip \cs_end: #1	13609
13610	}	13610
13611	\cs_generate_variant:Nn \muskip_new:N { c }	13611
13612	\cs_new_protected:Npn \muskip_const:Nn #1#2	13612
13613	{	13613
13614	\muskip_new:N #1	13614
13615	\tex_global:D #1 = \muskip_eval:n {#2} \scan_stop:	13615

13616	}	13616
13617	\cs_generate_variant:Nn \muskip_const:Nn { c }	13617
13618	\cs_new_protected:Npn \muskip_zero:N #1	13618
13619	{ #1 = \c_zero_muskip }	13619
13620	\cs_new_protected:Npn \muskip_gzero:N #1	13620
13621	{ \tex_global:D #1 = \c_zero_muskip }	13621
13622	\cs_generate_variant:Nn \muskip_zero:N { c }	13622
13623	\cs_generate_variant:Nn \muskip_gzero:N { c }	13623
13624	\cs_new_protected:Npn \muskip_zero_new:N #1	13624
13625	{ \muskip_if_exist:NTF #1 { \muskip_zero:N #1 } { \muskip_new:N #1 } }	13625
13626	\cs_new_protected:Npn \muskip_gzero_new:N #1	13626
13627	{ \muskip_if_exist:NTF #1 { \muskip_gzero:N #1 } { \muskip_new:N #1 } }	13627
13628	\cs_generate_variant:Nn \muskip_zero_new:N { c }	13628
13629	\cs_generate_variant:Nn \muskip_gzero_new:N { c }	13629
13630	\prg_new_eq_conditional:NNn \muskip_if_exist:N \cs_if_exist:N	13630
13631	{ TF , T , F , p }	13631
13632	\prg_new_eq_conditional:NNn \muskip_if_exist:c \cs_if_exist:c	13632
13633	{ TF , T , F , p }	13633
13634	\cs_new_protected:Npn \muskip_set:Nn #1#2	13634
13635	{ #1 = \tex_muexpr:D #2 \scan_stop: }	13635
13636	\cs_new_protected:Npn \muskip_gset:Nn #1#2	13636
13637	{ \tex_global:D #1 = \tex_muexpr:D #2 \scan_stop: }	13637
13638	\cs_generate_variant:Nn \muskip_set:Nn { NV , c , cV }	13638
13639	\cs_generate_variant:Nn \muskip_gset:Nn { NV , c , cV }	13639
13640	\cs_new_protected:Npn \muskip_set_eq:NN #1#2 { #1 = #2 }	13640
13641	\cs_generate_variant:Nn \muskip_set_eq:NN { c , Nc , cc }	13641
13642	\cs_new_protected:Npn \muskip_gset_eq:NN #1#2 { \tex_global:D #1 = #2 }	13642
13643	\cs_generate_variant:Nn \muskip_gset_eq:NN { c , Nc , cc }	13643
13644	\cs_new_protected:Npn \muskip_add:Nn #1#2	13644
13645	{ \tex_advance:D #1 \tex_muexpr:D #2 \scan_stop: }	13645
13646	\cs_new_protected:Npn \muskip_gadd:Nn #1#2	13646
13647	{ \tex_global:D \tex_advance:D #1 \tex_muexpr:D #2 \scan_stop: }	13647
13648	\cs_generate_variant:Nn \muskip_add:Nn { c }	13648
13649	\cs_generate_variant:Nn \muskip_gadd:Nn { c }	13649
13650	\cs_new_protected:Npn \muskip_sub:Nn #1#2	13650
13651	{ \tex_advance:D #1 - \tex_muexpr:D #2 \scan_stop: }	13651
13652	\cs_new_protected:Npn \muskip_gsub:Nn #1#2	13652
13653	{ \tex_global:D \tex_advance:D #1 - \tex_muexpr:D #2 \scan_stop: }	13653
13654	\cs_generate_variant:Nn \muskip_sub:Nn { c }	13654
13655	\cs_generate_variant:Nn \muskip_gsub:Nn { c }	13655
13656	\cs_new:Npn \muskip_eval:n #1	13656
13657	{ \muskip_use:N \tex_muexpr:D #1 \scan_stop: }	13657
13658	\cs_new_eq:NN \muskip_use:N \dim_use:N	13658
13659	\cs_new_eq:NN \muskip_use:c \dim_use:c	13659
13660	\cs_new_eq:NN \muskip_show:N __kernel_register_show:N	13660
13661	\cs_generate_variant:Nn \muskip_show:N { c }	13661

```
13662 \cs_new_protected:Npn \muskip_show:n 13662
13663 { \__kernel_msg_show_eval:Nn \muskip_eval:n } 13663
13664 \cs_new_eq:NN \muskip_log:N \__kernel_register_log:N 13664
13665 \cs_new_eq:NN \muskip_log:c \__kernel_register_log:c 13665
13666 \cs_new_protected:Npn \muskip_log:n 13666
13667 { \__kernel_msg_log_eval:Nn \muskip_eval:n } 13667
13668 \muskip_const:Nn \c_zero_muskip { 0 mu } 13668
13669 \muskip_const:Nn \c_max_muskip { 16383.99999 mu } 13669
13670 \muskip_new:N \l_tmpa_muskip 13670
13671 \muskip_new:N \l_tmpb_muskip 13671
13672 \muskip_new:N \g_tmpa_muskip 13672
13673 \muskip_new:N \g_tmpb_muskip 13673
13674 %% File: l3keys.dtx 13674
13675 \scan_new:N \s__keyval_nil 13675
13676 \scan_new:N \s__keyval_mark 13676
13677 \scan_new:N \s__keyval_stop 13677
13678 \scan_new:N \s__keyval_tail 13678
13679 \bool_new:N \l__kernel_keyval_allow_blank_keys_bool 13679
13680 \group_begin: 13680
13681 \cs_set_protected:Npn \__keyval_tmp:w #1#2 13681
13682 { 13682
13683 \cs_new:Npn \keyval_parse:nnn ##1 ##2 ##3 13683
13684 { 13684
13685 \__kernel_exp_not:w \tex_expanded:D 13685
13686 { 13686
13687 { 13687
13688 \__keyval_loop_active:nnw {##1} {##2} 13688
13689 \s__keyval_mark ##3 #1 \s__keyval_tail #1 13689
13690 } 13690
13691 } 13691
13692 } 13692
13693 \cs_new_eq:NN \keyval_parse:NNn \keyval_parse:nnn 13693
13694 \cs_new:Npn \__keyval_loop_active:nnw ##1 ##2 ##3 #1 13694
13695 { 13695
13696 \__keyval_if_recursion_tail:w ##3 13696
13697 \__keyval_end_loop_active:w \s__keyval_tail 13697
13698 \__keyval_loop_other:nnw {##1} {##2} ##3 , \s__keyval_tail , 13698
13699 } 13699
13700 \cs_new:Npn \__keyval_split_other:w ##1 = ##2 \s__keyval_mark ##3 13700
13701 { ##3 ##1 \s__keyval_stop \s__keyval_mark ##2 } 13701
13702 \cs_new:Npn \__keyval_split_active:w ##1 #2 ##2 \s__keyval_mark ##3 13702
13703 { ##3 ##1 \s__keyval_stop \s__keyval_mark ##2 } 13703
13704 \cs_new:Npn \__keyval_loop_other:nnw ##1 ##2 ##3 , 13704
13705 { 13705
13706 \__keyval_if_recursion_tail:w ##3 13706
13707 \__keyval_end_loop_other:w \s__keyval_tail 13707
```

```
13708     \__keyval_split_active:w ##3 \s__keyval_nil 13708
13709     \s__keyval_mark \__keyval_split_active_auxi:w 13709
13710     #2 \s__keyval_mark \__keyval_clean_up_active:w 13710
13711     {##1} {##2} 13711
13712     \s__keyval_mark 13712
13713 } 13713
13714 \cs_new:Npn \__keyval_split_active_auxi:w ##1 \s__keyval_stop 13714
13715 { 13715
13716     \__keyval_split_other:w ##1 \s__keyval_nil 13716
13717     \s__keyval_mark \__keyval_misplaced_equal_after_active_error:w 13717
13718     = \s__keyval_mark \__keyval_split_active_auxii:w 13718
13719 } 13719
13720 \cs_new:Npn \__keyval_split_active_auxii:w 13720
13721     ##1 \s__keyval_nil \s__keyval_mark 13721
13722     \__keyval_misplaced_equal_after_active_error:w 13722
13723     \s__keyval_stop \s__keyval_mark 13722
13724     ##2 \s__keyval_nil #2 \s__keyval_mark \__keyval_clean_up_active:w 13723
13725     { \__keyval_trim:nN {##1} \__keyval_split_active_auxiii:w ##2 \s__keyval_nil } 13724
13726 \cs_new:Npn \__keyval_split_active_auxiii:w ##1 ##2 \s__keyval_nil 13725
13727 { 13726
13728     \__keyval_split_active:w ##2 \s__keyval_nil 13727
13729     \s__keyval_mark \__keyval_misplaced_equal_in_split_error:w 13728
13730     #2 \s__keyval_mark \__keyval_split_active_auxiv:w 13729
13731     {##1} 13730
13732 } 13731
13733 \cs_new:Npn \__keyval_split_active_auxiv:w 13732
13734     ##1 \s__keyval_nil \s__keyval_mark \__keyval_misplaced_equal_in_split_error:w 13733
13735     \s__keyval_stop \s__keyval_mark 13734
13736 { 13735
13737     \__keyval_split_other:w ##1 \s__keyval_nil 13736
13738     \s__keyval_mark \__keyval_misplaced_equal_in_split_error:w 13737
13739     = \s__keyval_mark \__keyval_split_active_auxv:w 13738
13740 } 13739
13741 \cs_new:Npn \__keyval_split_active_auxv:w 13740
13742     ##1 \s__keyval_nil \s__keyval_mark \__keyval_misplaced_equal_in_split_error:w 13741
13743     \s__keyval_stop \s__keyval_mark 13742
13744     { \__keyval_trim:nN { ##1 } \__keyval_pair:nnnn } 13743
13745 \cs_new:Npn \__keyval_clean_up_active:w 13744
13746     ##1 \s__keyval_nil \s__keyval_mark \__keyval_split_active_auxi:w 13745
13747     \s__keyval_stop \s__keyval_mark 13746
13748 { 13747
13749     \__keyval_split_other:w ##1 \s__keyval_nil 13747
13750     \s__keyval_mark \__keyval_split_other_auxi:w 13748
13751     = \s__keyval_mark \__keyval_clean_up_other:w 13749
13752 } 13750
13753 \cs_new:Npn \__keyval_split_other_auxi:w ##1 \s__keyval_stop 13751
```



```
13752 { \__keyval_trim:nN { ##1 } \__keyval_split_other_auxii:w } 13752
13753 \cs_new:Npn \__keyval_split_other_auxii:w 13753
13754 ##1 ##2 \s__keyval_nil = \s__keyval_mark \__keyval_clean_up_other:w 13754
13755 { 13755
13756 \__keyval_split_other:w ##2 \s__keyval_nil 13756
13757 \s__keyval_mark \__keyval_misplaced_equal_in_split_error:w 13757
13758 = \s__keyval_mark \__keyval_split_other_auxiii:w 13758
13759 { ##1 } 13759
13760 } 13760
13761 \cs_new:Npn \__keyval_split_other_auxiii:w 13761
13762 ##1 \s__keyval_nil \s__keyval_mark \__keyval_misplaced_equal_in_split_error:w 13762
13763 \s__keyval_stop \s__keyval_mark 13763
13764 { \__keyval_trim:nN { ##1 } \__keyval_pair:nnnn } 13764
13765 \cs_new:Npn \__keyval_clean_up_other:w 13765
13766 ##1 \s__keyval_nil \s__keyval_mark \__keyval_split_other_auxi:w ✓ 13766
\s__keyval_stop \s__keyval_mark 13766
13767 { 13767
13768 \__keyval_if_blank:w ##1 \s__keyval_nil \s__keyval_stop \__keyval_blank_true:w 13768
13769 \s__keyval_mark \s__keyval_stop 13769
13770 \__keyval_trim:nN { ##1 } \__keyval_key:nn 13770
13771 } 13771
13772 \cs_new:Npn \__keyval_misplaced_equal_after_active_error:w 13772
13773 \s__keyval_mark ##1 \s__keyval_stop \s__keyval_mark ##2 \s__keyval_nil 13773
13774 = \s__keyval_mark \__keyval_split_active_auxii:w 13774
13775 \s__keyval_mark ##3 \s__keyval_nil 13775
13776 #2 \s__keyval_mark \__keyval_clean_up_active:w 13776
13777 { 13777
13778 \msg_expandable_error:nn 13778
13779 { keyval } { misplaced-equals-sign } 13779
13780 \__keyval_loop_other:nnw 13780
13781 } 13781
13782 \cs_new:Npn \__keyval_misplaced_equal_in_split_error:w 13782
13783 \s__keyval_mark ##1 \s__keyval_stop \s__keyval_mark ##2 \s__keyval_nil 13783
13784 ##3 \s__keyval_mark ##4 ##5 13784
13785 { 13785
13786 \msg_expandable_error:nn 13786
13787 { keyval } { misplaced-equals-sign } 13787
13788 \__keyval_loop_other:nnw 13788
13789 } 13789
13790 \cs_new:Npn \__keyval_end_loop_other:w 13790
13791 \s__keyval_tail 13791
13792 \__keyval_split_active:w 13792
13793 \s__keyval_mark \s__keyval_tail 13793
13794 \s__keyval_nil \s__keyval_mark 13794
13795 \__keyval_split_active_auxi:w 13795
13796 #2 \s__keyval_mark \__keyval_clean_up_active:w 13796
```



```
13797 { \__keyval_loop_active:nnw } 13797
13798 \cs_new:Npn \__keyval_end_loop_active:w 13798
13799 \s_keyval_tail 13799
13800 \__keyval_loop_other:nnw ##1 \s_keyval_mark \s_keyval_tail , \s_keyval_tail ✓ 13800
, 13800
13801 { } 13801
13802 } 13802
13803 \char_set_catcode_active:n { \, } 13803
13804 \char_set_catcode_active:n { \= } 13804
13805 \__keyval_tmp:w , = 13805
13806 \group_end: 13806
13807 \cs_generate_variant:Nn \keyval_parse:NNn { NNv , NNv } 13807
13808 \cs_generate_variant:Nn \keyval_parse:nnn { nnV , nnv } 13808
13809 \group_begin: 13809
13810 \cs_set_protected:Npn \__keyval_tmp:w #1#2 13810
13811 { 13811
13812 \cs_new:Npn \__keyval_pair:nnnn ##1 ##2 ##3 ##4 13812
13813 { 13813
13814 \__keyval_if_blank:w \s_keyval_mark ##2 \s_keyval_nil \s_keyval_stop ✓ 13814
\__keyval_blank_key_error:w 13814
13815 \s_keyval_mark \s_keyval_stop 13815
13816 #1 13816
13817 \exp_not:n { ##4 {##2} {##1} } 13817
13818 #2 13818
13819 \__keyval_loop_other:nnw {##3} {##4} 13819
13820 } 13820
13821 \cs_new:Npn \__keyval_key:nn ##1 ##2 13821
13822 { 13822
13823 \__keyval_if_blank:w \s_keyval_mark ##1 \s_keyval_nil \s_keyval_stop ✓ 13823
\__keyval_blank_key_error:w 13823
13824 \s_keyval_mark \s_keyval_stop 13824
13825 #1 13825
13826 \exp_not:n { ##2 {##1} } 13826
13827 #2 13827
13828 \__keyval_loop_other:nnw {##2} 13828
13829 } 13829
13830 } 13830
13831 \__keyval_tmp:w { } { } 13831
13832 \group_end: 13832
13833 \cs_new:Npn \__keyval_if_empty:w #1 \s_keyval_mark \s_keyval_stop { } 13833
13834 \cs_new:Npn \__keyval_if_blank:w \s_keyval_mark #1 { \__keyval_if_empty:w ✓ 13834
\s_keyval_mark } 13834
13835 \cs_new:Npn \__keyval_if_recursion_tail:w \s_keyval_mark #1 \s_keyval_tail { } 13835
13836 \cs_new:Npn \__keyval_blank_true:w \s_keyval_mark \s_keyval_stop \__keyval_trim:nN #1 ✓ 13836
\__keyval_key:nn 13836
13837 { \__keyval_loop_other:nnw } 13837
```

```

13838 \cs_new:Npn \__keyval_blank_key_error:w \s__keyval_mark \s__keyval_stop #1 ↙
13839 \__keyval_loop_other:nnw 13838
13839 { 13839
13840 \bool_if:NTF \l__kernel_keyval_allow_blank_keys_bool 13840
13841 { #1 } 13841
13842 { \msg_expandable_error:nn { keyval } { blank-key-name } } 13842
13843 \__keyval_loop_other:nnw 13843
13844 } 13844
13845 \msg_new:nnn { keyval } { misplaced-equals-sign } 13845
13846 { Misplaced~'='~in~key~value~input~\msg_line_context: } 13846
13847 \msg_new:nnn { keyval } { blank-key-name } 13847
13848 { Blank~key~name~in~key~value~input~\msg_line_context: } 13848
13849 \prop_gput:Nnn \g_msg_module_name_prop { keyval } { LaTeX } 13849
13850 \prop_gput:Nnn \g_msg_module_type_prop { keyval } { } 13850
13851 \group_begin: 13851
13852 \cs_set_protected:Npn \__keyval_tmp:w #1 13852
13853 { 13853
13854 \cs_new:Npn \__keyval_trim:nN ##1 13854
13855 { 13855
13856 \__keyval_trim_auxi:w 13856
13857 ##1 13857
13858 \s__keyval_nil 13858
13859 \s__keyval_mark #1 { } 13859
13860 \s__keyval_mark \__keyval_trim_auxii:w 13860
13861 \__keyval_trim_auxiii:w 13861
13862 #1 \s__keyval_nil 13862
13863 \__keyval_trim_auxiv:w 13863
13864 } 13864
13865 \cs_new:Npn \__keyval_trim_auxi:w ##1 \s__keyval_mark #1 ##2 \s__keyval_mark ##3 13865
13866 { 13866
13867 ##3 13867
13868 \__keyval_trim_auxi:w 13868
13869 \s__keyval_mark 13869
13870 ##2 13870
13871 \s__keyval_mark #1 {##1} 13871
13872 } 13872
13873 \cs_new:Npn \__keyval_trim_auxii:w \__keyval_trim_auxi:w \s__keyval_mark ↙
13874 \s__keyval_mark ##1 13873
13874 { 13874
13875 \__keyval_trim_auxiii:w 13875
13876 ##1 13876
13877 } 13877
13878 \cs_new:Npn \__keyval_trim_auxiii:w ##1 #1 \s__keyval_nil ##2 13878
13879 { 13879
13880 ##2 13880
13881 ##1 \s__keyval_nil 13881

```

13882	__keyval_trim_auxiii:w	13882
13883	}	13883
13884	\cs_new:Npn __keyval_trim_auxiv:w	13884
13885	\s__keyval_mark ##1 \s__keyval_nil	13885
13886	__keyval_trim_auxiii:w \s__keyval_nil __keyval_trim_auxiii:w	13886
13887	##2	13887
13888	{ ##2 { ##1 } }	13888
13889	}	13889
13890	__keyval_tmp:w { ~ }	13890
13891	\group_end:	13891
13892	\str_const:Nn \c__keys_code_root_str { key~code~>~ }	13892
13893	\str_const:Nn \c__keys_check_root_str { key~check~>~ }	13893
13894	\str_const:Nn \c__keys_default_root_str { key~default~>~ }	13894
13895	\str_const:Nn \c__keys_groups_root_str { key~groups~>~ }	13895
13896	\str_const:Nn \c__keys_inherit_root_str { key~inherit~>~ }	13896
13897	\str_const:Nn \c__keys_type_root_str { key~type~>~ }	13897
13898	\str_const:Nn \c__keys_props_root_str { key~prop~>~ }	13898
13899	\int_new:N \l_keys_choice_int	13899
13900	\tl_new:N \l_keys_choice_tl	13900
13901	\clist_new:N \l__keys_groups_clist	13901
13902	\bool_new:N \l__keys_inherit_bool	13902
13903	\clist_new:N \l__keys_inherit_clist	13903
13904	\str_new:N \l_keys_key_str	13904
13905	\tl_new:N \l_keys_key_tl	13905
13906	\str_new:N \l__keys_module_str	13906
13907	\bool_new:N \l__keys_no_value_bool	13907
13908	\bool_new:N \l__keys_only_known_bool	13908
13909	\str_new:N \l_keys_path_str	13909
13910	\tl_new:N \l_keys_path_tl	13910
13911	\str_new:N \l__keys_inherit_str	13911
13912	\tl_new:N \l__keys_relative_tl	13912
13913	\tl_set:Nn \l__keys_relative_tl { \q__keys_no_value }	13913
13914	\str_new:N \l__keys_property_str	13914
13915	\bool_new:N \l__keys_selective_bool	13915
13916	\bool_new:N \l__keys_exclude_bool	13916
13917	\clist_new:N \l__keys_selective_clist	13917
13918	\clist_new:N \l__keys_tmp_clist	13918
13919	\clist_new:N \l__keys_unused_clist	13919
13920	\tl_new:N \l_keys_value_tl	13920
13921	\bool_new:N \l__keys_tmp_bool	13921
13922	\tl_new:N \l__keys_tmpa_tl	13922
13923	\tl_new:N \l__keys_tmpb_tl	13923
13924	\bool_new:N \l__keys_precompile_bool	13924
13925	\tl_new:N \l__keys_precompile_tl	13925
13926	\prop_new:N \l_keys_usage_load_prop	13926
13927	\prop_new:N \l_keys_usage_preamble_prop	13927

```

13928 \scan_new:N \s__keys_nil 13928
13929 \scan_new:N \s__keys_mark 13929
13930 \scan_new:N \s__keys_stop 13930
13931 \quark_new:N \q__keys_no_value 13931
13932 \__kernel_quark_new_conditional:Nn \__keys_quark_if_no_value:N { TF } 13932
13933 \cs_new_protected:Npn \__keys_precompile:n #1 13933
13934 { 13934
13935     \bool_if:NTF \l__keys_precompile_bool 13935
13936     { \tl_put_right:Nn \l__keys_precompile_tl } 13936
13937     { \use:n } 13937
13938     {#1} 13938
13939 } 13939
13940 \cs_new_protected:Npn \__keys_cs_undefine:c #1 13940
13941 { 13941
13942     \if_cs_exist:w #1 \cs_end: 13942
13943     \else: 13943
13944         \use_i:nnnn 13944
13945         \fi: 13945
13946         \cs_set_eq:cN {#1} \tex_undefined:D 13946
13947     } 13947
13948 \cs_new_protected:Npn \keys_define:nn #1#2 13948
13949 { 13949
13950     \use:e 13950
13951     { 13951
13952         \exp_not:n 13952
13953         { 13953
13954             \str_set:Ne \l__keys_module_str { \__keys_trim_spaces:n {#1} } 13954
13955             \keyval_parse:NNn \__keys_define:n \__keys_define:nn {#2} 13955
13956         } 13956
13957         \__keys_reset_var:N \l__keys_module_str 13957
13958         \__keys_reset_var:N \l__keys_inherit_str 13958
13959         \__keys_reset_var:N \l_keys_choice_tl 13959
13960         \__keys_reset_var:N \l_keys_key_tl 13960
13961         \__keys_reset_var:N \l_keys_key_str 13961
13962         \__keys_reset_var:N \l_keys_path_tl 13962
13963         \__keys_reset_var:N \l_keys_path_str 13963
13964         \__keys_reset_var:N \l_keys_value_tl 13964
13965         \int_set:Nn \l_keys_choice_int { \int_use:N \l_keys_choice_int } 13965
13966     } 13966
13967 } 13967
13968 \cs_generate_variant:Nn \keys_define:nn { ne , nx } 13968
13969 \cs_new_protected:Npn \__keys_define:n #1 13969
13970 { 13970
13971     \bool_set_true:N \l__keys_no_value_bool 13971
13972     \__keys_define_aux:nn {#1} { } 13972
13973 } 13973

```

```
13974 \cs_new_protected:Npn \__keys_define:nn #1#2
13975 {
13976     \bool_set_false:N \l__keys_no_value_bool
13977     \__keys_define_aux:nn {#1} {#2}
13978 }
13979 \cs_new_protected:Npn \__keys_define_aux:nn #1#2
13980 {
13981     \__keys_property_find:n {#1}
13982     \cs_if_exist:cTF { \c__keys_props_root_str \l__keys_property_str }
13983     { \__keys_define_code:n {#2} }
13984     {
13985         \str_if_empty:NF \l__keys_property_str
13986         {
13987             \msg_error:nnee { keys } { property-unknown }
13988             \l__keys_property_str \l_keys_path_str
13989         }
13990     }
13991 }
13992 \cs_new_protected:Npn \__keys_property_find:n #1
13993 {
13994     \exp_after:wN \__keys_property_find_auxi:w \tl_to_str:n {#1}
13995     \s__keys_nil \__keys_property_find_auxii:w
13996     . \s__keys_nil \__keys_property_find_err:w
13997 }
13998 \cs_new:Npn \__keys_property_find_auxi:w #1 . #2 \s__keys_nil #3
13999 {
14000     #3 #1 \s__keys_mark #2 \s__keys_nil #3
14001 }
14002 \cs_new_protected:Npn \__keys_property_find_auxii:w
14003     #1 \s__keys_mark #2 \s__keys_nil \__keys_property_find_auxii:w . \s__keys_nil
14004     \__keys_property_find_err:w
14005 {
14006     \cs_set_nopar:Npe \l_keys_path_str
14007     {
14008         \str_if_empty:NF \l__keys_module_str { \l__keys_module_str / }
14009         \exp_after:wN \__keys_trim_spaces:n \tex_expanded:D {{
14010             #1
14011             \if_false: } } } \fi:
14012         \__keys_property_find_auxi:w #2 \s__keys_nil \__keys_property_find_auxiii:w
14013         . \s__keys_nil \__keys_property_find_auxiv:w
14014     }
14015 \cs_new:Npn \__keys_property_find_auxiii:w #1 \s__keys_mark #2 . #3 \s__keys_nil #4
14016 {
14017     . #1 #4 #2 \s__keys_mark #3 \s__keys_nil #4
14018 }
14019 \cs_new:Npn \__keys_property_find_auxiv:w
```

14020	#1 \s__keys_nil __keys_property_find_auxiii:w	14020
14021	\s__keys_mark \s__keys_nil __keys_property_find_auxiv:w	14021
14022	{	14022
14023	\if_false: {{{ \fi: }}}	14023
14024	\cs_set_nopar:Npe \l__keys_property_str { . #1 }	14024
14025	\tl_set_eq:NN \l_keys_path_tl \l_keys_path_str	14025
14026	}	14026
14027	\cs_new_protected:Npn __keys_property_find_err:w	14027
14028	#1 \s__keys_nil #2 __keys_property_find_err:w	14028
14029	{	14029
14030	\str_clear:N \l__keys_property_str	14030
14031	\msg_error:nnn { keys } { no-property } {#1}	14031
14032	}	14032
14033	\cs_new_protected:Npn __keys_define_code:n #1	14033
14034	{	14034
14035	\bool_if:NTF \l__keys_no_value_bool	14035
14036	{	14036
14037	__keys_define_code:nnn	14037
14038	{ \use:c { \c__keys_props_root_str \l__keys_property_str } {#1} }	14038
14039	{ \use:c { \c__keys_props_root_str \l__keys_property_str } }	14039
14040	{	14040
14041	\msg_error:nnee { keys } { property-requires-value }	14041
14042	\l__keys_property_str \l_keys_path_str	14042
14043	}	14043
14044	}	14044
14045	{ \use:c { \c__keys_props_root_str \l__keys_property_str } {#1} }	14045
14046	}	14046
14047	\cs_new:Npe __keys_define_code:nnn	14047
14048	{	14048
14049	\exp_not:N \exp_after:wN \exp_not:N __keys_define_code:w	14049
14050	\exp_not:N \l__keys_property_str	14050
14051	\c_colon_str \c_colon_str	14051
14052	\exp_not:N \s__keys_stop	14052
14053	}	14053
14054	\use:e	14054
14055	{	14055
14056	\cs_new:Npn \exp_not:N __keys_define_code:w	14056
14057	#1 \c_colon_str #2 \c_colon_str #3 \exp_not:N \s__keys_stop	14057
14058	}	14058
14059	{	14059
14060	\tl_if_empty:nTF {#3}	14060
14061	{ \use_i:nnn }	14061
14062	{	14062
14063	\tl_if_empty:nTF {#2}	14063
14064	{ \use_ii:nnn }	14064
14065	{ \use_iii:nnn }	14065

```
14066     }
14067 }
14068 \cs_new_protected:Npn \__keys_bool_set:Nn #1#2
14069 { \__keys_bool_set:Nnnn #1 {#2} { true } { false } }
14070 \cs_generate_variant:Nn \__keys_bool_set:Nn { c }
14071 \cs_new_protected:Npn \__keys_bool_set_inverse:Nn #1#2
14072 { \__keys_bool_set:Nnnn #1 {#2} { false } { true } }
14073 \cs_generate_variant:Nn \__keys_bool_set_inverse:Nn { c }
14074 \cs_new_protected:Npn \__keys_bool_set:Nnnn #1#2#3#4
14075 {
14076   \bool_if_exist:NF #1 { \bool_new:N #1 }
14077   \__keys_choice_make:
14078   \__keys_cmd_set:ne { \l_keys_path_str / true }
14079   { \exp_not:c { bool_ #2 set_ #3 :N } \exp_not:N #1 }
14080   \__keys_cmd_set:ne { \l_keys_path_str / false }
14081   { \exp_not:c { bool_ #2 set_ #4 :N } \exp_not:N #1 }
14082   \__keys_cmd_set_direct:nn { \l_keys_path_str / unknown }
14083   {
14084     \msg_error:nne { keys } { boolean-values-only }
14085     \l_keys_path_str
14086   }
14087   \__keys_default_set:n { true }
14088 }
14089 \cs_generate_variant:Nn \__keys_bool_set:Nn { c }
14090 \cs_new_protected:Npn \__keys_choice_make:
14091 { \__keys_choice_make:N \__keys_choice_find:n }
14092 \cs_new_protected:Npn \__keys_multichoice_make:
14093 { \__keys_choice_make:N \__keys_multichoice_find:n }
14094 \cs_new_protected:Npn \__keys_choice_make:N #1
14095 {
14096   \cs_if_exist:cTF
14097   { \c__keys_type_root_str \__keys_parent:o \l_keys_path_str }
14098   {
14099     \str_if_eq:vnTF
14100     { \c__keys_type_root_str \__keys_parent:o \l_keys_path_str }
14101     { choice }
14102     {
14103       \msg_error:nnee { keys } { nested-choice-key }
14104       \l_keys_path_tl { \__keys_parent:o \l_keys_path_str }
14105     }
14106     { \__keys_choice_make_aux:N #1 }
14107   }
14108   { \__keys_choice_make_aux:N #1 }
14109 }
14110 \cs_new_protected:Npn \__keys_choice_make_aux:N #1
14111 {
```



```
14112 \cs_set_nopar:cpn { \c__keys_type_root_str \l_keys_path_str } 14112
14113 { choice } 14113
14114 \__keys_cmd_set_direct:nn \l_keys_path_str { #1 {##1} } 14114
14115 \__keys_cmd_set_direct:nn { \l_keys_path_str / unknown } 14115
14116 { 14116
14117 \msg_error:nnee { keys } { choice-unknown } 14117
14118 \l_keys_path_str {##1} 14118
14119 } 14119
14120 } 14120
14121 \cs_new_protected:Npn \__keys_choices_make:nn 14121
14122 { \__keys_choices_make:Nnn \__keys_choice_make: } 14122
14123 \cs_new_protected:Npn \__keys_multichoice_make:nn 14123
14124 { \__keys_choices_make:Nnn \__keys_multichoice_make: } 14124
14125 \cs_new_protected:Npn \__keys_choices_make:Nnn #1#2#3 14125
14126 { 14126
14127 #1 14127
14128 \int_zero:N \l_keys_choice_int 14128
14129 \clist_map_inline:nn {#2} 14129
14130 { 14130
14131 \int_incr:N \l_keys_choice_int 14131
14132 \__keys_cmd_set:ne 14132
14133 { \l_keys_path_str / \__keys_trim_spaces:n {##1} } 14133
14134 { 14134
14135 \tl_set:Nn \exp_not:N \l_keys_choice_tl {##1} 14135
14136 \int_set:Nn \exp_not:N \l_keys_choice_int 14136
14137 { \int_use:N \l_keys_choice_int } 14137
14138 \exp_not:n {#3} 14138
14139 } 14139
14140 } 14140
14141 } 14141
14142 \cs_new_protected:Npn \__keys_cmd_set:nn #1#2 14142
14143 { \__keys_cmd_set_direct:nn {#1} { \__keys_precompile:n {#2} } } 14143
14144 \cs_generate_variant:Nn \__keys_cmd_set:nn { ne , Vn , Vo } 14144
14145 \cs_new_protected:Npn \__keys_cmd_set_direct:nn #1#2 14145
14146 { \cs_set_protected:cpn { \c__keys_code_root_str #1 } ##1 {#2} } 14146
14147 \cs_new_protected:Npn \__keys_cs_set:NNpn #1#2#3# 14147
14148 { 14148
14149 \cs_set_protected:cpe { \c__keys_code_root_str \l_keys_path_str } ##1 14149
14150 { 14150
14151 \__keys_precompile:n 14151
14152 { #1 \exp_not:N #2 \exp_not:n {#3} {##1} } 14152
14153 } 14153
14154 \use_none:n 14154
14155 } 14155
14156 \cs_generate_variant:Nn \__keys_cs_set:NNpn { Nc } 14156
14157 \cs_new_protected:Npn \__keys_default_set:n #1 14157
```

```
14158 {
14159     \tl_if_empty:nTF {#1}
14160     {
14161         \__keys_cs_undefine:c
14162         { \c__keys_default_root_str \l_keys_path_str }
14163     }
14164     {
14165         \cs_set_nopar:cpe
14166         { \c__keys_default_root_str \l_keys_path_str }
14167         { \exp_not:n {#1} }
14168         \__keys_value_requirement:nn { required } { false }
14169     }
14170 }
14171 \cs_new_protected:Npn \__keys_groups_set:n #1
14172 {
14173     \clist_set:Nc \l__keys_groups_clist { \tl_to_str:n {#1} }
14174     \clist_if_empty:NTF \l__keys_groups_clist
14175     {
14176         \__keys_cs_undefine:c
14177         { \c__keys_groups_root_str \l_keys_path_str }
14178     }
14179     {
14180         \cs_set_eq:cN { \c__keys_groups_root_str \l_keys_path_str }
14181         \l__keys_groups_clist
14182     }
14183 }
14184 \cs_new_protected:Npn \__keys_inherit:n #1
14185 {
14186     \__keys_undefine:
14187     \clist_set:Nn \l__keys_inherit_clist {#1}
14188     \cs_set_eq:cN { \c__keys_inherit_root_str \l_keys_path_str }
14189     \l__keys_inherit_clist
14190 }
14191 \cs_new_protected:Npn \__keys_initialise:n #1
14192 {
14193     \typeout{----->}
14194     \exp_after:wN \__keys_find_key_module:wNN
14195     \l_keys_path_str \s__keys_stop
14196     \l_keys_key_tl \l_keys_key_str
14197     \tl_set_eq:NN \l_keys_key_tl \l_keys_key_str
14198     \tl_set:Nn \l_keys_value_tl {#1}
14199     \cs_if_exist:cTF { \c__keys_code_root_str \l_keys_path_str }
14200     {
14201         \str_clear:N \l__keys_inherit_str
14202         \__keys_execute:nn \l_keys_path_str {#1}
14203     }
```

```
14204 {
14205     \cs_if_exist:cT
14206         { \c_keys_inherit_root_str \__keys_parent:o \l_keys_path_str }
14207         { \__keys_execute_inherit: }
14208     }
14209 }
14210 \cs_new_protected:Npn \__keys_legacy_if_set:nn #1#2
14211 { \__keys_legacy_if_set:nnnn {#1} {#2} { true } { false } }
14212 \cs_new_protected:Npn \__keys_legacy_if_set_inverse:nn #1#2
14213 { \__keys_legacy_if_set:nnnn {#1} {#2} { false } { true } }
14214 \cs_new_protected:Npn \__keys_legacy_if_set:nnnn #1#2#3#4
14215 {
14216     \__keys_choice_make:
14217     \__keys_cmd_set:ne { \l_keys_path_str / true }
14218     { \exp_not:c { legacy_if_#2 set_ #3 :n } { \exp_not:n {#1} } }
14219     \__keys_cmd_set:ne { \l_keys_path_str / false }
14220     { \exp_not:c { legacy_if_#2 set_ #4 :n } { \exp_not:n {#1} } }
14221     \__keys_cmd_set:nn { \l_keys_path_str / unknown }
14222     {
14223         \msg_error:nne { keys } { boolean-values-only }
14224         \l_keys_path_str
14225     }
14226     \__keys_default_set:n { true }
14227     \cs_if_exist:cF { if#1 }
14228     {
14229         \cs:w newif \exp_after:wN \cs_end:
14230         \cs:w if#1 \cs_end:
14231     }
14232 }
14233 \cs_new_protected:Npn \__keys_meta_make:n
14234 { \exp_args:NV \__keys_meta_make:nn \l__keys_module_str }
14235 \cs_new_protected:Npn \__keys_meta_make:nn #1#2
14236 {
14237     \exp_args:NV \__keys_cmd_set_direct:nn
14238     \l_keys_path_str { \__keys_set:nn {#1} {#2} }
14239 }
14240 \cs_new_protected:Npn \__keys_prop_put:Nn #1#2
14241 {
14242     \prop_if_exist:NF #1 { \prop_new:N #1 }
14243     \exp_after:wN \__keys_find_key_module:wNN \l_keys_path_str \s__keys_stop
14244     \l__keys_tmpa_tl \l__keys_tmpb_tl
14245     \__keys_cmd_set:ne \l_keys_path_str
14246     {
14247         \exp_not:c { prop_ #2 put:Nnn }
14248         \exp_not:N #1
14249         { \l__keys_tmpb_tl }
```

```
14250 \exp_not:n { {##1} } 14250
14251 } 14251
14252 } 14252
14253 \cs_generate_variant:Nn \__keys_prop_put:Nn { c } 14253
14254 \cs_new_protected:Npn \__keys_undefine: 14254
14255 { 14255
14256 \clist_map_inline:nn 14256
14257 { code , default , groups , inherit , type , check } 14257
14258 { 14258
14259 \__keys_cs_undefine:c 14259
14260 { \tl_use:c { c__keys_ ##1 _root_str } \l_keys_path_str } 14260
14261 } 14261
14262 } 14262
14263 \cs_new_protected:Npn \__keys_value_requirement:nn #1#2 14263
14264 { 14264
14265 \str_case:nnF {#2} 14265
14266 { 14266
14267 { true } 14267
14268 { 14268
14269 \cs_set_eq:cc 14269
14270 { \c__keys_check_root_str \l_keys_path_str } 14270
14271 { __keys_check_ #1 : } 14271
14272 } 14272
14273 { false } 14273
14274 { 14274
14275 \cs_if_eq:ccT 14275
14276 { \c__keys_check_root_str \l_keys_path_str } 14276
14277 { __keys_check_ #1 : } 14277
14278 { 14278
14279 \__keys_cs_undefine:c 14279
14280 { \c__keys_check_root_str \l_keys_path_str } 14280
14281 } 14281
14282 } 14282
14283 } 14283
14284 { 14284
14285 \msg_error:nne { keys } 14285
14286 { boolean-values-only } 14286
14287 { .value_ #1 :n } 14287
14288 } 14288
14289 } 14289
14290 \cs_new_protected:Npn \__keys_check_forbidden: 14290
14291 { 14291
14292 \bool_if:NF \l__keys_no_value_bool 14292
14293 { 14293
14294 \msg_error:nnee { keys } { value-forbidden } 14294
14295 \l_keys_path_str \l_keys_value_tl 14295
```

14296	\use_none:nnn	14296
14297	}	14297
14298	}	14298
14299	\cs_new_protected:Npn __keys_check_required:	14299
14300	{	14300
14301	\bool_if:NT \l__keys_no_value_bool	14301
14302	{	14302
14303	\msg_error:nne { keys } { value-required }	14303
14304	\l_keys_path_str	14304
14305	\use_none:nnn	14305
14306	}	14306
14307	}	14307
14308	\cs_new_protected:Npn __keys_usage:n #1	14308
14309	{	14309
14310	\str_case:nnF {#1}	14310
14311	{	14311
14312	{ general }	14312
14313	{	14313
14314	__keys_usage:NN \l_keys_usage_load_prop	14314
14315	\c_false_bool	14315
14316	__keys_usage:NN \l_keys_usage_preamble_prop	14316
14317	\c_false_bool	14317
14318	}	14318
14319	{ load }	14319
14320	{	14320
14321	__keys_usage:NN \l_keys_usage_load_prop	14321
14322	\c_true_bool	14322
14323	__keys_usage:NN \l_keys_usage_preamble_prop	14323
14324	\c_false_bool	14324
14325	}	14325
14326	{ preamble }	14326
14327	{	14327
14328	__keys_usage:NN \l_keys_usage_load_prop	14328
14329	\c_false_bool	14329
14330	__keys_usage:NN \l_keys_usage_preamble_prop	14330
14331	\c_true_bool	14331
14332	}	14332
14333	}	14333
14334	{	14334
14335	\msg_error:nnnn { keys }	14335
14336	{ choice-unknown }	14336
14337	{ .usage:n }	14337
14338	{#1}	14338
14339	}	14339
14340	}	14340
14341	\cs_new_protected:Npn __keys_usage:NN #1#2	14341

```
14342 { 14342
14343 \prop_get:NVNF #1 \l__keys_module_str \l__keys_tmpa_tl 14343
14344 { \tl_clear:N \l__keys_tmpa_tl } 14344
14345 \tl_set:Ne \l__keys_tmpb_tl 14345
14346 { 14346
14347 \exp_after:wN \exp_after:wN \exp_after:wN \__keys_usage:w \exp_after:wN 14347
14348 \l_keys_path_str \exp_after:wN / \exp_after:wN \s__keys_stop 14348
14349 \exp_after:wN { \l_keys_path_str } 14349
14350 } 14350
14351 \bool_if:NTF #2 14351
14352 { \clist_put_right:NV \l__keys_tmpa_tl \l__keys_tmpb_tl } 14352
14353 { \clist_remove_all:NV \l__keys_tmpa_tl \l__keys_tmpb_tl } 14353
14354 \prop_put:NVV #1 \l__keys_module_str 14354
14355 \l__keys_tmpa_tl 14355
14356 } 14356
14357 \cs_new:Npn \__keys_usage:w #1 / #2 \s__keys_stop #3 14357
14358 { 14358
14359 \tl_if_blank:nTF {#2} 14359
14360 {#1} 14360
14361 { \__keys_usage_aux:w #3 \s__keys_stop } 14361
14362 } 14362
14363 \cs_new:Npn \__keys_usage_aux:w #1 / #2 \s__keys_stop {#2} 14363
14364 \cs_new_protected:Npn \__keys_variable_set:NnnN #1#2#3#4 14364
14365 { 14365
14366 \use:c { #2_if_exist:NF } #1 { \use:c { #2_new:N } #1 } 14366
14367 \__keys_cmd_set:ne \l_keys_path_str 14367
14368 { 14368
14369 \exp_not:c { #2 _ #3 set:N #4 } 14369
14370 \exp_not:N #1 14370
14371 \exp_not:n { {##1} } 14371
14372 } 14372
14373 } 14373
14374 \cs_generate_variant:Nn \__keys_variable_set:NnnN { c } 14374
14375 \cs_new_protected:Npn \__keys_variable_set_required:NnnN #1#2#3#4 14375
14376 { 14376
14377 \__keys_variable_set:NnnN #1 {#2} {#3} #4 14377
14378 \__keys_value_requirement:nn { required } { true } 14378
14379 } 14379
14380 \cs_generate_variant:Nn \__keys_variable_set_required:NnnN { c } 14380
14381 \cs_new_protected:cpn { \c__keys_props_root_str .bool_set:N } #1 14381
14382 { \__keys_bool_set:Nn #1 { } } 14382
14383 \cs_new_protected:cpn { \c__keys_props_root_str .bool_set:c } #1 14383
14384 { \__keys_bool_set:cn {#1} { } } 14384
14385 \cs_new_protected:cpn { \c__keys_props_root_str .bool_gset:N } #1 14385
14386 { \__keys_bool_set:Nn #1 { g } } 14386
14387 \cs_new_protected:cpn { \c__keys_props_root_str .bool_gset:c } #1 14387
```

14388	{ __keys_bool_set:cn {#1} { g } }	14388
14389	\cs_new_protected:cpn { \c__keys_props_root_str .bool_set_inverse:N } #1	14389
14390	{ __keys_bool_set_inverse:Nn #1 { } }	14390
14391	\cs_new_protected:cpn { \c__keys_props_root_str .bool_set_inverse:c } #1	14391
14392	{ __keys_bool_set_inverse:cn {#1} { } }	14392
14393	\cs_new_protected:cpn { \c__keys_props_root_str .bool_gset_inverse:N } #1	14393
14394	{ __keys_bool_set_inverse:Nn #1 { g } }	14394
14395	\cs_new_protected:cpn { \c__keys_props_root_str .bool_gset_inverse:c } #1	14395
14396	{ __keys_bool_set_inverse:cn {#1} { g } }	14396
14397	\cs_new_protected:cpn { \c__keys_props_root_str .choice: }	14397
14398	{ __keys_choice_make: }	14398
14399	\cs_new_protected:cpn { \c__keys_props_root_str .choices:nn } #1	14399
14400	{ __keys_choices_make:nn #1 }	14400
14401	\cs_new_protected:cpn { \c__keys_props_root_str .choices:Vn } #1	14401
14402	{ \exp_args:NV __keys_choices_make:nn #1 }	14402
14403	\cs_new_protected:cpn { \c__keys_props_root_str .choices:en } #1	14403
14404	{ \exp_args:Ne __keys_choices_make:nn #1 }	14404
14405	\cs_new_protected:cpn { \c__keys_props_root_str .choices:on } #1	14405
14406	{ \exp_args:No __keys_choices_make:nn #1 }	14406
14407	\cs_new_protected:cpn { \c__keys_props_root_str .choices:xn } #1	14407
14408	{ \exp_args:Nx __keys_choices_make:nn #1 }	14408
14409	\cs_new_protected:cpn { \c__keys_props_root_str .code:n } #1	14409
14410	{ __keys_cmd_set:nn \l_keys_path_str {#1} }	14410
14411	\cs_new_protected:cpn { \c__keys_props_root_str .clist_set:N } #1	14411
14412	{ __keys_variable_set:NnnN #1 { clist } { } n }	14412
14413	\cs_new_protected:cpn { \c__keys_props_root_str .clist_set:c } #1	14413
14414	{ __keys_variable_set:cnN {#1} { clist } { } n }	14414
14415	\cs_new_protected:cpn { \c__keys_props_root_str .clist_gset:N } #1	14415
14416	{ __keys_variable_set:NnnN #1 { clist } { g } n }	14416
14417	\cs_new_protected:cpn { \c__keys_props_root_str .clist_gset:c } #1	14417
14418	{ __keys_variable_set:cnN {#1} { clist } { g } n }	14418
14419	\cs_new_protected:cpn { \c__keys_props_root_str .cs_set:Np } #1	14419
14420	{ __keys_cs_set:NNpn \cs_set:Npn #1 { } }	14420
14421	\cs_new_protected:cpn { \c__keys_props_root_str .cs_set:cp } #1	14421
14422	{ __keys_cs_set:Ncpn \cs_set:Npn #1 { } }	14422
14423	\cs_new_protected:cpn { \c__keys_props_root_str .cs_set_protected:Np } #1	14423
14424	{ __keys_cs_set:NNpn \cs_set_protected:Npn #1 { } }	14424
14425	\cs_new_protected:cpn { \c__keys_props_root_str .cs_set_protected:cp } #1	14425
14426	{ __keys_cs_set:Ncpn \cs_set_protected:Npn #1 { } }	14426
14427	\cs_new_protected:cpn { \c__keys_props_root_str .cs_gset:Np } #1	14427
14428	{ __keys_cs_set:NNpn \cs_gset:Npn #1 { } }	14428
14429	\cs_new_protected:cpn { \c__keys_props_root_str .cs_gset:cp } #1	14429
14430	{ __keys_cs_set:Ncpn \cs_gset:Npn #1 { } }	14430
14431	\cs_new_protected:cpn { \c__keys_props_root_str .cs_gset_protected:Np } #1	14431
14432	{ __keys_cs_set:NNpn \cs_gset_protected:Npn #1 { } }	14432
14433	\cs_new_protected:cpn { \c__keys_props_root_str .cs_gset_protected:cp } #1	14433

14434	{ __keys_cs_set:Ncpn \cs_gset_protected:Npn #1 { } }	14434
14435	\cs_new_protected:cpn { \c__keys_props_root_str .default:n } #1	14435
14436	{ __keys_default_set:n {#1} }	14436
14437	\cs_new_protected:cpn { \c__keys_props_root_str .default:V } #1	14437
14438	{ \exp_args:NV __keys_default_set:n #1 }	14438
14439	\cs_new_protected:cpn { \c__keys_props_root_str .default:e } #1	14439
14440	{ \exp_args:Ne __keys_default_set:n {#1} }	14440
14441	\cs_new_protected:cpn { \c__keys_props_root_str .default:o } #1	14441
14442	{ \exp_args:No __keys_default_set:n {#1} }	14442
14443	\cs_new_protected:cpn { \c__keys_props_root_str .default:x } #1	14443
14444	{ \exp_args:Nx __keys_default_set:n {#1} }	14444
14445	\cs_new_protected:cpn { \c__keys_props_root_str .dim_set:N } #1	14445
14446	{ __keys_variable_set_required:NnnN #1 { dim } { } n }	14446
14447	\cs_new_protected:cpn { \c__keys_props_root_str .dim_set:c } #1	14447
14448	{ __keys_variable_set_required:cnnN {#1} { dim } { } n }	14448
14449	\cs_new_protected:cpn { \c__keys_props_root_str .dim_gset:N } #1	14449
14450	{ __keys_variable_set_required:NnnN #1 { dim } { g } n }	14450
14451	\cs_new_protected:cpn { \c__keys_props_root_str .dim_gset:c } #1	14451
14452	{ __keys_variable_set_required:cnnN {#1} { dim } { g } n }	14452
14453	\cs_new_protected:cpn { \c__keys_props_root_str .fp_set:N } #1	14453
14454	{ __keys_variable_set_required:NnnN #1 { fp } { } n }	14454
14455	\cs_new_protected:cpn { \c__keys_props_root_str .fp_set:c } #1	14455
14456	{ __keys_variable_set_required:cnnN {#1} { fp } { } n }	14456
14457	\cs_new_protected:cpn { \c__keys_props_root_str .fp_gset:N } #1	14457
14458	{ __keys_variable_set_required:NnnN #1 { fp } { g } n }	14458
14459	\cs_new_protected:cpn { \c__keys_props_root_str .fp_gset:c } #1	14459
14460	{ __keys_variable_set_required:cnnN {#1} { fp } { g } n }	14460
14461	\cs_new_protected:cpn { \c__keys_props_root_str .groups:n } #1	14461
14462	{ __keys_groups_set:n {#1} }	14462
14463	\cs_new_protected:cpn { \c__keys_props_root_str .inherit:n } #1	14463
14464	{ __keys_inherit:n {#1} }	14464
14465	\cs_new_protected:cpn { \c__keys_props_root_str .initial:n } #1	14465
14466	{ __keys_initialise:n {#1} }	14466
14467	\cs_new_protected:cpn { \c__keys_props_root_str .initial:V } #1	14467
14468	{ \exp_args:NV __keys_initialise:n #1 }	14468
14469	\cs_new_protected:cpn { \c__keys_props_root_str .initial:e } #1	14469
14470	{ \exp_args:Ne __keys_initialise:n {#1} }	14470
14471	\cs_new_protected:cpn { \c__keys_props_root_str .initial:o } #1	14471
14472	{ \exp_args:No __keys_initialise:n {#1} }	14472
14473	\cs_new_protected:cpn { \c__keys_props_root_str .initial:x } #1	14473
14474	{ \exp_args:Nx __keys_initialise:n {#1} }	14474
14475	\cs_new_protected:cpn { \c__keys_props_root_str .int_set:N } #1	14475
14476	{ __keys_variable_set_required:NnnN #1 { int } { } n }	14476
14477	\cs_new_protected:cpn { \c__keys_props_root_str .int_set:c } #1	14477
14478	{ __keys_variable_set_required:cnnN {#1} { int } { } n }	14478
14479	\cs_new_protected:cpn { \c__keys_props_root_str .int_gset:N } #1	14479

14480	{ __keys_variable_set_required:NnnN #1 { int } { g } n }	14480
14481	\cs_new_protected:cpn { \c__keys_props_root_str .int_gset:c } #1	14481
14482	{ __keys_variable_set_required:cnnN {#1} { int } { g } n }	14482
14483	\cs_new_protected:cpn { \c__keys_props_root_str .legacy_if_set:n } #1	14483
14484	{ __keys_legacy_if_set:nn {#1} { } }	14484
14485	\cs_new_protected:cpn { \c__keys_props_root_str .legacy_if_gset:n } #1	14485
14486	{ __keys_legacy_if_set:nn {#1} { g } }	14486
14487	\cs_new_protected:cpn { \c__keys_props_root_str .legacy_if_set_inverse:n } #1	14487
14488	{ __keys_legacy_if_set_inverse:nn {#1} { } }	14488
14489	\cs_new_protected:cpn { \c__keys_props_root_str .legacy_if_gset_inverse:n } #1	14489
14490	{ __keys_legacy_if_set_inverse:nn {#1} { g } }	14490
14491	\cs_new_protected:cpn { \c__keys_props_root_str .meta:n } #1	14491
14492	{ __keys_meta_make:n {#1} }	14492
14493	\cs_new_protected:cpn { \c__keys_props_root_str .meta:nn } #1	14493
14494	{ __keys_meta_make:nn #1 }	14494
14495	\cs_new_protected:cpn { \c__keys_props_root_str .multichoice: }	14495
14496	{ __keys_multichoice_make: }	14496
14497	\cs_new_protected:cpn { \c__keys_props_root_str .multichoices:nn } #1	14497
14498	{ __keys_multichoices_make:nn #1 }	14498
14499	\cs_new_protected:cpn { \c__keys_props_root_str .multichoices:Vn } #1	14499
14500	{ \exp_args:NV __keys_multichoices_make:nn #1 }	14500
14501	\cs_new_protected:cpn { \c__keys_props_root_str .multichoices:en } #1	14501
14502	{ \exp_args:Ne __keys_multichoices_make:nn #1 }	14502
14503	\cs_new_protected:cpn { \c__keys_props_root_str .multichoices:on } #1	14503
14504	{ \exp_args:No __keys_multichoices_make:nn #1 }	14504
14505	\cs_new_protected:cpn { \c__keys_props_root_str .multichoices:xn } #1	14505
14506	{ \exp_args:Nx __keys_multichoices_make:nn #1 }	14506
14507	\cs_new_protected:cpn { \c__keys_props_root_str .muskip_set:N } #1	14507
14508	{ __keys_variable_set_required:NnnN #1 { muskip } { } n }	14508
14509	\cs_new_protected:cpn { \c__keys_props_root_str .muskip_set:c } #1	14509
14510	{ __keys_variable_set_required:cnnN {#1} { muskip } { } n }	14510
14511	\cs_new_protected:cpn { \c__keys_props_root_str .muskip_gset:N } #1	14511
14512	{ __keys_variable_set_required:NnnN #1 { muskip } { g } n }	14512
14513	\cs_new_protected:cpn { \c__keys_props_root_str .muskip_gset:c } #1	14513
14514	{ __keys_variable_set_required:cnnN {#1} { muskip } { g } n }	14514
14515	\cs_new_protected:cpn { \c__keys_props_root_str .prop_put:N } #1	14515
14516	{ __keys_prop_put:Nn #1 { } }	14516
14517	\cs_new_protected:cpn { \c__keys_props_root_str .prop_put:c } #1	14517
14518	{ __keys_prop_put:cn {#1} { } }	14518
14519	\cs_new_protected:cpn { \c__keys_props_root_str .prop_gput:N } #1	14519
14520	{ __keys_prop_put:Nn #1 { g } }	14520
14521	\cs_new_protected:cpn { \c__keys_props_root_str .prop_gput:c } #1	14521
14522	{ __keys_prop_put:cn {#1} { g } }	14522
14523	\cs_new_protected:cpn { \c__keys_props_root_str .skip_set:N } #1	14523
14524	{ __keys_variable_set_required:NnnN #1 { skip } { } n }	14524
14525	\cs_new_protected:cpn { \c__keys_props_root_str .skip_set:c } #1	14525

14526	{ __keys_variable_set_required:cnnN {#1} { skip } { } n }	14526
14527	\cs_new_protected:cpn { \c__keys_props_root_str .skip_gset:N } #1	14527
14528	{ __keys_variable_set_required:NnnN #1 { skip } { g } n }	14528
14529	\cs_new_protected:cpn { \c__keys_props_root_str .skip_gset:c } #1	14529
14530	{ __keys_variable_set_required:cnnN {#1} { skip } { g } n }	14530
14531	\cs_new_protected:cpn { \c__keys_props_root_str .str_set:N } #1	14531
14532	{ __keys_variable_set:NnnN #1 { str } { } n }	14532
14533	\cs_new_protected:cpn { \c__keys_props_root_str .str_set:c } #1	14533
14534	{ __keys_variable_set:cnnN {#1} { str } { } n }	14534
14535	\cs_new_protected:cpn { \c__keys_props_root_str .str_set_e:N } #1	14535
14536	{ __keys_variable_set:NnnN #1 { str } { } e }	14536
14537	\cs_new_protected:cpn { \c__keys_props_root_str .str_set_e:c } #1	14537
14538	{ __keys_variable_set:cnnN {#1} { str } { } e }	14538
14539	\cs_new_protected:cpn { \c__keys_props_root_str .str_gset:N } #1	14539
14540	{ __keys_variable_set:NnnN #1 { str } { g } n }	14540
14541	\cs_new_protected:cpn { \c__keys_props_root_str .str_gset:c } #1	14541
14542	{ __keys_variable_set:cnnN {#1} { str } { g } n }	14542
14543	\cs_new_protected:cpn { \c__keys_props_root_str .str_gset_e:N } #1	14543
14544	{ __keys_variable_set:NnnN #1 { str } { g } e }	14544
14545	\cs_new_protected:cpn { \c__keys_props_root_str .str_gset_e:c } #1	14545
14546	{ __keys_variable_set:cnnN {#1} { str } { g } e }	14546
14547	\cs_new_protected:cpn { \c__keys_props_root_str .tl_set:N } #1	14547
14548	{ __keys_variable_set:NnnN #1 { tl } { } n }	14548
14549	\cs_new_protected:cpn { \c__keys_props_root_str .tl_set:c } #1	14549
14550	{ __keys_variable_set:cnnN {#1} { tl } { } n }	14550
14551	\cs_new_protected:cpn { \c__keys_props_root_str .tl_set_e:N } #1	14551
14552	{ __keys_variable_set:NnnN #1 { tl } { } e }	14552
14553	\cs_new_protected:cpn { \c__keys_props_root_str .tl_set_e:c } #1	14553
14554	{ __keys_variable_set:cnnN {#1} { tl } { } e }	14554
14555	\cs_new_protected:cpn { \c__keys_props_root_str .tl_gset:N } #1	14555
14556	{ __keys_variable_set:NnnN #1 { tl } { g } n }	14556
14557	\cs_new_protected:cpn { \c__keys_props_root_str .tl_gset:c } #1	14557
14558	{ __keys_variable_set:cnnN {#1} { tl } { g } n }	14558
14559	\cs_new_protected:cpn { \c__keys_props_root_str .tl_gset_e:N } #1	14559
14560	{ __keys_variable_set:NnnN #1 { tl } { g } e }	14560
14561	\cs_new_protected:cpn { \c__keys_props_root_str .tl_gset_e:c } #1	14561
14562	{ __keys_variable_set:cnnN {#1} { tl } { g } e }	14562
14563	\cs_new_protected:cpn { \c__keys_props_root_str .undefine: }	14563
14564	{ __keys_undefine: }	14564
14565	\cs_new_protected:cpn { \c__keys_props_root_str .usage:n } #1	14565
14566	{ __keys_usage:n {#1} }	14566
14567	\cs_new_protected:cpn { \c__keys_props_root_str .value_forbidden:n } #1	14567
14568	{ __keys_value_requirement:nn { forbidden } {#1} }	14568
14569	\cs_new_protected:cpn { \c__keys_props_root_str .value_required:n } #1	14569
14570	{ __keys_value_requirement:nn { required } {#1} }	14570
14571	\cs_new_protected:Npn __keys_set:nnnnNn #1#2#3#4#5#6	14571

```

14572 {
14573     \use:e
14574     {
14575         \exp_not:n
14576         {
14577             \clist_clear:N \l__keys_unused_clist
14578             \clist_set:Ne \l__keys_selective_clist { \tl_to_str:n {#2} }
14579             \tl_set:Nn \l__keys_relative_tl {#4}
14580             #6
14581             \__keys_set:nn {#1} {#3}
14582             \clist_set_eq:NN #5 \l__keys_unused_clist
14583         }
14584         \__keys_reset_bool:N \l__keys_only_known_bool
14585         \__keys_reset_bool:N \l__keys_exclude_bool
14586         \__keys_reset_bool:N \l__keys_selective_bool
14587         \__keys_reset_var:N \l__keys_unused_clist
14588         \__keys_reset_var:N \l__keys_selective_clist
14589         \__keys_reset_var:N \l__keys_relative_tl
14590         \__keys_reset_var:N \l__keys_inherit_str
14591         \__keys_reset_var:N \l_keys_choice_tl
14592         \__keys_reset_var:N \l_keys_key_tl
14593         \__keys_reset_var:N \l_keys_key_str
14594         \__keys_reset_var:N \l_keys_path_tl
14595         \__keys_reset_var:N \l_keys_path_str
14596         \__keys_reset_var:N \l_keys_value_tl
14597         \int_set:Nn \l_keys_choice_int { \int_use:N \l_keys_choice_int }
14598     }
14599 }
14600 \cs_new:Npn \__keys_reset_bool:N #1
14601 {
14602     \exp_not:c
14603     { bool_set_ \bool_if:NTF #1 { true } { false } :N }
14604     \exp_not:N #1
14605 }
14606 \cs_new:Npn \__keys_reset_var:N #1
14607 {
14608     \exp_not:n
14609     { \__kernel_tl_set:Nx #1 }
14610     { \exp_not:N \exp_not:n { \exp_not:o { #1 } } }
14611 }
14612 \cs_new_protected:Npn \__keys_set:nn #1#2
14613 { \exp_args:No \__keys_set:nnn \l__keys_module_str {#1} {#2} }
14614 \cs_new_protected:Npn \__keys_set:nnn #1#2#3
14615 {
14616     \str_set:Ne \l__keys_module_str { \__keys_trim_spaces:n {#2} }
14617     \keyval_parse:NNn \__keys_set_keyval:n \__keys_set_keyval:nn {#3}

```

14618	\str_set:Nn \l__keys_module_str {#1}	14618
14619	}	14619
14620	\cs_new_protected:Npn \keys_set:nn #1#2	14620
14621	{	14621
14622	__keys_set:nnnnNn	14622
14623	{#1} { } {#2} { \q_keys_no_value } \l__keys_tmp_clist	14623
14624	{	14624
14625	\bool_set_false:N \l__keys_only_known_bool	14625
14626	\bool_set_false:N \l__keys_exclude_bool	14626
14627	\bool_set_false:N \l__keys_selective_bool	14627
14628	}	14628
14629	}	14629
14630	\cs_generate_variant:Nn \keys_set:nn { nV , nv , ne , no , nx }	14630
14631	\cs_new_protected:Npn \keys_set_known:nnnN #1#2#3#4	14631
14632	{	14632
14633	__keys_set:nnnnNn	14633
14634	{#1} { } {#2} {#3} #4	14634
14635	{	14635
14636	\bool_set_true:N \l__keys_only_known_bool	14636
14637	\bool_set_false:N \l__keys_exclude_bool	14637
14638	\bool_set_false:N \l__keys_selective_bool	14638
14639	}	14639
14640	}	14640
14641	\cs_generate_variant:Nn \keys_set_known:nnnN { nV , nv , ne , no }	14641
14642	\cs_new_protected:Npn \keys_set_known:nnN #1#2#3	14642
14643	{ \keys_set_known:nnnN {#1} {#2} { \q_keys_no_value } #3 }	14643
14644	\cs_generate_variant:Nn \keys_set_known:nnN { nV , nv , ne , no }	14644
14645	\cs_new_protected:Npn \keys_set_known:nn #1#2	14645
14646	{ \keys_set_known:nnnN {#1} {#2} { \q_keys_no_value } \l__keys_tmp_clist }	14646
14647	\cs_generate_variant:Nn \keys_set_known:nn { nV , nv , ne , no }	14647
14648	\cs_new_protected:Npn \keys_set_exclude_groups:nnnnN #1#2#3#4#5	14648
14649	{	14649
14650	__keys_set:nnnnNn	14650
14651	{#1} {#2} {#3} {#4} #5	14651
14652	{	14652
14653	\bool_set_false:N \l__keys_only_known_bool	14653
14654	\bool_set_true:N \l__keys_exclude_bool	14654
14655	\bool_set_true:N \l__keys_selective_bool	14655
14656	}	14656
14657	}	14657
14658	\cs_generate_variant:Nn \keys_set_exclude_groups:nnnnN { nnV , nnv , nno }	14658
14659	\cs_new_protected:Npn \keys_set_exclude_groups:nnnN #1#2#3#4	14659
14660	{ \keys_set_exclude_groups:nnnnN {#1} {#2} {#3} { \q_keys_no_value } #4 }	14660
14661	\cs_generate_variant:Nn \keys_set_exclude_groups:nnnN { nnV , nnv , nno }	14661
14662	\cs_new_protected:Npn \keys_set_exclude_groups:nnn #1#2#3	14662
14663	{	14663


```
14664 \keys_set_exclude_groups:nnnnN {#1} {#2} {#3} 14664
14665 { \q__keys_no_value } \l__keys_tmp_clist 14665
14666 } 14666
14667 \cs_generate_variant:Nn \keys_set_exclude_groups:nnn { nnV , nnv , nno } 14667
14668 \cs_new_protected:Npn \keys_set_groups:nnnnN #1#2#3#4#5 14668
14669 { 14669
14670 \__keys_set:nnnnNn 14670
14671 {#1} {#2} {#3} {#4} #5 14671
14672 { 14672
14673 \bool_set_false:N \l__keys_only_known_bool 14673
14674 \bool_set_false:N \l__keys_exclude_bool 14674
14675 \bool_set_true:N \l__keys_selective_bool 14675
14676 } 14676
14677 } 14677
14678 \cs_generate_variant:Nn \keys_set_groups:nnnnN { nnV , nnv , nno } 14678
14679 \cs_new_protected:Npn \keys_set_groups:nnnN #1#2#3#4 14679
14680 { \keys_set_groups:nnnnN {#1} {#2} {#3} { \q__keys_no_value } #4 } 14680
14681 \cs_generate_variant:Nn \keys_set_groups:nnnN { nnV , nnv , nno } 14681
14682 \cs_new_protected:Npn \keys_set_groups:nnn #1#2#3 14682
14683 { 14683
14684 \keys_set_groups:nnnnN {#1} {#2} {#3} 14684
14685 { \q__keys_no_value } \l__keys_tmp_clist 14685
14686 } 14686
14687 \cs_generate_variant:Nn \keys_set_groups:nnn { nnV , nnv , nno } 14687
14688 \cs_new_protected:Npn \keys_precompile:nnN #1#2#3 14688
14689 { 14689
14690 \bool_set_true:N \l__keys_precompile_bool 14690
14691 \tl_clear:N \l__keys_precompile_tl 14691
14692 \keys_set:nn {#1} {#2} 14692
14693 \bool_set_false:N \l__keys_precompile_bool 14693
14694 \tl_set_eq:NN #3 \l__keys_precompile_tl 14694
14695 } 14695
14696 \cs_new_protected:Npn \__keys_set_keyval:n #1 14696
14697 { 14697
14698 \bool_set_true:N \l__keys_no_value_bool 14698
14699 \__keys_set_keyval:onn \l__keys_module_str {#1} { } 14699
14700 } 14700
14701 \cs_new_protected:Npn \__keys_set_keyval:nn #1#2 14701
14702 { 14702
14703 \bool_set_false:N \l__keys_no_value_bool 14703
14704 \__keys_set_keyval:onn \l__keys_module_str {#1} {#2} 14704
14705 } 14705
14706 \cs_new_protected:Npn \__keys_set_keyval:nnn #1#2#3 14706
14707 { 14707
14708 \__kernel_tl_set:Nx \l_keys_path_str 14708
14709 { 14709
```

```
14710         \tl_if_blank:nF {#1}
14711         { #1 / }
14712         \__keys_trim_spaces:n {#2}
14713     }
14714     \str_clear:N \l__keys_module_str
14715     \str_clear:N \l__keys_inherit_str
14716     \exp_after:wN \__keys_find_key_module:wNN \l_keys_path_str \s__keys_stop
14717         \l__keys_module_str \l_keys_key_str
14718     \tl_set_eq:NN \l_keys_key_tl \l_keys_key_str
14719     \__keys_value_or_default:n {#3}
14720     \bool_if:NTF \l__keys_selective_bool
14721         \__keys_set_selective:
14722         \__keys_execute:
14723     \str_set:Nn \l__keys_module_str {#1}
14724 }
14725 \cs_generate_variant:Nn \__keys_set_keyval:nnn { o }
14726 \cs_new_protected:Npn \__keys_find_key_module:wNN #1 \s__keys_stop #2 #3
14727 {
14728     \__keys_find_key_module_auxi:Nw #2 #1 \s__keys_nil \__keys_find_key_module_auxii:Nw
14729     / \s__keys_nil \__keys_find_key_module_auxiv:Nw #3
14730 }
14731 \cs_new_protected:Npn \__keys_find_key_module_auxi:Nw #1 #2 / #3 \s__keys_nil #4
14732 {
14733     #4 #1 #2 \s__keys_mark #3 \s__keys_nil #4
14734 }
14735 \cs_new_protected:Npn \__keys_find_key_module_auxii:Nw
14736     #1 #2 \s__keys_mark #3 \s__keys_nil \__keys_find_key_module_auxii:Nw
14737 {
14738     \cs_set_nopar:Npe #1 { \tl_if_empty:NF #1 { #1 / } #2 }
14739     \__keys_find_key_module_auxi:Nw #1 #3 \s__keys_nil \__keys_find_key_module_auxiii:Nw
14740 }
14741 \cs_new_protected:Npn \__keys_find_key_module_auxiii:Nw #1 #2 \s__keys_mark
14742 {
14743     \cs_set_nopar:Npe #1 { \tl_if_empty:NF #1 { #1 / } #2 }
14744     \__keys_find_key_module_auxi:Nw #1
14745 }
14746 \cs_new_protected:Npn \__keys_find_key_module_auxiv:Nw
14747     #1 #2 \s__keys_nil #3 \s__keys_mark
14748     \s__keys_nil \__keys_find_key_module_auxiv:Nw #4
14749 {
14750     \cs_set_nopar:Npn #4 { #2 }
14751 }
14752 \cs_new_protected:Npn \__keys_set_selective:
14753 {
14754     \cs_if_exist:cTF { \c__keys_groups_root_str \l_keys_path_str }
14755     {
```



```
14756         \clist_set_eq:Nc \l__keys_groups_clist 14756
14757         { \c__keys_groups_root_str \l_keys_path_str } 14757
14758     \__keys_check_groups: 14758
14759 } 14759
14760 { 14760
14761     \bool_if:NTF \l__keys_exclude_bool 14761
14762     \__keys_execute: 14762
14763     \__keys_store_unused: 14763
14764 } 14764
14765 } 14765
14766 \cs_new_protected:Npn \__keys_check_groups: 14766
14767 { 14767
14768     \bool_set_false:N \l__keys_tmp_bool 14768
14769     \clist_map_inline:Nn \l__keys_selective_clist 14769
14770     { 14770
14771         \clist_if_in:NnT \l__keys_groups_clist {##1} 14771
14772         { 14772
14773             \bool_set_true:N \l__keys_tmp_bool 14773
14774             \clist_map_break: 14774
14775         } 14775
14776     } 14776
14777     \bool_if:NTF \l__keys_tmp_bool 14777
14778     { 14778
14779         \bool_if:NTF \l__keys_exclude_bool 14779
14780         \__keys_store_unused: 14780
14781         \__keys_execute: 14781
14782     } 14782
14783     { 14783
14784         \bool_if:NTF \l__keys_exclude_bool 14784
14785         \__keys_execute: 14785
14786         \__keys_store_unused: 14786
14787     } 14787
14788 } 14788
14789 \cs_new_protected:Npn \__keys_value_or_default:n #1 14789
14790 { 14790
14791     \bool_if:NTF \l__keys_no_value_bool 14791
14792     { 14792
14793         \cs_if_exist:cTF { \c__keys_default_root_str \l_keys_path_str } 14793
14794         { 14794
14795             \tl_set_eq:Nc 14795
14796             \l_keys_value_tl 14796
14797             { \c__keys_default_root_str \l_keys_path_str } 14797
14798         } 14798
14799     } 14799
14800     \tl_clear:N \l_keys_value_tl 14800
14801     \cs_if_exist:cT 14801
```

```
14802         { \c__keys_inherit_root_str \__keys_parent:o \l_keys_path_str } 14802
14803         { \__keys_default_inherit: } 14803
14804     } 14804
14805 } 14805
14806 { \tl_set:Nn \l_keys_value_tl {#1} } 14806
14807 } 14807
14808 \cs_new_protected:Npn \__keys_default_inherit: 14808
14809 { 14809
14810     \clist_map_inline:cn 14810
14811     { \c__keys_inherit_root_str \__keys_parent:o \l_keys_path_str } 14811
14812     { 14812
14813         \cs_if_exist:cT 14813
14814         { \c__keys_default_root_str ##1 / \l_keys_key_str } 14814
14815         { 14815
14816             \tl_set_eq:Nc 14816
14817             \l_keys_value_tl 14817
14818             { \c__keys_default_root_str ##1 / \l_keys_key_str } 14818
14819             \clist_map_break: 14819
14820         } 14820
14821     } 14821
14822 } 14822
14823 \cs_new_protected:Npn \__keys_execute: 14823
14824 { 14824
14825     \cs_if_exist:cTF { \c__keys_code_root_str \l_keys_path_str } 14825
14826     { 14826
14827         \cs_if_exist_use:c { \c__keys_check_root_str \l_keys_path_str } 14827
14828         \__keys_execute:no \l_keys_path_str \l_keys_value_tl 14828
14829     } 14829
14830     { 14830
14831         \cs_if_exist:cTF 14831
14832         { \c__keys_inherit_root_str \__keys_parent:o \l_keys_path_str } 14832
14833         { \__keys_execute_inherit: } 14833
14834         { \__keys_execute_unknown: } 14834
14835     } 14835
14836 } 14836
14837 \cs_new_protected:Npn \__keys_execute_inherit: 14837
14838 { 14838
14839     \bool_set_false:N \l__keys_inherit_bool 14839
14840     \clist_map_inline:cn 14840
14841     { \c__keys_inherit_root_str \__keys_parent:o \l_keys_path_str } 14841
14842     { \__keys_excute_inherit:n {##1} } 14842
14843     \bool_if:NF \l__keys_inherit_bool 14843
14844     { \__keys_execute_unknown: } 14844
14845 } 14845
14846 \cs_new_protected:Npn \__keys_excute_inherit:n #1 14846
14847 { 14847
```

```
14848 \cs_if_exist:cTF
14849 { \c__keys_inherit_root_str #1 }
14850 {
14851   \clist_map_inline:cn { \c__keys_inherit_root_str #1 }
14852     { \__keys_excute_inherit:n {##1} }
14853   \bool_if:NT \l__keys_inherit_bool
14854     { \clist_map_break: }
14855 }
14856 {
14857   \cs_if_exist:cT
14858     { \c__keys_code_root_str #1 / \l_keys_key_str }
14859     {
14860       \str_set:Nn \l__keys_inherit_str {#1}
14861       \cs_if_exist_use:c { \c__keys_check_root_str #1 / \l_keys_key_str }
14862       \__keys_execute:no { #1 / \l_keys_key_str } \l_keys_value_tl
14863       \clist_map_break:n
14864         { \bool_set_true:N \l__keys_inherit_bool }
14865     }
14866 }
14867 }
14868 \cs_new_protected:Npn \__keys_execute_unknown:
14869 {
14870   \bool_if:NTF \l__keys_only_known_bool
14871     { \__keys_store_unused: }
14872     {
14873       \cs_if_exist:cTF
14874         { \c__keys_code_root_str \l__keys_module_str / unknown }
14875         {
14876           \bool_if:NT \l__keys_no_value_bool
14877             {
14878               \cs_if_exist:cT
14879                 { \c__keys_default_root_str \l__keys_module_str / unknown }
14880                 {
14881                   \tl_set_eq:Nc
14882                     \l_keys_value_tl
14883                     { \c__keys_default_root_str \l__keys_module_str / unknown }
14884                 }
14885             }
14886           \__keys_execute:no { \l__keys_module_str / unknown } \l_keys_value_tl
14887         }
14888       {
14889         \msg_error:nnee { keys } { unknown }
14890         \l_keys_path_str \l__keys_module_str
14891       }
14892     }
14893 }
```

```
14894 \cs_new:Npn \__keys_execute:nn #1#2 14894
14895 { \__keys_execute:no {#1} { \prg_do_nothing: #2 } } 14895
14896 \cs_new:Npn \__keys_execute:no #1#2 14896
14897 { 14897
14898 \exp_args:NNo \exp_args:No \use:n 14898
14899 { 14899
14900 \cs:w \c__keys_code_root_str #1 \exp_after:wN \cs_end: 14900
14901 \exp_after:wN {#2} 14901
14902 } 14902
14903 } 14903
14904 \cs_new_protected:Npn \__keys_store_unused: 14904
14905 { 14905
14906 \__keys_quark_if_no_value:NTF \l__keys_relative_tl 14906
14907 { 14907
14908 \clist_put_right:Ne \l__keys_unused_clist 14908
14909 { 14909
14910 \l_keys_key_str 14910
14911 \bool_if:NF \l__keys_no_value_bool 14911
14912 { = { \exp_not:o \l_keys_value_tl } } 14912
14913 } 14913
14914 } 14914
14915 { 14915
14916 \tl_if_empty:NTF \l__keys_relative_tl 14916
14917 { 14917
14918 \clist_put_right:Ne \l__keys_unused_clist 14918
14919 { 14919
14920 \l_keys_path_str 14920
14921 \bool_if:NF \l__keys_no_value_bool 14921
14922 { = { \exp_not:o \l_keys_value_tl } } 14922
14923 } 14923
14924 } 14924
14925 { \__keys_store_unused_aux: } 14925
14926 } 14926
14927 } 14927
14928 \cs_new_protected:Npn \__keys_store_unused_aux: 14928
14929 { 14929
14930 \__kernel_tl_set:Nx \l__keys_relative_tl 14930
14931 { \exp_args:No \__keys_trim_spaces:n \l__keys_relative_tl } 14931
14932 \use:e 14932
14933 { 14933
14934 \cs_set_protected:Npn \__keys_store_unused:w 14934
14935 ##1 \l__keys_relative_tl / 14935
14936 ##2 \l__keys_relative_tl / 14936
14937 ##3 \s__keys_stop 14937
14938 } 14938
14939 { 14939
```

```
14940         \tl_if_blank:nF {##1}
14941     {
14942         \msg_error:nnee { keys } { bad-relative-key-path }
14943         \l_keys_path_str
14944         \l__keys_relative_tl
14945     }
14946     \clist_put_right:Ne \l__keys_unused_clist
14947     {
14948         \exp_not:n {##2}
14949         \bool_if:NF \l__keys_no_value_bool
14950         { = { \exp_not:o \l_keys_value_tl } }
14951     }
14952 }
14953 \use:e
14954 {
14955     \__keys_store_unused:w \l_keys_path_str
14956     \l_keys_relative_tl / \l_keys_relative_tl /
14957     \s_keys_stop
14958 }
14959 }
14960 \cs_new_protected:Npn \__keys_store_unused:w { }
14961 \cs_new:Npn \__keys_choice_find:n #1
14962 {
14963     \str_if_empty:NTF \l__keys_inherit_str
14964     { \__keys_choice_find:nn \l_keys_path_str {#1} }
14965     {
14966         \__keys_choice_find:nn
14967         { \l_keys_inherit_str / \l_keys_key_str } {#1}
14968     }
14969 }
14970 \cs_new:Npn \__keys_choice_find:nn #1#2
14971 {
14972     \cs_if_exist:cTF { \c__keys_code_root_str #1 / \__keys_trim_spaces:n {#2} }
14973     { \__keys_execute:nn { #1 / \__keys_trim_spaces:n {#2} } {#2} }
14974     { \__keys_execute:nn { #1 / unknown } {#2} }
14975 }
14976 \cs_new:Npn \__keys_multichoice_find:n #1
14977 { \clist_map_function:nN {#1} \__keys_choice_find:n }
14978 \cs_new:Npn \__keys_parent:o #1
14979 {
14980     \exp_after:wN \__keys_parent_auxi:w #1 \q_nil \__keys_parent_auxii:w
14981     / \q_nil \__keys_parent_auxiv:w
14982 }
14983 \cs_new:Npn \__keys_parent_auxi:w #1 / #2 \q_nil #3
14984 {
14985     #3 { #1 } #2 \q_nil #3
```

14986	}	14986
14987	\cs_new:Npn __keys_parent_auxii:w #1 #2 \q_nil __keys_parent_auxii:w	14987
14988	{	14988
14989	#1 __keys_parent_auxi:w #2 \q_nil __keys_parent_auxiii:n	14989
14990	}	14990
14991	\cs_new:Npn __keys_parent_auxiii:n #1	14991
14992	{	14992
14993	/ #1 __keys_parent_auxi:w	14993
14994	}	14994
14995	\cs_new:Npn __keys_parent_auxiv:w #1 \q_nil __keys_parent_auxiv:w	14995
14996	{	14996
14997	}	14997
14998	\group_begin:	14998
14999	\cs_set:Npn __keys_tmp:w #1	14999
15000	{	15000
15001	\cs_new:Npn __keys_trim_spaces:n ##1	15001
15002	{	15002
15003	\exp_after:wN __keys_trim_spaces_auxi:w \tl_to_str:n { / ##1 } /	15003
15004	\s_keys_nil __keys_trim_spaces_auxi:w	15004
15005	\s_keys_mark __keys_trim_spaces_auxii:w	15005
15006	#1 / #1	15006
15007	\s_keys_nil __keys_trim_spaces_auxii:w	15007
15008	\s_keys_mark __keys_trim_spaces_auxiii:w	15008
15009	}	15009
15010	}	15010
15011	__keys_tmp:w { ~ }	15011
15012	\group_end:	15012
15013	\cs_new:Npn __keys_trim_spaces_auxi:w #1 ~ / #2 \s_keys_nil #3	15013
15014	{	15014
15015	#3 #1 / #2 \s_keys_nil #3	15015
15016	}	15016
15017	\cs_new:Npn __keys_trim_spaces_auxii:w #1 / ~ #2 \s_keys_mark #3	15017
15018	{	15018
15019	#3 #1 / #2 \s_keys_mark #3	15019
15020	}	15020
15021	\cs_new:Npn __keys_trim_spaces_auxiii:w	15021
15022	/ #1 /	15022
15023	\s_keys_nil __keys_trim_spaces_auxi:w	15023
15024	\s_keys_mark __keys_trim_spaces_auxii:w	15024
15025	/	15025
15026	\s_keys_nil __keys_trim_spaces_auxii:w	15026
15027	\s_keys_mark __keys_trim_spaces_auxiii:w	15027
15028	{	15028
15029	#1	15029
15030	}	15030
15031	\prg_new_conditional:Npnn \keys_if_exist:nn #1#2 { p , T , F , TF }	15031

15032	{	15032
15033	__keys_if_exist:ee	15033
15034	{ __keys_trim_spaces:n {#1} }	15034
15035	{ __keys_trim_spaces:n {#2} }	15035
15036	}	15036
15037	\prg_generate_conditional_variant:Nnn \keys_if_exist:nn { ne } { p , T , F , TF }	15037
15038	\cs_new:Npn __keys_if_exist:nn #1#2	15038
15039	{	15039
15040	\cs_if_exist:cTF	15040
15041	{ \c__keys_code_root_str #1 \tl_if_blank:nF {#1} { / } #2 }	15041
15042	{ \prg_return_true: }	15042
15043	{ \prg_return_false: }	15043
15044	}	15044
15045	\cs_generate_variant:Nn __keys_if_exist:nn { ee }	15045
15046	\prg_new_conditional:Npnn \keys_if_choice_exist:nnn #1#2#3	15046
15047	{ p , T , F , TF }	15047
15048	{	15048
15049	__keys_if_exist:eee	15049
15050	{ __keys_trim_spaces:n {#1} }	15050
15051	{ __keys_trim_spaces:n {#2} }	15051
15052	{ __keys_trim_spaces:n {#3} }	15052
15053	}	15053
15054	\cs_new:Npn __keys_if_exist:nnn #1#2#3	15054
15055	{	15055
15056	\cs_if_exist:cTF	15056
15057	{	15057
15058	\c__keys_code_root_str	15058
15059	#1 \tl_if_blank:nF {#1} { / }	15059
15060	#2 \tl_if_blank:nF {#2} { / }	15060
15061	#3	15061
15062	}	15062
15063	{ \prg_return_true: }	15063
15064	{ \prg_return_false: }	15064
15065	}	15065
15066	\cs_generate_variant:Nn __keys_if_exist:nnn { eee }	15066
15067	\cs_new_protected:Npn \keys_show:nn	15067
15068	{ __keys_show:Nnn \msg_show:nneeee }	15068
15069	\cs_new_protected:Npn \keys_log:nn	15069
15070	{ __keys_show:Nnn \msg_log:nneeee }	15070
15071	\cs_new_protected:Npn __keys_show:Nnn #1#2#3	15071
15072	{	15072
15073	__keys_show_aux:Nee	15073
15074	#1	15074
15075	{ __keys_trim_spaces:n {#2} }	15075
15076	{ __keys_trim_spaces:n {#3} }	15076
15077	}	15077


```
15078 \cs_new_protected:Npn \__keys_show_aux:Nnn #1#2#3
15079 {
15080     #1 { keys } { show-key }
15081     { #2 \tl_if_blank:nF {#2} { / } #3 }
15082     {
15083         \keys_if_exist:nnT {#2} {#3}
15084         {
15085             \exp_args:Nnf \msg_show_item_unbraced:nn { code }
15086             {
15087                 \exp_args:Ne \__keys_show:n
15088                 {
15089                     \exp_args:Nc \cs_replacement_spec:N
15090                     {
15091                         \c__keys_code_root_str
15092                         #2 \tl_if_blank:nF {#2} { / } #3
15093                     }
15094                 }
15095             }
15096         }
15097     } { } { }
15098 }
15099
15100 \cs_generate_variant:Nn \__keys_show_aux:Nnn { Nee }
15101 \cs_new:Npe \__keys_show:n #1
15102 {
15103     \exp_not:N \__keys_show:w
15104     #1
15105     \tl_to_str:n { \__keys_precompile:n }
15106     #1
15107     \tl_to_str:n { \__keys_precompile:n }
15108     \exp_not:N \s__keys_stop
15109 }
15110 \use:e
15111 {
15112     \cs_new:Npn \exp_not:N \__keys_show:w
15113     #1 \tl_to_str:n { \__keys_precompile:n }
15114     #2 \tl_to_str:n { \__keys_precompile:n }
15115     #3 \exp_not:N \s__keys_stop
15116 }
15117 {
15118     \tl_if_blank:nTF {#2}
15119     {#1}
15120     { \__keys_show:Nw #2 \s__keys_stop }
15121 }
15122 \use:e
15123 {
```

```
15124 \cs_new:Npn \exp_not:N \__keys_show:Nw #1#2 15124
15125 \c_right_brace_str \exp_not:N \s__keys_stop 15125
15126 } 15126
15127 {#2} 15127
15128 \msg_new:nnnn { keys } { bad-relative-key-path } 15128
15129 { The~key~'#1'~is~not~inside~the~'#2'~path. } 15129
15130 { The~key~'#1'~cannot~be~expressed~relative~to~path~'#2'. } 15130
15131 \msg_new:nnnn { keys } { boolean-values-only } 15131
15132 { Key~'#1'~accepts~boolean~values~only. } 15132
15133 { The~key~'#1'~only~accepts~the~values~'true'~and~'false'. } 15133
15134 \msg_new:nnnn { keys } { choice-unknown } 15134
15135 { Key~'#1'~accepts~only~a~fixed~set~of~choices. } 15135
15136 { 15136
15137 The~key~'#1'~only~accepts~predefined~values,~ 15137
15138 and~'#2'~is~not~one~of~these. 15138
15139 } 15139
15140 \msg_new:nnnn { keys } { unknown } 15140
15141 { The~key~'#1'~is~unknown~and~is~being~ignored. } 15141
15142 { 15142
15143 The~module~'#2'~does~not~have~a~key~called~'#1'.\\ 15143
15144 Check~that~you~have~spelled~the~key~name~correctly. 15144
15145 } 15145
15146 \msg_new:nnnn { keys } { nested-choice-key } 15146
15147 { Attempt~to~define~'#1'~as~a~nested~choice~key. } 15147
15148 { 15148
15149 The~key~'#1'~cannot~be~defined~as~a~choice~as~the~parent~key~'#2'~is~ 15149
15150 itself~a~choice. 15150
15151 } 15151
15152 \msg_new:nnnn { keys } { value-forbidden } 15152
15153 { The~key~'#1'~does~not~take~a~value. } 15153
15154 { 15154
15155 The~key~'#1'~should~be~given~without~a~value.\\ 15155
15156 The~value~'#2'~was~present:~the~key~will~be~ignored. 15156
15157 } 15157
15158 \msg_new:nnnn { keys } { value-required } 15158
15159 { The~key~'#1'~requires~a~value. } 15159
15160 { 15160
15161 The~key~'#1'~must~have~a~value.\\ 15161
15162 No~value~was~present:~the~key~will~be~ignored. 15162
15163 } 15163
15164 \msg_new:nnn { keys } { show-key } 15164
15165 { 15165
15166 The~key~#1~ 15166
15167 \tl_if_empty:nTF {#2} 15167
15168 { is~undefined. } 15168
15169 { has~the~properties: #2 . } 15169
```

```

15170 }
15171 \prop_gput:Nnn \g_msg_module_name_prop { keys } { LaTeX }
15172 \prop_gput:Nnn \g_msg_module_type_prop { keys } { }
15173 %% File: l3intarray.dtx
15174 \msg_new:nnn { kernel } { negative-array-size }
15175 { Size-of-array-may-not-be-negative:~#1 }
15176 \cs_new_eq:NN \__intarray_sep: \__kernel_int_sep:
15177 \int_new:N \l__intarray_loop_int
15178 \cs_if_exist:NTF \__intarray_gset_count:Nw
15179 {
15180 \int_new:N \g__intarray_table_int
15181 \int_new:N \l__intarray_bad_index_int
15182 \cs_new_protected:Npn \__intarray_new:N #1
15183 {
15184 \__kernel_chk_if_free_cs:N #1
15185 \int_gincr:N \g__intarray_table_int
15186 \cs_gset_nopar:Npe #1 { \__intarray:w \int_use:N \g__intarray_table_int
\c_space_tl }
15187 }
15188 \cs_new_protected:Npn \intarray_new:Nn #1#2
15189 {
15190 \__intarray_new:N #1
15191 \__intarray_gset_count:Nw #1 \int_eval:n {#2} \scan_stop:
15192 \int_compare:nNnT { \intarray_count:N #1 } < 0
15193 {
15194 \msg_error:nne { kernel } { negative-array-size }
15195 { \intarray_count:N #1 }
15196 }
15197 }
15198 \cs_generate_variant:Nn \intarray_new:Nn { c }
15199 \cs_generate_variant:Nn \intarray_count:N { c }
15200 \cs_new_protected:Npn \__kernel_intarray_gset:Nnn #1#2#3
15201 { \__intarray_gset:w #2 #1 #3 \scan_stop: }
15202 \cs_new_protected:Npn \intarray_gset:Nnn #1#2#3
15203 {
15204 \__intarray_gset:wF \int_eval:n {#2} #1 \int_eval:n{#3}
15205 {
15206 \msg_error:nneee { kernel } { out-of-bounds }
15207 { \token_to_str:N #1 } { \int_use:N \l__intarray_bad_index_int } {
\intarray_count:N #1 }
15208 }
15209 }
15210 \cs_generate_variant:Nn \intarray_gset:Nnn { c }
15211 \cs_generate_variant:Nn \intarray_gzero:N { c }
15212 \cs_new:Npn \__kernel_intarray_item:Nn #1#2
15213 { \__intarray_item:w #2 #1 }

```

```
15214 \cs_new:Npn \intarray_item:Nn #1#2 15214
15215 { 15215
15216 \__intarray_item:wF \int_eval:n {#2} #1 15216
15217 { 15217
15218 \msg_expandable_error:nnfff { kernel } { out-of-bounds } 15218
15219 { \token_to_str:N #1 } { \int_use:N \l__intarray_bad_index_int } { 15219 ✓
\intarray_count:N #1 }
15220 0 15220
15221 } 15221
15222 } 15222
15223 \cs_generate_variant:Nn \intarray_item:Nn { c } 15223
15224 \cs_new:Npn \intarray_rand_item:N #1 15224
15225 { \intarray_item:Nn #1 { \int_rand:n { \intarray_count:N #1 } } } 15225
15226 \cs_generate_variant:Nn \intarray_rand_item:N { c } 15226
15227 \cs_new_protected:Npn \intarray_const_from_clist:Nn #1#2 15227
15228 { 15228
15229 \__intarray_new:N #1 15229
15230 \int_zero:N \l__intarray_loop_int 15230
15231 \clist_map_inline:nn {#2} 15231
15232 { 15232
15233 \int_incr:N \l__intarray_loop_int 15233
15234 \__kernel_intarray_gset:Nnn #1 \l__intarray_loop_int { \int_eval:n {##1} } } 15234
15235 } 15235
15236 \cs_generate_variant:Nn \intarray_const_from_clist:Nn { c } 15236
15237 \cs_new:Npn \__kernel_intarray_range_to_clist:Nnn #1#2#3 15237
15238 { 15238
15239 \__intarray_range_to_clist:w #1 15239
15240 \int_eval:n {#2} ~ \int_eval:n {#3} ~ 15240
15241 } 15241
15242 \cs_new_protected:Npn \__kernel_intarray_gset_range_from_clist:Nnn #1#2#3 15242
15243 { 15243
15244 \__intarray_gset_range:w \int_eval:w #2 #1 #3 , , \scan_stop: 15244
15245 } 15245
15246 \cs_new_protected:Npn \__intarray_gset_overflow_test:nw #1 15246
15247 { 15247
15248 } 15248
15249 } 15249
15250 { 15250
15251 \cs_new_eq:NN \__intarray_entry:w \tex_fontdimen:D 15251
15252 \cs_new_eq:NN \__intarray_count:w \tex_hyphenchar:D 15252
15253 \dim_const:Nn \c__intarray_sp_dim { 1 sp } 15253
15254 \int_new:N \g__intarray_font_int 15254
15255 \cs_new_protected:Npn \__intarray_new:N #1 15255
15256 { 15256
15257 \__kernel_chk_if_free_cs:N #1 15257
15258 \int_gincr:N \g__intarray_font_int 15258
```

```

15259 \tex_global:D\tex_font:D #1
15260 = cmr10~at~ \g__intarray_font_int \c__intarray_sp_dim \scan_stop:
15261 \int_step_inline:nn { 8 }
15262 { \__kernel_intarray_gset:Nnn #1 {##1} \c_zero_int }
15263 }
15264 \cs_new_protected:Npn \intarray_new:Nn #1#2
15265 {
15266 \__intarray_new:N #1
15267 \__intarray_count:w #1 = \int_eval:n {#2} \scan_stop:
15268 \int_compare:nNnT { \intarray_count:N #1 } < 0
15269 {
15270 \msg_error:nne { kernel } { negative-array-size }
15271 { \intarray_count:N #1 }
15272 }
15273 \int_compare:nNnT { \intarray_count:N #1 } > 0
15274 { \__kernel_intarray_gset:Nnn #1 { \intarray_count:N #1 } { 0 } }
15275 }
15276 \cs_generate_variant:Nn \intarray_new:Nn { c }
15277 \cs_new:Npn \intarray_count:N #1 { \int_value:w \__intarray_count:w #1 }
15278 \cs_generate_variant:Nn \intarray_count:N { c }
15279 \cs_new:Npn \__intarray_signed_max_dim:n #1
15280 { \int_value:w \int_compare:nNnT {#1} < 0 { - } \c_max_dim }
15281 \cs_new:Npn \__intarray_bounds:NNnTF #1#2#3
15282 {
15283 \if_int_compare:w 1 > #3 \exp_stop_f:
15284 \__intarray_bounds_error:NNnw #1 #2 {#3}
15285 \else:
15286 \if_int_compare:w #3 > \intarray_count:N #2 \exp_stop_f:
15287 \__intarray_bounds_error:NNnw #1 #2 {#3}
15288 \fi:
15289 \fi:
15290 \use_i:nn
15291 }
15292 \cs_new:Npn \__intarray_bounds_error:NNnw #1#2#3#4 \use_i:nn #5#6
15293 {
15294 #4
15295 #1 { kernel } { out-of-bounds }
15296 { \token_to_str:N #2 } {#3} { \intarray_count:N #2 }
15297 #6
15298 }
15299 \cs_new_protected:Npn \__kernel_intarray_gset:Nnn #1#2#3
15300 { \__intarray_entry:w #2 #1 #3 \c__intarray_sp_dim }
15301 \cs_new_protected:Npn \intarray_gset:Nnn #1#2#3
15302 {
15303 \exp_after:wN \__intarray_gset:Nww
15304 \exp_after:wN #1

```

```
15305 \int_value:w \int_eval:n {#2} \exp_after:wN \__intarray_sep: 15305
15306 \int_value:w \int_eval:n {#3} \__intarray_sep: 15306
15307 } 15307
15308 \cs_generate_variant:Nn \intarray_gset:Nnn { c } 15308
15309 \cs_new_protected:Npn \__intarray_gset:Nww #1#2 \__intarray_sep: #3 \__intarray_sep: 15309
15310 { 15310
15311 \__intarray_bounds:NNnTF \msg_error:nneee #1 {#2} 15311
15312 { 15312
15313 \__intarray_gset_overflow_test:nw {#3} 15313
15314 \__kernel_intarray_gset:Nnn #1 {#2} {#3} 15314
15315 } 15315
15316 { } 15316
15317 } 15317
15318 \cs_if_exist:NTF \tex_ifabsnum:D 15318
15319 { 15319
15320 \cs_new_protected:Npn \__intarray_gset_overflow_test:nw #1 15320
15321 { 15321
15322 \tex_ifabsnum:D #1 > \c_max_dim 15322
15323 \exp_after:wN \__intarray_gset_overflow:NNnn 15323
15324 \fi: 15324
15325 } 15325
15326 } 15326
15327 { 15327
15328 \cs_new_protected:Npn \__intarray_gset_overflow_test:nw #1 15328
15329 { 15329
15330 \if_int_compare:w \int_abs:n {#1} > \c_max_dim 15330
15331 \exp_after:wN \__intarray_gset_overflow:NNnn 15331
15332 \fi: 15332
15333 } 15333
15334 } 15334
15335 \cs_new_protected:Npn \__intarray_gset_overflow:NNnn #1#2#3#4 15335
15336 { 15336
15337 \msg_error:nneeee { kernel } { overflow } 15337
15338 { \token_to_str:N #2 } {#3} {#4} { \__intarray_signed_max_dim:n {#4} } 15338
15339 #1 #2 {#3} { \__intarray_signed_max_dim:n {#4} } 15339
15340 } 15340
15341 \cs_new_protected:Npn \intarray_gzero:N #1 15341
15342 { 15342
15343 \int_zero:N \l__intarray_loop_int 15343
15344 \prg_replicate:nn { \intarray_count:N #1 } 15344
15345 { 15345
15346 \int_incr:N \l__intarray_loop_int 15346
15347 \__intarray_entry:w \l__intarray_loop_int #1 \c_zero_dim 15347
15348 } 15348
15349 } 15349
15350 \cs_generate_variant:Nn \intarray_gzero:N { c } 15350
```

```
15351 \cs_new:Npn \__kernel_intarray_item:Nn #1#2 15351
15352 { \int_value:w \__intarray_entry:w #2 #1 } 15352
15353 \cs_new:Npn \intarray_item:Nn #1#2 15353
15354 { 15354
15355 \exp_after:wN \__intarray_item:Nw 15355
15356 \exp_after:wN #1 15356
15357 \int_value:w \int_eval:n {#2} \__intarray_sep: 15357
15358 } 15358
15359 \cs_generate_variant:Nn \intarray_item:Nn { c } 15359
15360 \cs_new:Npn \__intarray_item:Nw #1#2 \__intarray_sep: 15360
15361 { 15361
15362 \__intarray_bounds:NNnTF \msg_expandable_error:nnfff #1 {#2} 15362
15363 { \__kernel_intarray_item:Nn #1 {#2} } 15363
15364 { 0 } 15364
15365 } 15365
15366 \cs_new:Npn \intarray_rand_item:N #1 15366
15367 { \intarray_item:Nn #1 { \int_rand:n { \intarray_count:N #1 } } } 15367
15368 \cs_generate_variant:Nn \intarray_rand_item:N { c } 15368
15369 \cs_new_protected:Npn \intarray_const_from_clist:Nn #1#2 15369
15370 { 15370
15371 \__intarray_new:N #1 15371
15372 \int_zero:N \l__intarray_loop_int 15372
15373 \clist_map_inline:nn {#2} 15373
15374 { \exp_args:Nf \__intarray_const_from_clist:nN { \int_eval:n {##1} } #1 } 15374
15375 \__intarray_count:w #1 \l__intarray_loop_int 15375
15376 } 15376
15377 \cs_generate_variant:Nn \intarray_const_from_clist:Nn { c } 15377
15378 \cs_new_protected:Npn \__intarray_const_from_clist:nN #1#2 15378
15379 { 15379
15380 \int_incr:N \l__intarray_loop_int 15380
15381 \__intarray_gset_overflow_test:nw {#1} 15381
15382 \__kernel_intarray_gset:Nnn #2 \l__intarray_loop_int {#1} 15382
15383 } 15383
15384 \cs_new:Npn \__intarray_to_clist:Nn #1#2 15384
15385 { 15385
15386 \int_compare:nNnF { \intarray_count:N #1 } = \c_zero_int 15386
15387 { 15387
15388 \exp_last_unbraced:Nf \use_none:n 15388
15389 { \__intarray_to_clist:w 1 \__intarray_sep: #1 {#2} \prg_break_point: } 15389
15390 } 15390
15391 } 15391
15392 \cs_new:Npn \__intarray_to_clist:w #1 \__intarray_sep: #2#3 15392
15393 { 15393
15394 \if_int_compare:w #1 > \__intarray_count:w #2 15394
15395 \prg_break:n 15395
15396 \fi: 15396
```


15397	#3 __kernel_intarray_item:Nn #2 {#1}	15397
15398	\exp_after:wN __intarray_to_clist:w	15398
15399	\int_value:w \int_eval:w #1 + \c_one_int __intarray_sep: #2 {#3}	15399
15400	}	15400
15401	\cs_new:Npn __kernel_intarray_range_to_clist:Nnn #1#2#3	15401
15402	{	15402
15403	\exp_last_unbraced:Nf \use_none:n	15403
15404	{	15404
15405	\exp_after:wN __intarray_range_to_clist:ww	15405
15406	\int_value:w \int_eval:w #2 \exp_after:wN __intarray_sep:	15406
15407	\int_value:w \int_eval:w #3 __intarray_sep:	15407
15408	#1 \prg_break_point:	15408
15409	}	15409
15410	}	15410
15411	\cs_new:Npn __intarray_range_to_clist:ww #1 __intarray_sep: #2 __intarray_sep: #3	15411
15412	{	15412
15413	\if_int_compare:w #1 > #2 \exp_stop_f:	15413
15414	\prg_break:n	15414
15415	\fi:	15415
15416	, __kernel_intarray_item:Nn #3 {#1}	15416
15417	\exp_after:wN __intarray_range_to_clist:ww	15417
15418	\int_value:w \int_eval:w #1 + \c_one_int __intarray_sep: #2 __intarray_sep: #3	15418
15419	}	15419
15420	\cs_new_protected:Npn __kernel_intarray_gset_range_from_clist:Nnn #1#2#3	15420
15421	{	15421
15422	\int_set:Nn \l__intarray_loop_int {#2}	15422
15423	__intarray_gset_range:Nw #1 #3 , , \prg_break_point:	15423
15424	}	15424
15425	\cs_new_protected:Npn __intarray_gset_range:Nw #1 #2 ,	15425
15426	{	15426
15427	\if_catcode:w \scan_stop: \tl_to_str:n {#2} \scan_stop:	15427
15428	\prg_break:n	15428
15429	\fi:	15429
15430	__kernel_intarray_gset:Nnn #1 \l__intarray_loop_int {#2}	15430
15431	\int_incr:N \l__intarray_loop_int	15431
15432	__intarray_gset_range:Nw #1	15432
15433	}	15433
15434	}	15434
15435	\prg_new_eq_conditional:NNn \intarray_if_exist:N \cs_if_exist:N	15435
15436	{ TF , T , F , p }	15436
15437	\prg_new_eq_conditional:NNn \intarray_if_exist:c \cs_if_exist:c	15437
15438	{ TF , T , F , p }	15438
15439	\cs_new_protected:Npn \intarray_show:N { __intarray_show:NN \msg_show:nneeee }	15439
15440	\cs_generate_variant:Nn \intarray_show:N { c }	15440
15441	\cs_new_protected:Npn \intarray_log:N { __intarray_show:NN \msg_log:nneeee }	15441
15442	\cs_generate_variant:Nn \intarray_log:N { c }	15442

```
15443 \cs_new_protected:Npn \__intarray_show:NN #1#2
15444 {
15445     \__kernel_chk_defined:NT #2
15446     {
15447         #1 { intarray } { show }
15448         { \token_to_str:N #2 }
15449         { \intarray_count:N #2 }
15450         { >~ \__intarray_to_clist:Nn #2 { , ~ } }
15451         { }
15452     }
15453 }
15454 %% File: l3fp.dtx
15455 %% File: l3fp-aux.dtx
15456 \cs_new_eq:NN \__fp_int_eval:w \tex_numexpr:D
15457 \cs_new_eq:NN \__fp_int_eval_end: \scan_stop:
15458 \cs_new_eq:NN \__fp_int_to_roman:w \tex_romannumeral:D
15459 \cs_new_eq:NN \__fp_sep: \__kernel_int_sep:
15460 \cs_new:Npn \__fp_use_none_stop_f:n #1 { \exp_stop_f: }
15461 \cs_new:Npn \__fp_use_s:n #1 { #1\__fp_sep: }
15462 \cs_new:Npn \__fp_use_s:nn #1#2 { #1#2\__fp_sep: }
15463 \cs_new:Npn \__fp_use_none_until_s:w #1\__fp_sep: { }
15464 \cs_new:Npn \__fp_use_i_until_s:nw #1#2\__fp_sep: {#1}
15465 \cs_new:Npn \__fp_use_ii_until_s:nnw #1#2#3\__fp_sep: {#2}
15466 \cs_new:Npn \__fp_reverse_args:Nww #1 #2\__fp_sep: #3\__fp_sep:
15467 { #1 #3\__fp_sep: #2\__fp_sep: }
15468 \cs_new:Npn \__fp_rrot:www #1\__fp_sep: #2\__fp_sep: #3\__fp_sep:
15469 { #2\__fp_sep: #3\__fp_sep: #1\__fp_sep: }
15470 \cs_new:Npn \__fp_use_i:ww #1\__fp_sep: #2\__fp_sep: { #1\__fp_sep: }
15471 \cs_new:Npn \__fp_use_i:www #1\__fp_sep: #2\__fp_sep: #3\__fp_sep: { #1\__fp_sep: }
15472 \cs_new_protected:Npn \__fp_misused:n #1
15473 { \msg_error:nne { fp } { misused } { \fp_to_tl:n {#1} } }
15474 \scan_new:N \s__fp
15475 \cs_new_protected:Npn \__fp_chk:w #1 \__fp_sep:
15476 { \__fp_misused:n { \s__fp \__fp_chk:w #1 \__fp_sep: } }
15477 \scan_new:N \s__fp_expr_mark
15478 \scan_new:N \s__fp_expr_stop
15479 \scan_new:N \s__fp_mark
15480 \scan_new:N \s__fp_stop
15481 \cs_new:Npn \__fp_use_i_delimit_by_s_stop:nw #1 #2 \s__fp_stop {#1}
15482 \scan_new:N \s__fp_invalid
15483 \scan_new:N \s__fp_underflow
15484 \scan_new:N \s__fp_overflow
15485 \scan_new:N \s__fp_division
15486 \scan_new:N \s__fp_exact
15487 \tl_const:Nn \c_zero_fp { \s__fp \__fp_chk:w 0 0 \s__fp_exact \__fp_sep: }
15488 \tl_const:Nn \c_minus_zero_fp { \s__fp \__fp_chk:w 0 2 \s__fp_exact \__fp_sep: }
```

```
15489 \tl_const:Nn \c_inf_fp { \s__fp \__fp_chk:w 2 0 \s__fp_exact \__fp_sep: } 15489
15490 \tl_const:Nn \c_minus_inf_fp { \s__fp \__fp_chk:w 2 2 \s__fp_exact \__fp_sep: } 15490
15491 \tl_const:Nn \c_nan_fp { \s__fp \__fp_chk:w 3 1 \s__fp_exact \__fp_sep: } 15491
15492 \int_const:Nn \c__fp_prec_int { 16 } 15492
15493 \int_const:Nn \c__fp_half_prec_int { 8 } 15493
15494 \int_const:Nn \c__fp_block_int { 4 } 15494
15495 \int_const:Nn \c__fp_myriad_int { 10000 } 15495
15496 \int_const:Nn \c__fp_minus_min_exponent_int { 10000 } 15496
15497 \int_const:Nn \c__fp_max_exponent_int { 10000 } 15497
15498 \int_const:Nn \c__fp_max_exp_exponent_int { 5 } 15498
15499 \tl_const:Ne \c__fp_overflowing_fp 15499
15500 { 15500
15501 \s__fp \__fp_chk:w 1 0 15501
15502 { \int_eval:n { \c__fp_max_exponent_int + 1 } } 15502
15503 {1000} {0000} {0000} {0000} \__fp_sep: 15503
15504 } 15504
15505 \cs_new:Npn \__fp_zero_fp:N #1 15505
15506 { \s__fp \__fp_chk:w 0 #1 \s__fp_underflow \__fp_sep: } 15506
15507 \cs_new:Npn \__fp_inf_fp:N #1 15507
15508 { \s__fp \__fp_chk:w 2 #1 \s__fp_overflow \__fp_sep: } 15508
15509 \cs_new:Npn \__fp_exponent:w \s__fp \__fp_chk:w #1 15509
15510 { 15510
15511 \if_meaning:w 1 #1 15511
15512 \exp_after:wN \__fp_use_ii_until_s:nnw 15512
15513 \else: 15513
15514 \exp_after:wN \__fp_use_i_until_s:nw 15514
15515 \exp_after:wN 0 15515
15516 \fi: 15516
15517 } 15517
15518 \cs_new:Npn \__fp_neg_sign:N #1 15518
15519 { \__fp_int_eval:w 2 - #1 \__fp_int_eval_end: } 15519
15520 \cs_new:Npn \__fp_kind:w #1 15520
15521 { 15521
15522 \__fp_if_type_fp:NTwFw 15522
15523 #1 \__fp_use_ii_until_s:nnw 15523
15524 \s__fp { \__fp_use_i_until_s:nw 4 } 15524
15525 \s__fp_stop 15525
15526 } 15526
15527 \cs_new:Npn \__fp_sanitize:Nw #1 #2\__fp_sep: 15527
15528 { 15528
15529 \if_case:w 15529
15530 \if_int_compare:w #2 > \c__fp_max_exponent_int 1 ~ \else: 15530
15531 \if_int_compare:w #2 < - \c__fp_minus_min_exponent_int 2 ~ \else: 15531
15532 \if_meaning:w 1 #1 3 ~ \fi: \fi: \fi: 0 ~ 15532
15533 \or: \exp_after:wN \__fp_overflow:w 15533
15534 \or: \exp_after:wN \__fp_underflow:w 15534
```

15535	\or: \exp_after:wN __fp_sanitiz	15535
15536	\fi:	15536
15537	\s__fp __fp_chk:w 1 #1 {#2}	15537
15538	}	15538
15539	\cs_new:Npn __fp_sanitiz:wN #1__fp_sep: #2 { __fp_sanitiz:Nw #2 #1__fp_sep: }	15539
15540	\cs_new:Npn __fp_sanitiz_zero:w \s__fp __fp_chk:w #1 #2 #3__fp_sep:	15540
15541	{ \c_zero_fp }	15541
15542	\cs_new:Npn __fp_exp_after_o:w \s__fp __fp_chk:w #1	15542
15543	{	15543
15544	\if_meaning:w 1 #1	15544
15545	\exp_after:wN __fp_exp_after_normal:nNNw	15545
15546	\else:	15546
15547	\exp_after:wN __fp_exp_after_special:nNNw	15547
15548	\fi:	15548
15549	{ }	15549
15550	#1	15550
15551	}	15551
15552	\cs_new:Npn __fp_exp_after_f:nw #1 \s__fp __fp_chk:w #2	15552
15553	{	15553
15554	\if_meaning:w 1 #2	15554
15555	\exp_after:wN __fp_exp_after_normal:nNNw	15555
15556	\else:	15556
15557	\exp_after:wN __fp_exp_after_special:nNNw	15557
15558	\fi:	15558
15559	{ \exp:w \exp_end_continue_f:w #1 }	15559
15560	#2	15560
15561	}	15561
15562	\cs_new:Npn __fp_exp_after_special:nNNw #1#2#3#4__fp_sep:	15562
15563	{	15563
15564	\exp_after:wN \s__fp	15564
15565	\exp_after:wN __fp_chk:w	15565
15566	\exp_after:wN #2	15566
15567	\exp_after:wN #3	15567
15568	\exp_after:wN #4	15568
15569	\exp_after:wN __fp_sep:	15569
15570	#1	15570
15571	}	15571
15572	\cs_new:Npn __fp_exp_after_normal:nNNw #1 1 #2 #3 #4#5#6#7__fp_sep:	15572
15573	{	15573
15574	\exp_after:wN __fp_exp_after_normal:Nwwwww	15574
15575	\exp_after:wN #2	15575
15576	\int_value:w #3 \exp_after:wN __fp_sep:	15576
15577	\int_value:w 1 #4 \exp_after:wN __fp_sep:	15577
15578	\int_value:w 1 #5 \exp_after:wN __fp_sep:	15578
15579	\int_value:w 1 #6 \exp_after:wN __fp_sep:	15579
15580	\int_value:w 1 #7 \exp_after:wN __fp_sep: #1	15580

15581	}	15581
15582	\cs_new:Npn __fp_exp_after_normal:Nwwwww	15582
15583	#1 #2__fp_sep: 1 #3 __fp_sep: 1 #4 __fp_sep: 1 #5 __fp_sep: 1 #6 __fp_sep:	15583
15584	{ \s__fp __fp_chk:w 1 #1 {#2} {#3} {#4} {#5} {#6} __fp_sep: }	15584
15585	\scan_new:N \s__fp_tuple	15585
15586	\cs_new_protected:Npn __fp_tuple_chk:w #1 __fp_sep:	15586
15587	{ __fp_misused:n { \s__fp_tuple __fp_tuple_chk:w #1 __fp_sep: } }	15587
15588	\tl_const:Nn \c__fp_empty_tuple_fp	15588
15589	{ \s__fp_tuple __fp_tuple_chk:w { } __fp_sep: }	15589
15590	\cs_new:Npn __fp_array_count:n #1	15590
15591	{ __fp_tuple_count:w \s__fp_tuple __fp_tuple_chk:w {#1} __fp_sep: }	15591
15592	\cs_new:Npn __fp_tuple_count:w \s__fp_tuple __fp_tuple_chk:w #1 __fp_sep:	15592
15593	{	15593
15594	\int_value:w __fp_int_eval:w 0	15594
15595	__fp_tuple_count_loop:Nw #1 { ? \prg_break: } __fp_sep:	15595
15596	\prg_break_point:	15596
15597	__fp_int_eval_end:	15597
15598	}	15598
15599	\cs_new:Npn __fp_tuple_count_loop:Nw #1#2__fp_sep:	15599
15600	{ \use_none:n #1 + 1 __fp_tuple_count_loop:Nw }	15600
15601	\cs_new:Npn __fp_if_type_fp:NTwFw #1 \s__fp #2 #3 \s__fp_stop {#2}	15601
15602	\cs_new:Npn __fp_array_if_all_fp:nTF #1	15602
15603	{	15603
15604	__fp_array_if_all_fp_loop:w #1 { \s__fp \prg_break: } __fp_sep:	15604
15605	\prg_break_point: \use_i:nn	15605
15606	}	15606
15607	\cs_new:Npn __fp_array_if_all_fp_loop:w #1#2 __fp_sep:	15607
15608	{	15608
15609	__fp_if_type_fp:NTwFw	15609
15610	#1 __fp_array_if_all_fp_loop:w	15610
15611	\s__fp { \prg_break:n \use_iii:nnn }	15611
15612	\s__fp_stop	15612
15613	}	15613
15614	\cs_new:Npn __fp_type_from_scan:N #1	15614
15615	{	15615
15616	__fp_if_type_fp:NTwFw	15616
15617	#1 { }	15617
15618	\s__fp { __fp_type_from_scan_other:N #1 }	15618
15619	\s__fp_stop	15619
15620	}	15620
15621	\cs_new:Npe __fp_type_from_scan_other:N #1	15621
15622	{	15622
15623	\exp_not:N \exp_after:wN \exp_not:N __fp_type_from_scan:w	15623
15624	\exp_not:N \token_to_str:N #1 \s__fp_mark	15624
15625	\tl_to_str:n { s__fp _? } \s__fp_mark \s__fp_stop	15625
15626	}	15626

15627	\exp_last_unbraced:NNNNNo	15627
15628	\cs_new:Npn __fp_type_from_scan:w #1	15628
15629	{ \tl_to_str:n { s__fp } } #2 \s__fp_mark #3 \s__fp_stop {#2}	15629
15630	\cs_new:Npn __fp_change_func_type:NNN #1#2#3	15630
15631	{	15631
15632	__fp_if_type_fp:NTwFw	15632
15633	#1 #2	15633
15634	\s__fp	15634
15635	{	15635
15636	\exp_after:wN __fp_change_func_type_chk:NNN	15636
15637	\cs:w	15637
15638	__fp __fp_type_from_scan_other:N #1	15638
15639	\exp_after:wN __fp_change_func_type_aux:w \token_to_str:N #2	15639
15640	\cs_end:	15640
15641	#2 #3	15641
15642	}	15642
15643	\s__fp_stop	15643
15644	}	15644
15645	\exp_last_unbraced:NNNNNo	15645
15646	\cs_new:Npn __fp_change_func_type_aux:w #1 { \tl_to_str:n { __fp } } { }	15646
15647	\cs_new:Npn __fp_change_func_type_chk:NNN #1#2#3	15647
15648	{	15648
15649	\if_meaning:w \scan_stop: #1	15649
15650	\exp_after:wN #3 \exp_after:wN #2	15650
15651	\else:	15651
15652	\exp_after:wN #1	15652
15653	\fi:	15653
15654	}	15654
15655	\cs_new:Npn __fp_exp_after_any_f:Nnw #1	15655
15656	{ \cs:w __fp_exp_after __fp_type_from_scan_other:N #1 _f:nw \cs_end: }	15656
15657	\cs_new:Npn __fp_exp_after_any_f:nw #1#2	15657
15658	{	15658
15659	__fp_if_type_fp:NTwFw	15659
15660	#2 __fp_exp_after_f:nw	15660
15661	\s__fp { __fp_exp_after_any_f:Nnw #2 }	15661
15662	\s__fp_stop	15662
15663	{#1} #2	15663
15664	}	15664
15665	\cs_new_eq:NN __fp_exp_after_expr_stop_f:nw \use_none:nn	15665
15666	\cs_new:Npn __fp_exp_after_tuple_o:w	15666
15667	{ __fp_exp_after_tuple_f:nw { \exp_after:wN \exp_stop_f: } }	15667
15668	\cs_new:Npn __fp_exp_after_tuple_f:nw	15668
15669	#1 \s__fp_tuple __fp_tuple_chk:w #2 __fp_sep:	15669
15670	{	15670
15671	\exp_after:wN \s__fp_tuple	15671
15672	\exp_after:wN __fp_tuple_chk:w	15672

15673	\exp_after:wN {	15673
15674	\exp:w \exp_end_continue_f:w	15674
15675	_fp_exp_after_array_f:w #2 \s_fp_expr_stop	15675
15676	\exp_after:wN }	15676
15677	\exp_after:wN _fp_sep:	15677
15678	\exp:w \exp_end_continue_f:w #1	15678
15679	}	15679
15680	\cs_new:Npn _fp_exp_after_array_f:w	15680
15681	{ _fp_exp_after_any_f:nw { _fp_exp_after_array_f:w } }	15681
15682	\int_const:Nn \c_fp_leading_shift_int { - 5 0000 }	15682
15683	\int_const:Nn \c_fp_middle_shift_int { 5 0000 * 9999 }	15683
15684	\int_const:Nn \c_fp_trailing_shift_int { 5 0000 * 10000 }	15684
15685	\cs_new:Npn _fp_pack:NNNNNw #1 #2#3#4#5 #6_fp_sep:	15685
15686	{ + #1#2#3#4#5 _fp_sep: {#6} }	15686
15687	\int_const:Nn \c_fp_big_leading_shift_int { - 15 2374 }	15687
15688	\int_const:Nn \c_fp_big_middle_shift_int { 15 2374 * 9999 }	15688
15689	\int_const:Nn \c_fp_big_trailing_shift_int { 15 2374 * 10000 }	15689
15690	\cs_new:Npn _fp_pack_big:NNNNNNw #1#2 #3#4#5#6 #7_fp_sep:	15690
15691	{ + #1#2#3#4#5#6 _fp_sep: {#7} }	15691
15692	\int_const:Nn \c_fp_Bigg_leading_shift_int { - 20 0000 }	15692
15693	\int_const:Nn \c_fp_Bigg_middle_shift_int { 20 0000 * 9999 }	15693
15694	\int_const:Nn \c_fp_Bigg_trailing_shift_int { 20 0000 * 10000 }	15694
15695	\cs_new:Npn _fp_pack_Bigg:NNNNNNw #1#2 #3#4#5#6 #7_fp_sep:	15695
15696	{ + #1#2#3#4#5#6 _fp_sep: {#7} }	15696
15697	\cs_new:Npn _fp_pack_twice_four:wNNNNNNNN #1_fp_sep: #2#3#4#5 #6#7#8#9	15697
15698	{ #1 {#2#3#4#5} {#6#7#8#9} _fp_sep: }	15698
15699	\cs_new:Npn _fp_pack_eight:wNNNNNNNN #1_fp_sep: #2#3#4#5 #6#7#8#9	15699
15700	{ #1 {#2#3#4#5#6#7#8#9} _fp_sep: }	15700
15701	\cs_new:Npn _fp_basics_pack_low:NNNNNw #1 #2#3#4#5 #6_fp_sep:	15701
15702	{ + #1 - 1 _fp_sep: {#2#3#4#5} {#6} _fp_sep: }	15702
15703	\cs_new:Npn _fp_basics_pack_high:NNNNNw #1 #2#3#4#5 #6_fp_sep:	15703
15704	{	15704
15705	\if_meaning:w 2 #1	15705
15706	_fp_basics_pack_high_carry:w	15706
15707	\fi:	15707
15708	_fp_sep: {#2#3#4#5} {#6}	15708
15709	}	15709
15710	\cs_new:Npn _fp_basics_pack_high_carry:w \fi: _fp_sep: #1	15710
15711	{ \fi: + 1 _fp_sep: {1000} }	15711
15712	\cs_new:Npn _fp_basics_pack_weird_low:NNNNw #1 #2#3#4 #5_fp_sep:	15712
15713	{	15713
15714	\if_meaning:w 2 #1	15714
15715	+ 1	15715
15716	\fi:	15716
15717	_fp_int_eval_end:	15717
15718	#2#3#4_fp_sep: {#5} _fp_sep:	15718


```
15719 } 15719
15720 \cs_new:Npn \__fp_basics_pack_weird_high:NNNNNNNNw 15720
15721 1 #1#2#3#4 #5#6#7#8 #9\__fp_sep: { \__fp_sep: {#1#2#3#4} {#5#6#7#8} {#9} } 15721
15722 \cs_new:Npn \__fp_decimate:nNnnnn #1 15722
15723 { 15723
15724 \cs:w 15724
15725 __fp_decimate_ 15725
15726 \if_int_compare:w \__fp_int_eval:w #1 > \c__fp_prec_int 15726
15727 tiny 15727
15728 \else: 15728
15729 \__fp_int_to_roman:w \__fp_int_eval:w #1 15729
15730 \fi: 15730
15731 :Nnnnn 15731
15732 \cs_end: 15732
15733 } 15733
15734 \cs_new:Npn \__fp_decimate_:Nnnnn #1 #2#3#4#5 15734
15735 { #1 0 {#2#3} {#4#5} \__fp_sep: } 15735
15736 \cs_new:Npn \__fp_decimate_tiny:Nnnnn #1 #2#3#4#5 15736
15737 { #1 1 { 0000 0000 } { 0000 0000 } 0 #2#3#4#5 \__fp_sep: } 15737
15738 \cs_new:Npn \__fp_tmp:w #1 #2 #3 15738
15739 { 15739
15740 \cs_new:cpn { __fp_decimate_ #1 :Nnnnn } ##1 ##2##3##4##5 15740
15741 { 15741
15742 \exp_after:wN ##1 15742
15743 \int_value:w 15743
15744 \exp_after:wN \__fp_round_digit:Nw #2 \__fp_sep: 15744
15745 \__fp_decimate_pack:nnnnnnnnnnw #3 \__fp_sep: 15745
15746 } 15746
15747 } 15747
15748 \__fp_tmp:w {i} {\use_none:nnn #50}{ 0{#2}#3{#4}#5 } 15748
15749 \__fp_tmp:w {ii} {\use_none:nn #5 }{ 00{#2}#3{#4}#5 } 15749
15750 \__fp_tmp:w {iii} {\use_none:n #5 }{ 000{#2}#3{#4}#5 } 15750
15751 \__fp_tmp:w {iv} { #5 }{ {0000}#2{#3}#4 #5 } 15751
15752 \__fp_tmp:w {v} {\use_none:nnn #4#5 }{ 0{0000}#2{#3}#4 #5 } 15752
15753 \__fp_tmp:w {vi} {\use_none:nn #4#5 }{ 00{0000}#2{#3}#4 #5 } 15753
15754 \__fp_tmp:w {vii} {\use_none:n #4#5 }{ 000{0000}#2{#3}#4 #5 } 15754
15755 \__fp_tmp:w {viii}{ #4#5 }{ {0000}0000{#2}#3 #4 #5 } 15755
15756 \__fp_tmp:w {ix} {\use_none:nnn #3#4+#5}{ 0{0000}0000{#2}#3 #4 #5 } 15756
15757 \__fp_tmp:w {x} {\use_none:nn #3#4+#5}{ 00{0000}0000{#2}#3 #4 #5 } 15757
15758 \__fp_tmp:w {xi} {\use_none:n #3#4+#5}{ 000{0000}0000{#2}#3 #4 #5 } 15758
15759 \__fp_tmp:w {xii} { #3#4+#5}{ {0000}0000{0000}#2 #3 #4 #5 } 15759
15760 \__fp_tmp:w {xiii}{\use_none:nnn#2#3+#4#5}{ 0{0000}0000{0000}#2 #3 #4 #5 } 15760
15761 \__fp_tmp:w {xiv} {\use_none:nn #2#3+#4#5}{ 00{0000}0000{0000}#2 #3 #4 #5 } 15761
15762 \__fp_tmp:w {xv} {\use_none:n #2#3+#4#5}{ 000{0000}0000{0000}#2 #3 #4 #5 } 15762
15763 \__fp_tmp:w {xvi} { #2#3+#4#5}{ {0000}0000{0000}0000 #2 #3 #4 #5 } 15763
15764 \cs_new:Npn \__fp_decimate_pack:nnnnnnnnnnw #1#2#3#4#5 15764
```

15765	{ __fp_decimate_pack:nnnnnnw { #1#2#3#4#5 } }	15765
15766	\cs_new:Npn __fp_decimate_pack:nnnnnnw #1 #2#3#4#5#6	15766
15767	{ {#1} {#2#3#4#5#6} }	15767
15768	\cs_new:Npn __fp_case_use:nw #1#2 \fi: #3 \s__fp { \fi: #1 \s__fp }	15768
15769	\cs_new:Npn __fp_case_return:nw #1#2 \fi: #3 __fp_sep: { \fi: #1 }	15769
15770	\cs_new:Npn __fp_case_return_o:Nw #1#2 \fi: #3 \s__fp #4 __fp_sep:	15770
15771	{ \fi: \exp_after:wN #1 }	15771
15772	\cs_new:Npn __fp_case_return_same_o:w #1 \fi: #2 \s__fp	15772
15773	{ \fi: __fp_exp_after_o:w \s__fp }	15773
15774	\cs_new:Npn __fp_case_return_o:Nww #1#2 \fi: #3 \s__fp #4 __fp_sep: #5 __fp_sep:	15774
15775	{ \fi: \exp_after:wN #1 }	15775
15776	\cs_new:Npn __fp_case_return_i_o:ww	15776
15777	#1 \fi: #2 \s__fp #3 __fp_sep: \s__fp #4 __fp_sep:	15777
15778	{ \fi: __fp_exp_after_o:w \s__fp #3 __fp_sep: }	15778
15779	\cs_new:Npn __fp_case_return_ii_o:ww #1 \fi: #2 \s__fp #3 __fp_sep:	15779
15780	{ \fi: __fp_exp_after_o:w }	15780
15781	\prg_new_conditional:Npnn __fp_int:w \s__fp __fp_chk:w #1 #2 #3 #4__fp_sep:	15781
15782	{ TF , T , F , p }	15782
15783	{	15783
15784	\if_case:w #1 \exp_stop_f:	15784
15785	\prg_return_true:	15785
15786	\or:	15786
15787	\if_charcode:w 0	15787
15788	__fp_decimate:nNnnnn { \c__fp_prec_int - #3 }	15788
15789	__fp_use_i_until_s:nw #4	15789
15790	\prg_return_true:	15790
15791	\else:	15791
15792	\prg_return_false:	15792
15793	\fi:	15793
15794	\else: \prg_return_false:	15794
15795	\fi:	15795
15796	}	15796
15797	\cs_new:Npn __fp_small_int:wTF \s__fp __fp_chk:w #1#2	15797
15798	{	15798
15799	\if_case:w #1 \exp_stop_f:	15799
15800	__fp_case_return:nw { __fp_small_int_true:wTF 0 __fp_sep: }	15800
15801	\or: \exp_after:wN __fp_small_int_normal:NnwTF	15801
15802	\or:	15802
15803	__fp_case_return:nw	15803
15804	{	15804
15805	\exp_after:wN __fp_small_int_true:wTF \int_value:w	15805
15806	\if_meaning:w 2 #2 - \fi: 1 0000 0000 __fp_sep:	15806
15807	}	15807
15808	\else: __fp_case_return:nw \use_ii:nn	15808
15809	\fi:	15809
15810	#2	15810

```

15811 } 15811
15812 \cs_new:Npn \__fp_small_int_true:wTF #1\__fp_sep: #2#3 { #2 {#1} } 15812
15813 \cs_new:Npn \__fp_small_int_normal:NnwTF #1#2#3\__fp_sep: 15813
15814 { 15814
15815 \__fp_decimate:nNnnnn { \c__fp_prec_int - #2 } 15815
15816 \__fp_small_int_test:NnnwNw 15816
15817 #3 #1 15817
15818 } 15818
15819 \cs_new:Npn \__fp_small_int_test:NnnwNw #1#2#3#4\__fp_sep: #5 15819
15820 { 15820
15821 \if_meaning:w 0 #1 15821
15822 \exp_after:wN \__fp_small_int_true:wTF 15822
15823 \int_value:w \if_meaning:w 2 #5 - \fi: 15823
15824 \if_int_compare:w #2 > \c_zero_int 15824
15825 1 0000 0000 15825
15826 \else: 15826
15827 #3 15827
15828 \fi: 15828
15829 \exp_after:wN \__fp_sep: 15829
15830 \else: 15830
15831 \exp_after:wN \use_ii:nn 15831
15832 \fi: 15832
15833 } 15833
15834 \cs_new_eq:NN \__fp_str_if_eq:nn \tex_strcmp:D 15834
15835 \cs_new:Npn \__fp_func_to_name:N #1 15835
15836 { 15836
15837 \exp_last_unbraced:Nf 15837
15838 \__fp_func_to_name_aux:w { \cs_to_str:N #1 } X 15838
15839 } 15839
15840 \cs_set_protected:Npn \__fp_tmp:w #1 #2 15840
15841 { \cs_new:Npn \__fp_func_to_name_aux:w ##1 #1 ##2 #2 ##3 X {##2} } 15841
15842 \exp_args:Nff \__fp_tmp:w { \tl_to_str:n { __fp_ } } 15842
15843 { \tl_to_str:n { _o: } } 15843
15844 \msg_new:nnnn { fp } { misused } 15844
15845 { A~floating~point~with~value~'#1'~was~misused. } 15845
15846 { 15846
15847 To~obtain~the~value~of~a~floating~point~variable,~use~ 15847
15848 '\token_to_str:N \fp_to_decimal:N',~ 15848
15849 '\token_to_str:N \fp_to_tl:N',~or~other~ 15849
15850 conversion~functions. 15850
15851 } 15851
15852 \prop_gput:Nnn \g_msg_module_name_prop { fp } { LaTeX } 15852
15853 \prop_gput:Nnn \g_msg_module_type_prop { fp } { } 15853
15854 %% File: l3fp-traps.dtx 15854
15855 \flag_new:N \l_fp_invalid_operation_flag 15855
15856 \flag_new:N \l_fp_division_by_zero_flag 15856

```

```
15857 \flag_new:N \l_fp_overflow_flag 15857
15858 \flag_new:N \l_fp_underflow_flag 15858
15859 \cs_new_protected:Npn \fp_trap:nn #1#2 15859
15860 { 15860
15861     \cs_if_exist_use:cF { __fp_trap_#1_set_#2: } 15861
15862     { 15862
15863         \clist_if_in:nnTF 15863
15864         { invalid_operation , division_by_zero , overflow , underflow } 15864
15865         {#1} 15865
15866         { 15866
15867             \msg_error:nnee { fp } 15867
15868             { unknown-fpu-trap-type } {#1} {#2} 15868
15869         } 15869
15870         { 15870
15871             \msg_error:nne 15871
15872             { fp } { unknown-fpu-exception } {#1} 15872
15873         } 15873
15874     } 15874
15875 } 15875
15876 \cs_new_protected:Npn \__fp_trap_invalid_operation_set_error: 15876
15877 { \__fp_trap_invalid_operation_set:N \prg_do_nothing: } 15877
15878 \cs_new_protected:Npn \__fp_trap_invalid_operation_set_flag: 15878
15879 { \__fp_trap_invalid_operation_set:N \use_none:nnnnn } 15879
15880 \cs_new_protected:Npn \__fp_trap_invalid_operation_set_none: 15880
15881 { \__fp_trap_invalid_operation_set:N \use_none:nnnnnnn } 15881
15882 \cs_new_protected:Npn \__fp_trap_invalid_operation_set:N #1 15882
15883 { 15883
15884     \exp_args:Nno \use:n 15884
15885     { \cs_set:Npn \__fp_invalid_operation:nnw ##1##2##3\__fp_sep: } 15885
15886     { 15886
15887         #1 15887
15888         \__fp_error:nnfn { invalid } {##2} { \fp_to_tl:n { ##3\__fp_sep: } } { } 15888
15889         \flag_ensure_raised:N \l_fp_invalid_operation_flag 15889
15890         ##1 15890
15891     } 15891
15892     \exp_args:Nno \use:n 15892
15893     { \cs_set:Npn \__fp_invalid_operation_o:Nww ##1##2\__fp_sep: ##3\__fp_sep: } 15893
15894     { 15894
15895         #1 15895
15896         \__fp_error:nffn { invalid-ii } 15896
15897         { \fp_to_tl:n { ##2\__fp_sep: } } 15897
15898         { \fp_to_tl:n { ##3\__fp_sep: } } 15898
15899         {##1} 15899
15900         \flag_ensure_raised:N \l_fp_invalid_operation_flag 15900
15901         \exp_after:wN \c_nan_fp 15901
15902     } 15902
```

```
15903 \exp_args:Nno \use:n
15904 { \cs_set:Npn \__fp_invalid_operation_tl_o:ff ##1##2 }
15905 {
15906 #1
15907 \__fp_error:nfn { invalid } {##1} {##2} { }
15908 \flag_ensure_raised:N \l_fp_invalid_operation_flag
15909 \exp_after:wN \c_nan_fp
15910 }
15911 }
15912 \cs_new_protected:Npn \__fp_trap_division_by_zero_set_error:
15913 { \__fp_trap_division_by_zero_set:N \prg_do_nothing: }
15914 \cs_new_protected:Npn \__fp_trap_division_by_zero_set_flag:
15915 { \__fp_trap_division_by_zero_set:N \use_none:nnnnn }
15916 \cs_new_protected:Npn \__fp_trap_division_by_zero_set_none:
15917 { \__fp_trap_division_by_zero_set:N \use_none:nnnnnnnn }
15918 \cs_new_protected:Npn \__fp_trap_division_by_zero_set:N #1
15919 {
15920 \exp_args:Nno \use:n
15921 { \cs_set:Npn \__fp_division_by_zero_o:Nnw ##1##2##3\__fp_sep: }
15922 {
15923 #1
15924 \__fp_error:nnfn { zero-div } {##2} { \fp_to_tl:n { ##3\__fp_sep: } } { }
15925 \flag_ensure_raised:N \l_fp_division_by_zero_flag
15926 \exp_after:wN ##1
15927 }
15928 \exp_args:Nno \use:n
15929 {
15930 \cs_set:Npn \__fp_division_by_zero_o:NNww ##1##2##3\__fp_sep: ##4\__fp_sep:
15931 }
15932 {
15933 #1
15934 \__fp_error:nfn { zero-div-ii }
15935 { \fp_to_tl:n { ##3\__fp_sep: } }
15936 { \fp_to_tl:n { ##4\__fp_sep: } }
15937 {##2}
15938 \flag_ensure_raised:N \l_fp_division_by_zero_flag
15939 \exp_after:wN ##1
15940 }
15941 }
15942 \cs_new_protected:Npn \__fp_trap_overflow_set_error:
15943 { \__fp_trap_overflow_set:N \prg_do_nothing: }
15944 \cs_new_protected:Npn \__fp_trap_overflow_set_flag:
15945 { \__fp_trap_overflow_set:N \use_none:nnnnn }
15946 \cs_new_protected:Npn \__fp_trap_overflow_set_none:
15947 { \__fp_trap_overflow_set:N \use_none:nnnnnnnn }
15948 \cs_new_protected:Npn \__fp_trap_overflow_set:N #1
```

```
15949 { \__fp_trap_overflow_set:NnNn #1 { overflow } \__fp_inf_fp:N { inf } } 15949
15950 \cs_new_protected:Npn \__fp_trap_underflow_set_error: 15950
15951 { \__fp_trap_underflow_set:N \prg_do_nothing: } 15951
15952 \cs_new_protected:Npn \__fp_trap_underflow_set_flag: 15952
15953 { \__fp_trap_underflow_set:N \use_none:nnnnn } 15953
15954 \cs_new_protected:Npn \__fp_trap_underflow_set_none: 15954
15955 { \__fp_trap_underflow_set:N \use_none:nnnnnnn } 15955
15956 \cs_new_protected:Npn \__fp_trap_underflow_set:N #1 15956
15957 { \__fp_trap_overflow_set:NnNn #1 { underflow } \__fp_zero_fp:N { 0 } } 15957
15958 \cs_new_protected:Npn \__fp_trap_overflow_set:NnNn #1#2#3#4 15958
15959 { 15959
15960 \exp_args:Nno \use:n 15960
15961 { \cs_set:cpn { __fp_ #2 :w } \s__fp \__fp_chk:w ##1##2##3\__fp_sep: } 15961
15962 { 15962
15963 #1 15963
15964 \__fp_error:nffn 15964
15965 { flow \if_meaning:w 1 ##1 -to \fi: } 15965
15966 { \fp_to_tl:n { \s__fp \__fp_chk:w ##1##2##3\__fp_sep: } } 15966
15967 { \token_if_eq_meaning:NNF 0 ##2 { - } #4 } 15967
15968 {#2} 15968
15969 \flag_ensure_raised:c { l_fp_#2_flag } 15969
15970 #3 ##2 15970
15971 } 15971
15972 } 15972
15973 \cs_new:Npn \__fp_invalid_operation:nnw #1#2#3\__fp_sep: { } 15973
15974 \cs_new:Npn \__fp_invalid_operation_o:Nww #1#2\__fp_sep: #3\__fp_sep: { } 15974
15975 \cs_new:Npn \__fp_invalid_operation_tl_o:ff #1 #2 { } 15975
15976 \cs_new:Npn \__fp_division_by_zero_o:Nnw #1#2#3\__fp_sep: { } 15976
15977 \cs_new:Npn \__fp_division_by_zero_o:NNww #1#2#3\__fp_sep: #4\__fp_sep: { } 15977
15978 \cs_new:Npn \__fp_overflow:w { } 15978
15979 \cs_new:Npn \__fp_underflow:w { } 15979
15980 \fp_trap:nn { invalid_operation } { error } 15980
15981 \fp_trap:nn { division_by_zero } { flag } 15981
15982 \fp_trap:nn { overflow } { flag } 15982
15983 \fp_trap:nn { underflow } { flag } 15983
15984 \cs_new:Npn \__fp_invalid_operation_o:nw 15984
15985 { \__fp_invalid_operation:nnw { \exp_after:wN \c_nan_fp } } 15985
15986 \cs_generate_variant:Nn \__fp_invalid_operation_o:nw { f } 15986
15987 \cs_new:Npn \__fp_error:nnnn 15987
15988 { \msg_expandable_error:nnnnn { fp } } 15988
15989 \cs_generate_variant:Nn \__fp_error:nnnn { nnf, nff , nfff } 15989
15990 \cs_new:Npn \__fp_error_num_args:nnnn #1#2#3#4 15990
15991 { 15991
15992 \int_compare:nNnTF {#2} = {#3} 15992
15993 { \msg_expandable_error:nnnnn { fp } { num-args-eq } {#1} {#2} {#4} } 15993
15994 { \msg_expandable_error:nnnnnn { fp } { num-args } {#1} {#2} {#3} {#4} } 15994
```

```
15995     }
15996     \cs_generate_variant:Nn \__fp_error_num_args:nnnn { ffff }
15997     \msg_new:nnn { fp } { num-args-eq }
15998     { #1()~needs~#2~arguments,~got~#3. }
15999     \msg_new:nnn { fp } { num-args }
16000     { #1()~needs~#2~to~#3~arguments,~got~#4. }
16001     \msg_new:nnnn { fp } { unknown-fpu-exception }
16002     {
16003         The~FPU~exception~'#1'~is~not~known:~
16004         that~trap~will~never~be~triggered.
16005     }
16006     {
16007         The~only~exceptions~to~which~traps~can~be~attached~are \\\
16008         \iow_indent:n
16009         {
16010             * ~ invalid_operation \\\
16011             * ~ division_by_zero \\\
16012             * ~ overflow \\\
16013             * ~ underflow
16014         }
16015     }
16016     \msg_new:nnnn { fp } { unknown-fpu-trap-type }
16017     { The~FPU~trap~type~'#2'~is~not~known. }
16018     {
16019         The~trap~type~must~be~one~of \\\
16020         \iow_indent:n
16021         {
16022             * ~ error \\\
16023             * ~ flag \\\
16024             * ~ none
16025         }
16026     }
16027     \msg_new:nnn { fp } { flow }
16028     { An ~ #3 ~ occurred. }
16029     \msg_new:nnn { fp } { flow-to }
16030     { #1 ~ #3 ed ~ to ~ #2 . }
16031     \msg_new:nnn { fp } { zero-div }
16032     { Division~by~zero~in~ #1 (#2) }
16033     \msg_new:nnn { fp } { zero-div-ii }
16034     { Division~by~zero~in~ (#1) #3 (#2) }
16035     \msg_new:nnn { fp } { invalid }
16036     { Invalid~operation~ #1 (#2) }
16037     \msg_new:nnn { fp } { invalid-ii }
16038     { Invalid~operation~ (#1) #3 (#2) }
16039     \msg_new:nnn { fp } { unknown-type }
16040     { Unknown~type~for~'#1' }
```



```
16041 %% File: l3fp-round.dtx 16041
16042 \cs_new:Npn \__fp_parse_word_trunc:N 16042
16043 { \__fp_parse_function:NNN \__fp_round_o:Nw \__fp_round_to_zero:NNN } 16043
16044 \cs_new:Npn \__fp_parse_word_floor:N 16044
16045 { \__fp_parse_function:NNN \__fp_round_o:Nw \__fp_round_to_ninf:NNN } 16045
16046 \cs_new:Npn \__fp_parse_word_ceil:N 16046
16047 { \__fp_parse_function:NNN \__fp_round_o:Nw \__fp_round_to_pinf:NNN } 16047
16048 \cs_new:Npn \__fp_parse_word_round:N #1#2 16048
16049 { 16049
16050 \__fp_parse_function:NNN 16050
16051 \__fp_round_o:Nw \__fp_round_to_nearest:NNN #1 16051
16052 #2 16052
16053 } 16053
16054 \cs_new:Npn \__fp_parse_round:Nw #1 #2 \__fp_round_to_nearest:NNN #3#4 16054
16055 { #2 #1 #3 } 16055
16056 \int_const:Nn \c__fp_five_int { 5 } 16056
16057 \cs_new:Npn \__fp_round_return_one: 16057
16058 { \exp_after:wN 1 \exp_after:wN \exp_stop_f: \exp:w } 16058
16059 \cs_new:Npn \__fp_round_to_ninf:NNN #1 #2 #3 16059
16060 { 16060
16061 \if_meaning:w 2 #1 16061
16062 \if_int_compare:w #3 > \c_zero_int 16062
16063 \__fp_round_return_one: 16063
16064 \fi: 16064
16065 \fi: 16065
16066 \c_zero_int 16066
16067 } 16067
16068 \cs_new:Npn \__fp_round_to_zero:NNN #1 #2 #3 { \c_zero_int } 16068
16069 \cs_new:Npn \__fp_round_to_pinf:NNN #1 #2 #3 16069
16070 { 16070
16071 \if_meaning:w 0 #1 16071
16072 \if_int_compare:w #3 > \c_zero_int 16072
16073 \__fp_round_return_one: 16073
16074 \fi: 16074
16075 \fi: 16075
16076 \c_zero_int 16076
16077 } 16077
16078 \cs_new:Npn \__fp_round_to_nearest:NNN #1 #2 #3 16078
16079 { 16079
16080 \if_int_compare:w #3 > \c__fp_five_int 16080
16081 \__fp_round_return_one: 16081
16082 \else: 16082
16083 \if_meaning:w 5 #3 16083
16084 \if_int_odd:w #2 \exp_stop_f: 16084
16085 \__fp_round_return_one: 16085
16086 \fi: 16086
```

16087	\fi:	16087
16088	\fi:	16088
16089	\c_zero_int	16089
16090	}	16090
16091	\cs_new:Npn __fp_round_to_nearest_ninf:NNN #1 #2 #3	16091
16092	{	16092
16093	\if_int_compare:w #3 > \c__fp_five_int	16093
16094	__fp_round_return_one:	16094
16095	\else:	16095
16096	\if_meaning:w 5 #3	16096
16097	\if_meaning:w 2 #1	16097
16098	__fp_round_return_one:	16098
16099	\fi:	16099
16100	\fi:	16100
16101	\fi:	16101
16102	\c_zero_int	16102
16103	}	16103
16104	\cs_new:Npn __fp_round_to_nearest_zero:NNN #1 #2 #3	16104
16105	{	16105
16106	\if_int_compare:w #3 > \c__fp_five_int	16106
16107	__fp_round_return_one:	16107
16108	\fi:	16108
16109	\c_zero_int	16109
16110	}	16110
16111	\cs_new:Npn __fp_round_to_nearest_pinf:NNN #1 #2 #3	16111
16112	{	16112
16113	\if_int_compare:w #3 > \c__fp_five_int	16113
16114	__fp_round_return_one:	16114
16115	\else:	16115
16116	\if_meaning:w 5 #3	16116
16117	\if_meaning:w 0 #1	16117
16118	__fp_round_return_one:	16118
16119	\fi:	16119
16120	\fi:	16120
16121	\fi:	16121
16122	\c_zero_int	16122
16123	}	16123
16124	\cs_new_eq:NN __fp_round:NNN __fp_round_to_nearest:NNN	16124
16125	\cs_new:Npn __fp_round_s:NNNw #1 #2 #3 #4__fp_sep:	16125
16126	{	16126
16127	\exp_after:wN __fp_round:NNN	16127
16128	\exp_after:wN #1	16128
16129	\exp_after:wN #2	16129
16130	\int_value:w __fp_int_eval:w	16130
16131	\if_int_odd:w 0 \if_meaning:w 0 #3 1 \fi:	16131
16132	\if_meaning:w 5 #3 1 \fi:	16132

16133	\exp_stop_f:	16133
16134	\if_int_compare:w __fp_int_eval:w #4 > \c_zero_int	16134
16135	1 +	16135
16136	\fi:	16136
16137	\fi:	16137
16138	#3	16138
16139	__fp_sep:	16139
16140	}	16140
16141	\cs_new:Npn __fp_round_digit:Nw #1 #2__fp_sep:	16141
16142	{	16142
16143	\if_int_odd:w \if_meaning:w 0 #1 1 \else:	16143
16144	\if_meaning:w 5 #1 1 \else:	16144
16145	0 \fi: \fi: \exp_stop_f:	16145
16146	\if_int_compare:w __fp_int_eval:w #2 > \c_zero_int	16146
16147	__fp_int_eval:w 1 +	16147
16148	\fi:	16148
16149	\fi:	16149
16150	#1	16150
16151	}	16151
16152	\cs_new_eq:NN __fp_round_to_ninf_neg:NNN __fp_round_to_pinf:NNN	16152
16153	\cs_new:Npn __fp_round_to_zero_neg:NNN #1 #2 #3	16153
16154	{	16154
16155	\if_int_compare:w #3 > \c_zero_int	16155
16156	__fp_round_return_one:	16156
16157	\fi:	16157
16158	\c_zero_int	16158
16159	}	16159
16160	\cs_new_eq:NN __fp_round_to_pinf_neg:NNN __fp_round_to_ninf:NNN	16160
16161	\cs_new_eq:NN __fp_round_to_nearest_neg:NNN __fp_round_to_nearest:NNN	16161
16162	\cs_new_eq:NN __fp_round_to_nearest_ninf_neg:NNN	16162
16163	__fp_round_to_nearest_pinf:NNN	16163
16164	\cs_new:Npn __fp_round_to_nearest_zero_neg:NNN #1 #2 #3	16164
16165	{	16165
16166	\if_int_compare:w #3 < \c__fp_five_int \else:	16166
16167	__fp_round_return_one:	16167
16168	\fi:	16168
16169	\c_zero_int	16169
16170	}	16170
16171	\cs_new_eq:NN __fp_round_to_nearest_pinf_neg:NNN	16171
16172	__fp_round_to_nearest_ninf:NNN	16172
16173	\cs_new_eq:NN __fp_round_neg:NNN __fp_round_to_nearest_neg:NNN	16173
16174	\cs_new:Npn __fp_round_o:Nw #1	16174
16175	{	16175
16176	__fp_parse_function_all_fp_o:fnw	16176
16177	{ __fp_round_name_from_cs:N #1 }	16177
16178	{ __fp_round_aux_o:Nw #1 }	16178

```

16179 }
16180 \cs_new:Npn \__fp_round_aux_o:Nw #1#2 @
16181 {
16182   \if_case:w
16183     \__fp_int_eval:w \__fp_array_count:n {#2} \__fp_int_eval_end:
16184     \__fp_round_no_arg_o:Nw #1 \exp:w
16185   \or: \__fp_round:Nwn #1 #2 {0} \exp:w
16186   \or: \__fp_round:Nww #1 #2 \exp:w
16187   \else: \__fp_round:Nwww #1 #2 @ \exp:w
16188   \fi:
16189   \exp_after:wN \exp_end:
16190 }
16191 \cs_new:Npn \__fp_round_no_arg_o:Nw #1
16192 {
16193   \cs_if_eq:NNTF #1 \__fp_round_to_nearest:NNN
16194   { \__fp_error_num_args:nnnn { round } { 1 } { 3 } { 0 } }
16195   {
16196     \__fp_error_num_args:ffff
16197     { \__fp_round_name_from_cs:N #1 } { 1 } { 2 } { 0 }
16198   }
16199   \exp_after:wN \c_nan_fp
16200 }
16201 \cs_new:Npn \__fp_round:Nwww
16202   #1#2 \__fp_sep: #3 \__fp_sep: \s_fp \__fp_chk:w #4#5#6 \__fp_sep: #7 @
16203 {
16204   \cs_if_eq:NNTF #1 \__fp_round_to_nearest:NNN
16205   {
16206     \tl_if_empty:nTF {#7}
16207     {
16208       \exp_args:Nc \__fp_round:Nww
16209       {
16210         __fp_round_to_nearest
16211         \if_meaning:w 0 #4 _zero \else:
16212         \if_case:w #5 \exp_stop_f: _pinf \or: \else: _ninf \fi: \fi:
16213         :NNN
16214       }
16215       #2 \__fp_sep: #3 \__fp_sep:
16216     }
16217     {
16218       \__fp_error_num_args:ffff { round } { 1 } { 3 }
16219       { \int_eval:n { 3 + \__fp_array_count:n {#7} } }
16220       \exp_after:wN \c_nan_fp
16221     }
16222   }
16223   {
16224     \__fp_error_num_args:ffff

```

```
16225         { \__fp_round_name_from_cs:N #1 } { 1 } { 2 } 16225
16226         { \int_eval:n { 3 + \__fp_array_count:n {#7} } } 16226
16227         \exp_after:wN \c_nan_fp 16227
16228     } 16228
16229 } 16229
16230 \cs_new:Npn \__fp_round_name_from_cs:N #1 16230
16231 { 16231
16232     \cs_if_eq:NNTF #1 \__fp_round_to_zero:NNN { trunc } 16232
16233     { 16233
16234         \cs_if_eq:NNTF #1 \__fp_round_to_ninf:NNN { floor } 16234
16235         { 16235
16236             \cs_if_eq:NNTF #1 \__fp_round_to_pinf:NNN { ceil } 16236
16237             { round } 16237
16238         } 16238
16239     } 16239
16240 } 16240
16241 \cs_new:Npn \__fp_round:Nww #1#2 \__fp_sep: #3 \__fp_sep: 16241
16242 { 16242
16243     \__fp_small_int:wTF #3\__fp_sep: { \__fp_round:Nwn #1#2\__fp_sep: } 16243
16244     { 16244
16245         \if:w 3 \__fp_kind:w #3 \__fp_sep: 16245
16246             \exp_after:wN \use_i:nn 16246
16247         \else: 16247
16248             \exp_after:wN \use_ii:nn 16248
16249         \fi: 16249
16250         { \exp_after:wN \c_nan_fp } 16250
16251         { 16251
16252             \__fp_invalid_operation_tl_o:ff 16252
16253             { \__fp_round_name_from_cs:N #1 } 16253
16254             { \__fp_array_to_clist:n { #2\__fp_sep: #3\__fp_sep: } } 16254
16255         } 16255
16256     } 16256
16257 } 16257
16258 \cs_new:Npn \__fp_round:Nwn #1 \s__fp \__fp_chk:w #2#3#4\__fp_sep: #5 16258
16259 { 16259
16260     \if_meaning:w 1 #2 16260
16261         \exp_after:wN \__fp_round_normal:NwNNnw 16261
16262         \exp_after:wN #1 16262
16263         \int_value:w #5 16263
16264     \else: 16264
16265         \exp_after:wN \__fp_exp_after_o:w 16265
16266     \fi: 16266
16267     \s__fp \__fp_chk:w #2#3#4\__fp_sep: 16267
16268 } 16268
16269 \cs_new:Npn \__fp_round_normal:NwNNnw #1#2 \s__fp \__fp_chk:w 1#3#4#5\__fp_sep: 16269
16270 { 16270
```

```
16271 \__fp_decimate:nNnnnn { \c__fp_prec_int - #4 - #2 } 16271
16272 \__fp_round_normal:NnnwNNnn #5 #1 #3 {#4} {#2} 16272
16273 } 16273
16274 \cs_new:Npn \__fp_round_normal:NnnwNNnn #1#2#3#4\__fp_sep: #5#6 16274
16275 { 16275
16276 \exp_after:wN \__fp_round_normal:NNwNnn 16276
16277 \int_value:w \__fp_int_eval:w 16277
16278 \if_int_compare:w #2 > \c_zero_int 16278
16279 1 \int_value:w #2 16279
16280 \exp_after:wN \__fp_round_pack:Nw 16280
16281 \int_value:w \__fp_int_eval:w 1#3 + 16281
16282 \else: 16282
16283 \if_int_compare:w #3 > \c_zero_int 16283
16284 1 \int_value:w #3 + 16284
16285 \fi: 16285
16286 \fi: 16286
16287 \exp_after:wN #5 16287
16288 \exp_after:wN #6 16288
16289 \use_none:nnnnnnn #3 16289
16290 #1 16290
16291 \__fp_int_eval_end: 16291
16292 0000 0000 0000 0000 \__fp_sep: #6 16292
16293 } 16293
16294 \cs_new:Npn \__fp_round_pack:Nw #1 16294
16295 { \if_meaning:w 2 #1 + 1 \fi: \__fp_int_eval_end: } 16295
16296 \cs_new:Npn \__fp_round_normal:NNwNnn #1 #2 16296
16297 { 16297
16298 \if_meaning:w 0 #2 16298
16299 \exp_after:wN \__fp_round_special:NwwNnn 16299
16300 \exp_after:wN #1 16300
16301 \fi: 16301
16302 \__fp_pack_twice_four:wNNNNNNNN 16302
16303 \__fp_pack_twice_four:wNNNNNNNN 16303
16304 \__fp_round_normal_end:wwNnn 16304
16305 \__fp_sep: #2 16305
16306 } 16306
16307 \cs_new:Npn \__fp_round_normal_end:wwNnn #1\__fp_sep:#2\__fp_sep:#3#4#5 16307
16308 { 16308
16309 \exp_after:wN \__fp_exp_after_o:w \exp:w \exp_end_continue_f:w 16309
16310 \__fp_sanitizew #3 #4 \__fp_sep: #1 \__fp_sep: 16310
16311 } 16311
16312 \cs_new:Npn \__fp_round_special:NwwNnn #1#2\__fp_sep:#3\__fp_sep:#4#5#6 16312
16313 { 16313
16314 \if_meaning:w 0 #1 16314
16315 \__fp_case_return:nw 16315
16316 { \exp_after:wN \__fp_zero_fp:N \exp_after:wN #4 } 16316
```

```

16317 \else: 16317
16318 \exp_after:wN \__fp_round_special_aux:Nw 16318
16319 \exp_after:wN #4 16319
16320 \int_value:w \__fp_int_eval:w 1 16320
16321 \if_meaning:w 1 #1 -#6 \else: +#5 \fi: 16321
16322 \fi: 16322
16323 \__fp_sep: 16323
16324 } 16324
16325 \cs_new:Npn \__fp_round_special_aux:Nw #1#2\__fp_sep: 16325
16326 { 16326
16327 \exp_after:wN \__fp_exp_after_o:w \exp:w \exp_end_continue_f:w 16327
16328 \__fp_sanitizew:Nw #1#2\__fp_sep: {1000}{0000}{0000}{0000}\__fp_sep: 16328
16329 } 16329
16330 %% File: l3fp-parse.dtx 16330
16331 \int_const:Nn \c__fp_prec_func_int { 16 } 16331
16332 \int_const:Nn \c__fp_prec_hatii_int { 14 } 16332
16333 \int_const:Nn \c__fp_prec_hat_int { 13 } 16333
16334 \int_const:Nn \c__fp_prec_not_int { 12 } 16334
16335 \int_const:Nn \c__fp_prec_juxt_int { 11 } 16335
16336 \int_const:Nn \c__fp_prec_times_int { 10 } 16336
16337 \int_const:Nn \c__fp_prec_plus_int { 9 } 16337
16338 \int_const:Nn \c__fp_prec_comp_int { 7 } 16338
16339 \int_const:Nn \c__fp_prec_and_int { 6 } 16339
16340 \int_const:Nn \c__fp_prec_or_int { 5 } 16340
16341 \int_const:Nn \c__fp_prec_quest_int { 4 } 16341
16342 \int_const:Nn \c__fp_prec_colon_int { 3 } 16342
16343 \int_const:Nn \c__fp_prec_comma_int { 2 } 16343
16344 \int_const:Nn \c__fp_prec_tuple_int { 1 } 16344
16345 \int_const:Nn \c__fp_prec_end_int { 0 } 16345
16346 \cs_new:Npn \__fp_parse_expand:w #1 { \exp_end_continue_f:w #1 } 16346
16347 \cs_new:Npn \__fp_parse_return_sep:w 16347
16348 #1 \fi: \__fp_parse_expand:w { \fi: \__fp_sep: #1 } 16348
16349 \cs_set_protected:Npn \__fp_tmp:w #1 #2 #3 16349
16350 { 16350
16351 \cs_new:cpn { __fp_parse_digits_ #1 :N } ##1 16351
16352 { 16352
16353 \if_int_compare:w 9 < 1 \token_to_str:N ##1 \exp_stop_f: 16353
16354 \token_to_str:N ##1 \exp_after:wN #2 \exp:w 16354
16355 \else: 16355
16356 \__fp_parse_return_sep:w #3 ##1 16356
16357 \fi: 16357
16358 \__fp_parse_expand:w 16358
16359 } 16359
16360 } 16360
16361 \__fp_tmp:w {vii} \__fp_parse_digits_vi:N { 0000000 \__fp_sep: 7 } 16361
16362 \__fp_tmp:w {vi} \__fp_parse_digits_v:N { 000000 \__fp_sep: 6 } 16362

```



```

16363 \__fp_tmp:w {v} \__fp_parse_digits_iv:N { 00000 \__fp_sep: 5 }
16364 \__fp_tmp:w {iv} \__fp_parse_digits_iii:N { 0000 \__fp_sep: 4 }
16365 \__fp_tmp:w {iii} \__fp_parse_digits_ii:N { 000 \__fp_sep: 3 }
16366 \__fp_tmp:w {ii} \__fp_parse_digits_i:N { 00 \__fp_sep: 2 }
16367 \__fp_tmp:w {i} \__fp_parse_digits_:N { 0 \__fp_sep: 1 }
16368 \cs_new:Npn \__fp_parse_digits_:N { \__fp_sep: \__fp_sep: 0 }
16369 \cs_new:Npn \__fp_parse_one:Nw #1 #2
16370 {
16371     \if_catcode:w \scan_stop: \exp_not:N #2
16372     \exp_after:wN \if_meaning:w \exp_not:N #2 #2 \else:
16373         \exp_after:wN \reverse_if:N
16374     \fi:
16375     \if_meaning:w \scan_stop: #2
16376     \exp_after:wN \exp_after:wN
16377     \exp_after:wN \__fp_parse_one_fp:NN
16378 \else:
16379     \exp_after:wN \exp_after:wN
16380     \exp_after:wN \__fp_parse_one_register:NN
16381     \fi:
16382 \else:
16383     \if_int_compare:w 9 < 1 \token_to_str:N #2 \exp_stop_f:
16384     \exp_after:wN \exp_after:wN
16385     \exp_after:wN \__fp_parse_one_digit:NN
16386 \else:
16387     \exp_after:wN \exp_after:wN
16388     \exp_after:wN \__fp_parse_one_other:NN
16389     \fi:
16390     \fi:
16391     #1 #2
16392 }
16393 \cs_new:Npn \__fp_parse_one_fp:NN #1
16394 {
16395     \__fp_exp_after_any_f:nw
16396     {
16397         \exp_after:wN \__fp_parse_infix:NN
16398         \exp_after:wN #1 \exp:w \__fp_parse_expand:w
16399     }
16400 }
16401 \cs_new:Npn \__fp_exp_after_expr_mark_f:nw #1
16402 {
16403     \int_case:nnF { \exp_after:wN \use_i:nnn \use_none:nnn #1 }
16404     {
16405         \c__fp_prec_comma_int { }
16406         \c__fp_prec_tuple_int { }
16407         \c__fp_prec_end_int
16408         {

```

```
16409         \exp_after:wN \c__fp_empty_tuple_fp
16410         \exp:w \exp_end_continue_f:w
16411     }
16412 }
16413 {
16414     \msg_expandable_error:nn { fp } { early-end }
16415     \exp_after:wN \c_nan_fp \exp:w \exp_end_continue_f:w
16416 }
16417 #1
16418 }
16419 \cs_new:cpn { __fp_exp_after_?_f:nw } #1#2
16420 {
16421     \msg_expandable_error:nnn { kernel } { bad-variable }
16422     {#2}
16423     \exp_after:wN \c_nan_fp \exp:w \exp_end_continue_f:w #1
16424 }
16425 \cs_set_protected:Npn \__fp_tmp:w #1
16426 {
16427     \cs_if_exist:NT #1
16428     {
16429         \cs_gset:cpn { __fp_exp_after_?_f:nw } ##1##2
16430         {
16431             \exp_after:wN \c_nan_fp \exp:w \exp_end_continue_f:w ##1
16432             \str_if_eq:nnTF {##2} { \protect }
16433             {
16434                 \cs_if_eq:NNTF ##2 #1 { \use_i:nn } { \use:n }
16435                 {
16436                     \msg_expandable_error:nnn { fp }
16437                     { robust-cmd }
16438                 }
16439             }
16440             {
16441                 \msg_expandable_error:nnn { kernel }
16442                 { bad-variable } {##2}
16443             }
16444         }
16445     }
16446 }
16447 \exp_args:Nc \__fp_tmp:w { @unexpandable@protect }
16448 \cs_new:Npn \__fp_parse_one_register:NN #1#2
16449 {
16450     \exp_after:wN \__fp_parse_infix_after_operand:NwN
16451     \exp_after:wN #1
16452     \exp:w \exp_end_continue_f:w
16453     \__fp_parse_one_register_special:N #2
16454     \exp_after:wN \__fp_parse_one_register_aux:Nw
```

```

16455 \exp_after:wN #2
16456 \int_value:w
16457 \exp_after:wN \__fp_parse_exponent:N
16458 \exp:w \__fp_parse_expand:w
16459 }
16460 \cs_new:Npe \__fp_parse_one_register_aux:Nw #1
16461 {
16462 \exp_not:n
16463 {
16464 \exp_after:wN \use:nn
16465 \exp_after:wN \__fp_parse_one_register_auxii:wwwNw
16466 }
16467 \exp_not:N \exp_after:wN { \exp_not:N \tex_the:D #1 }
16468 \__fp_sep: \exp_not:N \__fp_parse_one_register_dim:ww
16469 \tl_to_str:n { pt } \__fp_sep: \exp_not:N \__fp_parse_one_register_mu:www
16470 . \tl_to_str:n { pt } \__fp_sep: \exp_not:N \__fp_parse_one_register_int:www
16471 \s__fp_stop
16472 }
16473 \exp_args:Nno \use:nn
16474 { \cs_new:Npn \__fp_parse_one_register_auxii:wwwNw #1 . #2 }
16475 { \tl_to_str:n { pt } #3 \__fp_sep: #4#5 \s__fp_stop }
16476 { #4 #1.#2\__fp_sep: }
16477 \exp_args:Nno \use:nn
16478 { \cs_new:Npn \__fp_parse_one_register_mu:www #1 }
16479 { \tl_to_str:n { mu } \__fp_sep: #2 \__fp_sep: }
16480 { \__fp_parse_one_register_dim:ww #1 \__fp_sep: }
16481 \cs_new:Npn \__fp_parse_one_register_int:www #1\__fp_sep: #2.\__fp_sep: #3\__fp_sep:
16482 { \__fp_parse:n { #1 e #3 } }
16483 \cs_new:Npn \__fp_parse_one_register_dim:ww #1\__fp_sep: #2\__fp_sep:
16484 {
16485 \exp_after:wN \__fp_from_dim_test:ww
16486 \int_value:w #2 \exp_after:wN ,
16487 \int_value:w \dim_to_decimal_in_sp:n { #1 pt } \__fp_sep:
16488 }
16489 \cs_new:Npn \__fp_parse_one_register_special:N #1
16490 {
16491 \if_meaning:w \box_wd:N #1 \__fp_parse_one_register_wd:w \fi:
16492 \if_meaning:w \box_ht:N #1 \__fp_parse_one_register_wd:w \fi:
16493 \if_meaning:w \box_dp:N #1 \__fp_parse_one_register_wd:w \fi:
16494 \if_meaning:w \infty #1
16495 \__fp_parse_one_register_math:NNw \infty #1
16496 \fi:
16497 \if_meaning:w \pi #1
16498 \__fp_parse_one_register_math:NNw \pi #1
16499 \fi:
16500 }

```

```
16501 \cs_new:Npn \__fp_parse_one_register_math:NNw 16501
16502 #1#2#3#4 \__fp_parse_expand:w 16502
16503 { 16503
16504 #3 16504
16505 \str_if_eq:nnTF {#1} {#2} 16505
16506 { 16506
16507 \msg_expandable_error:nnn 16507
16508 { fp } { infinity-pi } {#1} 16508
16509 \c_nan_fp 16509
16510 } 16510
16511 { #4 \__fp_parse_expand:w } 16511
16512 } 16512
16513 \cs_new:Npn \__fp_parse_one_register_wd:w 16513
16514 #1#2 \exp_after:wN #3#4 \__fp_parse_expand:w 16514
16515 { 16515
16516 #1 16516
16517 \exp_after:wN \__fp_parse_one_register_wd:Nw 16517
16518 #4 \__fp_parse_expand:w e 16518
16519 } 16519
16520 \cs_new:Npn \__fp_parse_one_register_wd:Nw #1#2 \__fp_sep: 16520
16521 { 16521
16522 \exp_after:wN \__fp_from_dim_test:ww 16522
16523 \exp_after:wN 0 \exp_after:wN , 16523
16524 \int_value:w \dim_to_decimal_in_sp:n { #1 #2 } \__fp_sep: 16524
16525 } 16525
16526 \cs_new:Npn \__fp_parse_one_digit:NN #1 16526
16527 { 16527
16528 \exp_after:wN \__fp_parse_infix_after_operand:NwN 16528
16529 \exp_after:wN #1 16529
16530 \exp:w \exp_end_continue_f:w 16530
16531 \exp_after:wN \__fp_sanitise:wN 16531
16532 \int_value:w \__fp_int_eval:w 0 \__fp_parse_trim_zeros:N 16532
16533 } 16533
16534 \cs_new:Npn \__fp_parse_one_other:NN #1 #2 16534
16535 { 16535
16536 \if_int_compare:w 16536
16537 \__fp_int_eval:w 16537
16538 ( `#2 \if_int_compare:w `#2 > `Z - 32 \fi: ) / 26 16538
16539 = 3 \exp_stop_f: 16539
16540 \exp_after:wN \__fp_parse_word:Nw 16540
16541 \exp_after:wN #1 16541
16542 \exp_after:wN #2 16542
16543 \exp:w \exp_after:wN \__fp_parse_letters:N 16543
16544 \exp:w 16544
16545 \else: 16545
16546 \exp_after:wN \__fp_parse_prefix:NNN 16546
```

16547	\exp_after:wN #1	16547
16548	\exp_after:wN #2	16548
16549	\cs:w	16549
16550	__fp_parse_prefix_ \token_to_str:N #2 :Nw	16550
16551	\exp_after:wN	16551
16552	\cs_end:	16552
16553	\exp:w	16553
16554	\fi:	16554
16555	__fp_parse_expand:w	16555
16556	}	16556
16557	\cs_new:Npn __fp_parse_word:Nw #1#2__fp_sep:	16557
16558	{	16558
16559	\cs_if_exist_use:cF { __fp_parse_word_#2:N }	16559
16560	{	16560
16561	\cs_if_exist_use:cF	16561
16562	{ __fp_parse_caseless_ \str_casefold:n {#2} :N }	16562
16563	{	16563
16564	\msg_expandable_error:nnn	16564
16565	{ fp } { unknown-fp-word } {#2}	16565
16566	\exp_after:wN \c_nan_fp \exp:w \exp_end_continue_f:w	16566
16567	__fp_parse_infix:NN	16567
16568	}	16568
16569	}	16569
16570	#1	16570
16571	}	16571
16572	\cs_new:Npn __fp_parse_letters:N #1	16572
16573	{	16573
16574	\exp_end_continue_f:w	16574
16575	\if_int_compare:w	16575
16576	\if_catcode:w \scan_stop: \exp_not:N #1	16576
16577	0	16577
16578	\else:	16578
16579	__fp_int_eval:w	16579
16580	(`#1 \if_int_compare:w `#1 > `Z - 32 \fi:) / 26	16580
16581	\fi:	16581
16582	= 3 \exp_stop_f:	16582
16583	\exp_after:wN #1	16583
16584	\exp:w \exp_after:wN __fp_parse_letters:N	16584
16585	\exp:w	16585
16586	\else:	16586
16587	__fp_parse_return_sep:w #1	16587
16588	\fi:	16588
16589	__fp_parse_expand:w	16589
16590	}	16590
16591	\cs_new:Npn __fp_parse_prefix:NNN #1#2#3	16591
16592	{	16592

16593	\if_meaning:w \scan_stop: #3	16593
16594	\exp_after:wN __fp_parse_prefix_unknown:NNN	16594
16595	\exp_after:wN #2	16595
16596	\fi:	16596
16597	#3 #1	16597
16598	}	16598
16599	\cs_new:Npn __fp_parse_prefix_unknown:NNN #1#2#3	16599
16600	{	16600
16601	\cs_if_exist:cTF { __fp_parse_infix_ \token_to_str:N #1 :N }	16601
16602	{	16602
16603	\msg_expandable_error:nnn	16603
16604	{ fp } { missing-number } {#1}	16604
16605	\exp_after:wN \c_nan_fp \exp:w \exp_end_continue_f:w	16605
16606	__fp_parse_infix:NN #3 #1	16606
16607	}	16607
16608	{	16608
16609	\msg_expandable_error:nnn	16609
16610	{ fp } { unknown-symbol } {#1}	16610
16611	__fp_parse_one:Nw #3	16611
16612	}	16612
16613	}	16613
16614	\cs_new:Npn __fp_parse_trim_zeros:N #1	16614
16615	{	16615
16616	\if:w 0 \exp_not:N #1	16616
16617	\exp_after:wN __fp_parse_trim_zeros:N	16617
16618	\exp:w	16618
16619	\else:	16619
16620	\if:w . \exp_not:N #1	16620
16621	\exp_after:wN __fp_parse_strim_zeros:N	16621
16622	\exp:w	16622
16623	\else:	16623
16624	__fp_parse_trim_end:w #1	16624
16625	\fi:	16625
16626	\fi:	16626
16627	__fp_parse_expand:w	16627
16628	}	16628
16629	\cs_new:Npn __fp_parse_trim_end:w #1 \fi: \fi: __fp_parse_expand:w	16629
16630	{	16630
16631	\fi:	16631
16632	\fi:	16632
16633	\if_int_compare:w 9 < 1 \token_to_str:N #1 \exp_stop_f:	16633
16634	\exp_after:wN __fp_parse_large:N	16634
16635	\else:	16635
16636	\exp_after:wN __fp_parse_zero:	16636
16637	\fi:	16637
16638	#1	16638

16639	}	16639
16640	\cs_new:Npn __fp_parse_strim_zeros:N #1	16640
16641	{	16641
16642	\if:w 0 \exp_not:N #1	16642
16643	- 1	16643
16644	\exp_after:wN __fp_parse_strim_zeros:N \exp:w	16644
16645	\else:	16645
16646	__fp_parse_strim_end:w #1	16646
16647	\fi:	16647
16648	__fp_parse_expand:w	16648
16649	}	16649
16650	\cs_new:Npn __fp_parse_strim_end:w #1 \fi: __fp_parse_expand:w	16650
16651	{	16651
16652	\fi:	16652
16653	\if_int_compare:w 9 < 1 \token_to_str:N #1 \exp_stop_f:	16653
16654	\exp_after:wN __fp_parse_small:N	16654
16655	\else:	16655
16656	\exp_after:wN __fp_parse_zero:	16656
16657	\fi:	16657
16658	#1	16658
16659	}	16659
16660	\cs_new:Npn __fp_parse_zero:	16660
16661	{	16661
16662	\exp_after:wN __fp_sep: \exp_after:wN 1	16662
16663	\int_value:w __fp_parse_exponent:N	16663
16664	}	16664
16665	\cs_new:Npn __fp_parse_small:N #1	16665
16666	{	16666
16667	\exp_after:wN __fp_parse_pack_leading:NNNNNww	16667
16668	\int_value:w __fp_int_eval:w 1 \token_to_str:N #1	16668
16669	\exp_after:wN __fp_parse_small_leading:wwNN	16669
16670	\int_value:w 1	16670
16671	\exp_after:wN __fp_parse_digits_vii:N	16671
16672	\exp:w __fp_parse_expand:w	16672
16673	}	16673
16674	\cs_new:Npn __fp_parse_small_leading:wwNN 1 #1 __fp_sep: #2__fp_sep: #3 #4	16674
16675	{	16675
16676	#1 #2	16676
16677	\exp_after:wN __fp_parse_pack_trailing:NNNNNww	16677
16678	\exp_after:wN 0	16678
16679	\int_value:w __fp_int_eval:w 1	16679
16680	\if_int_compare:w 9 < 1 \token_to_str:N #4 \exp_stop_f:	16680
16681	\token_to_str:N #4	16681
16682	\exp_after:wN __fp_parse_small_trailing:wwNN	16682
16683	\int_value:w 1	16683
16684	\exp_after:wN __fp_parse_digits_vi:N	16684

16685	\exp:w	16685
16686	\else:	16686
16687	0000 0000 __fp_parse_exponent:Nw #4	16687
16688	\fi:	16688
16689	__fp_parse_expand:w	16689
16690	}	16690
16691	\cs_new:Npn __fp_parse_small_trailing:wwNN 1 #1 __fp_sep: #2__fp_sep: #3 #4	16691
16692	{	16692
16693	#1 #2	16693
16694	\if_int_compare:w 9 < 1 \token_to_str:N #4 \exp_stop_f:	16694
16695	\token_to_str:N #4	16695
16696	\exp_after:wN __fp_parse_small_round:NN	16696
16697	\exp_after:wN #4	16697
16698	\exp:w	16698
16699	\else:	16699
16700	0 __fp_parse_exponent:Nw #4	16700
16701	\fi:	16701
16702	__fp_parse_expand:w	16702
16703	}	16703
16704	\cs_new:Npn __fp_parse_pack_trailing:NNNNNNww	16704
16705	#1 #2 #3#4#5#6 #7__fp_sep: #8 __fp_sep:	16705
16706	{	16706
16707	\if_meaning:w 2 #2 + 1 \fi:	16707
16708	__fp_sep: #8 + #1 __fp_sep: {#3#4#5#6} {#7}__fp_sep:	16708
16709	}	16709
16710	\cs_new:Npn __fp_parse_pack_leading:NNNNNww #1 #2#3#4#5 #6__fp_sep: #7__fp_sep:	16710
16711	{	16711
16712	+ #7	16712
16713	\if_meaning:w 2 #1 __fp_parse_pack_carry:w \fi:	16713
16714	__fp_sep: 0 {#2#3#4#5} {#6}	16714
16715	}	16715
16716	\cs_new:Npn __fp_parse_pack_carry:w \fi: __fp_sep: 0 #1	16716
16717	{ \fi: + 1 __fp_sep: 0 {1000} }	16717
16718	\cs_new:Npn __fp_parse_large:N #1	16718
16719	{	16719
16720	\exp_after:wN __fp_parse_large_leading:wwNN	16720
16721	\int_value:w 1 \token_to_str:N #1	16721
16722	\exp_after:wN __fp_parse_digits_vii:N	16722
16723	\exp:w __fp_parse_expand:w	16723
16724	}	16724
16725	\cs_new:Npn __fp_parse_large_leading:wwNN 1 #1 __fp_sep: #2__fp_sep: #3 #4	16725
16726	{	16726
16727	+ \c__fp_half_prec_int - #3	16727
16728	\exp_after:wN __fp_parse_pack_leading:NNNNNww	16728
16729	\int_value:w __fp_int_eval:w 1 #1	16729
16730	\if_int_compare:w 9 < 1 \token_to_str:N #4 \exp_stop_f:	16730

16731	\exp_after:wN __fp_parse_large_trailing:wwNN	16731
16732	\int_value:w 1 \token_to_str:N #4	16732
16733	\exp_after:wN __fp_parse_digits_vi:N	16733
16734	\exp:w	16734
16735	\else:	16735
16736	\if:w . \exp_not:N #4	16736
16737	\exp_after:wN __fp_parse_small_leading:wwNN	16737
16738	\int_value:w 1	16738
16739	\cs:w	16739
16740	__fp_parse_digits_	16740
16741	__fp_int_to_roman:w #3	16741
16742	:N \exp_after:wN	16742
16743	\cs_end:	16743
16744	\exp:w	16744
16745	\else:	16745
16746	#2	16746
16747	\exp_after:wN __fp_parse_pack_trailing:NNNNNNww	16747
16748	\exp_after:wN 0	16748
16749	\int_value:w 1 0000 0000	16749
16750	__fp_parse_exponent:Nw #4	16750
16751	\fi:	16751
16752	\fi:	16752
16753	__fp_parse_expand:w	16753
16754	}	16754
16755	\cs_new:Npn __fp_parse_large_trailing:wwNN 1 #1 __fp_sep: #2__fp_sep: #3 #4	16755
16756	{	16756
16757	\if_int_compare:w 9 < 1 \token_to_str:N #4 \exp_stop_f:	16757
16758	\exp_after:wN __fp_parse_pack_trailing:NNNNNNww	16758
16759	\exp_after:wN \c__fp_half_prec_int	16759
16760	\int_value:w __fp_int_eval:w 1 #1 \token_to_str:N #4	16760
16761	\exp_after:wN __fp_parse_large_round:NN	16761
16762	\exp_after:wN #4	16762
16763	\exp:w	16763
16764	\else:	16764
16765	\exp_after:wN __fp_parse_pack_trailing:NNNNNNww	16765
16766	\int_value:w __fp_int_eval:w 7 - #3 \exp_stop_f:	16766
16767	\int_value:w __fp_int_eval:w 1 #1	16767
16768	\if:w . \exp_not:N #4	16768
16769	\exp_after:wN __fp_parse_small_trailing:wwNN	16769
16770	\int_value:w 1	16770
16771	\cs:w	16771
16772	__fp_parse_digits_	16772
16773	__fp_int_to_roman:w #3	16773
16774	:N \exp_after:wN	16774
16775	\cs_end:	16775
16776	\exp:w	16776

16777	\else:	16777
16778	#2 0 __fp_parse_exponent:Nw #4	16778
16779	\fi:	16779
16780	\fi:	16780
16781	__fp_parse_expand:w	16781
16782	}	16782
16783	\cs_new:Npn __fp_parse_round_loop:N #1	16783
16784	{	16784
16785	\if_int_compare:w 9 < 1 \token_to_str:N #1 \exp_stop_f:	16785
16786	+ 1	16786
16787	\if:w 0 \token_to_str:N #1	16787
16788	\exp_after:wN __fp_parse_round_loop:N	16788
16789	\exp:w	16789
16790	\else:	16790
16791	\exp_after:wN __fp_parse_round_up:N	16791
16792	\exp:w	16792
16793	\fi:	16793
16794	\else:	16794
16795	__fp_parse_return_sep:w 0 #1	16795
16796	\fi:	16796
16797	__fp_parse_expand:w	16797
16798	}	16798
16799	\cs_new:Npn __fp_parse_round_up:N #1	16799
16800	{	16800
16801	\if_int_compare:w 9 < 1 \token_to_str:N #1 \exp_stop_f:	16801
16802	+ 1	16802
16803	\exp_after:wN __fp_parse_round_up:N	16803
16804	\exp:w	16804
16805	\else:	16805
16806	__fp_parse_return_sep:w 1 #1	16806
16807	\fi:	16807
16808	__fp_parse_expand:w	16808
16809	}	16809
16810	\cs_new:Npn __fp_parse_round_after:wN #1__fp_sep: #2	16810
16811	{	16811
16812	+ #2 \exp_after:wN __fp_sep:	16812
16813	\int_value:w __fp_int_eval:w #1 + __fp_parse_exponent:N	16813
16814	}	16814
16815	\cs_new:Npn __fp_parse_small_round:NN #1#2	16815
16816	{	16816
16817	\if_int_compare:w 9 < 1 \token_to_str:N #2 \exp_stop_f:	16817
16818	+	16818
16819	\exp_after:wN __fp_round_s:NNNw	16819
16820	\exp_after:wN 0	16820
16821	\exp_after:wN #1	16821
16822	\exp_after:wN #2	16822

16823	\int_value:w __fp_int_eval:w	16823
16824	\exp_after:wN __fp_parse_round_after:wN	16824
16825	\int_value:w __fp_int_eval:w 0 * __fp_int_eval:w 0	16825
16826	\exp_after:wN __fp_parse_round_loop:N	16826
16827	\exp:w	16827
16828	\else:	16828
16829	__fp_parse_exponent:Nw #2	16829
16830	\fi:	16830
16831	__fp_parse_expand:w	16831
16832	}	16832
16833	\cs_new:Npn __fp_parse_large_round:NN #1#2	16833
16834	{	16834
16835	\if_int_compare:w 9 < 1 \token_to_str:N #2 \exp_stop_f:	16835
16836	+	16836
16837	\exp_after:wN __fp_round_s:NNNw	16837
16838	\exp_after:wN 0	16838
16839	\exp_after:wN #1	16839
16840	\exp_after:wN #2	16840
16841	\int_value:w __fp_int_eval:w	16841
16842	\exp_after:wN __fp_parse_large_round_aux:wNN	16842
16843	\int_value:w __fp_int_eval:w 1	16843
16844	\exp_after:wN __fp_parse_round_loop:N	16844
16845	\else: %^^A could be dot, or e, or other	16845
16846	\exp_after:wN __fp_parse_large_round_test:NN	16846
16847	\exp_after:wN #1	16847
16848	\exp_after:wN #2	16848
16849	\fi:	16849
16850	}	16850
16851	\cs_new:Npn __fp_parse_large_round_test:NN #1#2	16851
16852	{	16852
16853	\if:w . \exp_not:N #2	16853
16854	\exp_after:wN __fp_parse_small_round:NN	16854
16855	\exp_after:wN #1	16855
16856	\exp:w	16856
16857	\else:	16857
16858	__fp_parse_exponent:Nw #2	16858
16859	\fi:	16859
16860	__fp_parse_expand:w	16860
16861	}	16861
16862	\cs_new:Npn __fp_parse_large_round_aux:wNN #1 __fp_sep: #2 #3	16862
16863	{	16863
16864	+ #2	16864
16865	\exp_after:wN __fp_parse_round_after:wN	16865
16866	\int_value:w __fp_int_eval:w #1	16866
16867	\if:w . \exp_not:N #3	16867
16868	+ 0 * __fp_int_eval:w 0	16868

```
16869         \exp_after:wN \__fp_parse_round_loop:N 16869
16870         \exp:w \exp_after:wN \__fp_parse_expand:w 16870
16871     \else: 16871
16872         \exp_after:wN \__fp_sep: 16872
16873         \exp_after:wN 0 16873
16874         \exp_after:wN #3 16874
16875     \fi: 16875
16876 } 16876
16877 \cs_new:Npn \__fp_parse_exponent:Nw #1 #2 \__fp_parse_expand:w 16877
16878 { 16878
16879     \exp_after:wN \__fp_sep: 16879
16880     \int_value:w #2 \__fp_parse_exponent:N #1 16880
16881 } 16881
16882 \cs_new:Npn \__fp_parse_exponent:N #1 16882
16883 { 16883
16884     \if:w e \if:w E \exp_not:N #1 e \else: \exp_not:N #1 \fi: 16884
16885     \exp_after:wN \__fp_parse_exponent_aux:NN 16885
16886     \exp_after:wN #1 16886
16887     \exp:w 16887
16888     \else: 16888
16889         0 \__fp_parse_return_sep:w #1 16889
16890     \fi: 16890
16891     \__fp_parse_expand:w 16891
16892 } 16892
16893 \cs_new:Npn \__fp_parse_exponent_aux:NN #1#2 16893
16894 { 16894
16895     \if_int_compare:w \if_catcode:w \scan_stop: \exp_not:N #2 16895
16896         0 \else: `#2 \fi: > `9 \exp_stop_f: 16896
16897         0 \exp_after:wN \__fp_sep: \exp_after:wN #1 16897
16898     \else: 16898
16899         \exp_after:wN \__fp_parse_exponent_sign:N 16899
16900     \fi: 16900
16901     #2 16901
16902 } 16902
16903 \cs_new:Npn \__fp_parse_exponent_sign:N #1 16903
16904 { 16904
16905     \if:w + \if:w - \exp_not:N #1 + \fi: \token_to_str:N #1 16905
16906     \exp_after:wN \__fp_parse_exponent_sign:N 16906
16907     \exp:w \exp_after:wN \__fp_parse_expand:w 16907
16908     \else: 16908
16909         \exp_after:wN \__fp_parse_exponent_body:N 16909
16910         \exp_after:wN #1 16910
16911     \fi: 16911
16912 } 16912
16913 \cs_new:Npn \__fp_parse_exponent_body:N #1 16913
16914 { 16914
```

16915	\if_int_compare:w 9 < 1 \token_to_str:N #1 \exp_stop_f:	16915
16916	\token_to_str:N #1	16916
16917	\exp_after:wN __fp_parse_exponent_digits:N	16917
16918	\exp:w	16918
16919	\else:	16919
16920	__fp_parse_exponent_keep:N TF #1	16920
16921	{ __fp_parse_return_sep:w #1 }	16921
16922	{	16922
16923	\exp_after:wN __fp_sep:	16923
16924	\exp:w	16924
16925	}	16925
16926	\fi:	16926
16927	__fp_parse_expand:w	16927
16928	}	16928
16929	\cs_new:Npn __fp_parse_exponent_digits:N #1	16929
16930	{	16930
16931	\if_int_compare:w 9 < 1 \token_to_str:N #1 \exp_stop_f:	16931
16932	\token_to_str:N #1	16932
16933	\exp_after:wN __fp_parse_exponent_digits:N	16933
16934	\exp:w	16934
16935	\else:	16935
16936	__fp_parse_return_sep:w #1	16936
16937	\fi:	16937
16938	__fp_parse_expand:w	16938
16939	}	16939
16940	\prg_new_conditional:Npnn __fp_parse_exponent_keep:N #1 { TF }	16940
16941	{	16941
16942	\if_catcode:w \scan_stop: \exp_not:N #1	16942
16943	\if_meaning:w \scan_stop: #1	16943
16944	\if:w 0 __fp_str_if_eq:nn { \s__fp } { \exp_not:N #1 }	16944
16945	0	16945
16946	\msg_expandable_error:nnn	16946
16947	{ fp } { after-e } { floating~point~ }	16947
16948	\prg_return_true:	16948
16949	\else:	16949
16950	0	16950
16951	\msg_expandable_error:nnn	16951
16952	{ kernel } { bad-variable } { #1 }	16952
16953	\prg_return_false:	16953
16954	\fi:	16954
16955	\else:	16955
16956	\if:w 0 __fp_str_if_eq:nn { \int_value:w #1 } { \tex_the:D #1 }	16956
16957	\int_value:w #1	16957
16958	\else:	16958
16959	0	16959
16960	\msg_expandable_error:nnn	16960

```

16961         { fp } { after-e } { dimension~#1 }
16962     \fi:
16963     \prg_return_false:
16964 \fi:
16965 \else:
16966     0
16967     \msg_expandable_error:nnn
16968         { fp } { missing } { exponent }
16969     \prg_return_true:
16970 \fi:
16971 }
16972 \cs_new_eq:cN { __fp_parse_prefix+:Nw } \__fp_parse_one:Nw
16973 \cs_new:Npn \__fp_parse_apply_function:NNNwN #1#2#3#4@#5
16974 {
16975     #3 #2 #4 @
16976     \exp:w \exp_end_continue_f:w #5 #1
16977 }
16978 \cs_new:Npn \__fp_parse_apply_unary:NNNwN #1#2#3#4@#5
16979 {
16980     \__fp_parse_apply_unary_chk:NwNw #4 @ \__fp_sep: . \s__fp_stop
16981     \__fp_parse_apply_unary_type:NNN
16982     #3 #2 #4 @
16983     \exp:w \exp_end_continue_f:w #5 #1
16984 }
16985 \cs_new:Npn \__fp_parse_apply_unary_chk:NwNw #1#2 \__fp_sep: #3#4 \s__fp_stop
16986 {
16987     \if_meaning:w @ #3 \else:
16988         \token_if_eq_meaning:NNTF . #3
16989         { \__fp_parse_apply_unary_chk:nNNNNw { no } }
16990         { \__fp_parse_apply_unary_chk:nNNNNw { multi } }
16991     \fi:
16992 }
16993 \cs_new:Npn \__fp_parse_apply_unary_chk:nNNNNw #1#2#3#4#5#6 @
16994 {
16995     #2
16996     \__fp_error:nffn { #1-arg } { \__fp_func_to_name:N #4 } { } { }
16997     \exp_after:wN #4 \exp_after:wN #5 \c_nan_fp @
16998 }
16999 \cs_new:Npn \__fp_parse_apply_unary_type:NNN #1#2#3
17000 {
17001     \__fp_change_func_type:NNN #3 #1 \__fp_parse_apply_unary_error:NNw
17002     #2 #3
17003 }
17004 \cs_new:Npn \__fp_parse_apply_unary_error:NNw #1#2#3 @
17005 { \__fp_invalid_operation_o:fw { \__fp_func_to_name:N #1 } #3 }
17006 \cs_set_protected:Npn \__fp_tmp:w #1#2#3#4

```



```
17007 { 17007
17008 \cs_new:cpn { __fp_parse_prefix_ #1 :Nw } ##1 17008
17009 { 17009
17010 \exp_after:wN \__fp_parse_apply_unary:NNNwN 17010
17011 \exp_after:wN ##1 17011
17012 \exp_after:wN #4 17012
17013 \exp_after:wN #3 17013
17014 \exp:w 17014
17015 \if_int_compare:w #2 < ##1 17015
17016 \__fp_parse_operand:Nw ##1 17016
17017 \else: 17017
17018 \__fp_parse_operand:Nw #2 17018
17019 \fi: 17019
17020 \__fp_parse_expand:w 17020
17021 } 17021
17022 } 17022
17023 \__fp_tmp:w - \c__fp_prec_not_int \__fp_set_sign_o:w 2 17023
17024 \__fp_tmp:w ! \c__fp_prec_not_int \__fp_not_o:w ? 17024
17025 \cs_new:cpn { __fp_parse_prefix_.:Nw } #1 17025
17026 { 17026
17027 \exp_after:wN \__fp_parse_infix_after_operand:NwN 17027
17028 \exp_after:wN #1 17028
17029 \exp:w \exp_end_continue_f:w 17029
17030 \exp_after:wN \__fp_sanitize:wN 17030
17031 \int_value:w \__fp_int_eval:w 0 \__fp_parse_strim_zeros:N 17031
17032 } 17032
17033 \cs_new:cpn { __fp_parse_prefix_( :Nw } #1 17033
17034 { 17034
17035 \exp_after:wN \__fp_parse_lparen_after:NwN 17035
17036 \exp_after:wN #1 17036
17037 \exp:w 17037
17038 \if_int_compare:w #1 = \c__fp_prec_func_int 17038
17039 \__fp_parse_operand:Nw \c__fp_prec_comma_int 17039
17040 \else: 17040
17041 \__fp_parse_operand:Nw \c__fp_prec_tuple_int 17041
17042 \fi: 17042
17043 \__fp_parse_expand:w 17043
17044 } 17044
17045 \cs_new:Npe \__fp_parse_lparen_after:NwN #1#2 @ #3 17045
17046 { 17046
17047 \exp_not:N \token_if_eq_meaning:NNTF #3 17047
17048 \exp_not:c { __fp_parse_infix_) :N } 17048
17049 { 17049
17050 \exp_not:N \__fp_exp_after_array_f:w #2 \s__fp_expr_stop 17050
17051 \exp_not:N \exp_after:wN 17051
17052 \exp_not:N \__fp_parse_infix_after_paren:NN 17052
```

17053	\exp_not:N \exp_after:wN #1	17053
17054	\exp_not:N \exp:w	17054
17055	\exp_not:N __fp_parse_expand:w	17055
17056	}	17056
17057	{	17057
17058	\exp_not:N \msg_expandable_error:nnn	17058
17059	{ fp } { missing } {) }	17059
17060	\exp_not:N \tl_if_empty:nT {#2} \exp_not:N \c__fp_empty_tuple_fp	17060
17061	#2 @	17061
17062	\exp_not:N \use_none:n #3	17062
17063	}	17063
17064	}	17064
17065	\cs_new:cpn { __fp_parse_prefix_):Nw } #1	17065
17066	{	17066
17067	\if_int_compare:w #1 = \c__fp_prec_comma_int	17067
17068	\else:	17068
17069	\if_int_compare:w #1 = \c__fp_prec_tuple_int	17069
17070	\exp_after:wN \c__fp_empty_tuple_fp \exp:w	17070
17071	\else:	17071
17072	\msg_expandable_error:nnn	17072
17073	{ fp } { missing-number } {) }	17073
17074	\exp_after:wN \c_nan_fp \exp:w	17074
17075	\fi:	17075
17076	\exp_end_continue_f:w	17076
17077	\fi:	17077
17078	__fp_parse_infix_after_paren:NN #1)	17078
17079	}	17079
17080	\cs_set_protected:Npn __fp_tmp:w #1 #2	17080
17081	{	17081
17082	\cs_new:cpn { __fp_parse_word_#1:N }	17082
17083	{ \exp_after:wN #2 \exp:w \exp_end_continue_f:w __fp_parse_infix:NN }	17083
17084	}	17084
17085	__fp_tmp:w { inf } \c_inf_fp	17085
17086	__fp_tmp:w { nan } \c_nan_fp	17086
17087	__fp_tmp:w { pi } \c_pi_fp	17087
17088	__fp_tmp:w { deg } \c_one_degree_fp	17088
17089	__fp_tmp:w { true } \c_one_fp	17089
17090	__fp_tmp:w { false } \c_zero_fp	17090
17091	\cs_new_eq:NN __fp_parse_caseless_inf:N __fp_parse_word_inf:N	17091
17092	\cs_new_eq:NN __fp_parse_caseless_infinity:N __fp_parse_word_inf:N	17092
17093	\cs_new_eq:NN __fp_parse_caseless_nan:N __fp_parse_word_nan:N	17093
17094	\cs_set_protected:Npn __fp_tmp:w #1 #2	17094
17095	{	17095
17096	\cs_new:cpn { __fp_parse_word_#1:N }	17096
17097	{	17097
17098	__fp_exp_after_f:nw { __fp_parse_infix:NN }	17098

```
17099 \s__fp __fp_chk:w 10 #2 __fp_sep: 17099
17100 } 17100
17101 } 17101
17102 __fp_tmp:w {pt} { {1} {1000} {0000} {0000} {0000} } 17102
17103 __fp_tmp:w {in} { {2} {7227} {0000} {0000} {0000} } 17103
17104 __fp_tmp:w {pc} { {2} {1200} {0000} {0000} {0000} } 17104
17105 __fp_tmp:w {cm} { {2} {2845} {2755} {9055} {1181} } 17105
17106 __fp_tmp:w {mm} { {1} {2845} {2755} {9055} {1181} } 17106
17107 __fp_tmp:w {dd} { {1} {1070} {0085} {6496} {0630} } 17107
17108 __fp_tmp:w {cc} { {2} {1284} {0102} {7795} {2756} } 17108
17109 __fp_tmp:w {nd} { {1} {1066} {9783} {4645} {6693} } 17109
17110 __fp_tmp:w {nc} { {2} {1280} {3740} {1574} {8031} } 17110
17111 __fp_tmp:w {bp} { {1} {1003} {7500} {0000} {0000} } 17111
17112 __fp_tmp:w {sp} { {-4} {1525} {8789} {0625} {0000} } 17112
17113 \tl_map_inline:nn { {em} {ex} } 17113
17114 { 17114
17115 \cs_new:cpn { __fp_parse_word_#1:N } 17115
17116 { 17116
17117 \exp_after:wN __fp_from_dim_test:ww 17117
17118 \exp_after:wN 0 \exp_after:wN , 17118
17119 \int_value:w \dim_to_decimal_in_sp:n { 1 #1 } \exp_after:wN __fp_sep: 17119
17120 \exp:w \exp_end_continue_f:w __fp_parse_infix:NN 17120
17121 } 17121
17122 } 17122
17123 \cs_new:Npn __fp_parse_unary_function:NNN #1#2#3 17123
17124 { 17124
17125 \exp_after:wN __fp_parse_apply_unary:NNNwN 17125
17126 \exp_after:wN #3 17126
17127 \exp_after:wN #2 17127
17128 \exp_after:wN #1 17128
17129 \exp:w 17129
17130 __fp_parse_operand:Nw \c__fp_prec_func_int __fp_parse_expand:w 17130
17131 } 17131
17132 \cs_new:Npn __fp_parse_function:NNN #1#2#3 17132
17133 { 17133
17134 \exp_after:wN __fp_parse_apply_function:NNNwN 17134
17135 \exp_after:wN #3 17135
17136 \exp_after:wN #2 17136
17137 \exp_after:wN #1 17137
17138 \exp:w 17138
17139 __fp_parse_operand:Nw \c__fp_prec_func_int __fp_parse_expand:w 17139
17140 } 17140
17141 \cs_new:Npn __fp_parse:n #1 17141
17142 { 17142
17143 \exp:w 17143
17144 \exp_after:wN __fp_parse_after:ww 17144
```

17145	\exp:w	17145
17146	__fp_parse_operand:Nw \c__fp_prec_end_int	17146
17147	__fp_parse_expand:w #1	17147
17148	\s__fp_expr_mark __fp_parse_infix_end:N	17148
17149	\s__fp_expr_stop	17149
17150	\exp_end:	17150
17151	}	17151
17152	\cs_new:Npn __fp_parse_after:ww	17152
17153	#1@ __fp_parse_infix_end:N \s__fp_expr_stop #2 { #2 #1 }	17153
17154	\cs_new:Npn __fp_parse_o:n #1	17154
17155	{	17155
17156	\exp:w	17156
17157	\exp_after:wN __fp_parse_after:ww	17157
17158	\exp:w	17158
17159	__fp_parse_operand:Nw \c__fp_prec_end_int	17159
17160	__fp_parse_expand:w #1	17160
17161	\s__fp_expr_mark __fp_parse_infix_end:N	17161
17162	\s__fp_expr_stop	17162
17163	{	17163
17164	\exp_end_continue_f:w	17164
17165	__fp_exp_after_any_f:nw { \exp_after:wN \exp_stop_f: }	17165
17166	}	17166
17167	}	17167
17168	\cs_new:Npn __fp_parse_operand:Nw #1	17168
17169	{	17169
17170	\exp_end_continue_f:w	17170
17171	\exp_after:wN __fp_parse_continue:NwN	17171
17172	\exp_after:wN #1	17172
17173	\exp:w \exp_end_continue_f:w	17173
17174	\exp_after:wN __fp_parse_one:Nw	17174
17175	\exp_after:wN #1	17175
17176	\exp:w	17176
17177	}	17177
17178	\cs_new:Npn __fp_parse_continue:NwN #1 #2 @ #3 { #3 #1 #2 @ }	17178
17179	\cs_new:Npn __fp_parse_apply_binary:NwNwN #1 #2#3@ #4 #5#6@ #7	17179
17180	{	17180
17181	\exp_after:wN __fp_parse_continue:NwN	17181
17182	\exp_after:wN #1	17182
17183	\exp:w \exp_end_continue_f:w	17183
17184	\exp_after:wN __fp_parse_apply_binary_chk:NN	17184
17185	\cs:w	17185
17186	__fp	17186
17187	__fp_type_from_scan:N #2	17187
17188	_#4	17188
17189	__fp_type_from_scan:N #5	17189
17190	_o:ww	17190

17191	\cs_end:	17191
17192	#4	17192
17193	#2#3 #5#6	17193
17194	\exp:w \exp_end_continue_f:w #7 #1	17194
17195	}	17195
17196	\cs_new:Npn __fp_parse_apply_binary_chk:NN #1#2	17196
17197	{	17197
17198	\if_meaning:w \scan_stop: #1	17198
17199	__fp_parse_apply_binary_error:NNN #2	17199
17200	\fi:	17200
17201	#1	17201
17202	}	17202
17203	\cs_new:Npn __fp_parse_apply_binary_error:NNN #1#2#3	17203
17204	{	17204
17205	#2	17205
17206	__fp_invalid_operation_o:Nww #1	17206
17207	}	17207
17208	\cs_new:Npn __fp_binary_type_o:Nww #1 #2#3 __fp_sep: #4	17208
17209	{	17209
17210	\exp_after:wN __fp_parse_apply_binary_chk:NN	17210
17211	\cs:w	17211
17212	__fp	17212
17213	__fp_type_from_scan:N #2	17213
17214	_ #1	17214
17215	__fp_type_from_scan:N #4	17215
17216	_o:ww	17216
17217	\cs_end:	17217
17218	#1	17218
17219	#2 #3 __fp_sep: #4	17219
17220	}	17220
17221	\cs_new:Npn __fp_binary_rev_type_o:Nww #1 #2#3 __fp_sep: #4#5 __fp_sep:	17221
17222	{	17222
17223	\exp_after:wN __fp_parse_apply_binary_chk:NN	17223
17224	\cs:w	17224
17225	__fp	17225
17226	__fp_type_from_scan:N #4	17226
17227	_ #1	17227
17228	__fp_type_from_scan:N #2	17228
17229	_o:ww	17229
17230	\cs_end:	17230
17231	#1	17231
17232	#4 #5 __fp_sep: #2 #3 __fp_sep:	17232
17233	}	17233
17234	\cs_new:Npn __fp_parse_infix_after_operand:NwN #1 #2__fp_sep:	17234
17235	{	17235
17236	__fp_exp_after_f:nw { __fp_parse_infix:NN #1 }	17236

17237	#2__fp_sep:	17237
17238	}	17238
17239	\cs_new:Npn __fp_parse_infix:NN #1 #2	17239
17240	{	17240
17241	\if_catcode:w \scan_stop: \exp_not:N #2	17241
17242	\if:w 0 __fp_str_if_eq:nn { \s__fp_expr_mark } { \exp_not:N #2 }	17242
17243	\exp_after:wN \exp_after:wN	17243
17244	\exp_after:wN __fp_parse_infix_mark:NNN	17244
17245	\else:	17245
17246	\exp_after:wN \exp_after:wN	17246
17247	\exp_after:wN __fp_parse_infix_juxt:N	17247
17248	\fi:	17248
17249	\else:	17249
17250	\if_int_compare:w	17250
17251	__fp_int_eval:w	17251
17252	(`#2 \if_int_compare:w `#2 > `Z - 32 \fi:) / 26	17252
17253	= 3 \exp_stop_f:	17253
17254	\exp_after:wN \exp_after:wN	17254
17255	\exp_after:wN __fp_parse_infix_juxt:N	17255
17256	\else:	17256
17257	\exp_after:wN __fp_parse_infix_check:NNN	17257
17258	\cs:w	17258
17259	__fp_parse_infix_ \token_to_str:N #2 :N	17259
17260	\exp_after:wN \exp_after:wN \exp_after:wN	17260
17261	\cs_end:	17261
17262	\fi:	17262
17263	\fi:	17263
17264	#1	17264
17265	#2	17265
17266	}	17266
17267	\cs_new:Npn __fp_parse_infix_check:NNN #1#2#3	17267
17268	{	17268
17269	\if_meaning:w \scan_stop: #1	17269
17270	\msg_expandable_error:nnn	17270
17271	{ fp } { missing } { * }	17271
17272	\exp_after:wN __fp_parse_infix_mul:N	17272
17273	\exp_after:wN #2	17273
17274	\exp_after:wN #3	17274
17275	\else:	17275
17276	\exp_after:wN #1	17276
17277	\exp_after:wN #2	17277
17278	\exp:w \exp_after:wN __fp_parse_expand:w	17278
17279	\fi:	17279
17280	}	17280
17281	\cs_new:Npn __fp_parse_infix_after_paren:NN #1 #2	17281
17282	{	17282

```
17283 \if_catcode:w \scan_stop: \exp_not:N #2 17283
17284 \if:w 0 \__fp_str_if_eq:nn { \s__fp_expr_mark } { \exp_not:N #2 } 17284
17285 \exp_after:wN \exp_after:wN 17285
17286 \exp_after:wN \__fp_parse_infix_mark:NNN 17286
17287 \else: 17287
17288 \exp_after:wN \exp_after:wN 17288
17289 \exp_after:wN \__fp_parse_infix_mul:N 17289
17290 \fi: 17290
17291 \else: 17291
17292 \if_int_compare:w 17292
17293 \__fp_int_eval:w 17293
17294 ( `#2 \if_int_compare:w `#2 > `Z - 32 \fi: ) / 26 17294
17295 = 3 \exp_stop_f: 17295
17296 \exp_after:wN \exp_after:wN 17296
17297 \exp_after:wN \__fp_parse_infix_mul:N 17297
17298 \else: 17298
17299 \exp_after:wN \__fp_parse_infix_check:NNN 17299
17300 \cs:w 17300
17301 \__fp_parse_infix_ \token_to_str:N #2 :N 17301
17302 \exp_after:wN \exp_after:wN \exp_after:wN 17302
17303 \cs_end: 17303
17304 \fi: 17304
17305 \fi: 17305
17306 #1 17306
17307 #2 17307
17308 } 17308
17309 \cs_new:Npn \__fp_parse_infix_mark:NNN #1#2#3 { #3 #1 } 17309
17310 \cs_new:Npn \__fp_parse_infix_end:N #1 17310
17311 { @ \use_none:n \__fp_parse_infix_end:N } 17311
17312 \cs_set_protected:Npn \__fp_tmp:w #1 17312
17313 { 17313
17314 \cs_new:Npn #1 ##1 17314
17315 { 17315
17316 \if_int_compare:w ##1 > \c__fp_prec_end_int 17316
17317 \exp_after:wN @ 17317
17318 \exp_after:wN \use_none:n 17318
17319 \exp_after:wN #1 17319
17320 \else: 17320
17321 \msg_expandable_error:nnn { fp } { extra } { ) } 17321
17322 \exp_after:wN \__fp_parse_infix:NN 17322
17323 \exp_after:wN ##1 17323
17324 \exp:w \exp_after:wN \__fp_parse_expand:w 17324
17325 \fi: 17325
17326 } 17326
17327 } 17327
17328 \exp_args:Nc \__fp_tmp:w { __fp_parse_infix_):N } 17328
```



```
17329 \cs_set_protected:Npn \__fp_tmp:w #1
17330 {
17331   \cs_new:Npn #1 ##1
17332   {
17333     \if_int_compare:w ##1 > \c__fp_prec_comma_int
17334     \exp_after:wN @
17335     \exp_after:wN \use_none:n
17336     \exp_after:wN #1
17337   \else:
17338     \if_int_compare:w ##1 < \c__fp_prec_comma_int
17339     \exp_after:wN @
17340     \exp_after:wN \__fp_parse_apply_comma:NwNwN
17341     \exp_after:wN ,
17342     \exp:w
17343   \else:
17344     \exp_after:wN \__fp_parse_infix_comma:w
17345     \exp:w
17346   \fi:
17347   \__fp_parse_operand:Nw \c__fp_prec_comma_int
17348   \exp_after:wN \__fp_parse_expand:w
17349   \fi:
17350 }
17351 }
17352 \exp_args:Nc \__fp_tmp:w { \__fp_parse_infix_,:N }
17353 \cs_new:Npn \__fp_parse_infix_comma:w #1 @
17354 { #1 @ \use_none:n }
17355 \cs_new:Npn \__fp_parse_apply_comma:NwNwN #1 #2@ #3 #4@ #5
17356 {
17357   \exp_after:wN \__fp_parse_continue:NwN
17358   \exp_after:wN #1
17359   \exp:w \exp_end_continue_f:w
17360   \__fp_exp_after_tuple_f:nw { }
17361   \s__fp_tuple \__fp_tuple_chk:w { #2 #4 } \__fp_sep:
17362   #5 #1
17363 }
17364 \cs_set_protected:Npn \__fp_tmp:w #1#2#3#4
17365 {
17366   \cs_new:Npn #1 ##1
17367   {
17368     \if_int_compare:w ##1 < #3
17369     \exp_after:wN @
17370     \exp_after:wN \__fp_parse_apply_binary:NwNwN
17371     \exp_after:wN #2
17372     \exp:w
17373     \__fp_parse_operand:Nw #4
17374     \exp_after:wN \__fp_parse_expand:w
```

17375	\else:	17375
17376	\exp_after:wN @	17376
17377	\exp_after:wN \use_none:n	17377
17378	\exp_after:wN #1	17378
17379	\fi:	17379
17380	}	17380
17381	}	17381
17382	\exp_args:Nc __fp_tmp:w { __fp_parse_infix_^:N } ^	17382
17383	\c__fp_prec_hatii_int \c__fp_prec_hat_int	17383
17384	\exp_args:Nc __fp_tmp:w { __fp_parse_infix_juxt:N } *	17384
17385	\c__fp_prec_juxt_int \c__fp_prec_juxt_int	17385
17386	\exp_args:Nc __fp_tmp:w { __fp_parse_infix_/:N } /	17386
17387	\c__fp_prec_times_int \c__fp_prec_times_int	17387
17388	\exp_args:Nc __fp_tmp:w { __fp_parse_infix_mul:N } *	17388
17389	\c__fp_prec_times_int \c__fp_prec_times_int	17389
17390	\exp_args:Nc __fp_tmp:w { __fp_parse_infix -:N } -	17390
17391	\c__fp_prec_plus_int \c__fp_prec_plus_int	17391
17392	\exp_args:Nc __fp_tmp:w { __fp_parse_infix_+:N } +	17392
17393	\c__fp_prec_plus_int \c__fp_prec_plus_int	17393
17394	\exp_args:Nc __fp_tmp:w { __fp_parse_infix_and:N } &	17394
17395	\c__fp_prec_and_int \c__fp_prec_and_int	17395
17396	\exp_args:Nc __fp_tmp:w { __fp_parse_infix_or:N }	17396
17397	\c__fp_prec_or_int \c__fp_prec_or_int	17397
17398	\cs_new:cpn { __fp_parse_infix_(:N } #1	17398
17399	{ __fp_parse_infix_mul:N #1 (}	17399
17400	\cs_set_protected:Npn __fp_tmp:w #1	17400
17401	{	17401
17402	\cs_new:cpn { __fp_parse_infix_*:N } ##1##2	17402
17403	{	17403
17404	\if:w * \exp_not:N ##2	17404
17405	\exp_after:wN #1	17405
17406	\exp_after:wN ##1	17406
17407	\else:	17407
17408	\exp_after:wN __fp_parse_infix_mul:N	17408
17409	\exp_after:wN ##1	17409
17410	\exp_after:wN ##2	17410
17411	\fi:	17411
17412	}	17412
17413	}	17413
17414	\exp_args:Nc __fp_tmp:w { __fp_parse_infix_^:N }	17414
17415	\cs_set_protected:Npn __fp_tmp:w #1#2#3	17415
17416	{	17416
17417	\cs_new:Npn #1 ##1##2	17417
17418	{	17418
17419	\if:w #2 \exp_not:N ##2	17419
17420	\exp_after:wN #1	17420

```
17421         \exp_after:wN ##1
17422         \exp:w \exp_after:wN \__fp_parse_expand:w
17423     \else:
17424         \exp_after:wN #3
17425         \exp_after:wN ##1
17426         \exp_after:wN ##2
17427     \fi:
17428 }
17429 }
17430 \exp_args:Nc \__fp_tmp:w { __fp_parse_infix_|:N } | \__fp_parse_infix_or:N
17431 \exp_args:Nc \__fp_tmp:w { __fp_parse_infix_&:N } & \__fp_parse_infix_and:N
17432 \cs_set_protected:Npn \__fp_tmp:w #1#2#3#4
17433 {
17434     \cs_new:Npn #1 ##1
17435     {
17436         \if_int_compare:w ##1 < \c__fp_prec_quest_int
17437             #4
17438             \exp_after:wN @
17439             \exp_after:wN #2
17440             \exp:w
17441             \__fp_parse_operand:Nw #3
17442             \exp_after:wN \__fp_parse_expand:w
17443         \else:
17444             \exp_after:wN @
17445             \exp_after:wN \use_none:n
17446             \exp_after:wN #1
17447         \fi:
17448     }
17449 }
17450 \exp_args:Nc \__fp_tmp:w { __fp_parse_infix_?:N }
17451 \__fp_ternary:NwwN \c__fp_prec_quest_int { }
17452 \exp_args:Nc \__fp_tmp:w { __fp_parse_infix_:N }
17453 \__fp_ternary_auxii:NwwN \c__fp_prec_colon_int
17454 {
17455     \msg_expandable_error:nnnn
17456     { fp } { missing } { ? } { ~for~?: }
17457 }
17458 \cs_new:cpn { __fp_parse_infix_<:N } #1
17459 { \__fp_parse_compare:NNNNNNN #1 1 0 0 0 0 < }
17460 \cs_new:cpn { __fp_parse_infix_=:N } #1
17461 { \__fp_parse_compare:NNNNNNN #1 1 0 0 0 0 = }
17462 \cs_new:cpn { __fp_parse_infix_>:N } #1
17463 { \__fp_parse_compare:NNNNNNN #1 1 0 0 0 0 > }
17464 \cs_new:cpn { __fp_parse_infix_!:N } #1
17465 {
17466     \exp_after:wN \__fp_parse_compare:NNNNNNN
```

17467	\exp_after:wN #1	17467
17468	\exp_after:wN 0	17468
17469	\exp_after:wN 1	17469
17470	\exp_after:wN 1	17470
17471	\exp_after:wN 1	17471
17472	\exp_after:wN 1	17472
17473	}	17473
17474	\cs_new:Npn __fp_parse_excl_error:	17474
17475	{	17475
17476	\msg_expandable_error:nnnn	17476
17477	{ fp } { missing } { = } { ~after~!. }	17477
17478	}	17478
17479	\cs_new:Npn __fp_parse_compare:NNNNNNN #1	17479
17480	{	17480
17481	\if_int_compare:w #1 < \c__fp_prec_comp_int	17481
17482	\exp_after:wN __fp_parse_compare_auxi:NNNNNNN	17482
17483	\exp_after:wN __fp_parse_excl_error:	17483
17484	\else:	17484
17485	\exp_after:wN @	17485
17486	\exp_after:wN \use_none:n	17486
17487	\exp_after:wN __fp_parse_compare:NNNNNNN	17487
17488	\fi:	17488
17489	}	17489
17490	\cs_new:Npn __fp_parse_compare_auxi:NNNNNNN #1#2#3#4#5#6#7	17490
17491	{	17491
17492	\if_case:w	17492
17493	__fp_int_eval:w \exp_after:wN ` \token_to_str:N #7 - `<	17493
17494	__fp_int_eval_end:	17494
17495	__fp_parse_compare_auxii:NNNNN #2#2#4#5#6	17495
17496	\or: __fp_parse_compare_auxii:NNNNN #2#3#2#5#6	17496
17497	\or: __fp_parse_compare_auxii:NNNNN #2#3#4#2#6	17497
17498	\or: __fp_parse_compare_auxii:NNNNN #2#3#4#5#2	17498
17499	\else: #1 __fp_parse_compare_end:NNNNw #3#4#5#6#7	17499
17500	\fi:	17500
17501	}	17501
17502	\cs_new:Npn __fp_parse_compare_auxii:NNNNN #1#2#3#4#5	17502
17503	{	17503
17504	\exp_after:wN __fp_parse_compare_auxi:NNNNNNN	17504
17505	\exp_after:wN \prg_do_nothing:	17505
17506	\exp_after:wN #1	17506
17507	\exp_after:wN #2	17507
17508	\exp_after:wN #3	17508
17509	\exp_after:wN #4	17509
17510	\exp_after:wN #5	17510
17511	\exp:w \exp_after:wN __fp_parse_expand:w	17511
17512	}	17512

17513	\cs_new:Npn __fp_parse_compare_end:NNNNw #1#2#3#4#5 \fi:	17513
17514	{	17514
17515	\fi:	17515
17516	\exp_after:wN @	17516
17517	\exp_after:wN __fp_parse_apply_compare:NwNNNNNwN	17517
17518	\exp_after:wN \c_one_fp	17518
17519	\exp_after:wN #1	17519
17520	\exp_after:wN #2	17520
17521	\exp_after:wN #3	17521
17522	\exp_after:wN #4	17522
17523	\exp:w	17523
17524	__fp_parse_operand:Nw \c__fp_prec_comp_int __fp_parse_expand:w #5	17524
17525	}	17525
17526	\cs_new:Npn __fp_parse_apply_compare:NwNNNNNwN	17526
17527	#1 #2@ #3 #4#5#6#7 #8@ #9	17527
17528	{	17528
17529	\if_int_odd:w	17529
17530	\if_meaning:w \c_zero_fp #3	17530
17531	0	17531
17532	\else:	17532
17533	\if_case:w __fp_compare_back_any:ww #8 #2 \exp_stop_f:	17533
17534	#5 \or: #6 \or: #7 \else: #4	17534
17535	\fi:	17535
17536	\fi:	17536
17537	\exp_stop_f:	17537
17538	\exp_after:wN __fp_parse_apply_compare_aux:NNwN	17538
17539	\exp_after:wN \c_one_fp	17539
17540	\else:	17540
17541	\exp_after:wN __fp_parse_apply_compare_aux:NNwN	17541
17542	\exp_after:wN \c_zero_fp	17542
17543	\fi:	17543
17544	#1 #8 #9	17544
17545	}	17545
17546	\cs_new:Npn __fp_parse_apply_compare_aux:NNwN #1 #2 #3__fp_sep: #4	17546
17547	{	17547
17548	\if_meaning:w __fp_parse_compare:NNNNNNN #4	17548
17549	\exp_after:wN __fp_parse_continue_compare:NNwNN	17549
17550	\exp_after:wN #1	17550
17551	\exp_after:wN #2	17551
17552	\exp:w \exp_end_continue_f:w	17552
17553	__fp_exp_after_o:w #3__fp_sep:	17553
17554	\exp:w \exp_end_continue_f:w	17554
17555	\else:	17555
17556	\exp_after:wN __fp_parse_continue:NwN	17556
17557	\exp_after:wN #2	17557
17558	\exp:w \exp_end_continue_f:w	17558

17559	\exp_after:wN #1	17559
17560	\exp:w \exp_end_continue_f:w	17560
17561	\fi:	17561
17562	#4 #2	17562
17563	}	17563
17564	\cs_new:Npn __fp_parse_continue_compare:NNwNN #1#2 #3@ #4#5	17564
17565	{ #4 #2 #3@ #1 }	17565
17566	\cs_new:Npn __fp_parse_function_all_fp_o:fnw #1#2#3 @	17566
17567	{	17567
17568	__fp_array_if_all_fp:nTF {#3}	17568
17569	{ #2 #3 @ }	17569
17570	{	17570
17571	__fp_error:nffn { bad-args }	17571
17572	{#1}	17572
17573	{ \fp_to_tl:n { \s__fp_tuple __fp_tuple_chk:w {#3} __fp_sep: } }	17573
17574	{ }	17574
17575	\exp_after:wN \c_nan_fp	17575
17576	}	17576
17577	}	17577
17578	\cs_new:Npn __fp_parse_function_one_two:nnw #1#2#3	17578
17579	{	17579
17580	__fp_if_type_fp:NTwFw	17580
17581	#3 { } \s__fp __fp_parse_function_one_two_error_o:w \s__fp_stop	17581
17582	__fp_parse_function_one_two_aux:nnw {#1} {#2} #3	17582
17583	}	17583
17584	\cs_new:Npn __fp_parse_function_one_two_error_o:w #1#2#3#4 @	17584
17585	{	17585
17586	__fp_error:nffn { bad-args }	17586
17587	{#2}	17587
17588	{ \fp_to_tl:n { \s__fp_tuple __fp_tuple_chk:w {#4} __fp_sep: } }	17588
17589	{ }	17589
17590	\exp_after:wN \c_nan_fp	17590
17591	}	17591
17592	\cs_new:Npn __fp_parse_function_one_two_aux:nnw #1#2 #3__fp_sep: #4	17592
17593	{	17593
17594	__fp_if_type_fp:NTwFw	17594
17595	#4 { }	17595
17596	\s__fp	17596
17597	{	17597
17598	\if_meaning:w @ #4	17598
17599	\exp_after:wN \use_iv:nnnn	17599
17600	\fi:	17600
17601	__fp_parse_function_one_two_error_o:w	17601
17602	}	17602
17603	\s__fp_stop	17603
17604	__fp_parse_function_one_two_auxii:nnw {#1} {#2} #3__fp_sep: #4	17604

```
17605 } 17605
17606 \cs_new:Npn \__fp_parse_function_one_two_auxii:nw #1#2#3\__fp_sep: #4\__fp_sep: #5 17606
17607 { 17607
17608     \if_meaning:w @ #5 \else: 17608
17609     \exp_after:wN \__fp_parse_function_one_two_error_o:w 17609
17610     \fi: 17610
17611     \use_ii:nn {#1} { \use_none:n #2 } #3\__fp_sep: #4\__fp_sep: #5 17611
17612 } 17612
17613 \cs_new:Npn \__fp_tuple_map_o:nw #1 \s__fp_tuple \__fp_tuple_chk:w #2 \__fp_sep: 17613
17614 { 17614
17615     \exp_after:wN \s__fp_tuple 17615
17616     \exp_after:wN \__fp_tuple_chk:w 17616
17617     \exp_after:wN { 17617
17618         \exp:w \exp_end_continue_f:w 17618
17619         \__fp_tuple_map_loop_o:nw {#1} #2 17619
17620         { \s__fp \prg_break: } \__fp_sep: 17620
17621         \prg_break_point: 17621
17622         \exp_after:wN } \exp_after:wN \__fp_sep: 17622
17623     } 17623
17624 \cs_new:Npn \__fp_tuple_map_loop_o:nw #1#2#3 \__fp_sep: 17624
17625 { 17625
17626     \use_none:n #2 17626
17627     #1 #2 #3 \__fp_sep: 17627
17628     \exp:w \exp_end_continue_f:w 17628
17629     \__fp_tuple_map_loop_o:nw {#1} 17629
17630 } 17630
17631 \cs_new:Npn \__fp_tuple_mapthread_o:nww #1 17631
17632     \s__fp_tuple \__fp_tuple_chk:w #2 \__fp_sep: 17632
17633     \s__fp_tuple \__fp_tuple_chk:w #3 \__fp_sep: 17633
17634 { 17634
17635     \exp_after:wN \s__fp_tuple 17635
17636     \exp_after:wN \__fp_tuple_chk:w 17636
17637     \exp_after:wN { 17637
17638         \exp:w \exp_end_continue_f:w 17638
17639         \__fp_tuple_mapthread_loop_o:nw {#1} 17639
17640         #2 { \s__fp \prg_break: } \__fp_sep: @ 17640
17641         #3 { \s__fp \prg_break: } \__fp_sep: 17641
17642         \prg_break_point: 17642
17643         \exp_after:wN } \exp_after:wN \__fp_sep: 17643
17644     } 17644
17645 \cs_new:Npn \__fp_tuple_mapthread_loop_o:nw #1#2#3 \__fp_sep: #4 @ #5#6 \__fp_sep: 17645
17646 { 17646
17647     \use_none:n #2 17647
17648     \use_none:n #5 17648
17649     #1 #2 #3 \__fp_sep: #5 #6 \__fp_sep: 17649
17650     \exp:w \exp_end_continue_f:w 17650
```



```

17651 \__fp_tuple_mapthread_loop_o:nw {#1} #4 @
17652 }
17653 \msg_new:nnn { fp } { deprecated }
17654 { '#1'~deprecated;~use~'#2' }
17655 \msg_new:nnn { fp } { unknown-fp-word }
17656 { Unknown~fp~word~#1. }
17657 \msg_new:nnn { fp } { missing }
17658 { Missing~#1~inserted #2. }
17659 \msg_new:nnn { fp } { extra }
17660 { Extra~#1~ignored. }
17661 \msg_new:nnn { fp } { early-end }
17662 { Premature~end~in~fp~expression. }
17663 \msg_new:nnn { fp } { after-e }
17664 { Cannot~use~#1 after~'e'. }
17665 \msg_new:nnn { fp } { missing-number }
17666 { Missing~number~before~'#1'. }
17667 \msg_new:nnn { fp } { unknown-symbol }
17668 { Unknown~symbol~#1~ignored. }
17669 \msg_new:nnn { fp } { extra-comma }
17670 { Unexpected~comma~turned~to~nan~result. }
17671 \msg_new:nnn { fp } { no-arg }
17672 { #1~got~no~argument;~used~nan. }
17673 \msg_new:nnn { fp } { multi-arg }
17674 { #1~got~more~than~one~argument;~used~nan. }
17675 \msg_new:nnn { fp } { bad-args }
17676 { Arguments~in~#1#2~are~invalid. }
17677 \msg_new:nnn { fp } { infty-pi }
17678 { Math~command~#1 is~not~an~fp }
17679 \cs_if_exist:cT { @unexpandable@protect }
17680 {
17681 \msg_new:nnn { fp } { robust-cmd }
17682 { Robust~command~#1 invalid~in~fp~expression! }
17683 }
17684 %% File: l3fp-assign.dtx
17685 \cs_new_protected:Npn \fp_new:N #1
17686 { \cs_new_eq:NN #1 \c_zero_fp }
17687 \cs_generate_variant:Nn \fp_new:N {c}
17688 \cs_new_protected:Npn \fp_set:Nn #1#2
17689 { \__kernel_tl_set:Nx #1 { \exp_not:f { \__fp_parse:n {#2} } } }
17690 \cs_new_protected:Npn \fp_gset:Nn #1#2
17691 { \__kernel_tl_gset:Nx #1 { \exp_not:f { \__fp_parse:n {#2} } } }
17692 \cs_new_protected:Npn \fp_const:Nn #1#2
17693 { \tl_const:Ne #1 { \exp_not:f { \__fp_parse:n {#2} } } }
17694 \cs_generate_variant:Nn \fp_set:Nn { NV , c , cV }
17695 \cs_generate_variant:Nn \fp_gset:Nn { NV , c , cV }
17696 \cs_generate_variant:Nn \fp_const:Nn {c}

```

```
17697 \cs_new_eq:NN \fp_set_eq:NN \tl_set_eq:NN 17697
17698 \cs_new_eq:NN \fp_gset_eq:NN \tl_gset_eq:NN 17698
17699 \cs_generate_variant:Nn \fp_set_eq:NN { c , Nc , cc } 17699
17700 \cs_generate_variant:Nn \fp_gset_eq:NN { c , Nc , cc } 17700
17701 \cs_new_protected:Npn \fp_zero:N #1 { \fp_set_eq:NN #1 \c_zero_fp } 17701
17702 \cs_new_protected:Npn \fp_gzero:N #1 { \fp_gset_eq:NN #1 \c_zero_fp } 17702
17703 \cs_generate_variant:Nn \fp_zero:N { c } 17703
17704 \cs_generate_variant:Nn \fp_gzero:N { c } 17704
17705 \cs_new_protected:Npn \fp_zero_new:N #1 17705
17706 { \fp_if_exist:NTF #1 { \fp_zero:N #1 } { \fp_new:N #1 } } 17706
17707 \cs_new_protected:Npn \fp_gzero_new:N #1 17707
17708 { \fp_if_exist:NTF #1 { \fp_gzero:N #1 } { \fp_new:N #1 } } 17708
17709 \cs_generate_variant:Nn \fp_zero_new:N { c } 17709
17710 \cs_generate_variant:Nn \fp_gzero_new:N { c } 17710
17711 \cs_new_protected:Npn \fp_add:Nn { \__fp_add:NNNn \fp_set:Nn + } 17711
17712 \cs_new_protected:Npn \fp_gadd:Nn { \__fp_add:NNNn \fp_gset:Nn + } 17712
17713 \cs_new_protected:Npn \fp_sub:Nn { \__fp_add:NNNn \fp_set:Nn - } 17713
17714 \cs_new_protected:Npn \fp_gsub:Nn { \__fp_add:NNNn \fp_gset:Nn - } 17714
17715 \cs_new_protected:Npn \__fp_add:NNNn #1#2#3#4 17715
17716 { #1 #3 { #3 #2 \__fp_parse:n {#4} } } 17716
17717 \cs_generate_variant:Nn \fp_add:Nn { c } 17717
17718 \cs_generate_variant:Nn \fp_gadd:Nn { c } 17718
17719 \cs_generate_variant:Nn \fp_sub:Nn { c } 17719
17720 \cs_generate_variant:Nn \fp_gsub:Nn { c } 17720
17721 \cs_new_protected:Npn \fp_show:N { \__fp_show:NN \tl_show:n } 17721
17722 \cs_generate_variant:Nn \fp_show:N { c } 17722
17723 \cs_new_protected:Npn \fp_log:N { \__fp_show:NN \tl_log:n } 17723
17724 \cs_generate_variant:Nn \fp_log:N { c } 17724
17725 \cs_new_protected:Npn \__fp_show:NN #1#2 17725
17726 { 17726
17727 \__kernel_chk_tl_type:NnnT #2 { fp } 17727
17728 { \exp_args:No \__fp_show_validate:n #2 } 17728
17729 { \exp_args:Ne #1 { \token_to_str:N #2 = \fp_to_tl:N #2 } } 17729
17730 } 17730
17731 \cs_new:Npn \__fp_show_validate:n #1 17731
17732 { 17732
17733 \__fp_show_validate:nn { #1 } 17733
17734 { 17734
17735 \s__fp \__fp_chk:w ??? \__fp_sep: or \iow_newline: 17735
17736 \s__fp_tuple \__fp_tuple_chk:w ? \__fp_sep: or \iow_newline: 17736
17737 \s__fp_symbolic \__fp_symbolic_chk:w ? , ? \__fp_sep: 17737
17738 } 17738
17739 } 17739
17740 \cs_new:Npn \__fp_show_validate_aux:n #1 17740
17741 { 17741
17742 \__fp_show_validate:nn { #1 } { } 17742
```

```
17743 }
17744 \cs_new:Npn \__fp_show_validate:nn #1#2
17745 {
17746   \tl_if_empty:nF { #1 }
17747   {
17748     \str_case:enF { \tl_head:n { #1 } }
17749     {
17750       { \s__fp }
17751       {
17752         \__fp_show_validate:w #1 \s__fp
17753         \__fp_chk:w ??? \__fp_sep: \s__fp_stop
17754       }
17755       { \s__fp_tuple }
17756       {
17757         \__fp_tuple_show_validate:w #1
17758         \s__fp_tuple \__fp_tuple_chk:w ?? \__fp_sep: \s__fp_stop
17759       }
17760       { \s__fp_symbolic }
17761       {
17762         \__fp_symbolic_show_validate:w #1
17763         \s__fp_symbolic \__fp_symbolic_chk:w ? , ?? \__fp_sep: \s__fp_stop
17764       }
17765     }
17766     { #2 }
17767   }
17768 }
17769 \cs_new:Npn \__fp_show_validate:w
17770 #1 \s__fp \__fp_chk:w #2#3#4#5 \__fp_sep: #6 \s__fp_stop
17771 {
17772   \str_if_eq:nnF { #2 } {?}
17773   {
17774     \token_if_eq_meaning:NNTF #2 1
17775     { \s__fp \__fp_chk:w #2 #3 { #4 } #5 \__fp_sep: }
17776     { \s__fp \__fp_chk:w #2 #3 #4 #5 \__fp_sep: }
17777     \__fp_show_validate_aux:n { #6 }
17778   }
17779 }
17780 \cs_new:Npn \__fp_tuple_show_validate:w
17781 #1 \s__fp_tuple \__fp_tuple_chk:w #2#3 \__fp_sep: #4 \s__fp_stop
17782 {
17783   \str_if_eq:nnF { #2 } {?}
17784   {
17785     \s__fp_tuple \__fp_tuple_chk:w { \__fp_show_validate_aux:n { #2 } }
17786     \__fp_sep:
17787   }
17788 }
```

```
17789 \cs_new:Npn \__fp_symbolic_show_validate:w 17789
17790     #1 \s__fp_symbolic \__fp_symbolic_chk:w #2 , #3#4 \__fp_sep: #5 \s__fp_stop 17790
17791 { 17791
17792     \str_if_eq:nnF { #2 } {?} 17792
17793     { 17793
17794         \s__fp_symbolic \__fp_symbolic_chk:w \exp_not:n { #2 } , 17794
17795         { \__fp_show_validate_aux:n { #3 } }\__fp_sep: 17795
17796         \__fp_show_validate_aux:n { #5 } 17796
17797     } 17797
17798 } 17798
17799 \cs_new_protected:Npn \fp_show:n 17799
17800     { \__kernel_msg_show_eval:Nn \fp_to_tl:n } 17800
17801 \cs_new_protected:Npn \fp_log:n 17801
17802     { \__kernel_msg_log_eval:Nn \fp_to_tl:n } 17802
17803 \fp_const:Nn \c_e_fp { 2.718 2818 2845 9045 } 17803
17804 \fp_const:Nn \c_one_fp { 1 } 17804
17805 \fp_const:Nn \c_pi_fp { 3.141 5926 5358 9793 } 17805
17806 \fp_const:Nn \c_one_degree_fp { 0.0 1745 3292 5199 4330 } 17806
17807 \fp_new:N \l_tmpa_fp 17807
17808 \fp_new:N \l_tmpb_fp 17808
17809 \fp_new:N \g_tmpa_fp 17809
17810 \fp_new:N \g_tmpb_fp 17810
17811 %% File: l3fp-logic.dtx 17811
17812 \cs_new:Npn \__fp_parse_word_max:N 17812
17813     { \__fp_parse_function:NNN \__fp_minmax_o:Nw 2 } 17813
17814 \cs_new:Npn \__fp_parse_word_min:N 17814
17815     { \__fp_parse_function:NNN \__fp_minmax_o:Nw 0 } 17815
17816 \prg_new_eq_conditional:NNn \fp_if_exist:N \cs_if_exist:N { TF , T , F , p } 17816
17817 \prg_new_eq_conditional:NNn \fp_if_exist:c \cs_if_exist:c { TF , T , F , p } 17817
17818 \prg_new_conditional:Npnn \fp_if_nan:n #1 { TF , T , F , p } 17818
17819 { 17819
17820     \if:w 3 \exp_last_unbraced:Nf \__fp_kind:w { \__fp_parse:n {#1} } 17820
17821     \prg_return_true: 17821
17822     \else: 17822
17823         \prg_return_false: 17823
17824     \fi: 17824
17825 } 17825
17826 \prg_new_conditional:Npnn \fp_compare:n #1 { p , T , F , TF } 17826
17827 { 17827
17828     \exp_after:wN \__fp_compare_return:w 17828
17829     \exp:w \exp_end_continue_f:w \__fp_parse:n {#1} 17829
17830 } 17830
17831 \cs_new:Npn \__fp_compare_return:w #1#2#3\__fp_sep: 17831
17832 { 17832
17833     \if_charcode:w 0 17833
17834         \__fp_if_type_fp:NTwFw 17834
```

17835	#1 { __fp_use_i_delimit_by_s_stop:nw #3 \s__fp_stop }	17835
17836	\s__fp 1 \s__fp_stop	17836
17837	\prg_return_false:	17837
17838	\else:	17838
17839	\prg_return_true:	17839
17840	\fi:	17840
17841	}	17841
17842	\prg_new_conditional:Npnn \fp_compare:nNn #1#2#3 { p , T , F , TF }	17842
17843	{	17843
17844	\if_int_compare:w	17844
17845	\exp_after:wN __fp_compare_aux:wn	17845
17846	\exp:w \exp_end_continue_f:w __fp_parse:n {#1} {#3}	17846
17847	= __fp_int_eval:w `#2 - `= __fp_int_eval_end:	17847
17848	\prg_return_true:	17848
17849	\else:	17849
17850	\prg_return_false:	17850
17851	\fi:	17851
17852	}	17852
17853	\cs_new:Npn __fp_compare_aux:wn #1__fp_sep: #2	17853
17854	{	17854
17855	\exp_after:wN __fp_compare_back_any:ww	17855
17856	\exp:w \exp_end_continue_f:w __fp_parse:n {#2} #1__fp_sep:	17856
17857	}	17857
17858	\cs_new:Npn __fp_compare_back:ww #1#2__fp_sep: #3#4__fp_sep:	17858
17859	{	17859
17860	\cs:w	17860
17861	__fp	17861
17862	__fp_type_from_scan:N #1	17862
17863	_bcmp	17863
17864	__fp_type_from_scan:N #3	17864
17865	:ww	17865
17866	\cs_end:	17866
17867	#1#2__fp_sep: #3#4__fp_sep:	17867
17868	}	17868
17869	\cs_new:Npn __fp_compare_back_any:ww #1#2__fp_sep: #3	17869
17870	{	17870
17871	__fp_if_type_fp:NTwFw	17871
17872	#1 { __fp_if_type_fp:NTwFw #3 \use_i:nn \s__fp \use_ii:nn \s__fp_stop }	17872
17873	\s__fp \use_ii:nn \s__fp_stop	17873
17874	__fp_compare_back:ww	17874
17875	{	17875
17876	\cs:w	17876
17877	__fp	17877
17878	__fp_type_from_scan:N #1	17878
17879	_compare_back	17879
17880	__fp_type_from_scan:N #3	17880

```

17881         :ww
17882     \cs_end:
17883 }
17884 #1#2 \__fp_sep: #3
17885 }
17886 \cs_new:Npn \__fp_bcmp:ww
17887     \s__fp \__fp_chk:w #1 #2 #3\__fp_sep:
17888     \s__fp \__fp_chk:w #4 #5 #6\__fp_sep:
17889 {
17890     \int_value:w
17891     \if_meaning:w 3 #1 \exp_after:wN \__fp_compare_nan:w \fi:
17892     \if_meaning:w 3 #4 \exp_after:wN \__fp_compare_nan:w \fi:
17893     \if_meaning:w 2 #5 - \fi:
17894     \if_meaning:w #2 #5
17895     \if_meaning:w #1 #4
17896     \if_meaning:w 1 #1
17897         \__fp_compare_npos:nwnw #6\__fp_sep: #3\__fp_sep:
17898     \else:
17899         0
17900     \fi:
17901     \else:
17902         \if_int_compare:w #4 < #1 - \fi: 1
17903     \fi:
17904     \else:
17905         \if_int_compare:w #1#4 = \c_zero_int
17906         0
17907     \else:
17908         1
17909     \fi:
17910     \fi:
17911     \exp_stop_f:
17912 }
17913 \cs_new:Npn \__fp_compare_nan:w #1 \fi: \exp_stop_f: { 2 \exp_stop_f: }
17914 \cs_new:Npn \__fp_compare_back_tuple:ww #1\__fp_sep: #2\__fp_sep: { 2 }
17915 \cs_new:Npn \__fp_tuple_compare_back:ww #1\__fp_sep: #2\__fp_sep: { 2 }
17916 \cs_new:Npn \__fp_tuple_compare_back_tuple:ww
17917     \s__fp_tuple \__fp_tuple_chk:w #1\__fp_sep:
17918     \s__fp_tuple \__fp_tuple_chk:w #2\__fp_sep:
17919 {
17920     \int_compare:nNnTF { \__fp_array_count:n {#1} } =
17921         { \__fp_array_count:n {#2} }
17922     {
17923         \int_value:w 0
17924         \__fp_tuple_compare_back_loop:w
17925             #1 { \s__fp \prg_break: } \__fp_sep: @
17926             #2 { \s__fp \prg_break: } \__fp_sep:

```

17927	\prg_break_point:	17927
17928	\exp_stop_f:	17928
17929	}	17929
17930	{ 2 }	17930
17931	}	17931
17932	\cs_new:Npn __fp_tuple_compare_back_loop:w #1#2 __fp_sep: #3 @ #4#5 __fp_sep:	17932
17933	{	17933
17934	\use_none:n #1	17934
17935	\use_none:n #4	17935
17936	\if_int_compare:w	17936
17937	__fp_compare_back_any:ww #1 #2 __fp_sep: #4 #5 __fp_sep: = \c_zero_int	17937
17938	\else:	17938
17939	2 \exp_after:wN \prg_break:	17939
17940	\fi:	17940
17941	__fp_tuple_compare_back_loop:w #3 @	17941
17942	}	17942
17943	\cs_new:Npn __fp_compare_npos:nwnw #1#2__fp_sep: #3#4__fp_sep:	17943
17944	{	17944
17945	\if_int_compare:w #1 = #3 \exp_stop_f:	17945
17946	__fp_compare_significand:nnnnnnnn #2 #4	17946
17947	\else:	17947
17948	\if_int_compare:w #1 < #3 - \fi: 1	17948
17949	\fi:	17949
17950	}	17950
17951	\cs_new:Npn __fp_compare_significand:nnnnnnnn #1#2#3#4#5#6#7#8	17951
17952	{	17952
17953	\if_int_compare:w #1#2 = #5#6 \exp_stop_f:	17953
17954	\if_int_compare:w #3#4 = #7#8 \exp_stop_f:	17954
17955	0	17955
17956	\else:	17956
17957	\if_int_compare:w #3#4 < #7#8 - \fi: 1	17957
17958	\fi:	17958
17959	\else:	17959
17960	\if_int_compare:w #1#2 < #5#6 - \fi: 1	17960
17961	\fi:	17961
17962	}	17962
17963	\cs_new:Npn \fp_do_until:nn #1#2	17963
17964	{	17964
17965	#2	17965
17966	\fp_compare:nF {#1}	17966
17967	{ \fp_do_until:nn {#1} {#2} }	17967
17968	}	17968
17969	\cs_new:Npn \fp_do_while:nn #1#2	17969
17970	{	17970
17971	#2	17971
17972	\fp_compare:nT {#1}	17972

17973	{ \fp_do_while:nn {#1} {#2} }	17973
17974	}	17974
17975	\cs_new:Npn \fp_until_do:nn #1#2	17975
17976	{	17976
17977	\fp_compare:nF {#1}	17977
17978	{	17978
17979	#2	17979
17980	\fp_until_do:nn {#1} {#2}	17980
17981	}	17981
17982	}	17982
17983	\cs_new:Npn \fp_while_do:nn #1#2	17983
17984	{	17984
17985	\fp_compare:nT {#1}	17985
17986	{	17986
17987	#2	17987
17988	\fp_while_do:nn {#1} {#2}	17988
17989	}	17989
17990	}	17990
17991	\cs_new:Npn \fp_do_until:nNnn #1#2#3#4	17991
17992	{	17992
17993	#4	17993
17994	\fp_compare:nNnF {#1} #2 {#3}	17994
17995	{ \fp_do_until:nNnn {#1} #2 {#3} {#4} }	17995
17996	}	17996
17997	\cs_new:Npn \fp_do_while:nNnn #1#2#3#4	17997
17998	{	17998
17999	#4	17999
18000	\fp_compare:nNnT {#1} #2 {#3}	18000
18001	{ \fp_do_while:nNnn {#1} #2 {#3} {#4} }	18001
18002	}	18002
18003	\cs_new:Npn \fp_until_do:nNnn #1#2#3#4	18003
18004	{	18004
18005	\fp_compare:nNnF {#1} #2 {#3}	18005
18006	{	18006
18007	#4	18007
18008	\fp_until_do:nNnn {#1} #2 {#3} {#4}	18008
18009	}	18009
18010	}	18010
18011	\cs_new:Npn \fp_while_do:nNnn #1#2#3#4	18011
18012	{	18012
18013	\fp_compare:nNnT {#1} #2 {#3}	18013
18014	{	18014
18015	#4	18015
18016	\fp_while_do:nNnn {#1} #2 {#3} {#4}	18016
18017	}	18017
18018	}	18018

```
18019 \cs_new:Npn \fp_step_function:nnnN #1#2#3
18020 {
18021   \exp_after:wN \__fp_step:wwwN
18022   \exp:w \exp_end_continue_f:w \__fp_parse_o:n {#1}
18023   \exp:w \exp_end_continue_f:w \__fp_parse_o:n {#2}
18024   \exp:w \exp_end_continue_f:w \__fp_parse:n {#3}
18025 }
18026 \cs_generate_variant:Nn \fp_step_function:nnnN { nnnC }
18027 \cs_new:Npn \__fp_step:wwwN #1#2\__fp_sep: #3#4\__fp_sep: #5#6\__fp_sep: #7
18028 {
18029   \__fp_if_type_fp:NTwFw #1 { } \s__fp \prg_break: \s__fp_stop
18030   \__fp_if_type_fp:NTwFw #3 { } \s__fp \prg_break: \s__fp_stop
18031   \__fp_if_type_fp:NTwFw #5 { } \s__fp \prg_break: \s__fp_stop
18032   \use_i:nnnn
18033   { \__fp_step_fp:wwwN #1#2\__fp_sep: #3#4\__fp_sep: #5#6\__fp_sep: #7 }
18034   \prg_break_point:
18035   \use:n
18036   {
18037     \__fp_error:nfff { step-tuple } { \fp_to_tl:n { #1#2 \__fp_sep: } }
18038     { \fp_to_tl:n { #3#4 \__fp_sep: } } { \fp_to_tl:n { #5#6 \__fp_sep: } }
18039   }
18040 }
18041 \cs_new:Npn \__fp_step_fp:wwwN
18042 #1 \__fp_sep: \s__fp \__fp_chk:w #2#3#4 \__fp_sep: #5\__fp_sep: #6
18043 {
18044   \token_if_eq_meaning:NNTF #2 1
18045   {
18046     \token_if_eq_meaning:NNTF #3 0
18047     { \__fp_step:NnnnnN > }
18048     { \__fp_step:NnnnnN < }
18049   }
18050   {
18051     \token_if_eq_meaning:NNTF #2 0
18052     {
18053       \msg_expandable_error:nnn { kernel }
18054       { zero-step } {#6}
18055     }
18056     {
18057       \__fp_error:nnfn { bad-step } { }
18058       { \fp_to_tl:n { \s__fp \__fp_chk:w #2#3#4 \__fp_sep: } } {#6}
18059     }
18060     \use_none:nnnnn
18061   }
18062   { #1 \__fp_sep: }
18063   { \c_nan_fp }
18064   { \s__fp \__fp_chk:w #2#3#4 \__fp_sep: }
```

```
18065         { #5 \__fp_sep: } 18065
18066         #6 18066
18067     } 18067
18068 \cs_new:Npn \__fp_step:NnnnnN #1#2#3#4#5#6 18068
18069 { 18069
18070     \fp_compare:nNnTF {#2} = {#3} 18070
18071     { 18071
18072         \__fp_error:nffn { tiny-step } 18072
18073         { \fp_to_tl:n {#3} } { \fp_to_tl:n {#4} } {#6} 18073
18074     } 18074
18075     { 18075
18076         \fp_compare:nNnF {#2} #1 {#5} 18076
18077         { 18077
18078             \exp_args:Nf #6 { \__fp_to_decimal_dispatch:w #2 } 18078
18079             \__fp_step:NfnnnN 18079
18080             #1 { \__fp_parse:n { #2 + #4 } } {#2} {#4} {#5} #6 18080
18081         } 18081
18082     } 18082
18083 } 18083
18084 \cs_generate_variant:Nn \__fp_step:NnnnnN { Nf } 18084
18085 \cs_new_protected:Npn \fp_step_inline:nnnn 18085
18086 { 18086
18087     \int_gincr:N \g__kernel_prg_map_int 18087
18088     \exp_args:NNc \__fp_step:NNnnnn 18088
18089     \cs_gset_protected:Npn 18089
18090     { __fp_map_ \int_use:N \g__kernel_prg_map_int :w } 18090
18091 } 18091
18092 \cs_new_protected:Npn \fp_step_variable:nnnNn #1#2#3#4#5 18092
18093 { 18093
18094     \int_gincr:N \g__kernel_prg_map_int 18094
18095     \exp_args:NNc \__fp_step:NNnnnn 18095
18096     \cs_gset_protected:Npe 18096
18097     { __fp_map_ \int_use:N \g__kernel_prg_map_int :w } 18097
18098     {#1} {#2} {#3} 18098
18099     { 18099
18100         \tl_set:Nn \exp_not:N #4 {##1} 18100
18101         \exp_not:n {#5} 18101
18102     } 18102
18103 } 18103
18104 \cs_new_protected:Npn \__fp_step:NNnnnn #1#2#3#4#5#6 18104
18105 { 18105
18106     #1 #2 ##1 {#6} 18106
18107     \fp_step_function:nnnN {#3} {#4} {#5} #2 18107
18108     \prg_break_point:Nn \scan_stop: { \int_gdecr:N \g__kernel_prg_map_int } 18108
18109 } 18109
18110 \msg_new:nnn { fp } { step-tuple } 18110
```

18111	{ Tuple~argument~in~fp_step...~{#1}{#2}{#3}. }	18111
18112	\msg_new:nnn { fp } { bad-step }	18112
18113	{ Invalid~step~size~#2~for~function~#3. }	18113
18114	\msg_new:nnn { fp } { tiny-step }	18114
18115	{ Tiny~step~size~(#1+#2=#1)~for~function~#3. }	18115
18116	\cs_new:Npn __fp_minmax_o:Nw #1	18116
18117	{	18117
18118	__fp_parse_function_all_fp_o:fnw	18118
18119	{ \token_if_eq_meaning:NNTF 0 #1 { min } { max } }	18119
18120	{ __fp_minmax_aux_o:Nw #1 }	18120
18121	}	18121
18122	\cs_new:Npn __fp_minmax_aux_o:Nw #1#2 @	18122
18123	{	18123
18124	\if_meaning:w 0 #1	18124
18125	\exp_after:wN __fp_minmax_loop:Nww \exp_after:wN +	18125
18126	\else:	18126
18127	\exp_after:wN __fp_minmax_loop:Nww \exp_after:wN -	18127
18128	\fi:	18128
18129	#2	18129
18130	\s__fp __fp_chk:w 2 #1 \s__fp_exact __fp_sep:	18130
18131	\s__fp __fp_chk:w { 3 __fp_minmax_break_o:w } __fp_sep:	18131
18132	}	18132
18133	\cs_new:Npn __fp_minmax_loop:Nww	18133
18134	#1 \s__fp __fp_chk:w #2#3__fp_sep: \s__fp __fp_chk:w #4#5__fp_sep:	18134
18135	{	18135
18136	\if_meaning:w 3 #4	18136
18137	\if_meaning:w 3 #2	18137
18138	__fp_minmax_auxi:ww	18138
18139	\else:	18139
18140	__fp_minmax_auxii:ww	18140
18141	\fi:	18141
18142	\else:	18142
18143	\if_int_compare:w	18143
18144	__fp_compare_back:ww	18144
18145	\s__fp __fp_chk:w #4#5__fp_sep:	18145
18146	\s__fp __fp_chk:w #2#3__fp_sep:	18146
18147	= #1 1 \exp_stop_f:	18147
18148	__fp_minmax_auxii:ww	18148
18149	\else:	18149
18150	__fp_minmax_auxi:ww	18150
18151	\fi:	18151
18152	\fi:	18152
18153	__fp_minmax_loop:Nww #1	18153
18154	\s__fp __fp_chk:w #2#3__fp_sep:	18154
18155	\s__fp __fp_chk:w #4#5__fp_sep:	18155
18156	}	18156

18157	\cs_new:Npn __fp_minmax_auxi:ww	18157
18158	#1 \fi: \fi: #2 \s__fp #3 __fp_sep: \s__fp #4__fp_sep:	18158
18159	{ \fi: \fi: #2 \s__fp #3 __fp_sep: }	18159
18160	\cs_new:Npn __fp_minmax_auxii:ww #1 \fi: \fi: #2 \s__fp #3 __fp_sep:	18160
18161	{ \fi: \fi: #2 }	18161
18162	\cs_new:Npn __fp_minmax_break_o:w #1 \fi: \fi: #2 \s__fp #3__fp_sep: #4__fp_sep:	18162
18163	{ \fi: __fp_exp_after_o:w \s__fp #3__fp_sep: }	18163
18164	\cs_new:Npn __fp_not_o:w #1 \s__fp __fp_chk:w #2#3__fp_sep: @	18164
18165	{	18165
18166	\if_meaning:w 0 #2	18166
18167	\exp_after:wN \exp_after:wN \exp_after:wN \c_one_fp	18167
18168	\else:	18168
18169	\exp_after:wN \exp_after:wN \exp_after:wN \c_zero_fp	18169
18170	\fi:	18170
18171	}	18171
18172	\cs_new:Npn __fp_tuple_not_o:w #1 @ { \exp_after:wN \c_zero_fp }	18172
18173	\group_begin:	18173
18174	\char_set_catcode_letter:N &	18174
18175	\char_set_catcode_letter:N	18175
18176	\cs_new:Npn __fp_&_o:ww #1 \s__fp __fp_chk:w #2#3__fp_sep:	18176
18177	{	18177
18178	\if_meaning:w 0 #2 #1	18178
18179	__fp_and_return:wNw \s__fp __fp_chk:w #2#3__fp_sep:	18179
18180	\fi:	18180
18181	__fp_exp_after_o:w	18181
18182	}	18182
18183	\cs_new:Npn __fp_&_tuple_o:ww #1 \s__fp __fp_chk:w #2#3__fp_sep:	18183
18184	{	18184
18185	\if_meaning:w 0 #2 #1	18185
18186	__fp_and_return:wNw \s__fp __fp_chk:w #2#3__fp_sep:	18186
18187	\fi:	18187
18188	__fp_exp_after_tuple_o:w	18188
18189	}	18189
18190	\cs_new:Npn __fp_tuple_&_o:ww #1__fp_sep: { __fp_exp_after_o:w }	18190
18191	\cs_new:Npn __fp_tuple_&_tuple_o:ww #1__fp_sep: { __fp_exp_after_tuple_o:w }	18191
18192	\cs_new:Npn __fp_ _o:ww { __fp_&_o:ww \else: }	18192
18193	\cs_new:Npn __fp_ _tuple_o:ww { __fp_&_tuple_o:ww \else: }	18193
18194	\cs_new:Npn __fp_tuple_ _o:ww #1__fp_sep: #2__fp_sep:	18194
18195	{ __fp_exp_after_tuple_o:w #1__fp_sep: }	18195
18196	\cs_new:Npn __fp_tuple_ _tuple_o:ww #1__fp_sep: #2__fp_sep:	18196
18197	{ __fp_exp_after_tuple_o:w #1__fp_sep: }	18197
18198	\group_end:	18198
18199	\cs_new:Npn __fp_and_return:wNw #1__fp_sep: \fi: #2__fp_sep:	18199
18200	{ \fi: __fp_exp_after_o:w #1__fp_sep: }	18200
18201	\cs_new:Npn __fp_ternary:NwwN #1 #2#3@ #4@ #5	18201
18202	{	18202

```
18203 \if_meaning:w \__fp_parse_infix_:N #5 18203
18204 \if_charcode:w 0 18204
18205 \__fp_if_type_fp:NTwFw 18205
18206 #2 { \use_i:nn \__fp_use_i_delimit_by_s_stop:nw #3 \s__fp_stop } 18206
18207 \s__fp 1 \s__fp_stop 18207
18208 \exp_after:wN \exp_after:wN \exp_after:wN \__fp_ternary_auxii:NwwN 18208
18209 \else: 18209
18210 \exp_after:wN \exp_after:wN \exp_after:wN \__fp_ternary_auxi:NwwN 18210
18211 \fi: 18211
18212 \exp_after:wN #1 18212
18213 \exp:w \exp_end_continue_f:w 18213
18214 \__fp_exp_after_array_f:w #4 \s__fp_expr_stop 18214
18215 \exp_after:wN @ 18215
18216 \exp:w 18216
18217 \__fp_parse_operand:Nw \c__fp_prec_colon_int 18217
18218 \__fp_parse_expand:w 18218
18219 \else: 18219
18220 \msg_expandable_error:nnnn 18220
18221 { fp } { missing } { : } { ~for~?: } 18221
18222 \exp_after:wN \__fp_parse_continue:NwN 18222
18223 \exp_after:wN #1 18223
18224 \exp:w \exp_end_continue_f:w 18224
18225 \__fp_exp_after_array_f:w #4 \s__fp_expr_stop 18225
18226 \exp_after:wN #5 18226
18227 \exp_after:wN #1 18227
18228 \fi: 18228
18229 } 18229
18230 \cs_new:Npn \__fp_ternary_auxi:NwwN #1#2@#3@#4 18230
18231 { 18231
18232 \exp_after:wN \__fp_parse_continue:NwN 18232
18233 \exp_after:wN #1 18233
18234 \exp:w \exp_end_continue_f:w 18234
18235 \__fp_exp_after_array_f:w #2 \s__fp_expr_stop 18235
18236 #4 #1 18236
18237 } 18237
18238 \cs_new:Npn \__fp_ternary_auxii:NwwN #1#2@#3@#4 18238
18239 { 18239
18240 \exp_after:wN \__fp_parse_continue:NwN 18240
18241 \exp_after:wN #1 18241
18242 \exp:w \exp_end_continue_f:w 18242
18243 \__fp_exp_after_array_f:w #3 \s__fp_expr_stop 18243
18244 #4 #1 18244
18245 } 18245
18246 %% File: l3fp-basics.dtx 18246
18247 \cs_new:Npn \__fp_parse_word_abs:N 18247
18248 { \__fp_parse_unary_function:NNN \__fp_set_sign_o:w 0 } 18248
```

```
18249 \cs_new:Npn \__fp_parse_word_logb:N 18249
18250 { \__fp_parse_unary_function:NNN \__fp_logb_o:w ? } 18250
18251 \cs_new:Npn \__fp_parse_word_sign:N 18251
18252 { \__fp_parse_unary_function:NNN \__fp_sign_o:w ? } 18252
18253 \cs_new:Npn \__fp_parse_word_sqrt:N 18253
18254 { \__fp_parse_unary_function:NNN \__fp_sqrt_o:w ? } 18254
18255 \cs_new:cpe { __fp_-_o:ww } \s__fp 18255
18256 { 18256
18257 \exp_not:c { __fp+_o:ww } 18257
18258 \exp_not:n { \s__fp \__fp_neg_sign:N } 18258
18259 } 18259
18260 \cs_new:cpn { __fp+_o:ww } 18260
18261 \s__fp #1 \__fp_chk:w #2 #3 \__fp_sep: \s__fp \__fp_chk:w #4 #5 18261
18262 { 18262
18263 \if_case:w 18263
18264 \if_meaning:w #2 #4 18264
18265 #2 18265
18266 \else: 18266
18267 \if_int_compare:w #2 > #4 \exp_stop_f: 18267
18268 3 18268
18269 \else: 18269
18270 4 18270
18271 \fi: 18271
18272 \fi: 18272
18273 \exp_stop_f: 18273
18274 \exp_after:wN \__fp_add_zeros_o:Nww \int_value:w 18274
18275 \or: \exp_after:wN \__fp_add_normal_o:Nww \int_value:w 18275
18276 \or: \exp_after:wN \__fp_add_inf_o:Nww \int_value:w 18276
18277 \or: \__fp_case_return_i_o:ww 18277
18278 \else: \exp_after:wN \__fp_add_return_ii_o:Nww \int_value:w 18278
18279 \fi: 18279
18280 #1 #5 18280
18281 \s__fp \__fp_chk:w #2 #3 \__fp_sep: 18281
18282 \s__fp \__fp_chk:w #4 #5 18282
18283 } 18283
18284 \cs_new:Npn \__fp_add_return_ii_o:Nww #1 #2 \__fp_sep: \s__fp \__fp_chk:w #3 #4 18284
18285 { \__fp_exp_after_o:w \s__fp \__fp_chk:w #3 #1 } 18285
18286 \cs_new:Npn \__fp_add_zeros_o:Nww #1 \s__fp \__fp_chk:w 0 #2 18286
18287 { 18287
18288 \if_int_compare:w #2 #1 = 20 \exp_stop_f: 18288
18289 \exp_after:wN \__fp_add_return_ii_o:Nww 18289
18290 \else: 18290
18291 \__fp_case_return_i_o:ww 18291
18292 \fi: 18292
18293 #1 18293
18294 \s__fp \__fp_chk:w 0 #2 18294
```


18295	}	18295
18296	\cs_new:Npn __fp_add_inf_o:Nww	18296
18297	#1 \s__fp __fp_chk:w 2 #2 #3__fp_sep: \s__fp __fp_chk:w 2 #4	18297
18298	{	18298
18299	\if_meaning:w #1 #2	18299
18300	__fp_case_return_i_o:ww	18300
18301	\else:	18301
18302	__fp_case_use:nw	18302
18303	{	18303
18304	\exp_last_unbraced:Nf __fp_invalid_operation_o:Nww	18304
18305	{ \token_if_eq_meaning:NNTF #1 #4 + - }	18305
18306	}	18306
18307	\fi:	18307
18308	\s__fp __fp_chk:w 2 #2 #3__fp_sep:	18308
18309	\s__fp __fp_chk:w 2 #4	18309
18310	}	18310
18311	\cs_new:Npn __fp_add_normal_o:Nww #1 \s__fp __fp_chk:w 1 #2	18311
18312	{	18312
18313	\if_meaning:w #1#2	18313
18314	\exp_after:wN __fp_add_npos_o:NnwNnw	18314
18315	\else:	18315
18316	\exp_after:wN __fp_sub_npos_o:NnwNnw	18316
18317	\fi:	18317
18318	#2	18318
18319	}	18319
18320	\cs_new:Npn __fp_add_npos_o:NnwNnw #1#2#3 __fp_sep: \s__fp __fp_chk:w 1 #4 #5	18320
18321	{	18321
18322	\exp_after:wN __fp_sanitize:Nw	18322
18323	\exp_after:wN #1	18323
18324	\int_value:w __fp_int_eval:w	18324
18325	\if_int_compare:w #2 > #5 \exp_stop_f:	18325
18326	#2	18326
18327	\exp_after:wN __fp_add_big_i_o:wNww \int_value:w -	18327
18328	\else:	18328
18329	#5	18329
18330	\exp_after:wN __fp_add_big_ii_o:wNww \int_value:w	18330
18331	\fi:	18331
18332	__fp_int_eval:w #5 - #2 __fp_sep: #1 #3__fp_sep:	18332
18333	}	18333
18334	\cs_new:Npn __fp_add_big_i_o:wNww #1__fp_sep: #2 #3__fp_sep: #4__fp_sep:	18334
18335	{	18335
18336	__fp_decimate:nNnnnn {#1}	18336
18337	__fp_add_significand_o:NnnwnnnnN	18337
18338	#4	18338
18339	#3	18339
18340	#2	18340

18341 } 18341
18342 \cs_new:Npn __fp_add_big_ii_o:wNww #1__fp_sep: #2 #3__fp_sep: #4__fp_sep: 18342
18343 { 18343
18344 __fp_decimate:nNnnnn {#1} 18344
18345 __fp_add_significand_o:NnnwnnnnN 18345
18346 #3 18346
18347 #4 18347
18348 #2 18348
18349 } 18349
18350 \cs_new:Npn __fp_add_significand_o:NnnwnnnnN #1 #2#3 #4__fp_sep: #5#6#7#8 18350
18351 { 18351
18352 \exp_after:wN __fp_add_significand_test_o:N 18352
18353 \int_value:w __fp_int_eval:w 1#5#6 + #2 18353
18354 \exp_after:wN __fp_add_significand_pack:NNNNNNN 18354
18355 \int_value:w __fp_int_eval:w 1#7#8 + #3 __fp_sep: #1 18355
18356 } 18356
18357 \cs_new:Npn __fp_add_significand_pack:NNNNNNN #1 #2#3#4#5#6#7 18357
18358 { 18358
18359 \if_meaning:w 2 #1 18359
18360 + 1 18360
18361 \fi: 18361
18362 __fp_sep: #2 #3 #4 #5 #6 #7 __fp_sep: 18362
18363 } 18363
18364 \cs_new:Npn __fp_add_significand_test_o:N #1 18364
18365 { 18365
18366 \if_meaning:w 2 #1 18366
18367 \exp_after:wN __fp_add_significand_carry_o:wwwNN 18367
18368 \else: 18368
18369 \exp_after:wN __fp_add_significand_no_carry_o:wwwNN 18369
18370 \fi: 18370
18371 } 18371
18372 \cs_new:Npn __fp_add_significand_no_carry_o:wwwNN 18372
18373 #1__fp_sep: #2__fp_sep: #3#4 __fp_sep: #5#6 18373
18374 { 18374
18375 \exp_after:wN __fp_basics_pack_high:NNNNNw 18375
18376 \int_value:w __fp_int_eval:w 1 #1 18376
18377 \exp_after:wN __fp_basics_pack_low:NNNNNw 18377
18378 \int_value:w __fp_int_eval:w 1 #2 #3#4 18378
18379 + __fp_round:NNN #6 #4 #5 18379
18380 \exp_after:wN __fp_sep: 18380
18381 } 18381
18382 \cs_new:Npn __fp_add_significand_carry_o:wwwNN 18382
18383 #1__fp_sep: #2__fp_sep: #3#4__fp_sep: #5#6 18383
18384 { 18384
18385 + 1 18385
18386 \exp_after:wN __fp_basics_pack_weird_high:NNNNNNNNw 18386

18387	\int_value:w __fp_int_eval:w 1 1 #1	18387
18388	\exp_after:wN __fp_basics_pack_weird_low:NNNNw	18388
18389	\int_value:w __fp_int_eval:w 1 #2#3 +	18389
18390	\exp_after:wN __fp_round:NNN	18390
18391	\exp_after:wN #6	18391
18392	\exp_after:wN #3	18392
18393	\int_value:w __fp_round_digit:Nw #4 #5 __fp_sep:	18393
18394	\exp_after:wN __fp_sep:	18394
18395	}	18395
18396	\cs_new:Npn __fp_sub_npos_o:NnwNnw	18396
18397	#1#2#3__fp_sep: \s__fp __fp_chk:w 1 #4#5#6__fp_sep:	18397
18398	{	18398
18399	\if_case:w	18399
18400	__fp_compare_npos:nwnw {#2} #3__fp_sep: {#5} #6__fp_sep: \exp_stop_f:	18400
18401	\exp_after:wN __fp_sub_eq_o:Nnwnw	18401
18402	\or:	18402
18403	\exp_after:wN __fp_sub_npos_i_o:Nnwnw	18403
18404	\else:	18404
18405	\exp_after:wN __fp_sub_npos_ii_o:Nnwnw	18405
18406	\fi:	18406
18407	#1 {#2} #3__fp_sep: {#5} #6__fp_sep:	18407
18408	}	18408
18409	\cs_new:Npn __fp_sub_eq_o:Nnwnw #1#2__fp_sep: #3__fp_sep:	18409
18410	{ \exp_after:wN \c_zero_fp }	18410
18411	\cs_new:Npn __fp_sub_npos_ii_o:Nnwnw #1 #2__fp_sep: #3__fp_sep:	18411
18412	{	18412
18413	\exp_after:wN __fp_sub_npos_i_o:Nnwnw	18413
18414	\int_value:w __fp_neg_sign:N #1	18414
18415	#3__fp_sep: #2__fp_sep:	18415
18416	}	18416
18417	\cs_new:Npn __fp_sub_npos_i_o:Nnwnw #1 #2#3__fp_sep: #4#5__fp_sep:	18417
18418	{	18418
18419	\exp_after:wN __fp_sanitize:Nw	18419
18420	\exp_after:wN #1	18420
18421	\int_value:w __fp_int_eval:w	18421
18422	#2	18422
18423	\if_int_compare:w #2 = #4 \exp_stop_f:	18423
18424	\exp_after:wN __fp_sub_back_near_o:nnnnnnnnN	18424
18425	\else:	18425
18426	\exp_after:wN __fp_decimate:nNnnnn \exp_after:wN	18426
18427	{ \int_value:w __fp_int_eval:w #2 - #4 - 1 \exp_after:wN }	18427
18428	\exp_after:wN __fp_sub_back_far_o:NnnwnnnnnN	18428
18429	\fi:	18429
18430	#5	18430
18431	#3	18431
18432	#1	18432

18433	}	18433
18434	\cs_new:Npn __fp_sub_back_near_o:nnnnnnnnN #1#2#3#4 #5#6#7#8 #9	18434
18435	{	18435
18436	\exp_after:wN __fp_sub_back_near_after:wNNNNw	18436
18437	\int_value:w __fp_int_eval:w 10#5#6 - #1#2 - 11	18437
18438	\exp_after:wN __fp_sub_back_near_pack:NNNNNNw	18438
18439	\int_value:w __fp_int_eval:w 11#7#8 - #3#4 \exp_after:wN __fp_sep:	18439
18440	}	18440
18441	\cs_new:Npn __fp_sub_back_near_pack:NNNNNNw #1#2#3#4#5#6#7 __fp_sep:	18441
18442	{ + #1#2 __fp_sep: {#3#4#5#6} {#7} __fp_sep: }	18442
18443	\cs_new:Npn __fp_sub_back_near_after:wNNNNw 10 #1#2#3#4 #5 __fp_sep:	18443
18444	{	18444
18445	\if_meaning:w 0 #1	18445
18446	\exp_after:wN __fp_sub_back_shift:wnnnn	18446
18447	\fi:	18447
18448	__fp_sep: {#1#2#3#4} {#5}	18448
18449	}	18449
18450	\cs_new:Npn __fp_sub_back_shift:wnnnn __fp_sep: #1#2	18450
18451	{	18451
18452	\exp_after:wN __fp_sub_back_shift_ii:ww	18452
18453	\int_value:w #1 #2 0 __fp_sep:	18453
18454	}	18454
18455	\cs_new:Npn __fp_sub_back_shift_ii:ww #1 0 __fp_sep: #2#3 __fp_sep:	18455
18456	{	18456
18457	\if_meaning:w @ #1 @	18457
18458	- 7	18458
18459	- \exp_after:wN \use_i:nnn	18459
18460	\exp_after:wN __fp_sub_back_shift_iii:NNNNNNNNw	18460
18461	\int_value:w #2#3 0 ~ 123456789__fp_sep:	18461
18462	\else:	18462
18463	- __fp_sub_back_shift_iii:NNNNNNNNw #1 123456789__fp_sep:	18463
18464	\fi:	18464
18465	\exp_after:wN __fp_pack_twice_four:wNNNNNNNN	18465
18466	\exp_after:wN __fp_pack_twice_four:wNNNNNNNN	18466
18467	\exp_after:wN __fp_sub_back_shift_iv:nnnnw	18467
18468	\exp_after:wN __fp_sep:	18468
18469	\int_value:w	18469
18470	#1 ~ #2#3 0 ~ 0000 0000 0000 000 __fp_sep:	18470
18471	}	18471
18472	\cs_new:Npn __fp_sub_back_shift_iii:NNNNNNNNw #1#2#3#4#5#6#7#8#9__fp_sep: {#8}	18472
18473	\cs_new:Npn __fp_sub_back_shift_iv:nnnnw #1 __fp_sep: #2 __fp_sep:	18473
18474	{ __fp_sep: #1 __fp_sep: }	18474
18475	\cs_new:Npn __fp_sub_back_far_o:NnnwnnnnN #1 #2#3 #4__fp_sep: #5#6#7#8	18475
18476	{	18476
18477	\if_case:w	18477
18478	\if_int_compare:w 1 #2 = #5#6 \use_i:nnnn #7 \exp_stop_f:	18478

```
18479         \if_int_compare:w #3 = \use_none:n #7#8 0 \exp_stop_f: 18479
18480             0 18480
18481         \else: 18481
18482             \if_int_compare:w #3 > \use_none:n #7#8 0 - \fi: 1 18482
18483             \fi: 18483
18484         \else: 18484
18485             \if_int_compare:w 1 #2 > #5#6 \use_i:nnnn #7 - \fi: 1 18485
18486             \fi: 18486
18487         \exp_stop_f: 18487
18488             \exp_after:wN \__fp_sub_back_quite_far_o:wwNN 18488
18489         \or: \exp_after:wN \__fp_sub_back_very_far_o:wwwNN 18489
18490         \else: \exp_after:wN \__fp_sub_back_not_far_o:wwwNN 18490
18491         \fi: 18491
18492         #2 ~ #3 \__fp_sep: #5 #6 ~ #7 #8 \__fp_sep: #1 18492
18493     } 18493
18494     \cs_new:Npn \__fp_sub_back_quite_far_o:wwNN #1\__fp_sep: #2\__fp_sep: #3#4 18494
18495     { 18495
18496         \exp_after:wN \__fp_sub_back_quite_far_ii:NN 18496
18497         \exp_after:wN #3 18497
18498         \exp_after:wN #4 18498
18499     } 18499
18500     \cs_new:Npn \__fp_sub_back_quite_far_ii:NN #1#2 18500
18501     { 18501
18502         \if_case:w \__fp_round_neg:NNN #2 0 #1 18502
18503             \exp_after:wN \use_i:nn 18503
18504         \else: 18504
18505             \exp_after:wN \use_ii:nn 18505
18506         \fi: 18506
18507         { \__fp_sep: {1000} {0000} {0000} {0000} \__fp_sep: } 18507
18508         { - 1 \__fp_sep: {9999} {9999} {9999} {9999} \__fp_sep: } 18508
18509     } 18509
18510     \cs_new:Npn \__fp_sub_back_not_far_o:wwwNN #1 ~ #2\__fp_sep: #3 ~ #4\__fp_sep: #5#6 18510
18511     { 18511
18512         - 1 18512
18513         \exp_after:wN \__fp_sub_back_near_after:wNNNNw 18513
18514         \int_value:w \__fp_int_eval:w 1#30 - #1 - 11 18514
18515         \exp_after:wN \__fp_sub_back_near_pack:NNNNNNw 18515
18516         \int_value:w \__fp_int_eval:w 11 0000 0000 + #40 - #2 18516
18517         - \exp_after:wN \__fp_round_neg:NNN 18517
18518         \exp_after:wN #6 18518
18519         \use_none:nnnnnnn #2 #5 18519
18520         \exp_after:wN \__fp_sep: 18520
18521     } 18521
18522     \cs_new:Npn \__fp_sub_back_very_far_o:wwwNN #1#2#3#4#5#6#7 18522
18523     { 18523
18524         \__fp_pack_eight:wNNNNNNNN 18524
```

```

18525 \__fp_sub_back_very_far_ii_o:nnNwwNN 18525
18526 { 0 #1#2#3 #4#5#6#7 } 18526
18527 \__fp_sep: 18527
18528 } 18528
18529 \cs_new:Npn \__fp_sub_back_very_far_ii_o:nnNwwNN 18529
18530 #1#2 \__fp_sep: #3 \__fp_sep: #4 ~ #5\__fp_sep: #6#7 18530
18531 { 18531
18532 \exp_after:wN \__fp_basics_pack_high:NNNNNw 18532
18533 \int_value:w \__fp_int_eval:w 1#4 - #1 - 1 18533
18534 \exp_after:wN \__fp_basics_pack_low:NNNNNw 18534
18535 \int_value:w \__fp_int_eval:w 2#5 - #2 18535
18536 - \exp_after:wN \__fp_round_neg:NNN 18536
18537 \exp_after:wN #7 18537
18538 \int_value:w 18538
18539 \if_int_odd:w \__fp_int_eval:w #5 - #2 \__fp_int_eval_end: 18539
18540 1 \else: 2 \fi: 18540
18541 \int_value:w \__fp_round_digit:Nw #3 #6 \__fp_sep: 18541
18542 \exp_after:wN \__fp_sep: 18542
18543 } 18543
18544 \cs_new:cpn { __fp*_o:ww } 18544
18545 { 18545
18546 \__fp_mul_cases_o:NnNww 18546
18547 * 18547
18548 { - 2 + } 18548
18549 \__fp_mul_npos_o:Nww 18549
18550 { } 18550
18551 } 18551
18552 \cs_new:Npn \__fp_mul_cases_o:NnNww 18552
18553 #1#2#3#4 \s__fp \__fp_chk:w #5#6#7\__fp_sep: \s__fp \__fp_chk:w #8#9 18553
18554 { 18554
18555 \if_case:w \__fp_int_eval:w 18555
18556 \if_int_compare:w #5 #8 = 11 ~ 18556
18557 1 18557
18558 \else: 18558
18559 \if_meaning:w 3 #8 18559
18560 3 18560
18561 \else: 18561
18562 \if_meaning:w 3 #5 18562
18563 2 18563
18564 \else: 18564
18565 \if_int_compare:w #5 #8 = 10 ~ 18565
18566 9 #2 - 2 18566
18567 \else: 18567
18568 (#5 #2 #8) / 2 * 2 + 7 18568
18569 \fi: 18569
18570 \fi: 18570

```

```

18571         \fi:
18572     \fi:
18573     \if_meaning:w #6 #9 - 1 \fi:
18574     \__fp_int_eval_end:
18575     \__fp_case_use:nw { #3 0 }
18576 \or: \__fp_case_use:nw { #3 2 }
18577 \or: \__fp_case_return_i_o:ww
18578 \or: \__fp_case_return_ii_o:ww
18579 \or: \__fp_case_return_o:Nww \c_zero_fp
18580 \or: \__fp_case_return_o:Nww \c_minus_zero_fp
18581 \or: \__fp_case_use:nw { \__fp_invalid_operation_o:Nww #1 }
18582 \or: \__fp_case_use:nw { \__fp_invalid_operation_o:Nww #1 }
18583 \or: \__fp_case_return_o:Nww \c_inf_fp
18584 \or: \__fp_case_return_o:Nww \c_minus_inf_fp
18585 #4
18586 \fi:
18587 \s__fp \__fp_chk:w #5 #6 #7\__fp_sep:
18588 \s__fp \__fp_chk:w #8 #9
18589 }
18590 \cs_new:Npn \__fp_mul_npos_o:Nww
18591 #1 \s__fp \__fp_chk:w #2 #3 #4 #5 \__fp_sep:
18592 \s__fp \__fp_chk:w #6 #7 #8 #9 \__fp_sep:
18593 {
18594 \exp_after:wN \__fp_sanitize:Nw
18595 \exp_after:wN #1
18596 \int_value:w \__fp_int_eval:w
18597 #4 + #8
18598 \__fp_mul_significand_o:nnnnNnnnn #5 #1 #9
18599 }
18600 \cs_new:Npn \__fp_mul_significand_o:nnnnNnnnn #1#2#3#4 #5 #6#7#8#9
18601 {
18602 \exp_after:wN \__fp_mul_significand_test_f:NNN
18603 \exp_after:wN #5
18604 \int_value:w \__fp_int_eval:w 99990000 + #1*#6 +
18605 \exp_after:wN \__fp_mul_significand_keep:NNNNNw
18606 \int_value:w \__fp_int_eval:w 99990000 + #1*#7 + #2*#6 +
18607 \exp_after:wN \__fp_mul_significand_keep:NNNNNw
18608 \int_value:w \__fp_int_eval:w 99990000 + #1*#8 + #2*#7 + #3*#6 +
18609 \exp_after:wN \__fp_mul_significand_drop:NNNNNw
18610 \int_value:w \__fp_int_eval:w 99990000 + #1*#9 + #2*#8 +
18611 #3*#7 + #4*#6 +
18612 \exp_after:wN \__fp_mul_significand_drop:NNNNNw
18613 \int_value:w \__fp_int_eval:w 99990000 + #2*#9 + #3*#8 +
18614 #4*#7 +
18615 \exp_after:wN \__fp_mul_significand_drop:NNNNNw
18616 \int_value:w \__fp_int_eval:w 99990000 + #3*#9 + #4*#8 +

```


18617	\exp_after:wN __fp_mul_significand_drop:NNNNNw	18617
18618	\int_value:w __fp_int_eval:w 100000000 + #4*#9 __fp_sep:	18618
18619	__fp_sep: \exp_after:wN __fp_sep:	18619
18620	}	18620
18621	\cs_new:Npn __fp_mul_significand_drop:NNNNNw #1#2#3#4#5 #6__fp_sep:	18621
18622	{ #1#2#3#4#5 __fp_sep: + #6 }	18622
18623	\cs_new:Npn __fp_mul_significand_keep:NNNNNw #1#2#3#4#5 #6__fp_sep:	18623
18624	{ #1#2#3#4#5 __fp_sep: #6 __fp_sep: }	18624
18625	\cs_new:Npn __fp_mul_significand_test_f:NNN #1 #2 #3	18625
18626	{	18626
18627	\if_meaning:w 0 #3	18627
18628	\exp_after:wN __fp_mul_significand_small_f:NNwwwN	18628
18629	\else:	18629
18630	\exp_after:wN __fp_mul_significand_large_f:NwwNNNN	18630
18631	\fi:	18631
18632	#1 #3	18632
18633	}	18633
18634	\cs_new:Npn __fp_mul_significand_large_f:NwwNNNN	18634
18635	#1 #2__fp_sep: #3__fp_sep: #4#5#6#7__fp_sep: +	18635
18636	{	18636
18637	\exp_after:wN __fp_basics_pack_high:NNNNNw	18637
18638	\int_value:w __fp_int_eval:w 1#2	18638
18639	\exp_after:wN __fp_basics_pack_low:NNNNNw	18639
18640	\int_value:w __fp_int_eval:w 1#3#4#5#6#7	18640
18641	+ \exp_after:wN __fp_round:NNN	18641
18642	\exp_after:wN #1	18642
18643	\exp_after:wN #7	18643
18644	\int_value:w __fp_round_digit:Nw	18644
18645	}	18645
18646	\cs_new:Npn __fp_mul_significand_small_f:NNwwwN	18646
18647	#1 #2#3__fp_sep: #4#5__fp_sep: #6__fp_sep: + #7	18647
18648	{	18648
18649	- 1	18649
18650	\exp_after:wN __fp_basics_pack_high:NNNNNw	18650
18651	\int_value:w __fp_int_eval:w 1#3#4	18651
18652	\exp_after:wN __fp_basics_pack_low:NNNNNw	18652
18653	\int_value:w __fp_int_eval:w 1#5#6#7	18653
18654	+ \exp_after:wN __fp_round:NNN	18654
18655	\exp_after:wN #1	18655
18656	\exp_after:wN #7	18656
18657	\int_value:w __fp_round_digit:Nw	18657
18658	}	18658
18659	\cs_new:cpn { __fp/_o:ww }	18659
18660	{	18660
18661	__fp_mul_cases_o:NnNnw	18661
18662	/	18662

18663	{ - }	18663
18664	__fp_div_npos_o:Nww	18664
18665	{	18665
18666	\or:	18666
18667	__fp_case_use:nw	18667
18668	{ __fp_division_by_zero_o:NNww \c_inf_fp / }	18668
18669	\or:	18669
18670	__fp_case_use:nw	18670
18671	{ __fp_division_by_zero_o:NNww \c_minus_inf_fp / }	18671
18672	}	18672
18673	}	18673
18674	\cs_new:Npn __fp_div_npos_o:Nww	18674
18675	#1 \s__fp __fp_chk:w 1 #2 #3 #4 __fp_sep:	18675
18676	\s__fp __fp_chk:w 1 #5 #6 #7#8#9__fp_sep:	18676
18677	{	18677
18678	\exp_after:wN __fp_sanitize:Nw	18678
18679	\exp_after:wN #1	18679
18680	\int_value:w __fp_int_eval:w	18680
18681	#3 - #6	18681
18682	\exp_after:wN __fp_div_significand_i_o:wnnw	18682
18683	\int_value:w __fp_int_eval:w #7 \use_i:nnnn #8 + 1 __fp_sep:	18683
18684	#4	18684
18685	{#7}{#8}#9 __fp_sep:	18685
18686	#1	18686
18687	}	18687
18688	\cs_new:Npn __fp_div_significand_i_o:wnnw #1 __fp_sep: #2#3 #4 __fp_sep:	18688
18689	{	18689
18690	\exp_after:wN __fp_div_significand_test_o:w	18690
18691	\int_value:w __fp_int_eval:w	18691
18692	\exp_after:wN __fp_div_significand_calc:wnnnnnnnn	18692
18693	\int_value:w __fp_int_eval:w 999999 + #2 #3 0 / #1 __fp_sep:	18693
18694	#2 #3 __fp_sep:	18694
18695	#4	18695
18696	{ \exp_after:wN __fp_div_significand_ii:wnn \int_value:w #1 }	18696
18697	{ \exp_after:wN __fp_div_significand_ii:wnn \int_value:w #1 }	18697
18698	{ \exp_after:wN __fp_div_significand_ii:wnn \int_value:w #1 }	18698
18699	{ \exp_after:wN __fp_div_significand_iii:wnnnnnn \int_value:w #1 }	18699
18700	}	18700
18701	\cs_new:Npn __fp_div_significand_calc:wnnnnnnnn 1#1	18701
18702	{	18702
18703	\if_meaning:w 1 #1	18703
18704	\exp_after:wN __fp_div_significand_calc_i:wnnnnnnnn	18704
18705	\else:	18705
18706	\exp_after:wN __fp_div_significand_calc_ii:wnnnnnnnn	18706
18707	\fi:	18707
18708	}	18708

```
18709 \cs_new:Npn \__fp_div_significand_calc_i:wwnnnnnnnn 18709
18710 #1\__fp_sep: #2\__fp_sep:#3#4 #5#6#7#8 #9 18710
18711 { 18711
18712 1 1 #1 18712
18713 #9 \exp_after:wN \__fp_sep: 18713
18714 \int_value:w \__fp_int_eval:w \c__fp_Bigg_leading_shift_int 18714
18715 + #2 - #1 * #5 - #5#60 18715
18716 \exp_after:wN \__fp_pack_Bigg:NNNNNNw 18716
18717 \int_value:w \__fp_int_eval:w \c__fp_Bigg_middle_shift_int 18717
18718 + #3 - #1 * #6 - #70 18718
18719 \exp_after:wN \__fp_pack_Bigg:NNNNNNw 18719
18720 \int_value:w \__fp_int_eval:w \c__fp_Bigg_middle_shift_int 18720
18721 + #4 - #1 * #7 - #80 18721
18722 \exp_after:wN \__fp_pack_Bigg:NNNNNNw 18722
18723 \int_value:w \__fp_int_eval:w \c__fp_Bigg_trailing_shift_int 18723
18724 - #1 * #8 \__fp_sep: 18724
18725 {#5}{#6}{#7}{#8} 18725
18726 } 18726
18727 \cs_new:Npn \__fp_div_significand_calc_ii:wwnnnnnnnn 18727
18728 #1\__fp_sep: #2\__fp_sep:#3#4 #5#6#7#8 #9 18728
18729 { 18729
18730 1 0 #1 18730
18731 #9 \exp_after:wN \__fp_sep: 18731
18732 \int_value:w \__fp_int_eval:w \c__fp_Bigg_leading_shift_int 18732
18733 + #2 - #1 * #5 18733
18734 \exp_after:wN \__fp_pack_Bigg:NNNNNNw 18734
18735 \int_value:w \__fp_int_eval:w \c__fp_Bigg_middle_shift_int 18735
18736 + #3 - #1 * #6 18736
18737 \exp_after:wN \__fp_pack_Bigg:NNNNNNw 18737
18738 \int_value:w \__fp_int_eval:w \c__fp_Bigg_middle_shift_int 18738
18739 + #4 - #1 * #7 18739
18740 \exp_after:wN \__fp_pack_Bigg:NNNNNNw 18740
18741 \int_value:w \__fp_int_eval:w \c__fp_Bigg_trailing_shift_int 18741
18742 - #1 * #8 \__fp_sep: 18742
18743 {#5}{#6}{#7}{#8} 18743
18744 } 18744
18745 \cs_new:Npn \__fp_div_significand_ii:wnn #1\__fp_sep: #2\__fp_sep:#3 18745
18746 { 18746
18747 \exp_after:wN \__fp_div_significand_pack:NNN 18747
18748 \int_value:w \__fp_int_eval:w 18748
18749 \exp_after:wN \__fp_div_significand_calc:wwnnnnnnnn 18749
18750 \int_value:w \__fp_int_eval:w 18750
18751 999999 + #2 #3 0 / #1 \__fp_sep: #2 #3 \__fp_sep: 18751
18752 } 18752
18753 \cs_new:Npn \__fp_div_significand_iii:wwnnnnn #1\__fp_sep: #2\__fp_sep:#3#4#5 #6#7 18753
18754 { 18754
```

```
18755 0
18756 \exp_after:wN \__fp_div_significand_iv:wwnnnnnnnn
18757 \int_value:w \__fp_int_eval:w ( 2 * #2 #3) / #6 #7 \__fp_sep: % <- P
18758 #2 \__fp_sep: {#3} {#4} {#5}
18759 {#6} {#7}
18760 }
18761 \cs_new:Npn \__fp_div_significand_iv:wwnnnnnnnn
18762 #1\__fp_sep: #2\__fp_sep:#3#4#5 #6#7#8#9
18763 {
18764 + 5 * #1
18765 \exp_after:wN \__fp_div_significand_vi:Nw
18766 \int_value:w \__fp_int_eval:w -50 + 2*#2#3 - #1*#6#7 +
18767 \exp_after:wN \__fp_div_significand_v:NN
18768 \int_value:w \__fp_int_eval:w 499950 + 2*#4 - #1*#8 +
18769 \exp_after:wN \__fp_div_significand_v:NN
18770 \int_value:w \__fp_int_eval:w 500000 + 2*#5 - #1*#9 \__fp_sep:
18771 }
18772 \cs_new:Npn \__fp_div_significand_v:NN #1#2 { #1#2 \__fp_int_eval_end: + }
18773 \cs_new:Npn \__fp_div_significand_vi:Nw #1#2\__fp_sep:
18774 {
18775 \if_meaning:w 0 #1
18776 \if_int_compare:w \__fp_int_eval:w #2 > 0 + 1 \fi:
18777 \else:
18778 \if_meaning:w - #1 - \else: + \fi: 1
18779 \fi:
18780 \__fp_sep:
18781 }
18782 \cs_new:Npn \__fp_div_significand_pack:NNN 1 #1 #2 { + #1 #2 \__fp_sep: }
18783 \cs_new:Npn \__fp_div_significand_test_o:w 10 #1
18784 {
18785 \if_meaning:w 0 #1
18786 \exp_after:wN \__fp_div_significand_small_o:wwwNNNNwN
18787 \else:
18788 \exp_after:wN \__fp_div_significand_large_o:wwwNNNNwN
18789 \fi:
18790 #1
18791 }
18792 \cs_new:Npn \__fp_div_significand_small_o:wwwNNNNwN
18793 0 #1\__fp_sep: #2\__fp_sep: #3\__fp_sep: #4#5#6#7#8\__fp_sep: #9
18794 {
18795 \exp_after:wN \__fp_basics_pack_high:NNNNw
18796 \int_value:w \__fp_int_eval:w 1 #1#2
18797 \exp_after:wN \__fp_basics_pack_low:NNNNw
18798 \int_value:w \__fp_int_eval:w 1 #3#4#5#6#7
18799 + \__fp_round:NNN #9 #7 #8
18800 \exp_after:wN \__fp_sep:
```

```
18801 } 18801
18802 \cs_new:Npn \__fp_div_significand_large_o:wwwNNNNwN 18802
18803 #1\__fp_sep: #2\__fp_sep: #3\__fp_sep: #4#5#6#7#8\__fp_sep: #9 18803
18804 { 18804
18805 + 1 18805
18806 \exp_after:wN \__fp_basics_pack_weird_high:NNNNNNNNw 18806
18807 \int_value:w \__fp_int_eval:w 1 #1 #2 18807
18808 \exp_after:wN \__fp_basics_pack_weird_low:NNNNw 18808
18809 \int_value:w \__fp_int_eval:w 1 #3 #4 #5 #6 + 18809
18810 \exp_after:wN \__fp_round:NNN 18810
18811 \exp_after:wN #9 18811
18812 \exp_after:wN #6 18812
18813 \int_value:w \__fp_round_digit:Nw #7 #8 \__fp_sep: 18813
18814 \exp_after:wN \__fp_sep: 18814
18815 } 18815
18816 \cs_new:Npn \__fp_sqrt_o:w #1 \s__fp \__fp_chk:w #2#3#4\__fp_sep: @ 18816
18817 { 18817
18818 \if_meaning:w 0 #2 \__fp_case_return_same_o:w \fi: 18818
18819 \if_meaning:w 2 #3 18819
18820 \__fp_case_use:nw { \__fp_invalid_operation_o:nw { sqrt } } 18820
18821 \fi: 18821
18822 \if_meaning:w 1 #2 \else: \__fp_case_return_same_o:w \fi: 18822
18823 \__fp_sqrt_npos_o:w 18823
18824 \s__fp \__fp_chk:w #2 #3 #4\__fp_sep: 18824
18825 } 18825
18826 \cs_new:Npn \__fp_sqrt_npos_o:w \s__fp \__fp_chk:w 1 0 #1#2#3#4#5\__fp_sep: 18826
18827 { 18827
18828 \exp_after:wN \__fp_sanitize:Nw 18828
18829 \exp_after:wN 0 18829
18830 \int_value:w \__fp_int_eval:w 18830
18831 \if_int_odd:w #1 \exp_stop_f: 18831
18832 \exp_after:wN \__fp_sqrt_npos_auxi_o:wwnnN 18832
18833 \fi: 18833
18834 #1 / 2 18834
18835 \__fp_sqrt_Newton_o:wnn 56234133\__fp_sep: 0\__fp_sep: {#2#3} {#4#5} 0 18835
18836 } 18836
18837 \cs_new:Npn \__fp_sqrt_npos_auxi_o:wwnnN #1 / 2 #2\__fp_sep: 0\__fp_sep: #3#4#5 18837
18838 { 18838
18839 ( #1 + 1 ) / 2 18839
18840 \__fp_pack_eight:wNNNNNNNNN 18840
18841 \__fp_sqrt_npos_auxii_o:wwnnNNNNN 18841
18842 \__fp_sep: 18842
18843 0 #3 #4 18843
18844 } 18844
18845 \cs_new:Npn \__fp_sqrt_npos_auxii_o:wwnnNNNNN #1\__fp_sep: #2#3#4#5#6#7#8#9 18845
18846 { \__fp_sqrt_Newton_o:wnn 17782794\__fp_sep: 0\__fp_sep: {#1}{#2#3#4#5#6#7#8#9} } 18846
```

```
18847 \cs_new:Npn \__fp_sqrt_Newton_o:wnn #1\__fp_sep: #2\__fp_sep: #3 18847
18848 { 18848
18849 \if_int_compare:w #1 = #2 \exp_stop_f: 18849
18850 \exp_after:wN \__fp_sqrt_auxi_o:NNNNwnnnN 18850
18851 \int_value:w \__fp_int_eval:w 9999 9999 + 18851
18852 \exp_after:wN \__fp_use_none_until_s:w 18852
18853 \fi: 18853
18854 \exp_after:wN \__fp_sqrt_Newton_o:wnn 18854
18855 \int_value:w \__fp_int_eval:w (#1 + #3 * 1 0000 0000 / #1) / 2 \__fp_sep: 18855
18856 #1\__fp_sep: {#3} 18856
18857 } 18857
18858 \cs_new:Npn \__fp_sqrt_auxi_o:NNNNwnnnN 1 #1#2#3#4#5\__fp_sep: 18858
18859 { 18859
18860 \__fp_sqrt_auxii_o:NnnnnnnnnN 18860
18861 \__fp_sqrt_auxiii_o:wnnnnnnnnn 18861
18862 {#1#2#3#4} {#5} {2499} {9988} {7500} 18862
18863 } 18863
18864 \cs_new:Npn \__fp_sqrt_auxii_o:NnnnnnnnnN #1 #2#3#4#5#6 #7#8#9 18864
18865 { 18865
18866 \exp_after:wN #1 18866
18867 \int_value:w \__fp_int_eval:w \c__fp_big_leading_shift_int 18867
18868 + #7 - #2 * #2 18868
18869 \exp_after:wN \__fp_pack_big:NNNNNNw 18869
18870 \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int 18870
18871 - 2 * #2 * #3 18871
18872 \exp_after:wN \__fp_pack_big:NNNNNNw 18872
18873 \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int 18873
18874 + #8 - #3 * #3 - 2 * #2 * #4 18874
18875 \exp_after:wN \__fp_pack_big:NNNNNNw 18875
18876 \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int 18876
18877 - 2 * #3 * #4 - 2 * #2 * #5 18877
18878 \exp_after:wN \__fp_pack_big:NNNNNNw 18878
18879 \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int 18879
18880 + #9 000 0000 - #4 * #4 - 2 * #3 * #5 - 2 * #2 * #6 18880
18881 \exp_after:wN \__fp_pack_big:NNNNNNw 18881
18882 \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int 18882
18883 - 2 * #4 * #5 - 2 * #3 * #6 18883
18884 \exp_after:wN \__fp_pack_big:NNNNNNw 18884
18885 \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int 18885
18886 - #5 * #5 - 2 * #4 * #6 18886
18887 \exp_after:wN \__fp_pack_big:NNNNNNw 18887
18888 \int_value:w \__fp_int_eval:w 18888
18889 \c__fp_big_middle_shift_int 18889
18890 - 2 * #5 * #6 18890
18891 \exp_after:wN \__fp_pack_big:NNNNNNw 18891
18892 \int_value:w \__fp_int_eval:w 18892
```



```
18893         \c_fp_big_trailing_shift_int 18893
18894         - #6 * #6 \__fp_sep: 18894
18895 % ( 18895
18896 - 257 ) * 5000 0000 / (#2#3 + 1) + 10 0000 0000 \__fp_sep: 18896
18897 {#2}{#3}{#4}{#5}{#6} {#7}{#8}#9 18897
18898 } 18898
18899 \cs_new:Npn \__fp_sqrt_auxiii_o:wnnnnnnnn 18899
18900 #1\__fp_sep: #2#3#4#5#6#7#8#9 18900
18901 { 18901
18902 \if_int_compare:w #1 > \c_one_int 18902
18903 \exp_after:wN \__fp_sqrt_auxiv_o:NNNNNw 18903
18904 \int_value:w \__fp_int_eval:w (#1#2 %) 18904
18905 \else: 18905
18906 \if_int_compare:w #1#2 > \c_one_int 18906
18907 \exp_after:wN \__fp_sqrt_auxv_o:NNNNNw 18907
18908 \int_value:w \__fp_int_eval:w (#1#2#3 %) 18908
18909 \else: 18909
18910 \if_int_compare:w #1#2#3 > \c_one_int 18910
18911 \exp_after:wN \__fp_sqrt_auxvi_o:NNNNNw 18911
18912 \int_value:w \__fp_int_eval:w (#1#2#3#4 %) 18912
18913 \else: 18913
18914 \exp_after:wN \__fp_sqrt_auxvii_o:NNNNNw 18914
18915 \int_value:w \__fp_int_eval:w (#1#2#3#4#5 %) 18915
18916 \fi: 18916
18917 \fi: 18917
18918 \fi: 18918
18919 } 18919
18920 \cs_new:Npn \__fp_sqrt_auxiv_o:NNNNNw 1#1#2#3#4#5#6\__fp_sep: 18920
18921 { \__fp_sqrt_auxviii_o:nnnnnnn {#1#2#3#4#5#6} {00000000} } 18921
18922 \cs_new:Npn \__fp_sqrt_auxv_o:NNNNNw 1#1#2#3#4#5#6\__fp_sep: 18922
18923 { \__fp_sqrt_auxviii_o:nnnnnnn {000#1#2#3#4#5} {#60000} } 18923
18924 \cs_new:Npn \__fp_sqrt_auxvi_o:NNNNNw 1#1#2#3#4#5#6\__fp_sep: 18924
18925 { \__fp_sqrt_auxviii_o:nnnnnnn {0000000#1} {#2#3#4#5#6} } 18925
18926 \cs_new:Npn \__fp_sqrt_auxvii_o:NNNNNw 1#1#2#3#4#5#6\__fp_sep: 18926
18927 { 18927
18928 \if_int_compare:w #1#2 = \c_zero_int 18928
18929 \exp_after:wN \__fp_sqrt_auxx_o:Nnnnnnnn 18929
18930 \fi: 18930
18931 \__fp_sqrt_auxviii_o:nnnnnnn {00000000} {000#1#2#3#4#5} 18931
18932 } 18932
18933 \cs_new:Npn \__fp_sqrt_auxviii_o:nnnnnnn #1#2 #3#4#5#6#7 18933
18934 { 18934
18935 \exp_after:wN \__fp_sqrt_auxix_o:wnnwnw 18935
18936 \int_value:w \__fp_int_eval:w #3 18936
18937 \exp_after:wN \__fp_basics_pack_low:NNNNNw 18937
18938 \int_value:w \__fp_int_eval:w #1 + 1#4#5 18938
```



```
18939         \exp_after:wN \__fp_basics_pack_low:NNNNNw 18939
18940         \int_value:w \__fp_int_eval:w #2 + 1#6#7 \__fp_sep: 18940
18941     } 18941
18942 \cs_new:Npn \__fp_sqrt_auxix_o:wnwnw #1\__fp_sep: #2#3\__fp_sep: #4#5\__fp_sep: 18942
18943 { 18943
18944     \__fp_sqrt_auxii_o:NnnnnnnnnN 18944
18945     \__fp_sqrt_auxiii_o:wnnnnnnnnn {#1}{#2}{#3}{#4}{#5} 18945
18946 } 18946
18947 \cs_new:Npn \__fp_sqrt_auxx_o:Nnnnnnnnn #1#2#3 #4#5#6#7#8 18947
18948 { 18948
18949     \exp_after:wN \__fp_sqrt_auxxi_o:wwnnN 18949
18950     \int_value:w \__fp_int_eval:w 18950
18951     (#8 + 2499) / 5000 * 5000 \__fp_sep: 18951
18952     {#4} {#5} {#6} {#7} \__fp_sep: 18952
18953 } 18953
18954 \cs_new:Npn \__fp_sqrt_auxxi_o:wwnnN #1\__fp_sep: #2\__fp_sep: #3#4#5 18954
18955 { 18955
18956     \__fp_sqrt_auxii_o:NnnnnnnnnN 18956
18957     \__fp_sqrt_auxxii_o:nnnnnnnnnw 18957
18958     #2 {#1} 18958
18959     {#3} { #4 + 1 } #5 18959
18960 } 18960
18961 \cs_new:Npn \__fp_sqrt_auxxii_o:nnnnnnnnnw 0\__fp_sep: #1#2#3#4#5#6#7#8 #9\__fp_sep: 18961
18962 { 18962
18963     \if_int_compare:w #1#2 > \c_zero_int 18963
18964     \if_int_compare:w #1#2 = \c_one_int 18964
18965     \if_int_compare:w #3#4 = \c_zero_int 18965
18966     \if_int_compare:w #5#6 = \c_zero_int 18966
18967     \if_int_compare:w #7#8 = \c_zero_int 18967
18968     \__fp_sqrt_auxxiii_o:w 18968
18969     \fi: 18969
18970     \fi: 18970
18971     \fi: 18971
18972     \fi: 18972
18973     \exp_after:wN \__fp_sqrt_auxxiv_o:wnnnnnnnnnN 18973
18974     \int_value:w 9998 18974
18975 \else: 18975
18976     \exp_after:wN \__fp_sqrt_auxxiv_o:wnnnnnnnnnN 18976
18977     \int_value:w 10000 18977
18978     \fi: 18978
18979     \__fp_sep: 18979
18980 } 18980
18981 \cs_new:Npn \__fp_sqrt_auxxiii_o:w \fi: \fi: \fi: \fi: #1 \fi: \__fp_sep: 18981
18982 { 18982
18983     \fi: \fi: \fi: \fi: \fi: 18983
18984     \__fp_sqrt_auxxiv_o:wnnnnnnnnnN 9999 \__fp_sep: 18984
```

```
18985 } 18985
18986 \cs_new:Npn \__fp_sqrt_auxxiv_o:wnnnnnnnnN #1\__fp_sep: #2#3#4#5#6 #7#8#9 18986
18987 { 18987
18988 \exp_after:wN \__fp_basics_pack_high:NNNNNw 18988
18989 \int_value:w \__fp_int_eval:w 1 0000 0000 + #2#3 18989
18990 \exp_after:wN \__fp_basics_pack_low:NNNNNw 18990
18991 \int_value:w \__fp_int_eval:w 1 0000 0000 18991
18992 + #4#5 18992
18993 \if_int_compare:w #6 > #1 \exp_stop_f: + 1 \fi: 18993
18994 + \exp_after:wN \__fp_round:NNN 18994
18995 \exp_after:wN 0 18995
18996 \exp_after:wN 0 18996
18997 \int_value:w 18997
18998 \exp_after:wN \use_i:nn 18998
18999 \exp_after:wN \__fp_round_digit:Nw 18999
19000 \int_value:w \__fp_int_eval:w #6 + 19999 - #1 \__fp_sep: 19000
19001 \exp_after:wN \__fp_sep: 19001
19002 } 19002
19003 \cs_new:Npn \__fp_logb_o:w ? \s__fp \__fp_chk:w #1#2\__fp_sep: @ 19003
19004 { 19004
19005 \if_case:w #1 \exp_stop_f: 19005
19006 \__fp_case_use:nw 19006
19007 { \__fp_division_by_zero_o:Nnw \c_minus_inf_fp { logb } } 19007
19008 \or: \exp_after:wN \__fp_logb_aux_o:w 19008
19009 \or: \__fp_case_return_o:Nw \c_inf_fp 19009
19010 \else: \__fp_case_return_same_o:w 19010
19011 \fi: 19011
19012 \s__fp \__fp_chk:w #1 #2\__fp_sep: 19012
19013 } 19013
19014 \cs_new:Npn \__fp_logb_aux_o:w \s__fp \__fp_chk:w #1 #2 #3 #4 \__fp_sep: 19014
19015 { 19015
19016 \exp_after:wN \__fp_parse:n \exp_after:wN 19016
19017 { \int_value:w \int_eval:w #3 - 1 \exp_after:wN } 19017
19018 } 19018
19019 \cs_new:Npn \__fp_sign_o:w ? \s__fp \__fp_chk:w #1#2\__fp_sep: @ 19019
19020 { 19020
19021 \if_case:w #1 \exp_stop_f: 19021
19022 \__fp_case_return_same_o:w 19022
19023 \or: \exp_after:wN \__fp_sign_aux_o:w 19023
19024 \or: \exp_after:wN \__fp_sign_aux_o:w 19024
19025 \else: \__fp_case_return_same_o:w 19025
19026 \fi: 19026
19027 \s__fp \__fp_chk:w #1 #2\__fp_sep: 19027
19028 } 19028
19029 \cs_new:Npn \__fp_sign_aux_o:w \s__fp \__fp_chk:w #1 #2 #3 \__fp_sep: 19029
19030 { \exp_after:wN \__fp_set_sign_o:w \exp_after:wN #2 \c_one_fp@ } 19030
```

19031	\cs_new:Npn __fp_set_sign_o:w #1 \s__fp __fp_chk:w #2#3#4__fp_sep: @	19031
19032	{	19032
19033	\exp_after:wN __fp_exp_after_o:w	19033
19034	\exp_after:wN \s__fp	19034
19035	\exp_after:wN __fp_chk:w	19035
19036	\exp_after:wN #2	19036
19037	\int_value:w	19037
19038	\if_case:w #3 \exp_stop_f: #1 \or: 1 \or: 0 \fi: \exp_stop_f:	19038
19039	#4__fp_sep:	19039
19040	}	19040
19041	\cs_new:Npn __fp_tuple_set_sign_o:w #1#2 @	19041
19042	{	19042
19043	\if_meaning:w 2 #1	19043
19044	\exp_after:wN __fp_tuple_set_sign_aux_o:Nnw	19044
19045	\fi:	19045
19046	__fp_invalid_operation_o:nw { abs }	19046
19047	#2	19047
19048	}	19048
19049	\cs_new:Npn __fp_tuple_set_sign_aux_o:Nnw #1#2	19049
19050	{ __fp_tuple_map_o:nw __fp_tuple_set_sign_aux_o:w }	19050
19051	\cs_new:Npn __fp_tuple_set_sign_aux_o:w #1#2 __fp_sep:	19051
19052	{	19052
19053	__fp_change_func_type:NNN #1 __fp_set_sign_o:w	19053
19054	__fp_parse_apply_unary_error:NNw	19054
19055	2 #1 #2 __fp_sep: @	19055
19056	}	19056
19057	\cs_new:cpn { __fp_*_tuple_o:ww } #1 __fp_sep:	19057
19058	{ __fp_tuple_map_o:nw { __fp_binary_type_o:Nww * #1 __fp_sep: } }	19058
19059	\cs_new:cpn { __fp_tuple_*_o:ww } #1 __fp_sep: #2 __fp_sep:	19059
19060	{	19060
19061	__fp_tuple_map_o:nw { __fp_binary_rev_type_o:Nww * #2 __fp_sep: }	19061
19062	#1 __fp_sep:	19062
19063	}	19063
19064	\cs_new:cpn { __fp_tuple/_o:ww } #1 __fp_sep: #2 __fp_sep:	19064
19065	{	19065
19066	__fp_tuple_map_o:nw { __fp_binary_rev_type_o:Nww / #2 __fp_sep: }	19066
19067	#1 __fp_sep:	19067
19068	}	19068
19069	\cs_set_protected:Npn __fp_tmp:w #1	19069
19070	{	19070
19071	\cs_new:cpn { __fp_tuple_#1_tuple_o:ww }	19071
19072	\s__fp_tuple __fp_tuple_chk:w ##1 __fp_sep:	19072
19073	\s__fp_tuple __fp_tuple_chk:w ##2 __fp_sep:	19073
19074	{	19074
19075	\int_compare:nNnTF	19075
19076	{ __fp_array_count:n {##1} } = { __fp_array_count:n {##2} }	19076

```

19077         { \__fp_tuple_mapthread_o:nww { \__fp_binary_type_o:Nww #1 } }
19078         { \__fp_invalid_operation_o:nww #1 }
19079         \s__fp_tuple \__fp_tuple_chk:w {##1} \__fp_sep:
19080         \s__fp_tuple \__fp_tuple_chk:w {##2} \__fp_sep:
19081     }
19082 }
19083 \__fp_tmp:w +
19084 \__fp_tmp:w -
19085 %% File: l3fp-extended.dtx
19086 \tl_const:Nn \c__fp_one_fixed_tl
19087     { {10000} {0000} {0000} {0000} {0000} {0000} \__fp_sep: }
19088 \cs_new:Npn \__fp_fixed_continue:wn #1\__fp_sep: #2 { #2 #1\__fp_sep: }
19089 \cs_new:Npn \__fp_fixed_add_one:wN #1#2\__fp_sep: #3
19090 {
19091     \exp_after:wN #3 \exp_after:wN
19092     { \int_value:w \__fp_int_eval:w \c__fp_myriad_int + #1 } #2 \__fp_sep:
19093 }
19094 \cs_new:Npn \__fp_fixed_div_myriad:wn #1#2#3#4#5#6\__fp_sep:
19095 {
19096     \exp_after:wN \__fp_fixed_mul_after:wnn
19097     \int_value:w \__fp_int_eval:w \c__fp_leading_shift_int
19098     \exp_after:wN \__fp_pack:NNNNNw
19099     \int_value:w \__fp_int_eval:w \c__fp_trailing_shift_int
19100     + #1 \__fp_sep: {#2}{#3}{#4}{#5}\__fp_sep:
19101 }
19102 \cs_new:Npn \__fp_fixed_mul_after:wnn #1\__fp_sep: #2\__fp_sep: #3
19103     { #3 {#1} #2\__fp_sep: }
19104 \cs_new:Npn \__fp_fixed_mul_short:wnn #1#2#3#4#5#6\__fp_sep: #7#8#9\__fp_sep:
19105 {
19106     \exp_after:wN \__fp_fixed_mul_after:wnn
19107     \int_value:w \__fp_int_eval:w \c__fp_leading_shift_int
19108     + #1*#7
19109     \exp_after:wN \__fp_pack:NNNNNw
19110     \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
19111     + #1*#8 + #2*#7
19112     \exp_after:wN \__fp_pack:NNNNNw
19113     \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
19114     + #1*#9 + #2*#8 + #3*#7
19115     \exp_after:wN \__fp_pack:NNNNNw
19116     \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
19117     + #2*#9 + #3*#8 + #4*#7
19118     \exp_after:wN \__fp_pack:NNNNNw
19119     \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
19120     + #3*#9 + #4*#8 + #5*#7
19121     \exp_after:wN \__fp_pack:NNNNNw
19122     \int_value:w \__fp_int_eval:w \c__fp_trailing_shift_int

```

```
19123         + #4*#9 + #5*#8 + #6*#7 19123
19124         + ( #5*#9 + #6*#8 + #6*#9 / \c__fp_myriad_int ) 19124
19125         / \c__fp_myriad_int \__fp_sep: \__fp_sep: 19125
19126     } 19126
19127 \cs_new:Npn \__fp_fixed_div_int:wnN #1#2#3#4#5#6 \__fp_sep: #7 \__fp_sep: #8 19127
19128 { 19128
19129     \exp_after:wN \__fp_fixed_div_int_after:Nw 19129
19130     \exp_after:wN #8 19130
19131     \int_value:w \__fp_int_eval:w - 1 19131
19132     \__fp_fixed_div_int:wnN 19132
19133     #1\__fp_sep: {#7} \__fp_fixed_div_int_auxi:wnn 19133
19134     #2\__fp_sep: {#7} \__fp_fixed_div_int_auxi:wnn 19134
19135     #3\__fp_sep: {#7} \__fp_fixed_div_int_auxi:wnn 19135
19136     #4\__fp_sep: {#7} \__fp_fixed_div_int_auxi:wnn 19136
19137     #5\__fp_sep: {#7} \__fp_fixed_div_int_auxi:wnn 19137
19138     #6\__fp_sep: {#7} \__fp_fixed_div_int_auxii:wnn \__fp_sep: 19138
19139 } 19139
19140 \cs_new:Npn \__fp_fixed_div_int:wnN #1\__fp_sep: #2 #3 19140
19141 { 19141
19142     \exp_after:wN #3 19142
19143     \int_value:w \__fp_int_eval:w #1 / #2 - 1 \__fp_sep: 19143
19144     {#2} 19144
19145     {#1} 19145
19146 } 19146
19147 \cs_new:Npn \__fp_fixed_div_int_auxi:wnn #1\__fp_sep: #2 #3 19147
19148 { 19148
19149     + #1 19149
19150     \exp_after:wN \__fp_fixed_div_int_pack:Nw 19150
19151     \int_value:w \__fp_int_eval:w 9999 19151
19152     \exp_after:wN \__fp_fixed_div_int:wnN 19152
19153     \int_value:w \__fp_int_eval:w #3 - #1*#2 \__fp_int_eval_end: 19153
19154 } 19154
19155 \cs_new:Npn \__fp_fixed_div_int_auxii:wnn #1\__fp_sep: #2 #3 { + #1 + 2 \__fp_sep: } 19155
19156 \cs_new:Npn \__fp_fixed_div_int_pack:Nw #1 #2\__fp_sep: { + #1\__fp_sep: {#2} } 19156
19157 \cs_new:Npn \__fp_fixed_div_int_after:Nw #1 #2\__fp_sep: { #1 {#2} } 19157
19158 \cs_new:Npn \__fp_fixed_add:wnn { \__fp_fixed_add:Nnnnnwnn + } 19158
19159 \cs_new:Npn \__fp_fixed_sub:wnn { \__fp_fixed_add:Nnnnnwnn - } 19159
19160 \cs_new:Npn \__fp_fixed_add:Nnnnnwnn #1 #2#3#4#5 #6\__fp_sep: #7#8 19160
19161 { 19161
19162     \exp_after:wN \__fp_fixed_add_after:NNNNNwn 19162
19163     \int_value:w \__fp_int_eval:w 9 9999 9998 + #2#3 #1 #7#8 19163
19164     \exp_after:wN \__fp_fixed_add_pack:NNNNNwn 19164
19165     \int_value:w \__fp_int_eval:w 1 9999 9998 + #4#5 19165
19166     \__fp_fixed_add:nnNnnwnn #6 #1 19166
19167 } 19167
19168 \cs_new:Npn \__fp_fixed_add:nnNnnwnn #1#2 #3 #4#5 #6#7 \__fp_sep: #8 19168
```

```

19169 {
19170     #3 #4#5
19171     \exp_after:wN \__fp_fixed_add_pack:NNNNNwn
19172     \int_value:w \__fp_int_eval:w
19173     2 0000 0000 #3 #6#7 + #1#2 \__fp_sep: {#8} \__fp_sep:
19174 }
19175 \cs_new:Npn \__fp_fixed_add_pack:NNNNNwn #1 #2#3#4#5 #6\__fp_sep: #7
19176 { + #1 \__fp_sep: {#7} {#2#3#4#5} {#6} }
19177 \cs_new:Npn \__fp_fixed_add_after:NNNNNwn 1 #1 #2#3#4#5 #6\__fp_sep: #7
19178 { #7 {#1#2#3#4#5} {#6} }
19179 \cs_new:Npn \__fp_fixed_mul:wwn #1#2#3#4 #5\__fp_sep: #6#7#8#9
19180 {
19181     \exp_after:wN \__fp_fixed_mul_after:wwn
19182     \int_value:w \__fp_int_eval:w \c__fp_leading_shift_int
19183     \exp_after:wN \__fp_pack:NNNNNw
19184     \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
19185     + #1*#6
19186     \exp_after:wN \__fp_pack:NNNNNw
19187     \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
19188     + #1*#7 + #2*#6
19189     \exp_after:wN \__fp_pack:NNNNNw
19190     \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
19191     + #1*#8 + #2*#7 + #3*#6
19192     \exp_after:wN \__fp_pack:NNNNNw
19193     \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
19194     + #1*#9 + #2*#8 + #3*#7 + #4*#6
19195     \exp_after:wN \__fp_pack:NNNNNw
19196     \int_value:w \__fp_int_eval:w \c__fp_trailing_shift_int
19197     + #2*#9 + #3*#8 + #4*#7
19198     + ( #3*#9 + #4*#8
19199         + \__fp_fixed_mul:nnnnnnnw #5 {#6}{#7} {#1}{#2}
19200     )
19201 \cs_new:Npn \__fp_fixed_mul:nnnnnnnw #1#2 #3#4 #5#6 #7#8 \__fp_sep:
19202 {
19203     #1*#4 + #2*#3 + #5*#8 + #6*#7 ) / \c__fp_myriad_int
19204     + #1*#3 + #5*#7 \__fp_sep: \__fp_sep:
19205 }
19206 \cs_new:Npn \__fp_fixed_mul_add:wwwn #1\__fp_sep: #2\__fp_sep: #3#4#5#6#7#8\__fp_sep:
19207 {
19208     \exp_after:wN \__fp_fixed_mul_after:wwn
19209     \int_value:w \__fp_int_eval:w \c__fp_big_leading_shift_int
19210     \exp_after:wN \__fp_pack_big:NNNNNNw
19211     \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int + #3 #4
19212     \__fp_fixed_mul_add:Nwnnnwnnn +
19213     + #5 #6 \__fp_sep: #2 \__fp_sep: #1 \__fp_sep: #2 \__fp_sep: +
19214     + #7 #8 \__fp_sep: \__fp_sep:

```



```
19215 } 19215
19216 \cs_new:Npn \__fp_fixed_mul_sub_back:wwn 19216
19217 #1\__fp_sep: #2\__fp_sep: #3#4#5#6#7#8\__fp_sep: 19217
19218 { 19218
19219 \exp_after:wN \__fp_fixed_mul_after:wwn 19219
19220 \int_value:w \__fp_int_eval:w \c__fp_big_leading_shift_int 19220
19221 \exp_after:wN \__fp_pack_big:NNNNNNw 19221
19222 \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int + #3 #4 19222
19223 \__fp_fixed_mul_add:Nwnnnwnnn - 19223
19224 + #5 #6 \__fp_sep: #2 \__fp_sep: #1 \__fp_sep: #2 \__fp_sep: - 19224
19225 + #7 #8 \__fp_sep: \__fp_sep: 19225
19226 } 19226
19227 \cs_new:Npn \__fp_fixed_one_minus_mul:wwn #1\__fp_sep: #2\__fp_sep: 19227
19228 { 19228
19229 \exp_after:wN \__fp_fixed_mul_after:wwn 19229
19230 \int_value:w \__fp_int_eval:w \c__fp_big_leading_shift_int 19230
19231 \exp_after:wN \__fp_pack_big:NNNNNNw 19231
19232 \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int + 19232
19233 1 0000 0000 19233
19234 \__fp_fixed_mul_add:Nwnnnwnnn - 19234
19235 \__fp_sep: #2 \__fp_sep: #1 \__fp_sep: #2 \__fp_sep: - 19235
19236 \__fp_sep: \__fp_sep: 19236
19237 } 19237
19238 \cs_new:Npn \__fp_fixed_mul_add:Nwnnnwnnn #1 #2\__fp_sep: #3#4#5#6\__fp_sep: #7#8#9 19238
19239 { 19239
19240 #1 #7*#3 19240
19241 \exp_after:wN \__fp_pack_big:NNNNNNw 19241
19242 \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int 19242
19243 #1 #7*#4 #1 #8*#3 19243
19244 \exp_after:wN \__fp_pack_big:NNNNNNw 19244
19245 \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int 19245
19246 #1 #7*#5 #1 #8*#4 #1 #9*#3 #2 19246
19247 \exp_after:wN \__fp_pack_big:NNNNNNw 19247
19248 \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int 19248
19249 #1 \__fp_fixed_mul_add:nnnnwnnnn {#7}{#8}{#9} 19249
19250 } 19250
19251 \cs_new:Npn \__fp_fixed_mul_add:nnnnwnnnn #1#2#3#4#5\__fp_sep: #6#7#8#9 19251
19252 { 19252
19253 ( #1*#9 + #2*#8 + #3*#7 + #4*#6 ) 19253
19254 \exp_after:wN \__fp_pack_big:NNNNNNw 19254
19255 \int_value:w \__fp_int_eval:w \c__fp_big_trailing_shift_int 19255
19256 \__fp_fixed_mul_add:nnnnwnnnwN 19256
19257 { #6 + #4*#7 + #3*#8 + #2*#9 + #1 } 19257
19258 { #7 + #4*#8 + #3*#9 + #2 } 19258
19259 {#1} #5\__fp_sep: 19259
19260 {#6} 19260
```


19261	}	19261
19262	\cs_new:Npn __fp_fixed_mul_add:nnnnwnnwN #1#2 #3#4#5__fp_sep: #6#7#8__fp_sep: #9	19262
19263	{	19263
19264	#9 (#4* #1 *#7)	19264
19265	#9 (#5*#6+#4* #2 *#7+#3*#8) / \c__fp_myriad_int	19265
19266	}	19266
19267	\cs_new:Npn __fp_ep_to_fixed:wwn #1,#2	19267
19268	{	19268
19269	\exp_after:wN __fp_ep_to_fixed_auxi:www	19269
19270	\int_value:w __fp_int_eval:w 1 0000 0000 + #2 \exp_after:wN __fp_sep:	19270
19271	\exp:w \exp_end_continue_f:w	19271
19272	\prg_replicate:nn { 4 - \int_max:nn {#1} { -32 } } { 0 } __fp_sep:	19272
19273	}	19273
19274	\cs_new:Npn __fp_ep_to_fixed_auxi:www	19274
19275	1#1__fp_sep: #2__fp_sep: #3#4#5#6#7__fp_sep:	19275
19276	{	19276
19277	__fp_pack_eight:wNNNNNNNNN	19277
19278	__fp_pack_twice_four:wNNNNNNNNN	19278
19279	__fp_pack_twice_four:wNNNNNNNNN	19279
19280	__fp_pack_twice_four:wNNNNNNNNN	19280
19281	__fp_ep_to_fixed_auxii:nnnnnnnwN __fp_sep:	19281
19282	#2 #1#3#4#5#6#7 0000 !	19282
19283	}	19283
19284	\cs_new:Npn __fp_ep_to_fixed_auxii:nnnnnnnwN #1#2#3#4#5#6#7__fp_sep: #8! #9	19284
19285	{ #9 {#1#2}{#3}{#4}{#5}{#6}{#7}__fp_sep: }	19285
19286	\cs_new:Npn __fp_ep_to_ep:wwN #1,#2#3#4#5#6#7__fp_sep: #8	19286
19287	{	19287
19288	\exp_after:wN #8	19288
19289	\int_value:w __fp_int_eval:w #1 + 4	19289
19290	\exp_after:wN \use_i:nn	19290
19291	\exp_after:wN __fp_ep_to_ep_loop:N	19291
19292	\int_value:w __fp_int_eval:w 1 0000 0000 + #2 __fp_int_eval_end:	19292
19293	#3#4#5#6#7 __fp_sep: __fp_sep: !	19293
19294	}	19294
19295	\cs_new:Npn __fp_ep_to_ep_loop:N #1	19295
19296	{	19296
19297	\if_meaning:w 0 #1	19297
19298	- 1	19298
19299	\else:	19299
19300	__fp_ep_to_ep_end:www #1	19300
19301	\fi:	19301
19302	__fp_ep_to_ep_loop:N	19302
19303	}	19303
19304	\cs_new:Npn __fp_ep_to_ep_end:www	19304
19305	#1 \fi:__fp_ep_to_ep_loop:N #2__fp_sep: #3!	19305
19306	{	19306

19307	\fi:	19307
19308	\if_meaning:w __fp_sep: #1	19308
19309	- 2 * \c__fp_max_exponent_int	19309
19310	__fp_ep_to_ep_zero:ww	19310
19311	\fi:	19311
19312	__fp_pack_twice_four:wNNNNNNNN	19312
19313	__fp_pack_twice_four:wNNNNNNNN	19313
19314	__fp_pack_twice_four:wNNNNNNNN	19314
19315	__fp_use_i:ww , __fp_sep:	19315
19316	#1 #2 0000 0000 0000 0000 0000 0000 __fp_sep:	19316
19317	}	19317
19318	\cs_new:Npn __fp_ep_to_ep_zero:ww \fi: #1__fp_sep: #2__fp_sep: #3__fp_sep:	19318
19319	{ \fi: , {1000}{0000}{0000}{0000}{0000}{0000} __fp_sep: }	19319
19320	\cs_new:Npn __fp_ep_compare:www #1,#2#3#4#5#6#7__fp_sep:	19320
19321	{ __fp_ep_compare_aux:www {#1}{#2}{#3}{#4}{#5}__fp_sep: #6#7__fp_sep: }	19321
19322	\cs_new:Npn __fp_ep_compare_aux:www	19322
19323	#1__fp_sep:#2__fp_sep:#3,#4#5#6#7#8#9__fp_sep:	19323
19324	{	19324
19325	\if_case:w	19325
19326	__fp_compare_npos:nwnw	19326
19327	#1__fp_sep: {#3}{#4}{#5}{#6}{#7}__fp_sep: \exp_stop_f:	19327
19328	\if_int_compare:w #2 = #8#9 \exp_stop_f:	19328
19329	0	19329
19330	\else:	19330
19331	\if_int_compare:w #2 < #8#9 - \fi: 1	19331
19332	\fi:	19332
19333	\or: 1	19333
19334	\else: -1	19334
19335	\fi:	19335
19336	}	19336
19337	\cs_new:Npn __fp_ep_mul:wwwn #1,#2__fp_sep: #3,#4__fp_sep:	19337
19338	{	19338
19339	__fp_ep_to_ep:wwN #3,#4__fp_sep:	19339
19340	__fp_fixed_continue:wn	19340
19341	{	19341
19342	__fp_ep_to_ep:wwN #1,#2__fp_sep:	19342
19343	__fp_ep_mul_raw:wwwN	19343
19344	}	19344
19345	__fp_fixed_continue:wn	19345
19346	}	19346
19347	\cs_new:Npn __fp_ep_mul_raw:wwwN #1,#2__fp_sep: #3,#4__fp_sep: #5	19347
19348	{	19348
19349	__fp_fixed_mul:wn #2__fp_sep: #4__fp_sep:	19349
19350	{ \exp_after:wN #5 \int_value:w __fp_int_eval:w #1 + #3 , }	19350
19351	}	19351
19352	\cs_new:Npn __fp_ep_div:wwwn #1,#2__fp_sep: #3,#4__fp_sep:	19352

```
19353 {
19354     \__fp_ep_to_ep:wwN #1,#2\__fp_sep:
19355     \__fp_fixed_continue:wn
19356 {
19357     \__fp_ep_to_ep:wwN #3,#4\__fp_sep:
19358     \__fp_ep_div_esti:wwwwn
19359 }
19360 }
19361 \cs_new:Npn \__fp_ep_div_esti:wwwwn #1,#2#3\__fp_sep: #4,
19362 {
19363     \exp_after:wN \__fp_ep_div_estii:wwnnwwn
19364     \int_value:w \__fp_int_eval:w 10 0000 0000 / ( #2 + 1 )
19365     \exp_after:wN \__fp_sep:
19366     \int_value:w \__fp_int_eval:w #4 - #1 + 1 ,
19367     {#2} #3\__fp_sep:
19368 }
19369 \cs_new:Npn \__fp_ep_div_estii:wwnnwwn
19370     #1\__fp_sep: #2,#3#4#5\__fp_sep: #6\__fp_sep: #7
19371 {
19372     \exp_after:wN \__fp_ep_div_estiii:NNNNNwwwn
19373     \int_value:w \__fp_int_eval:w 10 0000 0000 - 1750
19374     + #1 000 + (10 0000 0000 / #3 - #1) * (1000 - #4 / 10) \__fp_sep:
19375     {#3}{#4}#5\__fp_sep: #6\__fp_sep: { #7 #2, }
19376 }
19377 \cs_new:Npn \__fp_ep_div_estiii:NNNNNwwwn 1#1#2#3#4#5#6\__fp_sep: #7\__fp_sep:
19378 {
19379     \__fp_fixed_mul_short:wwn #7\__fp_sep: {#1}{#2#3#4#5}{#6}\__fp_sep:
19380     \__fp_ep_div_epsilon:wnNNNNNn {#1#2#3#4}#5#6
19381     \__fp_fixed_mul:wwn
19382 }
19383 \cs_new:Npn \__fp_ep_div_epsilon:wnNNNNNn #1#2#3#4#5#6\__fp_sep:
19384 {
19385     \exp_after:wN \__fp_ep_div_epsiloni:wnNNNNNn
19386     \int_value:w \__fp_int_eval:w 1 9998 - #2
19387     \exp_after:wN \__fp_ep_div_epsilon_pack:NNNNNw
19388     \int_value:w \__fp_int_eval:w 1 9999 9998 - #3#4
19389     \exp_after:wN \__fp_ep_div_epsilon_pack:NNNNNw
19390     \int_value:w \__fp_int_eval:w 2 0000 0000 - #5#6 \__fp_sep: \__fp_sep:
19391 }
19392 \cs_new:Npn \__fp_ep_div_epsilon_pack:NNNNNw #1#2#3#4#5#6\__fp_sep:
19393 { + #1 \__fp_sep: {#2#3#4#5} {#6} }
19394 \cs_new:Npn \__fp_ep_div_epsiloni:wnNNNNNn 1#1\__fp_sep: #2\__fp_sep: #3#4#5#6#7#8
19395 {
19396     \__fp_fixed_mul:wwn {0000}{#1}#2\__fp_sep: {0000}{#1}#2\__fp_sep:
19397     \__fp_fixed_add_one:wN
19398     \__fp_fixed_mul:wwn {10000} {#1} #2 \__fp_sep:
```

```
19399 { 19399
19400 \__fp_fixed_mul_short:wnn 19400
19401 {0000}{#1}#2\__fp_sep: {#3}{#4#5#6#7}{#8000}\__fp_sep: 19401
19402 \__fp_fixed_div_myriad:wn 19402
19403 \__fp_fixed_mul:wnn 19403
19404 } 19404
19405 \__fp_fixed_add:wnn {#3}{#4#5#6#7}{#8000}{0000}{0000}{0000}\__fp_sep: 19405
19406 } 19406
19407 \cs_new:Npn \__fp_ep_isqrt:wnn #1,#2\__fp_sep: 19407
19408 { 19408
19409 \__fp_ep_to_ep:wwN #1,#2\__fp_sep: 19409
19410 \__fp_ep_isqrt_auxi:wnn 19410
19411 } 19411
19412 \cs_new:Npn \__fp_ep_isqrt_auxi:wnn #1, 19412
19413 { 19413
19414 \exp_after:wN \__fp_ep_isqrt_auxii:wwnnwn 19414
19415 \int_value:w \__fp_int_eval:w 19415
19416 \int_if_odd:nTF {#1} 19416
19417 { (1 - #1) / 2 , 535 , { 0 } { } } 19417
19418 { 1 - #1 / 2 , 168 , { } { 0 } } 19418
19419 } 19419
19420 \cs_new:Npn \__fp_ep_isqrt_auxii:wwnnwn #1, #2, #3#4 #5#6\__fp_sep: #7 19420
19421 { 19421
19422 \__fp_ep_isqrt_esti:wwnnwn #2, 0, #5, {#3} {#4} 19422
19423 {#5} #6 \__fp_sep: { #7 #1 , } 19423
19424 } 19424
19425 \cs_new:Npn \__fp_ep_isqrt_esti:wwnnwn #1, #2, #3, #4 19425
19426 { 19426
19427 \if_int_compare:w #1 = #2 \exp_stop_f: 19427
19428 \exp_after:wN \__fp_ep_isqrt_estii:wwnnwn 19428
19429 \fi: 19429
19430 \exp_after:wN \__fp_ep_isqrt_esti:wwnnwn 19430
19431 \int_value:w \__fp_int_eval:w 19431
19432 (#1 + 1 0050 0000 #4 / (#1 * #3)) / 2 , 19432
19433 #1, #3, {#4} 19433
19434 } 19434
19435 \cs_new:Npn \__fp_ep_isqrt_estii:wwnnwn #1, #2, #3, #4#5 19435
19436 { 19436
19437 \exp_after:wN \__fp_ep_isqrt_estiii:NNNNNwwwn 19437
19438 \int_value:w \__fp_int_eval:w 1000 0000 + #2 * #2 #5 * 5 19438
19439 \exp_after:wN , \int_value:w \__fp_int_eval:w 10000 + #2 \__fp_sep: 19439
19440 } 19440
19441 \cs_new:Npn \__fp_ep_isqrt_estiii:NNNNNwwwn 19441
19442 1#1#2#3#4#5#6, 1#7#8\__fp_sep: #9\__fp_sep: 19442
19443 { 19443
19444 \__fp_fixed_mul_short:wnn #9\__fp_sep: {#1} {#2#3#4#5} {#600} \__fp_sep: 19444
```

```
19445 \__fp_ep_isqrt_epsilon:wN 19445
19446 \__fp_fixed_mul_short:wwn {#7} {#80} {0000} \__fp_sep: 19446
19447 } 19447
19448 \cs_new:Npn \__fp_ep_isqrt_epsilon:wN #1\__fp_sep: 19448
19449 { 19449
19450 \__fp_fixed_sub:wwn {15000}{0000}{0000}{0000}{0000}{0000}\__fp_sep: #1\__fp_sep: 19450
19451 \__fp_ep_isqrt_epsilon:wwN #1\__fp_sep: 19451
19452 \__fp_ep_isqrt_epsilon:wwN #1\__fp_sep: 19452
19453 \__fp_ep_isqrt_epsilon:wwN #1\__fp_sep: 19453
19454 } 19454
19455 \cs_new:Npn \__fp_ep_isqrt_epsilon:wwN #1\__fp_sep: #2\__fp_sep: 19455
19456 { 19456
19457 \__fp_fixed_mul:wwn #1\__fp_sep: #1\__fp_sep: 19457
19458 \__fp_fixed_mul_sub_back:wwn #2\__fp_sep: 19458
19459 {15000}{0000}{0000}{0000}{0000}{0000}\__fp_sep: 19459
19460 \__fp_fixed_mul:wwn #1\__fp_sep: 19460
19461 } 19461
19462 \cs_new:Npn \__fp_ep_to_float_o:wwN #1, 19462
19463 { + \__fp_int_eval:w #1 \__fp_fixed_to_float_o:wN } 19463
19464 \cs_new:Npn \__fp_ep_inv_to_float_o:wwN #1,#2\__fp_sep: 19464
19465 { 19465
19466 \__fp_ep_div:wwwwn 19466
19467 1,{1000}{0000}{0000}{0000}{0000}{0000}\__fp_sep: #1,#2\__fp_sep: 19467
19468 \__fp_ep_to_float_o:wwN 19468
19469 } 19469
19470 \cs_new:Npn \__fp_fixed_inv_to_float_o:wN 19470
19471 { \__fp_ep_inv_to_float_o:wwN 0, } 19471
19472 \cs_new:Npn \__fp_fixed_to_float_rad_o:wN #1\__fp_sep: 19472
19473 { 19473
19474 \__fp_fixed_mul:wwn #1\__fp_sep: {5729}{5779}{5130}{8232}{0876}{7981}\__fp_sep: 19474
19475 { \__fp_ep_to_float_o:wwN 2, } 19475
19476 } 19476
19477 \cs_new:Npn \__fp_fixed_to_float_o:Nw #1#2\__fp_sep: 19477
19478 { \__fp_fixed_to_float_o:wN #2\__fp_sep: #1 } 19478
19479 \cs_new:Npn \__fp_fixed_to_float_o:wN #1#2#3#4#5#6\__fp_sep: #7 19479
19480 { % for the 8-digit-at-the-start thing 19480
19481 + \__fp_int_eval:w \c__fp_block_int 19481
19482 \exp_after:wN \exp_after:wN 19482
19483 \exp_after:wN \__fp_fixed_to_loop:N 19483
19484 \exp_after:wN \use_none:n 19484
19485 \int_value:w \__fp_int_eval:w 19485
19486 1 0000 0000 + #1 \exp_after:wN \__fp_use_none_stop_f:n 19486
19487 \int_value:w 1#2 \exp_after:wN \__fp_use_none_stop_f:n 19487
19488 \int_value:w 1#3#4 \exp_after:wN \__fp_use_none_stop_f:n 19488
19489 \int_value:w 1#5#6 19489
19490 \exp_after:wN \__fp_sep: 19490
```

```
19491 \exp_after:wN \__fp_sep: 19491
19492 } 19492
19493 \cs_new:Npn \__fp_fixed_to_loop:N #1 19493
19494 { 19494
19495 \if_meaning:w 0 #1 19495
19496 - 1 19496
19497 \exp_after:wN \__fp_fixed_to_loop:N 19497
19498 \else: 19498
19499 \exp_after:wN \__fp_fixed_to_loop_end:w 19499
19500 \exp_after:wN #1 19500
19501 \fi: 19501
19502 } 19502
19503 \cs_new:Npn \__fp_fixed_to_loop_end:w #1 #2 \__fp_sep: 19503
19504 { 19504
19505 \if_meaning:w \__fp_sep: #1 19505
19506 \exp_after:wN \__fp_fixed_to_float_zero:w 19506
19507 \else: 19507
19508 \exp_after:wN \__fp_pack_twice_four:wNNNNNNNN 19508
19509 \exp_after:wN \__fp_pack_twice_four:wNNNNNNNN 19509
19510 \exp_after:wN \__fp_fixed_to_float_pack:ww 19510
19511 \exp_after:wN \__fp_sep: 19511
19512 \fi: 19512
19513 #1 #2 0000 0000 0000 0000 \__fp_sep: 19513
19514 } 19514
19515 \cs_new:Npn \__fp_fixed_to_float_zero:w \__fp_sep: 0000 0000 0000 0000 \__fp_sep: 19515
19516 { 19516
19517 - 2 * \c__fp_max_exponent_int \__fp_sep: 19517
19518 {0000} {0000} {0000} {0000} \__fp_sep: 19518
19519 } 19519
19520 \cs_new:Npn \__fp_fixed_to_float_pack:ww #1 \__fp_sep: #2#3 \__fp_sep: \__fp_sep: 19520
19521 { 19521
19522 \if_int_compare:w #2 > 4 \exp_stop_f: 19522
19523 \exp_after:wN \__fp_fixed_to_float_round_up:wnnnnw 19523
19524 \fi: 19524
19525 \__fp_sep: #1 \__fp_sep: 19525
19526 } 19526
19527 \cs_new:Npn \__fp_fixed_to_float_round_up:wnnnnw \__fp_sep: #1#2#3#4 \__fp_sep: 19527
19528 { 19528
19529 \exp_after:wN \__fp_basics_pack_high:NNNNNw 19529
19530 \int_value:w \__fp_int_eval:w 1 #1#2 19530
19531 \exp_after:wN \__fp_basics_pack_low:NNNNNw 19531
19532 \int_value:w \__fp_int_eval:w 1 #3#4 + 1 \__fp_sep: 19532
19533 } 19533
19534 %% File: l3fp-expo.dtx 19534
19535 \cs_new:Npn \__fp_parse_word_exp:N 19535
19536 { \__fp_parse_unary_function:NNN \__fp_exp_o:w ? } 19536
```

```
19537 \cs_new:Npn \__fp_parse_word_ln:N 19537
19538 { \__fp_parse_unary_function:NNN \__fp_ln_o:w ? } 19538
19539 \cs_new:Npn \__fp_parse_word_fact:N 19539
19540 { \__fp_parse_unary_function:NNN \__fp_fact_o:w ? } 19540
19541 \tl_const:Nn \c__fp_ln_i_fixed_tl 19541
19542 { {0000}{0000}{0000}{0000}{0000}{0000}\__fp_sep:} 19542
19543 \tl_const:Nn \c__fp_ln_ii_fixed_tl 19543
19544 { {6931}{4718}{0559}{9453}{0941}{7232}\__fp_sep:} 19544
19545 \tl_const:Nn \c__fp_ln_iii_fixed_tl 19545
19546 { {10986}{1228}{8668}{1096}{9139}{5245}\__fp_sep:} 19546
19547 \tl_const:Nn \c__fp_ln_iv_fixed_tl 19547
19548 { {13862}{9436}{1119}{8906}{1883}{4464}\__fp_sep:} 19548
19549 \tl_const:Nn \c__fp_ln_vi_fixed_tl 19549
19550 { {17917}{5946}{9228}{0550}{0081}{2477}\__fp_sep:} 19550
19551 \tl_const:Nn \c__fp_ln_vii_fixed_tl 19551
19552 { {19459}{1014}{9055}{3133}{0510}{5353}\__fp_sep:} 19552
19553 \tl_const:Nn \c__fp_ln_viii_fixed_tl 19553
19554 { {20794}{4154}{1679}{8359}{2825}{1696}\__fp_sep:} 19554
19555 \tl_const:Nn \c__fp_ln_ix_fixed_tl 19555
19556 { {21972}{2457}{7336}{2193}{8279}{0490}\__fp_sep:} 19556
19557 \tl_const:Nn \c__fp_ln_x_fixed_tl 19557
19558 { {23025}{8509}{2994}{0456}{8401}{7991}\__fp_sep:} 19558
19559 \cs_new:Npn \__fp_ln_o:w #1 \s__fp \__fp_chk:w #2#3#4\__fp_sep: @ 19559
19560 { 19560
19561 \if_meaning:w 2 #3 19561
19562 \__fp_case_use:nw { \__fp_invalid_operation_o:nw { ln } } 19562
19563 \fi: 19563
19564 \if_case:w #2 \exp_stop_f: 19564
19565 \__fp_case_use:nw 19565
19566 { \__fp_division_by_zero_o:Nnw \c_minus_inf_fp { ln } } 19566
19567 \or: 19567
19568 \else: 19568
19569 \__fp_case_return_same_o:w 19569
19570 \fi: 19570
19571 \__fp_ln_npos_o:w \s__fp \__fp_chk:w #2#3#4\__fp_sep: 19571
19572 } 19572
19573 \cs_new:Npn \__fp_ln_npos_o:w \s__fp \__fp_chk:w 10#1#2#3\__fp_sep: 19573
19574 { %^^A todo: ln(1) should be "exact zero", not "underflow" 19574
19575 \exp_after:wN \__fp_sanitize:Nw 19575
19576 \int_value:w % for the overall sign 19576
19577 \if_int_compare:w #1 < \c_one_int 19577
19578 2 19578
19579 \else: 19579
19580 0 19580
19581 \fi: 19581
19582 \exp_after:wN \exp_stop_f: 19582
```


19583	\int_value:w __fp_int_eval:w % for the exponent	19583
19584	__fp_ln_significand:NNNNnnnN #2#3	19584
19585	__fp_ln_exponent:wn {#1}	19585
19586	}	19586
19587	\cs_new:Npn __fp_ln_significand:NNNNnnnN #1#2#3#4	19587
19588	{	19588
19589	\exp_after:wN __fp_ln_x_ii:wnnnn	19589
19590	\int_value:w	19590
19591	\if_case:w #1 \exp_stop_f:	19591
19592	\or:	19592
19593	\if_int_compare:w #2 < 4 \exp_stop_f:	19593
19594	__fp_int_eval:w 10 - #2	19594
19595	\else:	19595
19596	6	19596
19597	\fi:	19597
19598	\or: 4	19598
19599	\or: 3	19599
19600	\or: 2	19600
19601	\or: 2	19601
19602	\or: 2	19602
19603	\else: 1	19603
19604	\fi:	19604
19605	__fp_sep: { #1 #2 #3 #4 }	19605
19606	}	19606
19607	\cs_new:Npn __fp_ln_x_ii:wnnnn #1__fp_sep: #2#3#4#5	19607
19608	{	19608
19609	\exp_after:wN __fp_ln_div_after:Nw	19609
19610	\cs:w c__fp_ln_ __fp_int_to_roman:w #1 _fixed_tl \exp_after:wN \cs_end:	19610
19611	\int_value:w	19611
19612	\exp_after:wN __fp_ln_x_iv:wnnnnnnnn	19612
19613	\int_value:w __fp_int_eval:w	19613
19614	\exp_after:wN __fp_ln_x_iii_var:NNNNNw	19614
19615	\int_value:w __fp_int_eval:w 9999 9990 + #1*#2#3 +	19615
19616	\exp_after:wN __fp_ln_x_iii:NNNNNNw	19616
19617	\int_value:w __fp_int_eval:w 10 0000 0000 + #1*#4#5 __fp_sep:	19617
19618	{20000} {0000} {0000} {0000}	19618
19619	} %^^A todo: reoptimize (a generalization attempt failed).	19619
19620	\cs_new:Npn __fp_ln_x_iii:NNNNNNw #1#2 #3#4#5#6 #7__fp_sep:	19620
19621	{ #1#2__fp_sep: {#3#4#5#6} {#7} }	19621
19622	\cs_new:Npn __fp_ln_x_iii_var:NNNNNw #1 #2#3#4#5 #6__fp_sep:	19622
19623	{	19623
19624	#1#2#3#4#5 + 1 __fp_sep:	19624
19625	{#1#2#3#4#5} {#6}	19625
19626	}	19626
19627	\cs_new:Npn __fp_ln_x_iv:wnnnnnnnn #1__fp_sep: #2#3#4#5 #6#7#8#9	19627
19628	{	19628

```
19629 \exp_after:wN \__fp_div_significand_pack:NNN 19629
19630 \int_value:w \__fp_int_eval:w 19630
19631 \__fp_ln_div_i:w #1 \__fp_sep: 19631
19632 #6 #7 \__fp_sep: {#8} {#9} 19632
19633 {#2} {#3} {#4} {#5} 19633
19634 { \exp_after:wN \__fp_ln_div_ii:wwn \int_value:w #1 } 19634
19635 { \exp_after:wN \__fp_ln_div_ii:wwn \int_value:w #1 } 19635
19636 { \exp_after:wN \__fp_ln_div_ii:wwn \int_value:w #1 } 19636
19637 { \exp_after:wN \__fp_ln_div_ii:wwn \int_value:w #1 } 19637
19638 { \exp_after:wN \__fp_ln_div_vi:wwn \int_value:w #1 } 19638
19639 } 19639
19640 \cs_new:Npn \__fp_ln_div_i:w #1\__fp_sep: 19640
19641 { 19641
19642 \exp_after:wN \__fp_div_significand_calc:wwnnnnnnnn 19642
19643 \int_value:w \__fp_int_eval:w 999999 + 2 0000 0000 / #1 \__fp_sep: % Q1 19643
19644 } 19644
19645 \cs_new:Npn \__fp_ln_div_ii:wwn 19645
19646 #1\__fp_sep: #2\__fp_sep:#3 % y\__fp_sep: B1\__fp_sep:B2 <- for k=1 19646
19647 { 19647
19648 \exp_after:wN \__fp_div_significand_pack:NNN 19648
19649 \int_value:w \__fp_int_eval:w 19649
19650 \exp_after:wN \__fp_div_significand_calc:wwnnnnnnnn 19650
19651 \int_value:w \__fp_int_eval:w 999999 + #2 #3 / #1 \__fp_sep: % Q2 19651
19652 #2 #3 \__fp_sep: 19652
19653 } 19653
19654 \cs_new:Npn \__fp_ln_div_vi:wwn 19654
19655 #1\__fp_sep: #2\__fp_sep:#3#4#5 #6#7#8#9 %y\__fp_sep:F1\__fp_sep:F2F3F4x1x2x3x4 19655
19656 { 19656
19657 \exp_after:wN \__fp_div_significand_pack:NNN 19657
19658 \int_value:w \__fp_int_eval:w 1000000 + #2 #3 / #1 \__fp_sep: % Q6 19658
19659 } 19659
19660 \cs_new:Npn \__fp_ln_div_after:Nw #1#2\__fp_sep: 19660
19661 { 19661
19662 \if_meaning:w 0 #2 19662
19663 \exp_after:wN \__fp_ln_t_small:Nw 19663
19664 \else: 19664
19665 \exp_after:wN \__fp_ln_t_large:NNw 19665
19666 \exp_after:wN - 19666
19667 \fi: 19667
19668 #1 19668
19669 } 19669
19670 \cs_new:Npn \__fp_ln_t_small:Nw 19670
19671 #1 #2\__fp_sep: #3\__fp_sep: #4\__fp_sep: #5\__fp_sep: #6\__fp_sep: #7\__fp_sep: 19671
19672 { 19672
19673 \exp_after:wN \__fp_ln_t_large:NNw 19673
19674 \exp_after:wN + % <sign> 19674
```

```
19675 \exp_after:wN #1 19675
19676 \int_value:w \__fp_int_eval:w 9999 - #2 \exp_after:wN \__fp_sep: 19676
19677 \int_value:w \__fp_int_eval:w 9999 - #3 \exp_after:wN \__fp_sep: 19677
19678 \int_value:w \__fp_int_eval:w 9999 - #4 \exp_after:wN \__fp_sep: 19678
19679 \int_value:w \__fp_int_eval:w 9999 - #5 \exp_after:wN \__fp_sep: 19679
19680 \int_value:w \__fp_int_eval:w 9999 - #6 \exp_after:wN \__fp_sep: 19680
19681 \int_value:w \__fp_int_eval:w 1 0000 - #7 \__fp_sep: 19681
19682 } 19682
19683 \cs_new:Npn \__fp_ln_t_large:NNw 19683
19684 #1 #2 19684
19685 #3\__fp_sep: #4\__fp_sep: #5\__fp_sep: #6\__fp_sep: #7\__fp_sep: #8\__fp_sep: 19685
19686 { 19686
19687 \exp_after:wN \__fp_ln_square_t_after:w 19687
19688 \int_value:w \__fp_int_eval:w 9999 0000 + #3*#3 19688
19689 \exp_after:wN \__fp_ln_square_t_pack:NNNNNw 19689
19690 \int_value:w \__fp_int_eval:w 9999 0000 + 2*#3*#4 19690
19691 \exp_after:wN \__fp_ln_square_t_pack:NNNNNw 19691
19692 \int_value:w \__fp_int_eval:w 9999 0000 + 2*#3*#5 + #4*#4 19692
19693 \exp_after:wN \__fp_ln_square_t_pack:NNNNNw 19693
19694 \int_value:w \__fp_int_eval:w 9999 0000 + 2*#3*#6 + 2*#4*#5 19694
19695 \exp_after:wN \__fp_ln_square_t_pack:NNNNNw 19695
19696 \int_value:w \__fp_int_eval:w 19696
19697 1 0000 0000 + 2*#3*#7 + 2*#4*#6 + #5*#5 19697
19698 + (2*#3*#8 + 2*#4*#7 + 2*#5*#6) / 1 0000 19698
19699 % \__fp_sep: \__fp_sep: \__fp_sep: 19699
19700 \exp_after:wN \__fp_ln_twice_t_after:w 19700
19701 \int_value:w \__fp_int_eval:w -1 + 2*#3 19701
19702 \exp_after:wN \__fp_ln_twice_t_pack:Nw 19702
19703 \int_value:w \__fp_int_eval:w 9999 + 2*#4 19703
19704 \exp_after:wN \__fp_ln_twice_t_pack:Nw 19704
19705 \int_value:w \__fp_int_eval:w 9999 + 2*#5 19705
19706 \exp_after:wN \__fp_ln_twice_t_pack:Nw 19706
19707 \int_value:w \__fp_int_eval:w 9999 + 2*#6 19707
19708 \exp_after:wN \__fp_ln_twice_t_pack:Nw 19708
19709 \int_value:w \__fp_int_eval:w 9999 + 2*#7 19709
19710 \exp_after:wN \__fp_ln_twice_t_pack:Nw 19710
19711 \int_value:w \__fp_int_eval:w 10000 + 2*#8 \__fp_sep: \__fp_sep: 19711
19712 { \__fp_ln_c:NwNw #1 } 19712
19713 #2 19713
19714 } 19714
19715 \cs_new:Npn \__fp_ln_twice_t_pack:Nw #1 #2\__fp_sep: { + #1 \__fp_sep: {#2} } 19715
19716 \cs_new:Npn \__fp_ln_twice_t_after:w #1\__fp_sep: 19716
19717 { \__fp_sep:\__fp_sep:\__fp_sep: {#1} } 19717
19718 \cs_new:Npn \__fp_ln_square_t_pack:NNNNNw #1 #2#3#4#5 #6\__fp_sep: 19718
19719 { + #1#2#3#4#5 \__fp_sep: {#6} } 19719
19720 \cs_new:Npn \__fp_ln_square_t_after:w 1 0 #1#2#3 #4\__fp_sep: 19720
```

```

19721 { \__fp_ln_Taylor:wwNw {0#1#2#3} {#4} }
19722 \cs_new:Npn \__fp_ln_Taylor:wwNw
19723 {
19724   \__fp_ln_Taylor_loop:www
19725   21 \__fp_sep: {0000}{0000}{0000}{0000}{0000}{0000} \__fp_sep:
19726 }
19727 \cs_new:Npn \__fp_ln_Taylor_loop:www #1\__fp_sep: #2\__fp_sep: #3\__fp_sep:
19728 {
19729   \if_int_compare:w #1 = \c_one_int
19730     \__fp_ln_Taylor_break:w
19731   \fi:
19732   \exp_after:wN \__fp_fixed_div_int:wwN \c__fp_one_fixed_tl #1\__fp_sep:
19733   \__fp_fixed_add:wwn #2\__fp_sep:
19734   \__fp_fixed_mul:wwn #3\__fp_sep:
19735   {
19736     \exp_after:wN \__fp_ln_Taylor_loop:www
19737     \int_value:w \__fp_int_eval:w #1 - 2 \__fp_sep:
19738   }
19739   #3\__fp_sep:
19740 }
19741 \cs_new:Npn \__fp_ln_Taylor_break:w
19742   \fi: #1 \__fp_fixed_add:wwn #2#3\__fp_sep: #4 \__fp_sep:\__fp_sep:
19743 {
19744   \fi:
19745   \exp_after:wN \__fp_fixed_mul:wwn
19746   \exp_after:wN { \int_value:w \__fp_int_eval:w 10000 + #2 } #3\__fp_sep:
19747 }
19748 \cs_new:Npn \__fp_ln_c:NwNw #1 #2\__fp_sep: #3
19749 {
19750   \if_meaning:w + #1
19751     \exp_after:wN \exp_after:wN \exp_after:wN \__fp_fixed_sub:wwn
19752   \else:
19753     \exp_after:wN \exp_after:wN \exp_after:wN \__fp_fixed_add:wwn
19754   \fi:
19755   #3 #2 \__fp_sep:
19756 }
19757 \cs_new:Npn \__fp_ln_exponent:wn #1\__fp_sep: #2
19758 {
19759   \if_case:w #2 \exp_stop_f:
19760     0 \__fp_case_return:nw { \__fp_fixed_to_float_o:Nw 2 }
19761   \or:
19762     \exp_after:wN \__fp_ln_exponent_one:ww \int_value:w
19763   \else:
19764     \if_int_compare:w #2 > \c_zero_int
19765       \exp_after:wN \__fp_ln_exponent_small:NNww
19766     \exp_after:wN 0

```

19767	\exp_after:wN __fp_fixed_sub:wwn \int_value:w	19767
19768	\else:	19768
19769	\exp_after:wN __fp_ln_exponent_small:NNww	19769
19770	\exp_after:wN 2	19770
19771	\exp_after:wN __fp_fixed_add:wwn \int_value:w -	19771
19772	\fi:	19772
19773	\fi:	19773
19774	#2__fp_sep: #1__fp_sep:	19774
19775	}	19775
19776	\cs_new:Npn __fp_ln_exponent_one:ww 1__fp_sep: #1__fp_sep:	19776
19777	{	19777
19778	0	19778
19779	\exp_after:wN __fp_fixed_sub:wwn \c__fp_ln_x_fixed_tl #1__fp_sep:	19779
19780	__fp_fixed_to_float_o:wN 0	19780
19781	}	19781
19782	\cs_new:Npn __fp_ln_exponent_small:NNww #1#2#3__fp_sep: #4#5#6#7#8#9__fp_sep:	19782
19783	{	19783
19784	4	19784
19785	\exp_after:wN __fp_fixed_mul:wwn	19785
19786	\c__fp_ln_x_fixed_tl	19786
19787	{#3}{0000}{0000}{0000}{0000}{0000} __fp_sep:	19787
19788	#2	19788
19789	{0000}{#4}{#5}{#6}{#7}{#8}__fp_sep:	19789
19790	__fp_fixed_to_float_o:wN #1	19790
19791	}	19791
19792	\cs_new:Npn __fp_exp_o:w #1 \s__fp __fp_chk:w #2#3#4__fp_sep: @	19792
19793	{	19793
19794	\if_case:w #2 \exp_stop_f:	19794
19795	__fp_case_return_o:Nw \c_one_fp	19795
19796	\or:	19796
19797	\exp_after:wN __fp_exp_normal_o:w	19797
19798	\or:	19798
19799	\if_meaning:w 0 #3	19799
19800	\exp_after:wN __fp_case_return_o:Nw	19800
19801	\exp_after:wN \c_inf_fp	19801
19802	\else:	19802
19803	\exp_after:wN __fp_case_return_o:Nw	19803
19804	\exp_after:wN \c_zero_fp	19804
19805	\fi:	19805
19806	\or:	19806
19807	__fp_case_return_same_o:w	19807
19808	\fi:	19808
19809	\s__fp __fp_chk:w #2#3#4__fp_sep:	19809
19810	}	19810
19811	\cs_new:Npn __fp_exp_normal_o:w \s__fp __fp_chk:w 1#1	19811
19812	{	19812

19813	\if_meaning:w 0 #1	19813
19814	_fp_exp_pos_o:NNwnw + _fp_fixed_to_float_o:wN	19814
19815	\else:	19815
19816	_fp_exp_pos_o:NNwnw - _fp_fixed_inv_to_float_o:wN	19816
19817	\fi:	19817
19818	}	19818
19819	\cs_new:Npn _fp_exp_pos_o:NNwnw #1#2#3 \fi: #4#5_fp_sep:	19819
19820	{	19820
19821	\fi:	19821
19822	\if_int_compare:w #4 > \c__fp_max_exp_exponent_int	19822
19823	\token_if_eq_charcode:NNTF + #1	19823
19824	{ _fp_exp_overflow:NN _fp_overflow:w \c_inf_fp }	19824
19825	{ _fp_exp_overflow:NN _fp_underflow:w \c_zero_fp }	19825
19826	\exp:w	19826
19827	\else:	19827
19828	\exp_after:wN _fp_sanitize:Nw	19828
19829	\exp_after:wN 0	19829
19830	\int_value:w #1 _fp_int_eval:w	19830
19831	\if_int_compare:w #4 < \c_zero_int	19831
19832	\exp_after:wN \use_i:nn	19832
19833	\else:	19833
19834	\exp_after:wN \use_ii:nn	19834
19835	\fi:	19835
19836	{	19836
19837	0	19837
19838	_fp_decimate:nNnnnn { - #4 }	19838
19839	_fp_exp_Taylor:Nnnwn	19839
19840	}	19840
19841	{	19841
19842	_fp_decimate:nNnnnn { \c__fp_prec_int - #4 }	19842
19843	_fp_exp_pos_large:NnnNwn	19843
19844	}	19844
19845	#5	19845
19846	{#4}	19846
19847	#1 #2 0	19847
19848	\exp:w	19848
19849	\fi:	19849
19850	\exp_after:wN \exp_end:	19850
19851	}	19851
19852	\cs_new:Npn _fp_exp_overflow:NN #1#2	19852
19853	{	19853
19854	\exp_after:wN \exp_after:wN	19854
19855	\exp_after:wN #1	19855
19856	\exp_after:wN #2	19856
19857	}	19857
19858	\cs_new:Npn _fp_exp_Taylor:Nnnwn #1#2#3 #4_fp_sep: #5 #6	19858

```
19859 { 19859
19860 #6 19860
19861 \__fp_pack_twice_four:wNNNNNNNN 19861
19862 \__fp_pack_twice_four:wNNNNNNNN 19862
19863 \__fp_pack_twice_four:wNNNNNNNN 19863
19864 \__fp_exp_Taylor_ii:ww 19864
19865 \__fp_sep: #2#3#4 0000 0000 \__fp_sep: 19865
19866 } 19866
19867 \cs_new:Npn \__fp_exp_Taylor_ii:ww #1\__fp_sep: #2\__fp_sep: 19867
19868 { \__fp_exp_Taylor_loop:www 10 \__fp_sep: #1 \__fp_sep: #1 \__fp_sep: \s__fp_stop } 19868
19869 \cs_new:Npn \__fp_exp_Taylor_loop:www #1\__fp_sep: #2\__fp_sep: #3\__fp_sep: 19869
19870 { 19870
19871 \if_int_compare:w #1 = \c_one_int 19871
19872 \exp_after:wN \__fp_exp_Taylor_break:Nww 19872
19873 \fi: 19873
19874 \__fp_fixed_div_int:wwN #3 \__fp_sep: #1 \__fp_sep: 19874
19875 \__fp_fixed_add_one:wN 19875
19876 \__fp_fixed_mul:wwn #2 \__fp_sep: 19876
19877 { 19877
19878 \exp_after:wN \__fp_exp_Taylor_loop:www 19878
19879 \int_value:w \__fp_int_eval:w #1 - 1 \__fp_sep: 19879
19880 #2 \__fp_sep: 19880
19881 } 19881
19882 } 19882
19883 \cs_new:Npn \__fp_exp_Taylor_break:Nww #1 #2\__fp_sep: #3 \s__fp_stop 19883
19884 { \__fp_fixed_add_one:wN #2 \__fp_sep: } 19884
19885 \intarray_const_from_clist:Nn \c__fp_exp_intarray 19885
19886 { 19886
19887 1 , 1 1105 1709 , 1 1807 5647 , 1 6248 1171 , 19887
19888 1 , 1 1221 4027 , 1 5816 0169 , 1 8339 2107 , 19888
19889 1 , 1 1349 8588 , 1 0757 6003 , 1 1039 8374 , 19889
19890 1 , 1 1491 8246 , 1 9764 1270 , 1 3178 2485 , 19890
19891 1 , 1 1648 7212 , 1 7070 0128 , 1 1468 4865 , 19891
19892 1 , 1 1822 1188 , 1 0039 0508 , 1 9748 7537 , 19892
19893 1 , 1 2013 7527 , 1 0747 0476 , 1 5216 2455 , 19893
19894 1 , 1 2225 5409 , 1 2849 2467 , 1 6045 7954 , 19894
19895 1 , 1 2459 6031 , 1 1115 6949 , 1 6638 0013 , 19895
19896 1 , 1 2718 2818 , 1 2845 9045 , 1 2353 6029 , 19896
19897 1 , 1 7389 0560 , 1 9893 0650 , 1 2272 3043 , 19897
19898 2 , 1 2008 5536 , 1 9231 8766 , 1 7740 9285 , 19898
19899 2 , 1 5459 8150 , 1 0331 4423 , 1 9078 1103 , 19899
19900 3 , 1 1484 1315 , 1 9102 5766 , 1 0342 1116 , 19900
19901 3 , 1 4034 2879 , 1 3492 7351 , 1 2260 8387 , 19901
19902 4 , 1 1096 6331 , 1 5842 8458 , 1 5992 6372 , 19902
19903 4 , 1 2980 9579 , 1 8704 1728 , 1 2747 4359 , 19903
19904 4 , 1 8103 0839 , 1 2757 5384 , 1 0077 1000 , 19904
```


19905	5 , 1 2202 6465 , 1 7948 0671 , 1 6516 9579 ,	19905
19906	9 , 1 4851 6519 , 1 5409 7902 , 1 7796 9107 ,	19906
19907	14 , 1 1068 6474 , 1 5815 2446 , 1 2146 9905 ,	19907
19908	18 , 1 2353 8526 , 1 6837 0199 , 1 8540 7900 ,	19908
19909	22 , 1 5184 7055 , 1 2858 7072 , 1 4640 8745 ,	19909
19910	27 , 1 1142 0073 , 1 8981 5684 , 1 2836 6296 ,	19910
19911	31 , 1 2515 4386 , 1 7091 9167 , 1 0062 6578 ,	19911
19912	35 , 1 5540 6223 , 1 8439 3510 , 1 0525 7117 ,	19912
19913	40 , 1 1220 4032 , 1 9431 7840 , 1 8020 0271 ,	19913
19914	44 , 1 2688 1171 , 1 4181 6135 , 1 4484 1263 ,	19914
19915	87 , 1 7225 9737 , 1 6812 5749 , 1 2581 7748 ,	19915
19916	131 , 1 1942 4263 , 1 9524 1255 , 1 9365 8421 ,	19916
19917	174 , 1 5221 4696 , 1 8976 4143 , 1 9505 8876 ,	19917
19918	218 , 1 1403 5922 , 1 1785 2837 , 1 4107 3977 ,	19918
19919	261 , 1 3773 0203 , 1 0092 9939 , 1 8234 0143 ,	19919
19920	305 , 1 1014 2320 , 1 5473 5004 , 1 5094 5533 ,	19920
19921	348 , 1 2726 3745 , 1 7211 2566 , 1 5673 6478 ,	19921
19922	391 , 1 7328 8142 , 1 2230 7421 , 1 7051 8866 ,	19922
19923	435 , 1 1970 0711 , 1 1401 7046 , 1 9938 8888 ,	19923
19924	869 , 1 3881 1801 , 1 9428 4368 , 1 5764 8232 ,	19924
19925	1303 , 1 7646 2009 , 1 8905 4704 , 1 8893 1073 ,	19925
19926	1738 , 1 1506 3559 , 1 7005 0524 , 1 9009 7592 ,	19926
19927	2172 , 1 2967 6283 , 1 8402 3667 , 1 0689 6630 ,	19927
19928	2606 , 1 5846 4389 , 1 5650 2114 , 1 7278 5046 ,	19928
19929	3041 , 1 1151 7900 , 1 5080 6878 , 1 2914 4154 ,	19929
19930	3475 , 1 2269 1083 , 1 0850 6857 , 1 8724 4002 ,	19930
19931	3909 , 1 4470 3047 , 1 3316 5442 , 1 6408 6591 ,	19931
19932	4343 , 1 8806 8182 , 1 2566 2921 , 1 5872 6150 ,	19932
19933	8686 , 1 7756 0047 , 1 2598 6861 , 1 0458 3204 ,	19933
19934	13029 , 1 6830 5723 , 1 7791 4884 , 1 1932 7351 ,	19934
19935	17372 , 1 6015 5609 , 1 3095 3052 , 1 3494 7574 ,	19935
19936	21715 , 1 5297 7951 , 1 6443 0315 , 1 3251 3576 ,	19936
19937	26058 , 1 4665 6719 , 1 0099 3379 , 1 5527 2929 ,	19937
19938	30401 , 1 4108 9724 , 1 3326 3186 , 1 5271 5665 ,	19938
19939	34744 , 1 3618 6973 , 1 3140 0875 , 1 3856 4102 ,	19939
19940	39087 , 1 3186 9209 , 1 6113 3900 , 1 6705 9685 ,	19940
19941	}	19941
19942	\cs_new:Npn __fp_exp_pos_large:NnnNwn #1#2#3 #4#5__fp_sep: #6	19942
19943	{	19943
19944	\exp_after:wN \exp_after:wN \exp_after:wN __fp_exp_large:NwN	19944
19945	\exp_after:wN \exp_after:wN \exp_after:wN #6	19945
19946	\exp_after:wN \c__fp_one_fixed_tl	19946
19947	\int_value:w #3 #4 \exp_stop_f:	19947
19948	#5 00000 __fp_sep:	19948
19949	}	19949
19950	\cs_new:Npn __fp_exp_large:NwN #1#2__fp_sep: #3	19950

```

19951 {
19952     \if_case:w #3 ~
19953         \exp_after:wN \__fp_fixed_continue:wn
19954     \else:
19955         \exp_after:wN \__fp_exp_intarray:w
19956         \int_value:w \__fp_int_eval:w 36 * #1 + 4 * #3 \exp_after:wN \__fp_sep:
19957     \fi:
19958     #2\__fp_sep:
19959     {
19960         \if_meaning:w 0 #1
19961         \exp_after:wN \__fp_exp_large_after:wwn
19962     \else:
19963         \exp_after:wN \__fp_exp_large:NwN
19964         \int_value:w \__fp_int_eval:w #1 - 1 \exp_after:wN \scan_stop:
19965     \fi:
19966     }
19967 }
19968 \cs_new:Npn \__fp_exp_intarray:w #1 \__fp_sep:
19969 {
19970     +
19971     \__kernel_intarray_item:Nn \c__fp_exp_intarray
19972     { \__fp_int_eval:w #1 - 3 \scan_stop: }
19973     \exp_after:wN \use_i:nnn
19974     \exp_after:wN \__fp_fixed_mul:wwn
19975     \int_value:w 0
19976     \exp_after:wN \__fp_exp_intarray_aux:w
19977     \int_value:w \__kernel_intarray_item:Nn
19978         \c__fp_exp_intarray { \__fp_int_eval:w #1 - 2 }
19979     \exp_after:wN \__fp_exp_intarray_aux:w
19980     \int_value:w \__kernel_intarray_item:Nn
19981         \c__fp_exp_intarray { \__fp_int_eval:w #1 - 1 }
19982     \exp_after:wN \__fp_exp_intarray_aux:w
19983     \int_value:w \__kernel_intarray_item:Nn
19984         \c__fp_exp_intarray {#1} \__fp_sep: \__fp_sep:
19985 }
19986 \cs_new:Npn \__fp_exp_intarray_aux:w 1 #1#2#3#4#5 \__fp_sep:
19987 { \__fp_sep: {#1#2#3#4} {#5} }
19988 \cs_new:Npn \__fp_exp_large_after:wwn #1\__fp_sep: #2\__fp_sep: #3
19989 {
19990     \__fp_exp_Taylor:Nnnwn ? { } { } 0 #2\__fp_sep: { } #3
19991     \__fp_fixed_mul:wwn #1\__fp_sep:
19992 }
19993 \cs_new:cpn { __fp_ \iow_char:N \^_o:ww }
19994     \s_fp \__fp_chk:w #1#2#3\__fp_sep: \s_fp \__fp_chk:w #4#5#6\__fp_sep:
19995 {
19996     \if_meaning:w 0 #4

```

19997	__fp_case_return_o:Nw \c_one_fp	19997
19998	\fi:	19998
19999	\if_case:w #2 \exp_stop_f:	19999
20000	\exp_after:wN \use_i:nn	20000
20001	\or:	20001
20002	__fp_case_return_o:Nw \c_nan_fp	20002
20003	\else:	20003
20004	\exp_after:wN __fp_pow_neg:www	20004
20005	\exp:w \exp_end_continue_f:w \exp_after:wN \use:nn	20005
20006	\fi:	20006
20007	{	20007
20008	\if_meaning:w 1 #1	20008
20009	\exp_after:wN __fp_pow_normal_o:ww	20009
20010	\else:	20010
20011	\exp_after:wN __fp_pow_zero_or_inf:ww	20011
20012	\fi:	20012
20013	\s__fp __fp_chk:w #1#2#3__fp_sep:	20013
20014	}	20014
20015	{ \s__fp __fp_chk:w #4#5#6__fp_sep: \s__fp __fp_chk:w #1#2#3__fp_sep: }	20015
20016	\s__fp __fp_chk:w #4#5#6__fp_sep:	20016
20017	}	20017
20018	\cs_new:Npn __fp_pow_zero_or_inf:ww	20018
20019	\s__fp __fp_chk:w #1#2__fp_sep: \s__fp __fp_chk:w #3#4	20019
20020	{	20020
20021	\if_meaning:w 1 #4	20021
20022	__fp_case_return_same_o:w	20022
20023	\fi:	20023
20024	\if_meaning:w #1 #4	20024
20025	__fp_case_return_o:Nw \c_zero_fp	20025
20026	\fi:	20026
20027	\if_meaning:w 2 #1	20027
20028	__fp_case_return_o:Nw \c_inf_fp	20028
20029	\fi:	20029
20030	\if_meaning:w 2 #3	20030
20031	__fp_case_return_o:Nw \c_inf_fp	20031
20032	\else:	20032
20033	__fp_case_use:nw	20033
20034	{	20034
20035	__fp_division_by_zero_o:NNww \c_inf_fp ^	20035
20036	\s__fp __fp_chk:w #1 #2 __fp_sep:	20036
20037	}	20037
20038	\fi:	20038
20039	\s__fp __fp_chk:w #3#4	20039
20040	}	20040
20041	\cs_new:Npn __fp_pow_normal_o:ww	20041
20042	\s__fp __fp_chk:w 1 #1#2#3__fp_sep: \s__fp __fp_chk:w #4#5	20042

20043	{	20043
20044	\if:w 0 __fp_str_if_eq:nn { #2 #3 } { 1 {1000} {0000} {0000} {0000} }	20044
20045	\if_int_compare:w #4 #1 = 32 \exp_stop_f:	20045
20046	\exp_after:wN __fp_case_return_ii_o:ww	20046
20047	\fi:	20047
20048	__fp_case_return_o:Nww \c_one_fp	20048
20049	\fi:	20049
20050	\if_case:w #4 \exp_stop_f:	20050
20051	\or:	20051
20052	\exp_after:wN __fp_pow_npos_o:Nww	20052
20053	\exp_after:wN #5	20053
20054	\or:	20054
20055	\if_meaning:w 2 #5 \exp_after:wN \reverse_if:N \fi:	20055
20056	\if_int_compare:w #2 > \c_zero_int	20056
20057	\exp_after:wN __fp_case_return_o:Nww	20057
20058	\exp_after:wN \c_inf_fp	20058
20059	\else:	20059
20060	\exp_after:wN __fp_case_return_o:Nww	20060
20061	\exp_after:wN \c_zero_fp	20061
20062	\fi:	20062
20063	\or:	20063
20064	__fp_case_return_ii_o:ww	20064
20065	\fi:	20065
20066	\s__fp __fp_chk:w 1 #1 {#2} #3 __fp_sep:	20066
20067	\s__fp __fp_chk:w #4 #5	20067
20068	}	20068
20069	\cs_new:Npn __fp_pow_npos_o:Nww #1 \s__fp __fp_chk:w 1#2#3	20069
20070	{	20070
20071	\exp_after:wN __fp_sanitize:Nw	20071
20072	\exp_after:wN 0	20072
20073	\int_value:w	20073
20074	\if:w #1 \if_int_compare:w #3 > \c_zero_int 0 \else: 2 \fi:	20074
20075	\exp_after:wN __fp_pow_npos_aux:NNnww	20075
20076	\exp_after:wN +	20076
20077	\exp_after:wN __fp_fixed_to_float_o:wN	20077
20078	\else:	20078
20079	\exp_after:wN __fp_pow_npos_aux:NNnww	20079
20080	\exp_after:wN -	20080
20081	\exp_after:wN __fp_fixed_inv_to_float_o:wN	20081
20082	\fi:	20082
20083	{#3}	20083
20084	}	20084
20085	\cs_new:Npn __fp_pow_npos_aux:NNnww	20085
20086	#1#2#3#4#5__fp_sep: \s__fp __fp_chk:w 1#6#7#8__fp_sep:	20086
20087	{	20087
20088	#1	20088

```

20089 \__fp_int_eval:w
20090 \__fp_ln_significand:NNNNnnnnN #4#5
20091 \__fp_pow_exponent:wnN {#3}
20092 \__fp_fixed_mul:wwn #8 {0000}{0000} \__fp_sep:
20093 \__fp_pow_B:wwN #7\__fp_sep:
20094 #1 #2 0 % fixed_to_float_o:wn
20095 }
20096 \cs_new:Npn \__fp_pow_exponent:wnN #1\__fp_sep: #2
20097 {
20098 \if_int_compare:w #2 > \c_zero_int
20099 \exp_after:wN \__fp_pow_exponent:Nwnnnnnw % n\ln(10) - (-\ln(x))
20100 \exp_after:wN +
20101 \else:
20102 \exp_after:wN \__fp_pow_exponent:Nwnnnnnw % -(|n|\ln(10) + (-\ln(x)))
20103 \exp_after:wN -
20104 \fi:
20105 #2\__fp_sep: #1\__fp_sep:
20106 }
20107 \cs_new:Npn \__fp_pow_exponent:Nwnnnnnw #1#2\__fp_sep: #3#4#5#6#7#8\__fp_sep:
20108 { %^A todo: use that in ln.
20109 \exp_after:wN \__fp_fixed_mul_after:wwn
20110 \int_value:w \__fp_int_eval:w \c_fp_leading_shift_int
20111 \exp_after:wN \__fp_pack:NNNNNw
20112 \int_value:w \__fp_int_eval:w \c_fp_middle_shift_int
20113 #1#2*23025 - #1 #3
20114 \exp_after:wN \__fp_pack:NNNNNw
20115 \int_value:w \__fp_int_eval:w \c_fp_middle_shift_int
20116 #1 #2*8509 - #1 #4
20117 \exp_after:wN \__fp_pack:NNNNNw
20118 \int_value:w \__fp_int_eval:w \c_fp_middle_shift_int
20119 #1 #2*2994 - #1 #5
20120 \exp_after:wN \__fp_pack:NNNNNw
20121 \int_value:w \__fp_int_eval:w \c_fp_middle_shift_int
20122 #1 #2*0456 - #1 #6
20123 \exp_after:wN \__fp_pack:NNNNNw
20124 \int_value:w \__fp_int_eval:w \c_fp_trailing_shift_int
20125 #1 #2*8401 - #1 #7
20126 #1 ( #2*7991 - #8 ) / 1 0000 \__fp_sep: \__fp_sep:
20127 }
20128 \cs_new:Npn \__fp_pow_B:wwN #1#2#3#4#5#6\__fp_sep: #7\__fp_sep:
20129 {
20130 \if_int_compare:w #7 < \c_zero_int
20131 \exp_after:wN \__fp_pow_C_neg:w \int_value:w -
20132 \else:
20133 \if_int_compare:w #7 < 22 \exp_stop_f:
20134 \exp_after:wN \__fp_pow_C_pos:w \int_value:w

```

```

20135         \else:
20136             \exp_after:wN \__fp_pow_C_overflow:w \int_value:w
20137         \fi:
20138     \fi:
20139     #7 \exp_after:wN \__fp_sep:
20140     \int_value:w \__fp_int_eval:w 10 0000 + #1 \__fp_int_eval_end:
20141     #2#3#4#5#6 0000 0000 0000 0000 0000 0000 \__fp_sep: %^^A todo: how many 0?
20142 }
20143 \cs_new:Npn \__fp_pow_C_overflow:w #1\__fp_sep: #2\__fp_sep: #3
20144 {
20145     + 2 * \c__fp_max_exponent_int
20146     \exp_after:wN \__fp_fixed_continue:wn \c__fp_one_fixed_tl
20147 }
20148 \cs_new:Npn \__fp_pow_C_neg:w #1 \__fp_sep: 1
20149 {
20150     \exp_after:wN \exp_after:wN \exp_after:wN \__fp_pow_C_pack:w
20151     \prg_replicate:nn {#1} {0}
20152 }
20153 \cs_new:Npn \__fp_pow_C_pos:w #1\__fp_sep: 1
20154 { \__fp_pow_C_pos_loop:wN #1\__fp_sep: }
20155 \cs_new:Npn \__fp_pow_C_pos_loop:wN #1\__fp_sep: #2
20156 {
20157     \if_meaning:w 0 #1
20158         \exp_after:wN \__fp_pow_C_pack:w
20159         \exp_after:wN #2
20160     \else:
20161         \if_meaning:w 0 #2
20162             \exp_after:wN \__fp_pow_C_pos_loop:wN \int_value:w
20163         \else:
20164             \exp_after:wN \__fp_pow_C_overflow:w \int_value:w
20165         \fi:
20166         \__fp_int_eval:w #1 - 1 \exp_after:wN \__fp_sep:
20167     \fi:
20168 }
20169 \cs_new:Npn \__fp_pow_C_pack:w
20170 {
20171     \exp_after:wN \__fp_exp_large:NwN
20172     \exp_after:wN 5
20173     \c__fp_one_fixed_tl
20174 }
20175 \cs_new:Npn \__fp_pow_neg:www
20176     \s__fp \__fp_chk:w #1#2\__fp_sep: #3\__fp_sep: #4\__fp_sep:
20177 {
20178     \if_case:w \__fp_pow_neg_case:w #4 \__fp_sep:
20179         \exp_after:wN \__fp_pow_neg_aux:wNN
20180     \or:

```

20181	\if_int_compare:w __fp_int_eval:w #1 / 2 = \c_one_int	20181
20182	__fp_invalid_operation_o:Nww ^ #3__fp_sep: #4__fp_sep:	20182
20183	\exp:w \exp_end_continue_f:w	20183
20184	\exp_after:wN \exp_after:wN	20184
20185	\exp_after:wN __fp_use_none_until_s:w	20185
20186	\fi:	20186
20187	\fi:	20187
20188	__fp_exp_after_o:w	20188
20189	\s__fp __fp_chk:w #1#2__fp_sep:	20189
20190	}	20190
20191	\cs_new:Npn __fp_pow_neg_aux:wNN #1 \s__fp __fp_chk:w #2#3	20191
20192	{	20192
20193	\exp_after:wN __fp_exp_after_o:w	20193
20194	\exp_after:wN \s__fp	20194
20195	\exp_after:wN __fp_chk:w	20195
20196	\exp_after:wN #2	20196
20197	\int_value:w __fp_int_eval:w 2 - #3 __fp_int_eval_end:	20197
20198	}	20198
20199	\cs_new:Npn __fp_pow_neg_case:w \s__fp __fp_chk:w #1#2#3__fp_sep:	20199
20200	{	20200
20201	\if_case:w #1 \exp_stop_f:	20201
20202	-1	20202
20203	\or: __fp_pow_neg_case_aux:nnnnn #3	20203
20204	\or: -1	20204
20205	\else: 1	20205
20206	\fi:	20206
20207	\exp_stop_f:	20207
20208	}	20208
20209	\cs_new:Npn __fp_pow_neg_case_aux:nnnnn #1#2#3#4#5	20209
20210	{	20210
20211	\if_int_compare:w #1 > \c__fp_prec_int	20211
20212	-1	20212
20213	\else:	20213
20214	__fp_decimate:nNnnnn { \c__fp_prec_int - #1 }	20214
20215	__fp_pow_neg_case_aux:Nnnw	20215
20216	{#2} {#3} {#4} {#5}	20216
20217	\fi:	20217
20218	}	20218
20219	\cs_new:Npn __fp_pow_neg_case_aux:Nnnw #1#2#3#4 __fp_sep:	20219
20220	{	20220
20221	\if_meaning:w 0 #1	20221
20222	\if_int_odd:w #3 \exp_stop_f:	20222
20223	0	20223
20224	\else:	20224
20225	-1	20225
20226	\fi:	20226


```
20227 \else: 20227
20228 1 20228
20229 \fi: 20229
20230 } 20230
20231 \int_const:Nn \c__fp_fact_max_arg_int { 3248 } 20231
20232 \cs_new:Npn \__fp_fact_o:w #1 \s__fp \__fp_chk:w #2#3#4\__fp_sep: @ 20232
20233 { 20233
20234 \if_case:w #2 \exp_stop_f: 20234
20235 \__fp_case_return_o:Nw \c_one_fp 20235
20236 \or: 20236
20237 \or: 20237
20238 \if_meaning:w 0 #3 20238
20239 \exp_after:wN \__fp_case_return_same_o:w 20239
20240 \fi: 20240
20241 \or: 20241
20242 \__fp_case_return_same_o:w 20242
20243 \fi: 20243
20244 \if_meaning:w 2 #3 20244
20245 \__fp_case_use:nw { \__fp_invalid_operation_o:fw { fact } } 20245
20246 \fi: 20246
20247 \__fp_fact_pos_o:w 20247
20248 \s__fp \__fp_chk:w #2 #3 #4 \__fp_sep: 20248
20249 } 20249
20250 \cs_new:Npn \__fp_fact_pos_o:w #1\__fp_sep: 20250
20251 { 20251
20252 \__fp_small_int:wTF #1\__fp_sep: 20252
20253 { \__fp_fact_int_o:n } 20253
20254 { \__fp_invalid_operation_o:fw { fact } #1\__fp_sep: } 20254
20255 } 20255
20256 \cs_new:Npn \__fp_fact_int_o:n #1 20256
20257 { 20257
20258 \if_int_compare:w #1 > \c__fp_fact_max_arg_int 20258
20259 \__fp_case_return:nw 20259
20260 { 20260
20261 \exp_after:wN \exp_after:wN \exp_after:wN \__fp_overflow:w 20261
20262 \exp_after:wN \c_inf_fp 20262
20263 } 20263
20264 \fi: 20264
20265 \exp_after:wN \__fp_sanitize:Nw 20265
20266 \exp_after:wN 0 20266
20267 \int_value:w \__fp_int_eval:w 20267
20268 \__fp_fact_loop_o:w #1 . 4 , { 1 } { } { } { } { } { } { } \__fp_sep: 20268
20269 } 20269
20270 \cs_new:Npn \__fp_fact_loop_o:w #1 . #2 \__fp_sep: 20270
20271 { 20271
20272 \if_int_compare:w #1 < 12 \exp_stop_f: 20272
```

```

20273 \__fp_fact_small_o:w #1
20274 \fi:
20275 \exp_after:wN \__fp_ep_mul:wwwwn
20276 \exp_after:wN 4 \exp_after:wN ,
20277 \exp_after:wN { \int_value:w \__fp_int_eval:w #1 * (#1 - 1) }
20278 { } { } { } { } { } { } \__fp_sep:
20279 #2 \__fp_sep:
20280 {
20281 \exp_after:wN \__fp_fact_loop_o:w
20282 \int_value:w \__fp_int_eval:w #1 - 2 .
20283 }
20284 }
20285 \cs_new:Npn \__fp_fact_small_o:w #1 \fi: #2 \__fp_sep: #3 \__fp_sep: #4
20286 {
20287 \fi:
20288 \exp_after:wN \__fp_ep_mul:wwwwn
20289 \exp_after:wN 4 \exp_after:wN ,
20290 \exp_after:wN
20291 {
20292 \int_value:w
20293 \if_case:w #1 \exp_stop_f:
20294 1 \or: 1 \or: 2 \or: 6 \or: 24 \or: 120 \or: 720 \or: 5040
20295 \or: 40320 \or: 362880 \or: 3628800 \or: 39916800
20296 \fi:
20297 { } { } { } { } { } { } { } \__fp_sep:
20298 #3 \__fp_sep:
20299 \__fp_ep_to_float_o:wwN 0
20300 }
20301 %% File: l3fp-trig.dtx
20302 \tl_map_inline:nn
20303 {
20304 {acos} {acsc} {asec} {asin}
20305 {cos} {cot} {csc} {sec} {sin} {tan}
20306 }
20307 {
20308 \cs_new:cpe { __fp_parse_word_#1:N }
20309 {
20310 \exp_not:N \__fp_parse_unary_function:NNN
20311 \exp_not:c { __fp_#1_o:w }
20312 \exp_not:N \use_i:nn
20313 }
20314 \cs_new:cpe { __fp_parse_word_#1d:N }
20315 {
20316 \exp_not:N \__fp_parse_unary_function:NNN
20317 \exp_not:c { __fp_#1_o:w }
20318 \exp_not:N \use_ii:nn

```

20319	}	20319
20320	}	20320
20321	\cs_new:Npn __fp_parse_word_acot:N	20321
20322	{ __fp_parse_function:NNN __fp_acot_o:Nw \use_i:nn }	20322
20323	\cs_new:Npn __fp_parse_word_acotd:N	20323
20324	{ __fp_parse_function:NNN __fp_acot_o:Nw \use_ii:nn }	20324
20325	\cs_new:Npn __fp_parse_word_atan:N	20325
20326	{ __fp_parse_function:NNN __fp_atan_o:Nw \use_i:nn }	20326
20327	\cs_new:Npn __fp_parse_word_atand:N	20327
20328	{ __fp_parse_function:NNN __fp_atan_o:Nw \use_ii:nn }	20328
20329	\cs_new:Npn __fp_sin_o:w #1 \s__fp __fp_chk:w #2#3#4__fp_sep: @	20329
20330	{	20330
20331	\if_case:w #2 \exp_stop_f:	20331
20332	__fp_case_return_same_o:w	20332
20333	\or: __fp_case_use:nw	20333
20334	{	20334
20335	__fp_trig:NNNNNwn #1 __fp_sin_series_o:NNwww	20335
20336	__fp_ep_to_float_o:wwN #3 0	20336
20337	}	20337
20338	\or: __fp_case_use:nw	20338
20339	{ __fp_invalid_operation_o:fw { #1 { sin } { sind } } }	20339
20340	\else: __fp_case_return_same_o:w	20340
20341	\fi:	20341
20342	\s__fp __fp_chk:w #2 #3 #4__fp_sep:	20342
20343	}	20343
20344	\cs_new:Npn __fp_cos_o:w #1 \s__fp __fp_chk:w #2#3__fp_sep: @	20344
20345	{	20345
20346	\if_case:w #2 \exp_stop_f:	20346
20347	__fp_case_return_o:Nw \c_one_fp	20347
20348	\or: __fp_case_use:nw	20348
20349	{	20349
20350	__fp_trig:NNNNNwn #1 __fp_sin_series_o:NNwww	20350
20351	__fp_ep_to_float_o:wwN 0 2	20351
20352	}	20352
20353	\or: __fp_case_use:nw	20353
20354	{ __fp_invalid_operation_o:fw { #1 { cos } { cosd } } }	20354
20355	\else: __fp_case_return_same_o:w	20355
20356	\fi:	20356
20357	\s__fp __fp_chk:w #2 #3__fp_sep:	20357
20358	}	20358
20359	\cs_new:Npn __fp_csc_o:w #1 \s__fp __fp_chk:w #2#3#4__fp_sep: @	20359
20360	{	20360
20361	\if_case:w #2 \exp_stop_f:	20361
20362	__fp_cot_zero_o:Nfw #3 { #1 { csc } { cscd } }	20362
20363	\or: __fp_case_use:nw	20363
20364	{	20364

20365	__fp_trig:NNNNNwn #1 __fp_sin_series_o:NNwww	20365
20366	__fp_ep_inv_to_float_o:wwN #3 0	20366
20367	}	20367
20368	\or: __fp_case_use:nw	20368
20369	{ __fp_invalid_operation_o:fw { #1 { csc } { cscd } } }	20369
20370	\else: __fp_case_return_same_o:w	20370
20371	\fi:	20371
20372	\s__fp __fp_chk:w #2 #3 #4__fp_sep:	20372
20373	}	20373
20374	\cs_new:Npn __fp_sec_o:w #1 \s__fp __fp_chk:w #2#3__fp_sep: @	20374
20375	{	20375
20376	\if_case:w #2 \exp_stop_f:	20376
20377	__fp_case_return_o:Nw \c_one_fp	20377
20378	\or: __fp_case_use:nw	20378
20379	{	20379
20380	__fp_trig:NNNNNwn #1 __fp_sin_series_o:NNwww	20380
20381	__fp_ep_inv_to_float_o:wwN 0 2	20381
20382	}	20382
20383	\or: __fp_case_use:nw	20383
20384	{ __fp_invalid_operation_o:fw { #1 { sec } { secd } } }	20384
20385	\else: __fp_case_return_same_o:w	20385
20386	\fi:	20386
20387	\s__fp __fp_chk:w #2 #3__fp_sep:	20387
20388	}	20388
20389	\cs_new:Npn __fp_tan_o:w #1 \s__fp __fp_chk:w #2#3#4__fp_sep: @	20389
20390	{	20390
20391	\if_case:w #2 \exp_stop_f:	20391
20392	__fp_case_return_same_o:w	20392
20393	\or: __fp_case_use:nw	20393
20394	{	20394
20395	__fp_trig:NNNNNwn #1	20395
20396	__fp_tan_series_o:NNwww 0 #3 1	20396
20397	}	20397
20398	\or: __fp_case_use:nw	20398
20399	{ __fp_invalid_operation_o:fw { #1 { tan } { tand } } }	20399
20400	\else: __fp_case_return_same_o:w	20400
20401	\fi:	20401
20402	\s__fp __fp_chk:w #2 #3 #4__fp_sep:	20402
20403	}	20403
20404	\cs_new:Npn __fp_cot_o:w #1 \s__fp __fp_chk:w #2#3#4__fp_sep: @	20404
20405	{	20405
20406	\if_case:w #2 \exp_stop_f:	20406
20407	__fp_cot_zero_o:Nfw #3 { #1 { cot } { cotd } }	20407
20408	\or: __fp_case_use:nw	20408
20409	{	20409
20410	__fp_trig:NNNNNwn #1	20410

20411 __fp_tan_series_o:NNwww 2 #3 3 20411

20412 } 20412

20413 \or: __fp_case_use:nw 20413

20414 { __fp_invalid_operation_o:fw { #1 { cot } { cotd } } } 20414

20415 \else: __fp_case_return_same_o:w 20415

20416 \fi: 20416

20417 \s__fp __fp_chk:w #2 #3 #4__fp_sep: 20417

20418 } 20418

20419 \cs_new:Npn __fp_cot_zero_o:Nfw #1#2#3 \fi: 20419

20420 { 20420

20421 \fi: 20421

20422 \token_if_eq_meaning:NNTF 0 #1 20422

20423 { \exp_args:NNf __fp_division_by_zero_o:Nnw \c_inf_fp } 20423

20424 { \exp_args:NNf __fp_division_by_zero_o:Nnw \c_minus_inf_fp } 20424

20425 {#2} 20425

20426 } 20426

20427 \cs_new:Npn __fp_trig:NNNNNwn #1#2#3#4#5 \s__fp __fp_chk:w 1#6#7#8__fp_sep: 20427

20428 { 20428

20429 \exp_after:wN #2 20429

20430 \exp_after:wN #3 20430

20431 \exp_after:wN #4 20431

20432 \int_value:w __fp_int_eval:w #5 20432

20433 \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN 20433

20434 \if_int_compare:w #7 > #1 0 1 \exp_stop_f: 20434

20435 #1 __fp_trig_large:ww __fp_trigd_large:ww 20435

20436 \else: 20436

20437 #1 __fp_trig_small:ww __fp_trigd_small:ww 20437

20438 \fi: 20438

20439 #7,#8{0000}{0000}__fp_sep: 20439

20440 } 20440

20441 \cs_new:Npn __fp_trig_small:ww #1,#2__fp_sep: 20441

20442 { __fp_ep_to_fixed:wwn #1,#2__fp_sep: . #1,#2__fp_sep: } 20442

20443 \cs_new:Npn __fp_trigd_small:ww #1,#2__fp_sep: 20443

20444 { 20444

20445 __fp_ep_mul_raw:wwwN 20445

20446 -1,{1745}{3292}{5199}{4329}{5769}{2369}__fp_sep: #1,#2__fp_sep: 20446

20447 __fp_trig_small:ww 20447

20448 } 20448

20449 \cs_new:Npn __fp_trigd_large:ww #1, #2#3#4#5#6#7__fp_sep: 20449

20450 { 20450

20451 \exp_after:wN __fp_pack_eight:wNNNNNNNN 20451

20452 \exp_after:wN __fp_pack_eight:wNNNNNNNN 20452

20453 \exp_after:wN __fp_pack_twice_four:wNNNNNNNN 20453

20454 \exp_after:wN __fp_pack_twice_four:wNNNNNNNN 20454

20455 \exp_after:wN __fp_trigd_large_auxi:nnnnwNNNN 20455

20456 \exp_after:wN __fp_sep: 20456

```
20457 \exp:w \exp_end_continue_f:w 20457
20458 \prg_replicate:nn { \int_max:nn { 22 - #1 } { 0 } } { 0 } 20458
20459 #2#3#4#5#6#7 0000 0000 0000 ! 20459
20460 } 20460
20461 \cs_new:Npn \__fp_trigd_large_auxi:nnnnwNNNN #1#2#3#4#5\__fp_sep: #6#7#8#9 20461
20462 { 20462
20463 \exp_after:wN \__fp_trigd_large_auxii:wNw 20463
20464 \int_value:w \__fp_int_eval:w #1 + #2 20464
20465 - (#1 + #2 - 4) / 9 * 9 \__fp_int_eval_end: 20465
20466 #3\__fp_sep: 20466
20467 #4\__fp_sep: #5{#6#7#8#9}\__fp_sep: 20467
20468 } 20468
20469 \cs_new:Npn \__fp_trigd_large_auxii:wNw #1\__fp_sep: #2#3\__fp_sep: 20469
20470 { 20470
20471 + (#1#2 - 4) / 9 * 2 20471
20472 \exp_after:wN \__fp_trigd_large_auxiii:www 20472
20473 \int_value:w \__fp_int_eval:w #1#2 20473
20474 - (#1#2 - 4) / 9 * 9 \__fp_int_eval_end: #3 \__fp_sep: 20474
20475 } 20475
20476 \cs_new:Npn \__fp_trigd_large_auxiii:www #1\__fp_sep: #2\__fp_sep: #3! 20476
20477 { 20477
20478 \if_int_compare:w #1 < 4500 \exp_stop_f: 20478
20479 \exp_after:wN \__fp_use_i_until_s:nw 20479
20480 \exp_after:wN \__fp_fixed_continue:wn 20480
20481 \else: 20481
20482 + 1 20482
20483 \fi: 20483
20484 \__fp_fixed_sub:wwn {9000}{0000}{0000}{0000}{0000}{0000}\__fp_sep: 20484
20485 {#1}#2{0000}{0000}\__fp_sep: 20485
20486 { \__fp_trigd_small:ww 2, } 20486
20487 } 20487
20488 \intarray_const_from_clist:Nn \c__fp_trig_intarray 20488
20489 { 20489
20490 100000000, 100000000, 115915494, 130918953, 135768883, 176337251, 20490
20491 143620344, 159645740, 145644874, 176673440, 158896797, 163422653, 20491
20492 150901138, 102766253, 108595607, 128427267, 157958036, 189291184, 20492
20493 161145786, 152877967, 141073169, 198392292, 139966937, 140907757, 20493
20494 130777463, 196925307, 168871739, 128962173, 197661693, 136239024, 20494
20495 117236290, 111832380, 111422269, 197557159, 140461890, 108690267, 20495
20496 139561204, 189410936, 193784408, 155287230, 199946443, 140024867, 20496
20497 123477394, 159610898, 132309678, 130749061, 166986462, 180469944, 20497
20498 186521878, 181574786, 156696424, 110389958, 174139348, 160998386, 20498
20499 180991999, 162442875, 158517117, 188584311, 117518767, 116054654, 20499
20500 175369880, 109739460, 136475933, 137680593, 102494496, 163530532, 20500
20501 171567755, 103220324, 177781639, 171660229, 146748119, 159816584, 20501
20502 106060168, 103035998, 113391198, 174988327, 186654435, 127975507, 20502
```

20503	100162406,	177564388,	184957131,	108801221,	199376147,	168137776,	20503
20504	147378906,	133068046,	145797848,	117613124,	127314069,	196077502,	20504
20505	145002977,	159857089,	105690279,	167851315,	125210016,	131774602,	20505
20506	109248116,	106240561,	145620314,	164840892,	148459191,	143521157,	20506
20507	154075562,	100871526,	160680221,	171591407,	157474582,	172259774,	20507
20508	162853998,	175155329,	139081398,	117724093,	158254797,	107332871,	20508
20509	190406999,	175907657,	170784934,	170393589,	182808717,	134256403,	20509
20510	166895116,	162545705,	194332763,	112686500,	126122717,	197115321,	20510
20511	112599504,	138667945,	103762556,	108363171,	116952597,	158128224,	20511
20512	194162333,	143145106,	112353687,	185631136,	136692167,	114206974,	20512
20513	169601292,	150578336,	105311960,	185945098,	139556718,	170995474,	20513
20514	165104316,	123815517,	158083944,	129799709,	199505254,	138756612,	20514
20515	194458833,	106846050,	178529151,	151410404,	189298850,	163881607,	20515
20516	176196993,	107341038,	199957869,	118905980,	193737772,	106187543,	20516
20517	122271893,	101366255,	126123878,	103875388,	181106814,	106765434,	20517
20518	108282785,	126933426,	179955607,	107903860,	160352738,	199624512,	20518
20519	159957492,	176297023,	159409558,	143011648,	129641185,	157771240,	20519
20520	157544494,	157021789,	176979240,	194903272,	194770216,	164960356,	20520
20521	153181535,	144003840,	168987471,	176915887,	163190966,	150696440,	20521
20522	147769706,	187683656,	177810477,	197954503,	153395758,	130188183,	20522
20523	186879377,	166124814,	195305996,	155802190,	183598751,	103512712,	20523
20524	190432315,	180498719,	168687775,	194656634,	162210342,	104440855,	20524
20525	149785037,	192738694,	129353661,	193778292,	187359378,	143470323,	20525
20526	102371458,	137923557,	111863634,	119294601,	183182291,	196416500,	20526
20527	187830793,	131353497,	179099745,	186492902,	167450609,	189368909,	20527
20528	145883050,	133703053,	180547312,	132158094,	131976760,	132283131,	20528
20529	141898097,	149822438,	133517435,	169898475,	101039500,	168388003,	20529
20530	197867235,	199608024,	100273901,	108749548,	154787923,	156826113,	20530
20531	199489032,	168997427,	108349611,	149208289,	103776784,	174303550,	20531
20532	145684560,	183671479,	130845672,	133270354,	185392556,	120208683,	20532
20533	193240995,	162211753,	131839402,	109707935,	170774965,	149880868,	20533
20534	160663609,	168661967,	103747454,	121028312,	119251846,	122483499,	20534
20535	111611495,	166556037,	196967613,	199312829,	196077608,	127799010,	20535
20536	107830360,	102338272,	198790854,	102387615,	157445430,	192601191,	20536
20537	100543379,	198389046,	154921248,	129516070,	172853005,	122721023,	20537
20538	160175233,	113173179,	175931105,	103281551,	109373913,	163964530,	20538
20539	157926071,	180083617,	195487672,	146459804,	173977292,	144810920,	20539
20540	109371257,	186918332,	189588628,	139904358,	168666639,	175673445,	20540
20541	114095036,	137327191,	174311388,	106638307,	125923027,	159734506,	20541
20542	105482127,	178037065,	133778303,	121709877,	134966568,	149080032,	20542
20543	169885067,	141791464,	168350828,	116168533,	114336160,	173099514,	20543
20544	198531198,	119733758,	144420984,	116559541,	152250643,	139431286,	20544
20545	144403838,	183561508,	179771645,	101706470,	167518774,	156059160,	20545
20546	187168578,	157939226,	123475633,	117111329,	198655941,	159689071,	20546
20547	198506887,	144230057,	151919770,	156900382,	118392562,	120338742,	20547
20548	135362568,	108354156,	151729710,	188117217,	195936832,	156488518,	20548

20549	174997487,	108553116,	159830610,	113921445,	144601614,	188452770,	20549
20550	125114110,	170248521,	173974510,	138667364,	103872860,	109967489,	20550
20551	131735618,	112071174,	104788993,	168886556,	192307848,	150230570,	20551
20552	157144063,	163863202,	136852010,	174100574,	185922811,	115721968,	20552
20553	100397824,	175953001,	166958522,	112303464,	118773650,	143546764,	20553
20554	164565659,	171901123,	108476709,	193097085,	191283646,	166919177,	20554
20555	169387914,	133315566,	150669813,	121641521,	100895711,	172862384,	20555
20556	126070678,	145176011,	113450800,	169947684,	122356989,	162488051,	20556
20557	157759809,	153397080,	185475059,	175362656,	149034394,	145420581,	20557
20558	178864356,	183042000,	131509559,	147434392,	152544850,	167491429,	20558
20559	108647514,	142303321,	133245695,	111634945,	167753939,	142403609,	20559
20560	105438335,	152829243,	142203494,	184366151,	146632286,	102477666,	20560
20561	166049531,	140657343,	157553014,	109082798,	180914786,	169343492,	20561
20562	127376026,	134997829,	195701816,	119643212,	133140475,	176289748,	20562
20563	140828911,	174097478,	126378991,	181699939,	148749771,	151989818,	20563
20564	172666294,	160183053,	195832752,	109236350,	168538892,	128468247,	20564
20565	125997252,	183007668,	156937583,	165972291,	198244297,	147406163,	20565
20566	181831139,	158306744,	134851692,	185973832,	137392662,	140243450,	20566
20567	119978099,	140402189,	161348342,	173613676,	144991382,	171541660,	20567
20568	163424829,	136374185,	106122610,	186132119,	198633462,	184709941,	20568
20569	183994274,	129559156,	128333990,	148038211,	175011612,	111667205,	20569
20570	119125793,	103552929,	124113440,	131161341,	112495318,	138592695,	20570
20571	184904438,	146807849,	109739828,	108855297,	104515305,	139914009,	20571
20572	188698840,	188365483,	166522246,	168624087,	125401404,	100911787,	20572
20573	142122045,	123075334,	173972538,	114940388,	141905868,	142311594,	20573
20574	163227443,	139066125,	116239310,	162831953,	123883392,	113153455,	20574
20575	163815117,	152035108,	174595582,	101123754,	135976815,	153401874,	20575
20576	107394340,	136339780,	138817210,	104531691,	182951948,	179591767,	20576
20577	139541778,	179243527,	161740724,	160593916,	102732282,	187946819,	20577
20578	136491289,	149714953,	143255272,	135916592,	198072479,	198580612,	20578
20579	169007332,	118844526,	179433504,	155801952,	149256630,	162048766,	20579
20580	116134365,	133992028,	175452085,	155344144,	109905129,	182727454,	20580
20581	165911813,	122232840,	151166615,	165070983,	175574337,	129548631,	20581
20582	120411217,	116380915,	160616116,	157320000,	183306114,	160618128,	20582
20583	103262586,	195951602,	146321661,	138576614,	180471993,	127077713,	20583
20584	116441201,	159496011,	106328305,	120759583,	148503050,	179095584,	20584
20585	198298218,	167402898,	138551383,	123957020,	180763975,	150429225,	20585
20586	198476470,	171016426,	197438450,	143091658,	164528360,	132493360,	20586
20587	143546572,	137557916,	113663241,	120457809,	196971566,	134022158,	20587
20588	180545794,	131328278,	100552461,	132088901,	187421210,	192448910,	20588
20589	141005215,	149680971,	113720754,	100571096,	134066431,	135745439,	20589
20590	191597694,	135788920,	179342561,	177830222,	137011486,	142492523,	20590
20591	192487287,	113132021,	176673607,	156645598,	127260957,	141566023,	20591
20592	143787436,	129132109,	174858971,	150713073,	191040726,	143541417,	20592
20593	197057222,	165479803,	181512759,	157912400,	125344680,	148220261,	20593
20594	173422990,	101020483,	106246303,	137964746,	178190501,	181183037,	20594

20595	151538028,	179523433,	141955021,	135689770,	191290561,	143178787,	20595
20596	192086205,	174499925,	178975690,	118492103,	124206471,	138519113,	20596
20597	188147564,	102097605,	154895793,	178514140,	141453051,	151583964,	20597
20598	128232654,	106020603,	131189158,	165702720,	186250269,	191639375,	20598
20599	115278873,	160608114,	155694842,	110322407,	177272742,	116513642,	20599
20600	134366992,	171634030,	194053074,	180652685,	109301658,	192136921,	20600
20601	141431293,	171341061,	157153714,	106203978,	147618426,	150297807,	20601
20602	186062669,	169960809,	118422347,	163350477,	146719017,	145045144,	20602
20603	161663828,	146208240,	186735951,	102371302,	190444377,	194085350,	20603
20604	134454426,	133413062,	163074595,	113830310,	122931469,	134466832,	20604
20605	185176632,	182415152,	110179422,	164439571,	181217170,	121756492,	20605
20606	119644493,	196532222,	118765848,	182445119,	109401340,	150443213,	20606
20607	198586286,	121083179,	139396084,	143898019,	114787389,	177233102,	20607
20608	186310131,	148695521,	126205182,	178063494,	157118662,	177825659,	20608
20609	188310053,	151552316,	165984394,	109022180,	163144545,	121212978,	20609
20610	197344714,	188741258,	126822386,	102360271,	109981191,	152056882,	20610
20611	134723983,	158013366,	106837863,	128867928,	161973236,	172536066,	20611
20612	185216856,	132011948,	197807339,	158419190,	166595838,	167852941,	20612
20613	124187182,	117279875,	106103946,	106481958,	157456200,	160892122,	20613
20614	184163943,	173846549,	158993202,	184812364,	133466119,	170732430,	20614
20615	195458590,	173361878,	162906318,	150165106,	126757685,	112163575,	20615
20616	188696307,	145199922,	100107766,	176830946,	198149756,	122682434,	20616
20617	179367131,	108412102,	119520899,	148191244,	140487511,	171059184,	20617
20618	141399078,	189455775,	118462161,	190415309,	134543802,	180893862,	20618
20619	180732375,	178615267,	179711433,	123241969,	185780563,	176301808,	20619
20620	184386640,	160717536,	183213626,	129671224,	126094285,	140110963,	20620
20621	121826276,	151201170,	122552929,	128965559,	146082049,	138409069,	20621
20622	107606920,	103954646,	119164002,	115673360,	117909631,	187289199,	20622
20623	186343410,	186903200,	157966371,	103128612,	135698881,	176403642,	20623
20624	152540837,	109810814,	183519031,	121318624,	172281810,	150845123,	20624
20625	169019064,	166322359,	138872454,	163073727,	128087898,	130041018,	20625
20626	194859136,	173742589,	141812405,	167291912,	138003306,	134499821,	20626
20627	196315803,	186381054,	124578934,	150084553,	128031351,	118843410,	20627
20628	107373060,	159565443,	173624887,	171292628,	198074235,	139074061,	20628
20629	178690578,	144431052,	174262641,	176783005,	182214864,	162289361,	20629
20630	192966929,	192033046,	169332843,	181580535,	164864073,	118444059,	20630
20631	195496893,	153773183,	167266131,	130108623,	158802128,	180432893,	20631
20632	144562140,	147978945,	142337360,	158506327,	104399819,	132635916,	20632
20633	168734194,	136567839,	101281912,	120281622,	195003330,	112236091,	20633
20634	185875592,	101959081,	122415367,	194990954,	148881099,	175891989,	20634
20635	108115811,	163538891,	163394029,	123722049,	184837522,	142362091,	20635
20636	100834097,	156679171,	100841679,	157022331,	178971071,	102928884,	20636
20637	189701309,	195339954,	124415335,	106062584,	139214524,	133864640,	20637
20638	134324406,	157317477,	155340540,	144810061,	177612569,	108474646,	20638
20639	114329765,	143900008,	138265211,	145210162,	136643111,	197987319,	20639
20640	102751191,	144121361,	169620456,	193602633,	161023559,	162140467,	20640

20641	102901215,	167964187,	135746835,	187317233,	110047459,	163339773,	20641
20642	124770449,	118885134,	141536376,	100915375,	164267438,	145016622,	20642
20643	113937193,	106748706,	128815954,	164819775,	119220771,	102367432,	20643
20644	189062690,	170911791,	194127762,	112245117,	123546771,	115640433,	20644
20645	135772061,	166615646,	174474627,	130562291,	133320309,	153340551,	20645
20646	138417181,	194605321,	150142632,	180008795,	151813296,	175497284,	20646
20647	167018836,	157425342,	150169942,	131069156,	134310662,	160434122,	20647
20648	105213831,	158797111,	150754540,	163290657,	102484886,	148697402,	20648
20649	187203725,	198692811,	149360627,	140384233,	128749423,	132178578,	20649
20650	177507355,	171857043,	178737969,	134023369,	102911446,	196144864,	20650
20651	197697194,	134527467,	144296030,	189437192,	154052665,	188907106,	20651
20652	162062575,	150993037,	199766583,	167936112,	181374511,	104971506,	20652
20653	115378374,	135795558,	167972129,	135876446,	130937572,	103221320,	20653
20654	124605656,	161129971,	131027586,	191128460,	143251843,	143269155,	20654
20655	129284585,	173495971,	150425653,	199302112,	118494723,	121323805,	20655
20656	116549802,	190991967,	168151180,	122483192,	151273721,	199792134,	20656
20657	133106764,	121874844,	126215985,	112167639,	167793529,	182985195,	20657
20658	185453921,	106957880,	158685312,	132775454,	133229161,	198905318,	20658
20659	190537253,	191582222,	192325972,	178133427,	181825606,	148823337,	20659
20660	160719681,	101448145,	131983362,	137910767,	112550175,	128826351,	20660
20661	183649210,	135725874,	110356573,	189469487,	154446940,	118175923,	20661
20662	106093708,	128146501,	185742532,	149692127,	164624247,	183221076,	20662
20663	154737505,	168198834,	156410354,	158027261,	125228550,	131543250,	20663
20664	139591848,	191898263,	104987591,	115406321,	103542638,	190012837,	20664
20665	142615518,	178773183,	175862355,	117537850,	169565995,	170028011,	20665
20666	158412588,	170150030,	117025916,	174630208,	142412449,	112839238,	20666
20667	105257725,	114737141,	123102301,	172563968,	130555358,	132628403,	20667
20668	183638157,	168682846,	143304568,	105994018,	170010719,	152092970,	20668
20669	117799058,	132164175,	179868116,	158654714,	177489647,	116547948,	20669
20670	183121404,	131836079,	184431405,	157311793,	149677763,	173989893,	20670
20671	102277656,	107058530,	140837477,	152640947,	143507039,	152145247,	20671
20672	101683884,	107090870,	161471944,	137225650,	128231458,	172995869,	20672
20673	173831689,	171268519,	139042297,	111072135,	107569780,	137262545,	20673
20674	181410950,	138270388,	198736451,	162848201,	180468288,	120582913,	20674
20675	153390138,	135649144,	130040157,	106509887,	192671541,	174507066,	20675
20676	186888783,	143805558,	135011967,	145862340,	180595327,	124727843,	20676
20677	182925939,	157715840,	136885940,	198993925,	152416883,	178793572,	20677
20678	179679516,	154076673,	192703125,	164187609,	162190243,	104699348,	20678
20679	159891990,	160012977,	174692145,	132970421,	167781726,	115178506,	20679
20680	153008552,	155999794,	102099694,	155431545,	127458567,	104403686,	20680
20681	168042864,	184045128,	181182309,	179349696,	127218364,	192935516,	20681
20682	120298724,	169583299,	148193297,	183358034,	159023227,	105261254,	20682
20683	121144370,	184359584,	194433836,	138388317,	175184116,	108817112,	20683
20684	151279233,	137457721,	193398208,	119005406,	132929377,	175306906,	20684
20685	160741530,	149976826,	147124407,	176881724,	186734216,	185881509,	20685
20686	191334220,	175930947,	117385515,	193408089,	157124410,	163472089,	20686

```
20687 131949128, 180783576, 131158294, 100549708, 191802336, 165960770, 20687
20688 170927599, 101052702, 181508688, 197828549, 143403726, 142729262, 20688
20689 110348701, 139928688, 153550062, 106151434, 130786653, 196085995, 20689
20690 100587149, 139141652, 106530207, 100852656, 124074703, 166073660, 20690
20691 153338052, 163766757, 120188394, 197277047, 122215363, 138511354, 20691
20692 183463624, 161985542, 159938719, 133367482, 104220974, 149956672, 20692
20693 170250544, 164232439, 157506869, 159133019, 137469191, 142980999, 20693
20694 134242305, 150172665, 121209241, 145596259, 160554427, 159095199, 20694
20695 168243130, 184279693, 171132070, 121049823, 123819574, 171759855, 20695
20696 119501864, 163094029, 175943631, 194450091, 191506160, 149228764, 20696
20697 132319212, 197034460, 193584259, 126727638, 168143633, 109856853, 20697
20698 127860243, 132141052, 133076065, 188414958, 158718197, 107124299, 20698
20699 159592267, 181172796, 144388537, 196763139, 127431422, 179531145, 20699
20700 100064922, 112650013, 132686230, 121550837, 20700
20701 } 20701
20702 \cs_new:Npn \__fp_trig_large:ww #1, #2#3#4#5#6\__fp_sep: 20702
20703 { 20703
20704 \exp_after:wN \__fp_trig_large_auxi:w 20704
20705 \int_value:w \__fp_int_eval:w (#1 - 4) / 8 \exp_after:wN , 20705
20706 \int_value:w #1 , \__fp_sep: 20706
20707 {#2}{#3}{#4}{#5} \__fp_sep: 20707
20708 } 20708
20709 \cs_new:Npn \__fp_trig_large_auxi:w #1, #2, 20709
20710 { 20710
20711 \exp_after:wN \exp_after:wN 20711
20712 \exp_after:wN \__fp_trig_large_auxiii:w 20712
20713 \cs:w 20713
20714 use_none:n \prg_replicate:nn { #2 - #1 * 8 } { n } 20714
20715 \exp_after:wN 20715
20716 \cs_end: 20716
20717 \int_value:w 20717
20718 \__kernel_intarray_item:Nn \c__fp_trig_intarray 20718
20719 { \__fp_int_eval:w #1 + 1 \scan_stop: } 20719
20720 \exp_after:wN \__fp_trig_large_auxiii:w \int_value:w 20720
20721 \__kernel_intarray_item:Nn \c__fp_trig_intarray 20721
20722 { \__fp_int_eval:w #1 + 2 \scan_stop: } 20722
20723 \exp_after:wN \__fp_trig_large_auxiii:w \int_value:w 20723
20724 \__kernel_intarray_item:Nn \c__fp_trig_intarray 20724
20725 { \__fp_int_eval:w #1 + 3 \scan_stop: } 20725
20726 \exp_after:wN \__fp_trig_large_auxiii:w \int_value:w 20726
20727 \__kernel_intarray_item:Nn \c__fp_trig_intarray 20727
20728 { \__fp_int_eval:w #1 + 4 \scan_stop: } 20728
20729 \exp_after:wN \__fp_trig_large_auxiii:w \int_value:w 20729
20730 \__kernel_intarray_item:Nn \c__fp_trig_intarray 20730
20731 { \__fp_int_eval:w #1 + 5 \scan_stop: } 20731
20732 \exp_after:wN \__fp_trig_large_auxiii:w \int_value:w 20732
```

```

20733 \__kernel_intarray_item:Nn \c__fp_trig_intarray 20733
20734 { \__fp_int_eval:w #1 + 6 \scan_stop: } 20734
20735 \exp_after:wN \__fp_trig_large_auxiii:w \int_value:w 20735
20736 \__kernel_intarray_item:Nn \c__fp_trig_intarray 20736
20737 { \__fp_int_eval:w #1 + 7 \scan_stop: } 20737
20738 \exp_after:wN \__fp_trig_large_auxiii:w \int_value:w 20738
20739 \__kernel_intarray_item:Nn \c__fp_trig_intarray 20739
20740 { \__fp_int_eval:w #1 + 8 \scan_stop: } 20740
20741 \exp_after:wN \__fp_trig_large_auxiii:w \int_value:w 20741
20742 \__kernel_intarray_item:Nn \c__fp_trig_intarray 20742
20743 { \__fp_int_eval:w #1 + 9 \scan_stop: } 20743
20744 \exp_stop_f: 20744
20745 } 20745
20746 \cs_new:Npn \__fp_trig_large_auxii:w 20746
20747 { 20747
20748 \__fp_pack_twice_four:wNNNNNNNN \__fp_pack_twice_four:wNNNNNNNN 20748
20749 \__fp_pack_twice_four:wNNNNNNNN \__fp_pack_twice_four:wNNNNNNNN 20749
20750 \__fp_pack_twice_four:wNNNNNNNN \__fp_pack_twice_four:wNNNNNNNN 20750
20751 \__fp_pack_twice_four:wNNNNNNNN \__fp_pack_twice_four:wNNNNNNNN 20751
20752 \__fp_trig_large_auxv:www \__fp_sep: 20752
20753 } 20753
20754 \cs_new:Npn \__fp_trig_large_auxiii:w 1 { \exp_stop_f: } 20754
20755 \cs_new:Npn \__fp_trig_large_auxv:www #1\__fp_sep: #2\__fp_sep: #3\__fp_sep: 20755
20756 { 20756
20757 \exp_after:wN \__fp_use_i_until_s:nw 20757
20758 \exp_after:wN \__fp_trig_large_auxvii:w 20758
20759 \int_value:w \__fp_int_eval:w \c__fp_leading_shift_int 20759
20760 \prg_replicate:nn { 13 } 20760
20761 { \__fp_trig_large_auxvi:wnnnnnnnn } 20761
20762 + \c__fp_trailing_shift_int - \c__fp_middle_shift_int 20762
20763 \__fp_use_i_until_s:nw 20763
20764 \__fp_sep: #3 #1 \__fp_sep: \__fp_sep: 20764
20765 } 20765
20766 \cs_new:Npn \__fp_trig_large_auxvi:wnnnnnnnn #1\__fp_sep: #2#3#4#5#6#7#8#9 20766
20767 { 20767
20768 \exp_after:wN \__fp_trig_large_pack:NNNNNw 20768
20769 \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int 20769
20770 + #2*#9 + #3*#8 + #4*#7 + #5*#6 20770
20771 #1\__fp_sep: {#2}{#3}{#4}{#5} {#7}{#8}{#9} 20771
20772 } 20772
20773 \cs_new:Npn \__fp_trig_large_pack:NNNNNw #1#2#3#4#5#6\__fp_sep: 20773
20774 { + #1#2#3#4#5 \__fp_sep: #6 } 20774
20775 \cs_new:Npn \__fp_trig_large_auxvii:w #1#2#3 20775
20776 { 20776
20777 \exp_after:wN \__fp_trig_large_auxviii:ww 20777
20778 \int_value:w \__fp_int_eval:w (#1#2#3 - 62) / 125 \__fp_sep: 20778

```

```

20779 #1#2#3
20780 }
20781 \cs_new:Npn \__fp_trig_large_auxviii:ww #1\__fp_sep:
20782 {
20783   + #1
20784   \if_int_odd:w #1 \exp_stop_f:
20785     \exp_after:wN \__fp_trig_large_auxix:Nw
20786     \exp_after:wN -
20787   \else:
20788     \exp_after:wN \__fp_trig_large_auxix:Nw
20789     \exp_after:wN +
20790   \fi:
20791 }
20792 \cs_new:Npn \__fp_trig_large_auxix:Nw
20793 {
20794   \exp_after:wN \__fp_use_i_until_s:nw
20795   \exp_after:wN \__fp_trig_large_auxxi:w
20796   \int_value:w \__fp_int_eval:w \c__fp_leading_shift_int
20797   \prg_replicate:nn { 13 }
20798   { \__fp_trig_large_auxx:wNNNNN }
20799   + \c__fp_trailing_shift_int - \c__fp_middle_shift_int
20800   \__fp_sep:
20801 }
20802 \cs_new:Npn \__fp_trig_large_auxx:wNNNNN #1\__fp_sep: #2 #3#4#5#6
20803 {
20804   \exp_after:wN \__fp_trig_large_pack:NNNNNw
20805   \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
20806   #2 8 * #3#4#5#6
20807   #1\__fp_sep: #2
20808 }
20809 \cs_new:Npn \__fp_trig_large_auxxi:w #1\__fp_sep:
20810 {
20811   \exp_after:wN \__fp_ep_mul_raw:wwwN
20812   \int_value:w \__fp_int_eval:w 0 \__fp_ep_to_ep_loop:N #1 \__fp_sep: \__fp_sep: !
20813   0,{7853}{9816}{3397}{4483}{0961}{5661}\__fp_sep:
20814   \__fp_trig_small:ww
20815 }
20816 \cs_new:Npn \__fp_sin_series_o:NNwww #1#2#3. #4\__fp_sep:
20817 {
20818   \__fp_fixed_mul:wwn #4\__fp_sep: #4\__fp_sep:
20819   {
20820     \exp_after:wN \__fp_sin_series_aux_o:NNwww
20821     \exp_after:wN #1
20822     \int_value:w
20823     \if_int_odd:w \__fp_int_eval:w (#3 + 2) / 4 \__fp_int_eval_end:
20824     #2

```

```
20825         \else:
20826             \if_meaning:w #2 0 2 \else: 0 \fi:
20827         \fi:
20828     {#3}
20829 }
20830 }
20831 \cs_new:Npn \__fp_sin_series_aux_o:NNnwww #1#2#3 #4\__fp_sep: #5,#6\__fp_sep:
20832 {
20833     \if_int_odd:w \__fp_int_eval:w #3 / 2 \__fp_int_eval_end:
20834         \exp_after:wN \use_i:nn
20835     \else:
20836         \exp_after:wN \use_ii:nn
20837     \fi:
20838     { % 1/18!
20839         \__fp_fixed_mul_sub_back:wwwn {0000}{0000}{0000}{0001}{5619}{2070}\__fp_sep:
20840             #4\__fp_sep:
20841             {0000}{0000}{0000}{0477}{9477}{3324}\__fp_sep:
20842         \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
20843             {0000}{0000}{0011}{4707}{4559}{7730}\__fp_sep:
20844         \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
20845             {0000}{0000}{2087}{6756}{9878}{6810}\__fp_sep:
20846         \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
20847             {0000}{0027}{5573}{1922}{3985}{8907}\__fp_sep:
20848         \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
20849             {0000}{2480}{1587}{3015}{8730}{1587}\__fp_sep:
20850         \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
20851             {0013}{8888}{8888}{8888}{8888}{8889}\__fp_sep:
20852         \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
20853             {0416}{6666}{6666}{6666}{6666}{6667}\__fp_sep:
20854         \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
20855             {5000}{0000}{0000}{0000}{0000}{0000}\__fp_sep:
20856         \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
20857             {10000}{0000}{0000}{0000}{0000}{0000}\__fp_sep:
20858         { \__fp_fixed_continue:wn 0, }
20859     }
20860     { % 1/17!
20861         \__fp_fixed_mul_sub_back:wwwn {0000}{0000}{0000}{0028}{1145}{7254}\__fp_sep:
20862             #4\__fp_sep:
20863             {0000}{0000}{0000}{7647}{1637}{3182}\__fp_sep:
20864         \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
20865             {0000}{0000}{0160}{5904}{3836}{8216}\__fp_sep:
20866         \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
20867             {0000}{0002}{5052}{1083}{8544}{1719}\__fp_sep:
20868         \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
20869             {0000}{0275}{5731}{9223}{9858}{9065}\__fp_sep:
20870         \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
```


20871 {0001}{9841}{2698}{4126}{9841}{2698}__fp_sep: 20871

20872 __fp_fixed_mul_sub_back:wwn #4__fp_sep: 20872

20873 {0083}{3333}{3333}{3333}{3333}{3333}__fp_sep: 20873

20874 __fp_fixed_mul_sub_back:wwn #4__fp_sep: 20874

20875 {1666}{6666}{6666}{6666}{6666}{6667}__fp_sep: 20875

20876 __fp_fixed_mul_sub_back:wwn#4__fp_sep: 20876

20877 {10000}{0000}{0000}{0000}{0000}{0000}__fp_sep: 20877

20878 { __fp_ep_mul:wwwn 0, } #5,#6__fp_sep: 20878

20879 } 20879

20880 { 20880

20881 \exp_after:wN __fp_sanitizew 20881

20882 \exp_after:wN #2 20882

20883 \int_value:w __fp_int_eval:w #1 20883

20884 } 20884

20885 #2 20885

20886 } 20886

20887 \cs_new:Npn __fp_tan_series_o:NNwww #1#2#3. #4__fp_sep: 20887

20888 { 20888

20889 __fp_fixed_mul:wwn #4__fp_sep: #4__fp_sep: 20889

20890 { 20890

20891 \exp_after:wN __fp_tan_series_aux_o:Nnwww 20891

20892 \int_value:w 20892

20893 \if_int_odd:w __fp_int_eval:w #3 / 2 __fp_int_eval_end: 20893

20894 \exp_after:wN \reverse_if:N 20894

20895 \fi: 20895

20896 \if_meaning:w #1#2 2 \else: 0 \fi: 20896

20897 {#3} 20897

20898 } 20898

20899 } 20899

20900 \cs_new:Npn __fp_tan_series_aux_o:Nnwww #1 #2 #3__fp_sep: #4,#5__fp_sep: 20900

20901 { 20901

20902 __fp_fixed_mul_sub_back:wwn {0000}{0000}{1527}{3493}{0856}{7059}__fp_sep: 20902

20903 #3__fp_sep: 20903

20904 {0000}{0159}{6080}{0274}{5257}{6472}__fp_sep: 20904

20905 __fp_fixed_mul_sub_back:wwn #3__fp_sep: 20905

20906 {0002}{4571}{2320}{0157}{2558}{8481}__fp_sep: 20906

20907 __fp_fixed_mul_sub_back:wwn #3__fp_sep: 20907

20908 {0115}{5830}{7533}{5397}{3168}{2147}__fp_sep: 20908

20909 __fp_fixed_mul_sub_back:wwn #3__fp_sep: 20909

20910 {1929}{8245}{6140}{3508}{7719}{2982}__fp_sep: 20910

20911 __fp_fixed_mul_sub_back:wwn #3__fp_sep: 20911

20912 {10000}{0000}{0000}{0000}{0000}{0000}__fp_sep: 20912

20913 { __fp_ep_mul:wwwn 0, } #4,#5__fp_sep: 20913

20914 { 20914

20915 __fp_fixed_mul_sub_back:wwn {0000}{0007}{0258}{0681}{9408}{4706}__fp_sep: 20915

20916 #3__fp_sep: 20916

```
20917             {0000}{2343}{7175}{1399}{6151}{7670}\__fp_sep: 20917
20918 \__fp_fixed_mul_sub_back:wwwn #3\__fp_sep: 20918
20919             {0019}{2638}{4588}{9232}{8861}{3691}\__fp_sep: 20919
20920 \__fp_fixed_mul_sub_back:wwwn #3\__fp_sep: 20920
20921             {0536}{6357}{0691}{4344}{6852}{4252}\__fp_sep: 20921
20922 \__fp_fixed_mul_sub_back:wwwn #3\__fp_sep: 20922
20923             {5263}{1578}{9473}{6842}{1052}{6315}\__fp_sep: 20923
20924 \__fp_fixed_mul_sub_back:wwwn#3\__fp_sep: 20924
20925             {10000}{0000}{0000}{0000}{0000}{0000}\__fp_sep: 20925
20926 { 20926
20927     \reverse_if:N \if_int_odd:w 20927
20928         \__fp_int_eval:w (#2 - 1) / 2 \__fp_int_eval_end: 20928
20929     \exp_after:wN \__fp_reverse_args:Nww 20929
20930     \fi: 20930
20931     \__fp_ep_div:wwwwn 0, 20931
20932 } 20932
20933 } 20933
20934 { 20934
20935     \exp_after:wN \__fp_sanitize:Nw 20935
20936     \exp_after:wN #1 20936
20937     \int_value:w \__fp_int_eval:w \__fp_ep_to_float_o:wwN 20937
20938 } 20938
20939 #1 20939
20940 } 20940
20941 \cs_new:Npn \__fp_atan_o:Nw #1 20941
20942 { 20942
20943     \__fp_parse_function_one_two:nnw 20943
20944     { #1 { atan } { atand } } 20944
20945     { \__fp_atan_default:w \__fp_atanii_o:Nww #1 } 20945
20946 } 20946
20947 \cs_new:Npn \__fp_acot_o:Nw #1 20947
20948 { 20948
20949     \__fp_parse_function_one_two:nnw 20949
20950     { #1 { acot } { acotd } } 20950
20951     { \__fp_atan_default:w \__fp_acotii_o:Nww #1 } 20951
20952 } 20952
20953 \cs_new:Npe \__fp_atan_default:w #1#2#3 @ { #1 #2 #3 \c_one_fp @ } 20953
20954 \cs_new:Npn \__fp_atanii_o:Nww 20954
20955     #1 \s__fp \__fp_chk:w #2#3#4\__fp_sep: \s__fp \__fp_chk:w #5 #6 @ 20955
20956 { 20956
20957     \if_meaning:w 3 #2 \__fp_case_return_i_o:ww \fi: 20957
20958     \if_meaning:w 3 #5 \__fp_case_return_ii_o:ww \fi: 20958
20959     \if_case:w 20959
20960         \if_meaning:w #2 #5 20960
20961             \if_meaning:w 1 #2 10 \else: 0 \fi: 20961
20962     \else: 20962
```

```
20963 \if_int_compare:w #2 > #5 \exp_stop_f: 1 \else: 2 \fi: 20963
20964 \fi: 20964
20965 \exp_stop_f: 20965
20966 \__fp_case_return:nw { \__fp_atan_inf_o:NNNw #1 #3 2 } 20966
20967 \or: \__fp_case_return:nw { \__fp_atan_inf_o:NNNw #1 #3 4 } 20967
20968 \or: \__fp_case_return:nw { \__fp_atan_inf_o:NNNw #1 #3 0 } 20968
20969 \fi: 20969
20970 \__fp_atan_normal_o:NNnwNnw #1 20970
20971 \s__fp \__fp_chk:w #2#3#4\__fp_sep: 20971
20972 \s__fp \__fp_chk:w #5 #6 20972
20973 } 20973
20974 \cs_new:Npn \__fp_acotii_o:Nww #1#2\__fp_sep: #3\__fp_sep: 20974
20975 { \__fp_atanii_o:Nww #1#3\__fp_sep: #2\__fp_sep: } 20975
20976 \cs_new:Npn \__fp_atan_inf_o:NNNw #1#2#3 \s__fp \__fp_chk:w #4#5#6\__fp_sep: 20976
20977 { 20977
20978 \exp_after:wN \__fp_atan_combine_o:NwwwwwN 20978
20979 \exp_after:wN #2 20979
20980 \int_value:w \__fp_int_eval:w 20980
20981 \if_meaning:w 2 #5 7 - \fi: #3 \exp_after:wN \__fp_sep: 20981
20982 \c__fp_one_fixed_tl 20982
20983 {0000}{0000}{0000}{0000}{0000}{0000}\__fp_sep: 20983
20984 0,{0000}{0000}{0000}{0000}{0000}{0000}\__fp_sep: #1 20984
20985 } 20985
20986 \cs_new_protected:Npn \__fp_atan_normal_o:NNnwNnw 20986
20987 #1 \s__fp \__fp_chk:w 1#2#3#4\__fp_sep: \s__fp \__fp_chk:w 1#5#6#7\__fp_sep: 20987
20988 { 20988
20989 \__fp_atan_test_o:NwwNwwN 20989
20990 #2 #3, #4{0000}{0000}\__fp_sep: 20990
20991 #5 #6, #7{0000}{0000}\__fp_sep: #1 20991
20992 } 20992
20993 \cs_new:Npn \__fp_atan_test_o:NwwNwwN #1#2,#3\__fp_sep: #4#5,#6\__fp_sep: 20993
20994 { 20994
20995 \exp_after:wN \__fp_atan_combine_o:NwwwwwN 20995
20996 \exp_after:wN #1 20996
20997 \int_value:w \__fp_int_eval:w 20997
20998 \if_meaning:w 2 #4 20998
20999 7 - \__fp_int_eval:w 20999
21000 \fi: 21000
21001 \if_int_compare:w 21001
21002 \__fp_ep_compare:www #2,#3\__fp_sep: #5,#6\__fp_sep: > \c_zero_int 21002
21003 3 - 21003
21004 \exp_after:wN \__fp_reverse_args:Nww 21004
21005 \fi: 21005
21006 \__fp_atan_div:wnwnw #2,#3\__fp_sep: #5,#6\__fp_sep: 21006
21007 } 21007
21008 \cs_new:Npn \__fp_atan_div:wnwnw #1,#2#3\__fp_sep: #4,#5#6\__fp_sep: 21008
```

21009	{	21009
21010	\if_int_compare:w	21010
21011	__fp_int_eval:w 41421 * #5 < #2 000	21011
21012	\if_case:w __fp_int_eval:w #4 - #1 __fp_int_eval_end:	21012
21013	00 \or: 0 \fi:	21013
21014	\exp_stop_f:	21014
21015	\exp_after:wN __fp_atan_near:wwn	21015
21016	\fi:	21016
21017	0	21017
21018	__fp_ep_div:wwwn #1,{#2}#3__fp_sep: #4,{#5}#6__fp_sep:	21018
21019	__fp_atan_auxi:ww	21019
21020	}	21020
21021	\cs_new:Npn __fp_atan_near:wwn	21021
21022	0 __fp_ep_div:wwwn #1,#2__fp_sep: #3,	21022
21023	{	21023
21024	1	21024
21025	__fp_ep_to_fixed:wn #1 - #3, #2__fp_sep:	21025
21026	__fp_atan_near_aux:wn	21026
21027	}	21027
21028	\cs_new:Npn __fp_atan_near_aux:wn #1__fp_sep: #2__fp_sep:	21028
21029	{	21029
21030	__fp_fixed_add:wn #1__fp_sep: #2__fp_sep:	21030
21031	{ __fp_fixed_sub:wn #2__fp_sep: #1__fp_sep: { __fp_ep_div:wwwn 0, } 0, }	21031
21032	}	21032
21033	\cs_new:Npn __fp_atan_auxi:ww #1,#2__fp_sep:	21033
21034	{ __fp_ep_to_fixed:wn #1,#2__fp_sep: __fp_atan_auxii:w #1,#2__fp_sep: }	21034
21035	\cs_new:Npn __fp_atan_auxii:w #1__fp_sep:	21035
21036	{	21036
21037	__fp_fixed_mul:wn #1__fp_sep: #1__fp_sep:	21037
21038	{	21038
21039	__fp_atan_Taylor_loop:ww 39 __fp_sep:	21039
21040	{0000}{0000}{0000}{0000}{0000}{0000} __fp_sep:	21040
21041	}	21041
21042	! #1__fp_sep:	21042
21043	}	21043
21044	\cs_new:Npn __fp_atan_Taylor_loop:ww #1__fp_sep: #2__fp_sep: #3__fp_sep:	21044
21045	{	21045
21046	\if_int_compare:w #1 = - \c_one_int	21046
21047	__fp_atan_Taylor_break:w	21047
21048	\fi:	21048
21049	\exp_after:wN __fp_fixed_div_int:wwN \c__fp_one_fixed_tl #1__fp_sep:	21049
21050	__fp_rrot:ww __fp_fixed_mul_sub_back:wwn #2__fp_sep: #3__fp_sep:	21050
21051	{	21051
21052	\exp_after:wN __fp_atan_Taylor_loop:ww	21052
21053	\int_value:w __fp_int_eval:w #1 - 2 __fp_sep:	21053
21054	}	21054

```
21055 #3\__fp_sep: 21055
21056 } 21056
21057 \cs_new:Npn \__fp_atan_Taylor_break:w 21057
21058 \fi: #1 \__fp_fixed_mul_sub_back:wwn #2\__fp_sep: #3 ! 21058
21059 { \fi: \__fp_sep: #2 \__fp_sep: } 21059
21060 \cs_new:Npn \__fp_atan_combine_o:NwwwwN 21060
21061 #1 #2\__fp_sep: #3\__fp_sep: #4\__fp_sep: #5,#6\__fp_sep: #7 21061
21062 { 21062
21063 \exp_after:wN \__fp_sanitize:Nw 21063
21064 \exp_after:wN #1 21064
21065 \int_value:w \__fp_int_eval:w 21065
21066 \if_meaning:w 0 #2 21066
21067 \exp_after:wN \use_i:nn 21067
21068 \else: 21068
21069 \exp_after:wN \use_ii:nn 21069
21070 \fi: 21070
21071 { #5 \__fp_fixed_mul:wwn #3\__fp_sep: #6\__fp_sep: } 21071
21072 { 21072
21073 \__fp_fixed_mul:wwn #3\__fp_sep: #4\__fp_sep: 21073
21074 { 21074
21075 \exp_after:wN \__fp_atan_combine_aux:ww 21075
21076 \int_value:w \__fp_int_eval:w #2 / 2 \__fp_sep: #2\__fp_sep: 21076
21077 } 21077
21078 } 21078
21079 { #7 \__fp_fixed_to_float_o:wN \__fp_fixed_to_float_rad_o:wN } 21079
21080 #1 21080
21081 } 21081
21082 \cs_new:Npn \__fp_atan_combine_aux:ww #1\__fp_sep: #2\__fp_sep: 21082
21083 { 21083
21084 \__fp_fixed_mul_short:wwn 21084
21085 {7853}{9816}{3397}{4483}{0961}{5661}\__fp_sep: 21085
21086 {#1}{0000}{0000}\__fp_sep: 21086
21087 { 21087
21088 \if_int_odd:w #2 \exp_stop_f: 21088
21089 \exp_after:wN \__fp_fixed_sub:wwn 21089
21090 \else: 21090
21091 \exp_after:wN \__fp_fixed_add:wwn 21091
21092 \fi: 21092
21093 } 21093
21094 } 21094
21095 \cs_new:Npn \__fp_asin_o:w #1 \s__fp \__fp_chk:w #2#3\__fp_sep: @ 21095
21096 { 21096
21097 \if_case:w #2 \exp_stop_f: 21097
21098 \__fp_case_return_same_o:w 21098
21099 \or: 21099
21100 \__fp_case_use:nw 21100
```

21101	{ __fp_asin_normal_o:NfwNnnnnw #1 { #1 { asin } { asind } } }	21101
21102	\or:	21102
21103	__fp_case_use:nw	21103
21104	{ __fp_invalid_operation_o:fw { #1 { asin } { asind } } }	21104
21105	\else:	21105
21106	__fp_case_return_same_o:w	21106
21107	\fi:	21107
21108	\s__fp __fp_chk:w #2 #3__fp_sep:	21108
21109	}	21109
21110	\cs_new:Npn __fp_acos_o:w #1 \s__fp __fp_chk:w #2#3__fp_sep: @	21110
21111	{	21111
21112	\if_case:w #2 \exp_stop_f:	21112
21113	__fp_case_use:nw { __fp_atan_inf_o:NNWw #1 0 4 }	21113
21114	\or:	21114
21115	__fp_case_use:nw	21115
21116	{	21116
21117	__fp_asin_normal_o:NfwNnnnnw #1 { #1 { acos } { acosd } }	21117
21118	__fp_reverse_args:Nww	21118
21119	}	21119
21120	\or:	21120
21121	__fp_case_use:nw	21121
21122	{ __fp_invalid_operation_o:fw { #1 { acos } { acosd } } }	21122
21123	\else:	21123
21124	__fp_case_return_same_o:w	21124
21125	\fi:	21125
21126	\s__fp __fp_chk:w #2 #3__fp_sep:	21126
21127	}	21127
21128	\cs_new:Npn __fp_asin_normal_o:NfwNnnnnw	21128
21129	#1#2#3 \s__fp __fp_chk:w 1#4#5#6#7#8#9__fp_sep:	21129
21130	{	21130
21131	\if_int_compare:w #5 < \c_one_int	21131
21132	\exp_after:wN __fp_use_none_until_s:w	21132
21133	\fi:	21133
21134	\if_int_compare:w __fp_int_eval:w #5 + #6#7 + #8#9 = 1000 0001 ~	21134
21135	\exp_after:wN __fp_use_none_until_s:w	21135
21136	\fi:	21136
21137	__fp_use_i:ww	21137
21138	__fp_invalid_operation_o:fw {#2}	21138
21139	\s__fp __fp_chk:w 1#4#{#5}{#6}{#7}{#8}{#9}__fp_sep:	21139
21140	__fp_asin_auxi_o:NnNww	21140
21141	#1 {#3} #4 #5,{#6}{#7}{#8}{#9}{0000}{0000}__fp_sep:	21141
21142	}	21142
21143	\cs_new:Npn __fp_asin_auxi_o:NnNww #1#2#3#4,#5__fp_sep:	21143
21144	{	21144
21145	__fp_ep_to_fixed:wwn #4,#5__fp_sep:	21145
21146	__fp_asin_isqrt:wn	21146

21147	__fp_ep_mul:wwwn #4,#5__fp_sep:	21147
21148	__fp_ep_to_ep:wwN	21148
21149	__fp_fixed_continue:wn	21149
21150	{ #2 __fp_atan_test_o:NwwNwwN #3 }	21150
21151	0 1,{1000}{0000}{0000}{0000}{0000}{0000}__fp_sep: #1	21151
21152	}	21152
21153	\cs_new:Npn __fp_asin_isqrt:wn #1__fp_sep:	21153
21154	{	21154
21155	\exp_after:wN __fp_fixed_sub:wwn \c__fp_one_fixed_tl #1__fp_sep:	21155
21156	{	21156
21157	__fp_fixed_add_one:wN #1__fp_sep:	21157
21158	__fp_fixed_continue:wn { __fp_ep_mul:wwwn 0, } 0,	21158
21159	}	21159
21160	__fp_ep_isqrt:wwn	21160
21161	}	21161
21162	\cs_new:Npn __fp_acsc_o:w #1 \s__fp __fp_chk:w #2#3#4__fp_sep: @	21162
21163	{	21163
21164	\if_case:w \if_meaning:w 2 #2 #3 \fi: #2 \exp_stop_f:	21164
21165	__fp_case_use:nw	21165
21166	{ __fp_invalid_operation_o:fw { #1 { acsc } { acscd } } }	21166
21167	\or: __fp_case_use:nw	21167
21168	{ __fp_acsc_normal_o:NfwNnw #1 { #1 { acsc } { acscd } } }	21168
21169	\or: __fp_case_return_o:Nw \c_zero_fp	21169
21170	\or: __fp_case_return_same_o:w	21170
21171	\else: __fp_case_return_o:Nw \c_minus_zero_fp	21171
21172	\fi:	21172
21173	\s__fp __fp_chk:w #2 #3 #4__fp_sep:	21173
21174	}	21174
21175	\cs_new:Npn __fp_asec_o:w #1 \s__fp __fp_chk:w #2#3__fp_sep: @	21175
21176	{	21176
21177	\if_case:w #2 \exp_stop_f:	21177
21178	__fp_case_use:nw	21178
21179	{ __fp_invalid_operation_o:fw { #1 { asec } { asecd } } }	21179
21180	\or:	21180
21181	__fp_case_use:nw	21181
21182	{	21182
21183	__fp_acsc_normal_o:NfwNnw #1 { #1 { asec } { asecd } }	21183
21184	__fp_reverse_args:Nww	21184
21185	}	21185
21186	\or: __fp_case_use:nw { __fp_atan_inf_o:NNnw #1 0 4 }	21186
21187	\else: __fp_case_return_same_o:w	21187
21188	\fi:	21188
21189	\s__fp __fp_chk:w #2 #3__fp_sep:	21189
21190	}	21190
21191	\cs_new:Npn __fp_acsc_normal_o:NfwNnw #1#2#3 \s__fp __fp_chk:w 1#4#5#6__fp_sep:	21191
21192	{	21192

21193	\int_compare:nNnTF {#5} < 1	21193
21194	{	21194
21195	_fp_invalid_operation_o:fw {#2}	21195
21196	\s__fp _fp_chk:w 1#4{#5}#6_fp_sep:	21196
21197	}	21197
21198	{	21198
21199	_fp_ep_div:wwwn	21199
21200	1,{1000}{0000}{0000}{0000}{0000}{0000}_fp_sep:	21200
21201	#5,#6{0000}{0000}_fp_sep:	21201
21202	{ _fp_asin_auxi_o:NnNww #1 {#3} #4 }	21202
21203	}	21203
21204	}	21204
21205	%% File: l3fp-convert.dtx	21205
21206	\cs_new:Npn _fp_tuple_convert:Nw #1 \s__fp_tuple _fp_tuple_chk:w #2 _fp_sep:	21206
21207	{	21207
21208	\int_case:nnF { _fp_array_count:n {#2} }	21208
21209	{	21209
21210	{ 0 } { () }	21210
21211	{ 1 } { _fp_tuple_convert_end:w @ { #1 #2 , } }	21211
21212	}	21212
21213	{	21213
21214	_fp_tuple_convert_loop:nNw { } #1	21214
21215	#2 { ? _fp_tuple_convert_end:w } _fp_sep:	21215
21216	@ { \use_none:nn }	21216
21217	}	21217
21218	}	21218
21219	\cs_new:Npn _fp_tuple_convert_loop:nNw #1#2#3#4_fp_sep: #5 @ #6	21219
21220	{	21220
21221	\use_none:n #3	21221
21222	\exp_args:Nf _fp_tuple_convert_loop:nNw { #2 #3#4 _fp_sep: } #2 #5	21222
21223	@ { #6 , ~ #1 }	21223
21224	}	21224
21225	\cs_new:Npn _fp_tuple_convert_end:w #1 @ #2	21225
21226	{ \exp_after:wN (\exp:w \exp_end_continue_f:w #2) }	21226
21227	\cs_new:Npn _fp_trim_zeros:w #1 _fp_sep:	21227
21228	{	21228
21229	_fp_trim_zeros_loop:w #1 _fp_sep:	21229
21230	_fp_trim_zeros_loop:w 0_fp_sep:	21230
21231	_fp_trim_zeros_dot:w ._fp_sep:	21231
21232	\s__fp_stop	21232
21233	}	21233
21234	\cs_new:Npn _fp_trim_zeros_loop:w #1 0_fp_sep: #2 { #2 #1 _fp_sep: #2 }	21234
21235	\cs_new:Npn _fp_trim_zeros_dot:w #1 ._fp_sep:	21235
21236	{ _fp_trim_zeros_end:w #1 _fp_sep: }	21236
21237	\cs_new:Npn _fp_trim_zeros_end:w #1 _fp_sep: #2 \s__fp_stop { #1 }	21237
21238	\cs_new:Npn \fp_to_scientific:N #1	21238

```
21239 { \exp_after:wN \__fp_to_scientific_dispatch:w #1 } 21239
21240 \cs_generate_variant:Nn \fp_to_scientific:N { c } 21240
21241 \cs_new:Npn \fp_to_scientific:n 21241
21242 { 21242
21243     \exp_after:wN \__fp_to_scientific_dispatch:w 21243
21244     \exp:w \exp_end_continue_f:w \__fp_parse:n 21244
21245 } 21245
21246 \cs_new:Npn \__fp_to_scientific_dispatch:w #1 21246
21247 { 21247
21248     \__fp_change_func_type:NNN 21248
21249     #1 \__fp_to_scientific:w \__fp_to_scientific_recover:w 21249
21250     #1 21250
21251 } 21251
21252 \cs_new:Npn \__fp_to_scientific_recover:w #1 #2 \__fp_sep: 21252
21253 { 21253
21254     \__fp_error:nffn { unknown-type } { \tl_to_str:n { #2 \__fp_sep: } } { } { } 21254
21255     nan 21255
21256 } 21256
21257 \cs_new:Npn \__fp_tuple_to_scientific:w 21257
21258 { \__fp_tuple_convert:Nw \__fp_to_scientific_dispatch:w } 21258
21259 \cs_new:Npn \__fp_to_scientific:w \s__fp \__fp_chk:w #1#2 21259
21260 { 21260
21261     \if_meaning:w 2 #2 \exp_after:wN - \exp:w \exp_end_continue_f:w \fi: 21261
21262     \if_case:w #1 \exp_stop_f: 21262
21263         \__fp_case_return:nw { 0.0000000000000000e0 } 21263
21264     \or: \exp_after:wN \__fp_to_scientific_normal:wnnnnn 21264
21265     \or: 21265
21266         \__fp_case_use:nw 21266
21267         { 21267
21268             \__fp_invalid_operation:nnw 21268
21269             { \fp_to_scientific:N \c__fp_overflowing_fp } 21269
21270             { fp_to_scientific } 21270
21271         } 21271
21272     \or: 21272
21273         \__fp_case_use:nw 21273
21274         { 21274
21275             \__fp_invalid_operation:nnw 21275
21276             { \fp_to_scientific:N \c_zero_fp } 21276
21277             { fp_to_scientific } 21277
21278         } 21278
21279     \fi: 21279
21280     \s__fp \__fp_chk:w #1 #2 21280
21281 } 21281
21282 \cs_new:Npn \__fp_to_scientific_normal:wnnnnn 21282
21283 \s__fp \__fp_chk:w 1 #1 #2 #3#4#5#6 \__fp_sep: 21283
21284 { 21284
```

```
21285 \exp_after:wN \__fp_to_scientific_normal:wNw 21285
21286 \exp_after:wN e 21286
21287 \int_value:w \__fp_int_eval:w #2 - 1 21287
21288 \__fp_sep: #3 #4 #5 #6 \__fp_sep: 21288
21289 } 21289
21290 \cs_new:Npn \__fp_to_scientific_normal:wNw #1 \__fp_sep: #2#3\__fp_sep: 21290
21291 { #2.#3 #1 } 21291
21292 \cs_new:Npn \fp_to_decimal:N #1 21292
21293 { \exp_after:wN \__fp_to_decimal_dispatch:w #1 } 21293
21294 \cs_generate_variant:Nn \fp_to_decimal:N { c } 21294
21295 \cs_new:Npn \fp_to_decimal:n 21295
21296 { 21296
21297 \exp_after:wN \__fp_to_decimal_dispatch:w 21297
21298 \exp:w \exp_end_continue_f:w \__fp_parse:n 21298
21299 } 21299
21300 \cs_new:Npn \__fp_to_decimal_dispatch:w #1 21300
21301 { 21301
21302 \__fp_change_func_type:NNN 21302
21303 #1 \__fp_to_decimal:w \__fp_to_decimal_recover:w 21303
21304 #1 21304
21305 } 21305
21306 \cs_new:Npn \__fp_to_decimal_recover:w #1 #2 \__fp_sep: 21306
21307 { 21307
21308 \__fp_error:nffn { unknown-type } { \tl_to_str:n { #2 \__fp_sep: } } { } { } 21308
21309 nan 21309
21310 } 21310
21311 \cs_new:Npn \__fp_tuple_to_decimal:w 21311
21312 { \__fp_tuple_convert:Nw \__fp_to_decimal_dispatch:w } 21312
21313 \cs_new:Npn \__fp_to_decimal:w \s__fp \__fp_chk:w #1#2 21313
21314 { 21314
21315 \if_meaning:w 2 #2 \exp_after:wN - \exp:w \exp_end_continue_f:w \fi: 21315
21316 \if_case:w #1 \exp_stop_f: 21316
21317 \__fp_case_return:nw { 0 } 21317
21318 \or: \exp_after:wN \__fp_to_decimal_normal:wnnnnn 21318
21319 \or: 21319
21320 \__fp_case_use:nw 21320
21321 { 21321
21322 \__fp_invalid_operation:nnw 21322
21323 { \fp_to_decimal:N \c__fp_overflowing_fp } 21323
21324 { fp_to_decimal } 21324
21325 } 21325
21326 \or: 21326
21327 \__fp_case_use:nw 21327
21328 { 21328
21329 \__fp_invalid_operation:nnw 21329
21330 { 0 } 21330
```

```

21331         { fp_to_decimal }
21332     }
21333     \fi:
21334     \s__fp \__fp_chk:w #1 #2
21335 }
21336 \cs_new:Npn \__fp_to_decimal_normal:wnnnnn
21337     \s__fp \__fp_chk:w 1 #1 #2 #3#4#5#6 \__fp_sep:
21338 {
21339     \int_compare:nNnTF {#2} > 0
21340     {
21341         \int_compare:nNnTF {#2} < \c__fp_prec_int
21342         {
21343             \__fp_decimate:nNnnnn { \c__fp_prec_int - #2 }
21344             \__fp_to_decimal_large:Nnnw
21345         }
21346         {
21347             \exp_after:wN \exp_after:wN
21348             \exp_after:wN \__fp_to_decimal_huge:wnnnn
21349             \prg_replicate:nn { #2 - \c__fp_prec_int } { 0 } \__fp_sep:
21350         }
21351         {#3} {#4} {#5} {#6}
21352     }
21353     {
21354         \exp_after:wN \__fp_trim_zeros:w
21355         \exp_after:wN 0
21356         \exp_after:wN .
21357         \exp:w \exp_end_continue_f:w \prg_replicate:nn { - #2 } { 0 }
21358         #3#4#5#6 \__fp_sep:
21359     }
21360 }
21361 \cs_new:Npn \__fp_to_decimal_large:Nnnw #1#2#3#4\__fp_sep:
21362 {
21363     \exp_after:wN \__fp_trim_zeros:w \int_value:w
21364     \if_int_compare:w #2 > \c_zero_int
21365         #2
21366     \fi:
21367     \exp_stop_f:
21368     #3.#4 \__fp_sep:
21369 }
21370 \cs_new:Npn \__fp_to_decimal_huge:wnnnn #1\__fp_sep: #2#3#4#5 { #2#3#4#5 #1 }
21371 \cs_new:Npn \fp_to_tl:N #1 { \exp_after:wN \__fp_to_tl_dispatch:w #1 }
21372 \cs_generate_variant:Nn \fp_to_tl:N { c }
21373 \cs_new:Npn \fp_to_tl:n
21374 {
21375     \exp_after:wN \__fp_to_tl_dispatch:w
21376     \exp:w \exp_end_continue_f:w \__fp_parse:n

```

```
21377 } 21377
21378 \cs_new:Npn \__fp_to_tl_dispatch:w #1 21378
21379 { \__fp_change_func_type:NNN #1 \__fp_to_tl:w \__fp_to_tl_recover:w #1 } 21379
21380 \cs_new:Npn \__fp_to_tl_recover:w #1 #2 \__fp_sep: 21380
21381 { 21381
21382 \__fp_error:nffn { unknown-type } { \tl_to_str:n { #2 \__fp_sep: } } { } { } 21382
21383 nan 21383
21384 } 21384
21385 \cs_new:Npn \__fp_tuple_to_tl:w 21385
21386 { \__fp_tuple_convert:Nw \__fp_to_tl_dispatch:w } 21386
21387 \cs_new:Npn \__fp_to_tl:w \s__fp \__fp_chk:w #1#2 21387
21388 { 21388
21389 \if_meaning:w 2 #2 \exp_after:wN - \exp:w \exp_end_continue_f:w \fi: 21389
21390 \if_case:w #1 \exp_stop_f: 21390
21391 \__fp_case_return:nw { 0 } 21391
21392 \or: \exp_after:wN \__fp_to_tl_normal:nnnnn 21392
21393 \or: \__fp_case_return:nw { inf } 21393
21394 \else: \__fp_case_return:nw { nan } 21394
21395 \fi: 21395
21396 } 21396
21397 \cs_new:Npn \__fp_to_tl_normal:nnnnn #1 21397
21398 { 21398
21399 \int_compare:nTF 21399
21400 { -2 <= #1 <= \c__fp_prec_int } 21400
21401 { \__fp_to_decimal_normal:wnnnnn } 21401
21402 { \__fp_to_tl_scientific:wnnnnn } 21402
21403 \s__fp \__fp_chk:w 1 0 {#1} 21403
21404 } 21404
21405 \cs_new:Npn \__fp_to_tl_scientific:wnnnnn 21405
21406 \s__fp \__fp_chk:w 1 #1 #2 #3#4#5#6 \__fp_sep: 21406
21407 { 21407
21408 \exp_after:wN \__fp_to_tl_scientific:wNw 21408
21409 \exp_after:wN e 21409
21410 \int_value:w \__fp_int_eval:w #2 - 1 21410
21411 \__fp_sep: #3 #4 #5 #6 \__fp_sep: 21411
21412 } 21412
21413 \cs_new:Npn \__fp_to_tl_scientific:wNw #1 \__fp_sep: #2#3\__fp_sep: 21413
21414 { \__fp_trim_zeros:w #2.#3 \__fp_sep: #1 } 21414
21415 \cs_new:Npn \fp_to_dim:N #1 21415
21416 { \exp_after:wN \__fp_to_dim_dispatch:w #1 } 21416
21417 \cs_generate_variant:Nn \fp_to_dim:N { c } 21417
21418 \cs_new:Npn \fp_to_dim:n 21418
21419 { 21419
21420 \exp_after:wN \__fp_to_dim_dispatch:w 21420
21421 \exp:w \exp_end_continue_f:w \__fp_parse:n 21421
21422 } 21422
```

```

21423 \cs_new:Npn \__fp_to_dim_dispatch:w #1#2 \__fp_sep: 21423
21424 { 21424
21425     \__fp_change_func_type:NNN #1 \__fp_to_dim:w \__fp_to_dim_recover:w 21425
21426     #1 #2 \__fp_sep: 21426
21427 } 21427
21428 \cs_new:Npn \__fp_to_dim_recover:w #1 21428
21429 { \__fp_invalid_operation:nnw { Opt } { fp_to_dim } } 21429
21430 \cs_new:Npn \__fp_to_dim:w #1 \__fp_sep: { \__fp_to_decimal:w #1 \__fp_sep: pt } 21430
21431 \cs_new:Npn \fp_to_int:N #1 { \exp_after:wN \__fp_to_int_dispatch:w #1 } 21431
21432 \cs_generate_variant:Nn \fp_to_int:N { c } 21432
21433 \cs_new:Npn \fp_to_int:n 21433
21434 { 21434
21435     \exp_after:wN \__fp_to_int_dispatch:w 21435
21436     \exp:w \exp_end_continue_f:w \__fp_parse:n 21436
21437 } 21437
21438 \cs_new:Npn \__fp_to_int_dispatch:w #1#2 \__fp_sep: 21438
21439 { 21439
21440     \__fp_change_func_type:NNN #1 \__fp_to_int:w \__fp_to_int_recover:w 21440
21441     #1 #2 \__fp_sep: 21441
21442 } 21442
21443 \cs_new:Npn \__fp_to_int_recover:w #1 21443
21444 { \__fp_invalid_operation:nnw { 0 } { fp_to_int } } 21444
21445 \cs_new:Npn \__fp_to_int:w #1\__fp_sep: 21445
21446 { 21446
21447     \exp_after:wN \__fp_to_decimal:w \exp:w \exp_end_continue_f:w 21447
21448     \__fp_round:Nwn \__fp_round_to_nearest:NNN #1\__fp_sep: { 0 } 21448
21449 } 21449
21450 \cs_new:Npn \dim_to_fp:n #1 21450
21451 { 21451
21452     \exp_after:wN \__fp_from_dim_test:ww 21452
21453     \exp_after:wN 0 21453
21454     \exp_after:wN , 21454
21455     \int_value:w \tex_glueexpr:D #1 \__fp_sep: 21455
21456 } 21456
21457 \cs_new:Npn \__fp_from_dim_test:ww #1, #2 21457
21458 { 21458
21459     \if_meaning:w 0 #2 21459
21460     \__fp_case_return:nw { \exp_after:wN \c_zero_fp } 21460
21461     \else: 21461
21462     \exp_after:wN \__fp_from_dim:wNw 21462
21463     \int_value:w \__fp_int_eval:w #1 - 4 21463
21464     \if_meaning:w - #2 21464
21465     \exp_after:wN , \exp_after:wN 2 \int_value:w 21465
21466     \else: 21466
21467     \exp_after:wN , \exp_after:wN 0 \int_value:w #2 21467
21468     \fi: 21468

```

```

21469 \fi:
21470 }
21471 \cs_new:Npn \__fp_from_dim:wNw #1,#2#3\__fp_sep:
21472 {
21473   \__fp_pack_twice_four:wNNNNNNNN \__fp_from_dim:wNNnnnnnn \__fp_sep:
21474   #3 000 0000 00 {10}987654321\__fp_sep: #2 {#1}
21475 }
21476 \cs_new:Npn \__fp_from_dim:wNNnnnnnn #1\__fp_sep: #2#3#4#5#6#7#8#9
21477 { \__fp_from_dim:wnnnnwNn #1 {#2#300} {0000} \__fp_sep: }
21478 \cs_new:Npn \__fp_from_dim:wnnnnwNn #1\__fp_sep: #2#3#4#5#6\__fp_sep: #7#8
21479 {
21480   \__fp_mul_npos_o:Nww #7
21481   \s__fp \__fp_chk:w 1 #7 {#5} #1 \__fp_sep:
21482   \s__fp \__fp_chk:w 1 0 {#8} {1525} {8789} {0625} {0000} \__fp_sep:
21483   \prg_do_nothing:
21484 }
21485 \cs_new_eq:NN \fp_use:N \fp_to_decimal:N
21486 \cs_generate_variant:Nn \fp_use:N { c }
21487 \cs_new_eq:NN \fp_eval:n \fp_to_decimal:n
21488 \cs_new:Npn \fp_sign:n #1
21489 { \fp_to_decimal:n { sign \__fp_parse:n {#1} } }
21490 \cs_new:Npn \fp_abs:n #1
21491 { \fp_to_decimal:n { abs \__fp_parse:n {#1} } }
21492 \cs_new:Npn \fp_max:nn #1#2
21493 { \fp_to_decimal:n { max ( \__fp_parse:n {#1} , \__fp_parse:n {#2} ) } }
21494 \cs_new:Npn \fp_min:nn #1#2
21495 { \fp_to_decimal:n { min ( \__fp_parse:n {#1} , \__fp_parse:n {#2} ) } }
21496 \cs_new:Npn \__fp_array_to_clist:n #1
21497 {
21498   \tl_if_empty:nF {#1}
21499   {
21500     \exp_last_unbraced:Ne \use_ii:nn
21501     {
21502       \__fp_array_to_clist_loop:Nw #1 { ? \prg_break: } \__fp_sep:
21503       \prg_break_point:
21504     }
21505   }
21506 }
21507 \cs_new:Npn \__fp_array_to_clist_loop:Nw #1#2\__fp_sep:
21508 {
21509   \use_none:n #1
21510   , ~
21511   \exp_not:f { \__fp_to_tl_dispatch:w #1 #2 \__fp_sep: }
21512   \__fp_array_to_clist_loop:Nw
21513 }
21514 %% File: l3fp-random.dtx

```



```

21515 \cs_new:Npn \__fp_parse_word_rand:N 21515
21516 { \__fp_parse_function:NNN \__fp_rand_o:Nw ? } 21516
21517 \cs_new:Npn \__fp_parse_word_randint:N 21517
21518 { \__fp_parse_function:NNN \__fp_randint_o:Nw ? } 21518
21519 \int_const:Nn \c__kernel_randint_max_int { 131071 } 21519
21520 \cs_new:Npn \__kernel_randint:n #1 21520
21521 { 21521
21522     (#1 * \tex_uniformdeviate:D 16384 21522
21523     + \tex_uniformdeviate:D #1 + 8192 ) / 16384 21523
21524 } 21524
21525 \cs_new:Npn \__fp_rand_myriads:n #1 21525
21526 { \__fp_rand_myriads_loop:w #1 \prg_break: X \prg_break_point: \__fp_sep: } 21526
21527 \cs_new:Npn \__fp_rand_myriads_loop:w #1 X 21527
21528 { 21528
21529     #1 21529
21530     \exp_after:wN \__fp_rand_myriads_get:w 21530
21531     \int_value:w \__fp_int_eval:w 9999 + 21531
21532     \__kernel_randint:n { 10000 } 21532
21533     \__fp_rand_myriads_loop:w 21533
21534 } 21534
21535 \cs_new:Npn \__fp_rand_myriads_get:w 1 #1 \__fp_sep: { \__fp_sep: {#1} } 21535
21536 \cs_new:Npn \__fp_rand_o:Nw ? #1 @ 21536
21537 { 21537
21538     \tl_if_empty:nTF {#1} 21538
21539     { 21539
21540         \exp_after:wN \__fp_rand_o:w 21540
21541         \exp:w \exp_end_continue_f:w 21541
21542         \__fp_rand_myriads:n { XXXX } { 0000 } { 0000 } \__fp_sep: 0 21542
21543     } 21543
21544     { 21544
21545         \__fp_error_num_args:ffff { rand } { 0 } { 0 } 21545
21546         { \__fp_array_count:n {#1} } 21546
21547         \exp_after:wN \c_nan_fp 21547
21548     } 21548
21549 } 21549
21550 \cs_new:Npn \__fp_rand_o:w \__fp_sep: 21550
21551 { 21551
21552     \exp_after:wN \__fp_sanitize:Nw 21552
21553     \exp_after:wN 0 21553
21554     \int_value:w \__fp_int_eval:w \c_zero_int 21554
21555     \__fp_fixed_to_float_o:wN 21555
21556 } 21556
21557 \cs_new:Npn \__fp_randint_o:Nw ? 21557
21558 { 21558
21559     \__fp_parse_function_one_two:nnw 21559
21560     { randint } 21560

```

21561	{ __fp_randint_default:w __fp_randint_o:w }	21561
21562	}	21562
21563	\cs_new:Npn __fp_randint_default:w #1 { \exp_after:wN #1 \c_one_fp }	21563
21564	\cs_new:Npn __fp_randint_badarg:w \s__fp __fp_chk:w #1#2#3__fp_sep:	21564
21565	{	21565
21566	__fp_int:wTF \s__fp __fp_chk:w #1#2#3__fp_sep:	21566
21567	{	21567
21568	\if_meaning:w 1 #1	21568
21569	\if_int_compare:w	21569
21570	__fp_use_i_until_s:nw #3 __fp_sep: > \c__fp_prec_int	21570
21571	\c_one_int	21571
21572	\fi:	21572
21573	\fi:	21573
21574	}	21574
21575	{ \c_one_int }	21575
21576	}	21576
21577	\cs_new:Npn __fp_randint_o:w #1__fp_sep: #2__fp_sep: @	21577
21578	{	21578
21579	\if_case:w	21579
21580	__fp_randint_badarg:w #1__fp_sep:	21580
21581	__fp_randint_badarg:w #2__fp_sep:	21581
21582	\if:w 1 __fp_compare_back:ww #2__fp_sep: #1__fp_sep: \c_one_int \fi:	21582
21583	\c_zero_int	21583
21584	__fp_randint_auxi_o:ww #1__fp_sep: #2__fp_sep:	21584
21585	\or:	21585
21586	__fp_invalid_operation_tl_o:ff	21586
21587	{ randint } { __fp_array_to_clist:n { #1__fp_sep: #2__fp_sep: } }	21587
21588	\exp:w	21588
21589	\fi:	21589
21590	\exp_after:wN \exp_end:	21590
21591	}	21591
21592	\cs_new:Npn __fp_randint_auxi_o:ww #1 __fp_sep: #2 __fp_sep: #3 \exp_end:	21592
21593	{	21593
21594	\fi:	21594
21595	__fp_randint_auxii:wn #2 __fp_sep:	21595
21596	{ __fp_randint_auxii:wn #1 __fp_sep: __fp_randint_auxiii_o:ww }	21596
21597	}	21597
21598	\cs_new:Npn __fp_randint_auxii:wn \s__fp __fp_chk:w #1#2#3#4 __fp_sep:	21598
21599	{	21599
21600	\if_meaning:w 0 #1	21600
21601	\exp_after:wN \use_i:nn	21601
21602	\else:	21602
21603	\exp_after:wN \use_ii:nn	21603
21604	\fi:	21604
21605	{ \exp_after:wN __fp_fixed_continue:wn \c__fp_one_fixed_tl }	21605
21606	{	21606

21607	\exp_after:wN __fp_ep_to_fixed:wwn	21607
21608	\int_value:w __fp_int_eval:w	21608
21609	#3 - \c__fp_prec_int , #4 {0000} {0000} __fp_sep:	21609
21610	{	21610
21611	\if_meaning:w 0 #2	21611
21612	\exp_after:wN \use_i:nnnn	21612
21613	\exp_after:wN __fp_fixed_add_one:wN	21613
21614	\fi:	21614
21615	\exp_after:wN __fp_fixed_sub:wwn \c__fp_one_fixed_tl	21615
21616	}	21616
21617	__fp_fixed_continue:wn	21617
21618	}	21618
21619	}	21619
21620	\cs_new:Npn __fp_randint_auxiii_o:ww #1 __fp_sep: #2 __fp_sep:	21620
21621	{	21621
21622	__fp_fixed_add:wwn #2 __fp_sep:	21622
21623	{0000} {0000} {0000} {0001} {0000} {0000} __fp_sep:	21623
21624	__fp_fixed_sub:wwn #1 __fp_sep:	21624
21625	{	21625
21626	\exp_after:wN \use_i:nn	21626
21627	\exp_after:wN __fp_fixed_mul_add:wwn	21627
21628	\exp:w \exp_end_continue_f:w __fp_rand_myriads:n { XXXXXX } __fp_sep:	21628
21629	}	21629
21630	#1 __fp_sep:	21630
21631	__fp_randint_auxiv_o:ww	21631
21632	#2 __fp_sep:	21632
21633	__fp_randint_auxv_o:w #1 __fp_sep: @	21633
21634	}	21634
21635	\cs_new:Npn __fp_randint_auxiv_o:ww #1#2#3#4#5 __fp_sep: #6#7#8#9	21635
21636	{	21636
21637	\if_int_compare:w	21637
21638	\if_int_compare:w #1#2 > #6#7 \exp_stop_f: 1 \else:	21638
21639	\if_int_compare:w #1#2 < #6#7 \exp_stop_f: - \fi: \fi:	21639
21640	#3#4 > #8#9 \exp_stop_f:	21640
21641	__fp_use_i_until_s:nw	21641
21642	\fi:	21642
21643	__fp_randint_auxv_o:w {#1}{#2}{#3}{#4}#5	21643
21644	}	21644
21645	\cs_new:Npn __fp_randint_auxv_o:w #1#2#3#4#5 __fp_sep: #6 @	21645
21646	{	21646
21647	\exp_after:wN __fp_sanitize:Nw	21647
21648	\int_value:w	21648
21649	\if_int_compare:w #1 < 10000 \exp_stop_f:	21649
21650	2	21650
21651	\else:	21651
21652	0	21652

```

21653     \exp_after:wN \exp_after:wN                                     21653
21654     \exp_after:wN \__fp_reverse_args:Nww                             21654
21655 \fi:                                                                    21655
21656 \exp_after:wN \__fp_fixed_sub:wwn \c__fp_one_fixed_tl               21656
21657 {#1} {#2} {#3} {#4} {0000} {0000} \__fp_sep:                         21657
21658 {                                                                        21658
21659     \exp_after:wN \exp_stop_f:                                         21659
21660     \int_value:w \__fp_int_eval:w \c__fp_prec_int                    21660
21661     \__fp_fixed_to_float_o:wN                                          21661
21662 }                                                                        21662
21663 0                                                                        21663
21664 \exp:w \exp_after:wN \exp_end:                                         21664
21665 }                                                                        21665
21666 \cs_new:Npn \int_rand:nn #1#2                                         21666
21667 {                                                                        21667
21668     \int_eval:n                                                         21668
21669     {                                                                    21669
21670         \exp_after:wN \__fp_randint:ww                                21670
21671         \int_value:w \int_eval:n {#1} \exp_after:wN \__fp_sep:        21671
21672         \int_value:w \int_eval:n {#2} \__fp_sep:                      21672
21673     }                                                                    21673
21674 }                                                                        21674
21675 \cs_new:Npn \__fp_randint:ww #1\__fp_sep: #2\__fp_sep:              21675
21676 {                                                                        21676
21677     \if_int_compare:w #1 > #2 \exp_stop_f:                             21677
21678     \msg_expandable_error:nnnn                                          21678
21679         { kernel } { randint-backward-range } {#1} {#2}              21679
21680     \__fp_randint:ww #2\__fp_sep: #1\__fp_sep:                       21680
21681 \else:                                                                    21681
21682     \if_int_compare:w \__fp_int_eval:w #2                             21682
21683         \if_int_compare:w #1 > \c_zero_int                             21683
21684             - #1 < \__fp_int_eval:w                                     21684
21685         \else:                                                            21685
21686             < \__fp_int_eval:w #1 +                                     21686
21687         \fi:                                                              21687
21688         \c__kernel_randint_max_int                                     21688
21689         \__fp_int_eval_end:                                             21689
21690         \__kernel_randint:n                                             21690
21691             { \__fp_int_eval:w #2 - #1 + 1 \__fp_int_eval_end: }      21691
21692         - 1 + #1                                                         21692
21693     \else:                                                                21693
21694         \__kernel_randint:nn {#1} {#2}                                 21694
21695     \fi:                                                                  21695
21696 \fi:                                                                      21696
21697 }                                                                        21697
21698 \cs_new:Npn \__kernel_randint:nn #1#2                                  21698

```

```

21699 {
21700 #1
21701 \exp_after:wN \__fp_randint_wide_aux:w
21702 \int_value:w
21703 \exp_after:wN \__fp_randint_split_o:Nw
21704 \tex_uniformdeviate:D 268435456 \__fp_sep:
21705 \int_value:w
21706 \exp_after:wN \__fp_randint_split_o:Nw
21707 \tex_uniformdeviate:D 268435456 \__fp_sep:
21708 \int_value:w
21709 \exp_after:wN \__fp_randint_split_o:Nw
21710 \int_value:w \__fp_int_eval:w 131072 +
21711 \exp_after:wN \__fp_randint_split_o:Nw
21712 \int_value:w
21713 \__kernel_int_add:nnn {#2} { -#1 } { -\c_max_int } \__fp_sep:
21714 .
21715 }
21716 \cs_new:Npn \__fp_randint_split_o:Nw #1#2 \__fp_sep:
21717 {
21718 \if_meaning:w 0 #1
21719 0 \exp_after:wN \__fp_sep: \int_value:w 0
21720 \else:
21721 \exp_after:wN \__fp_randint_split_aux:w
21722 \int_value:w \__fp_int_eval:w (#1#2 - 8192) / 16384 \__fp_sep:
21723 + #1#2
21724 \fi:
21725 \exp_after:wN \__fp_sep:
21726 }
21727 \cs_new:Npn \__fp_randint_split_aux:w #1 \__fp_sep:
21728 {
21729 #1 \exp_after:wN \__fp_sep:
21730 \int_value:w \__fp_int_eval:w - #1 * 16384
21731 }
21732 \cs_new:Npn \__fp_randint_wide_aux:w
21733 #1\__fp_sep:#2\__fp_sep: #3\__fp_sep:#4\__fp_sep:
21734 #5\__fp_sep:#6\__fp_sep:#7\__fp_sep: .
21735 {
21736 \exp_after:wN \__fp_randint_wide_auxii:w
21737 \int_value:w \__fp_int_eval:w #5 * #3 + #6 * #1 +
21738 (#5 * #4 + #6 * #3 + #7 * #1 +
21739 (#5 * #2 + #7 * #3 +
21740 (16384 * #6 + #7) * (16384 * #4 + #2) / 268435456) / 16384
21741 ) / 16384 \exp_after:wN \__fp_sep:
21742 \int_value:w \__fp_int_eval:w (#5 + #6) * 16384 + #7 \__fp_sep:
21743 #1 \__fp_sep: #5 \__fp_sep:
21744 }

```

```

21745 \cs_new:Npn \__fp_randint_wide_auxii:w 21745
21746 #1\__fp_sep: #2\__fp_sep: #3\__fp_sep: #4\__fp_sep: 21746
21747 { 21747
21748 \if_int_odd:w 0 21748
21749 \if_int_compare:w #1 = #2 \else: \exp_stop_f: \fi: 21749
21750 \if_int_compare:w #4 = \c_zero_int 1 \fi: 21750
21751 \if_int_compare:w #3 = 16383 ~ 1 \fi: 21751
21752 \exp_stop_f: 21752
21753 \exp_after:wN \prg_break: 21753
21754 \fi: 21754
21755 \if_int_compare:w #4 < 8 \exp_stop_f: 21755
21756 + #4 * #3 * 16384 21756
21757 \else: 21757
21758 + 8 * #3 * 16384 + (#4 - 8) * #3 * 16384 21758
21759 \fi: 21759
21760 + #1 21760
21761 \prg_break_point: 21761
21762 } 21762
21763 \cs_new:Npn \int_rand:n #1 21763
21764 { 21764
21765 \int_eval:n 21765
21766 { \exp_args:Nf \__fp_randint:n { \int_eval:n {#1} } } 21766
21767 } 21767
21768 \cs_new:Npn \__fp_randint:n #1 21768
21769 { 21769
21770 \if_int_compare:w #1 < \c_one_int 21770
21771 \msg_expandable_error:nnnn 21771
21772 { kernel } { randint-backward-range } { 1 } {#1} 21772
21773 \__fp_randint:ww #1\__fp_sep: 1\__fp_sep: 21773
21774 \else: 21774
21775 \if_int_compare:w #1 > \c__kernel_randint_max_int 21775
21776 \__kernel_randint:nn { 1 } {#1} 21776
21777 \else: 21777
21778 \__kernel_randint:n {#1} 21778
21779 \fi: 21779
21780 \fi: 21780
21781 } 21781
21782 %% File l3fp-types.dtx (C) Copyright 2012–2015,2017,2018,2020,2021,2023 The LaTeX Project 21782
21783 \cs_new:Npe \__fp_types_cs_to_op:N #1 21783
21784 { 21784
21785 \exp_not:N \exp_after:wN \exp_not:N \__fp_types_cs_to_op_auxi:wwwn 21785
21786 \exp_not:N \token_to_str:N #1 \s__fp_mark 21786
21787 \exp_not:N \__fp_use_i_delimit_by_s_stop:nw 21787
21788 \tl_to_str:n { __fp_o:w } \s__fp_mark 21788
21789 { \exp_not:N \__fp_use_i_delimit_by_s_stop:nw ? } 21789
21790 \s__fp_stop 21790

```

21791	}	21791
21792	\use:e	21792
21793	{	21793
21794	\cs_new:Npn \exp_not:N __fp_types_cs_to_op_auxi:wwwn	21794
21795	#1 \tl_to_str:n { __fp_ } #2	21795
21796	\tl_to_str:n { _o:w } #3 \s__fp_mark #4 { #4 {#2} }	21796
21797	}	21797
21798	\cs_new:Npn __fp_types_unary:NNw #1	21798
21799	{	21799
21800	\exp_args:Nf __fp_types_unary_auxi:nNw	21800
21801	{ __fp_types_cs_to_op:N #1 }	21801
21802	}	21802
21803	\cs_new:Npn __fp_types_unary_auxi:nNw #1#2#3	21803
21804	{	21804
21805	\exp_after:wN __fp_types_unary_auxii:NnNw	21805
21806	\cs:w __fp_#1 __fp_type_from_scan:N #3 _o:w \cs_end:	21806
21807	{#1}	21807
21808	#2#3	21808
21809	}	21809
21810	\cs_new:Npn __fp_types_unary_auxii:NnNw #1#2#3	21810
21811	{	21811
21812	\token_if_eq_meaning:NNTF \scan_stop: #1	21812
21813	{ __fp_invalid_operation_o:nw {#2} }	21813
21814	{ #1 #3 }	21814
21815	}	21815
21816	\cs_new:Npn __fp_types_binary:Nww #1	21816
21817	{	21817
21818	\exp_last_unbraced:Nf __fp_types_binary_auxi:Nww	21818
21819	{ __fp_types_cs_to_op:N #1 }	21819
21820	}	21820
21821	\cs_new:Npn __fp_types_binary_auxi:Nww #1#2#3__fp_sep: #4#5__fp_sep: @	21821
21822	{	21822
21823	\exp_after:wN __fp_types_binary_auxii:NNww	21823
21824	\cs:w	21824
21825	__fp	21825
21826	__fp_type_from_scan:N #2	21826
21827	_#1	21827
21828	__fp_type_from_scan:N #4	21828
21829	_o:ww	21829
21830	\cs_end:	21830
21831	#1 #2#3__fp_sep: #4#5__fp_sep:	21831
21832	}	21832
21833	\cs_new:Npn __fp_types_binary_auxii:NNww #1#2	21833
21834	{	21834
21835	\token_if_eq_meaning:NNTF \scan_stop: #1	21835
21836	{ __fp_invalid_operation_o:Nww #2 }	21836


```
21837     {#1}
21838   }
21839   %% File l3fp-symbolic.dtx (C) Copyright 2012-2025 The LaTeX Project
21840   \fp_new:N \l__fp_symbolic_fp
21841   \scan_new:N \s__fp_symbolic
21842   \cs_new_protected:Npn \__fp_symbolic_chk:w #1,#2#3\__fp_sep:
21843   {
21844     \msg_error:nne { fp } { misused-fp }
21845     {
21846       \__fp_to_tl_dispatch:w
21847       \s__fp_symbolic \__fp_symbolic_chk:w #1,{#2}\__fp_sep:
21848     }
21849   }
21850   \cs_new:Npn \__fp_if_has_symbolic:nTF #1
21851   {
21852     \__fp_if_has_symbolic_aux:w
21853     #1 \s__fp_mark \use_i:nn
21854     \s__fp_symbolic \s__fp_mark \use_ii:nn
21855     \s__fp_stop
21856   }
21857   \cs_new:Npn \__fp_if_has_symbolic_aux:w
21858     #1 \s__fp_symbolic #2 \s__fp_mark #3#4 \s__fp_stop { #3 }
21859   \cs_new:Npn \__fp_exp_after_symbolic_f:nw
21860     #1 \s__fp_symbolic \__fp_symbolic_chk:w #2, #3#4\__fp_sep:
21861   {
21862     \exp_after:wN \__fp_exp_after_symbolic_aux:w
21863     \exp:w
21864     \__fp_exp_after_symbolic_loop:N #2
21865     { , \exp:w \use_none:nn }
21866     \exp_after:wN \exp_end: \exp_after:wN
21867     {
21868       \exp:w \exp_end_continue_f:w
21869       \__fp_exp_after_array_f:w #3 \s__fp_expr_stop
21870       \exp_after:wN
21871     }
21872     \exp_after:wN \__fp_sep:
21873     \exp:w \exp_end_continue_f:w #1
21874   }
21875   \cs_new:Npn \__fp_exp_after_symbolic_aux:w #1, #2\__fp_sep:
21876   {
21877     \__fp_if_has_symbolic:nTF {#2}
21878     { \s__fp_symbolic \__fp_symbolic_chk:w #1, {#2} \__fp_sep: }
21879     { #1 #2 @ \prg_do_nothing: }
21880   }
21881   \cs_new:Npn \__fp_exp_after_symbolic_loop:N #1
21882   {
```

21883	\exp_after:wN \exp_end:	21883
21884	\exp_after:wN #1	21884
21885	\exp:w	21885
21886	__fp_exp_after_symbolic_loop:N	21886
21887	}	21887
21888	\cs_new:Npn __fp_symbolic_binary_o:Nww #1 #2__fp_sep: #3__fp_sep:	21888
21889	{	21889
21890	__fp_exp_after_symbolic_f:nw { \exp_after:wN \exp_stop_f: }	21890
21891	\s__fp_symbolic __fp_symbolic_chk:w	21891
21892	__fp_types_binary:Nww #1 , { #2__fp_sep: #3__fp_sep: } __fp_sep:	21892
21893	}	21893
21894	\cs_set_protected:Npn __fp_tmp:w #1#2	21894
21895	{	21895
21896	\cs_new:cpn	21896
21897	{ __fp_symbolic_#2_symbolic_o:ww }	21897
21898	{ __fp_symbolic_binary_o:Nww #1 }	21898
21899	\cs_new_eq:cc	21899
21900	{ __fp_symbolic_#2_o:ww }	21900
21901	{ __fp_symbolic_#2_symbolic_o:ww }	21901
21902	\cs_new_eq:cc	21902
21903	{ __fp_#2_symbolic_o:ww }	21903
21904	{ __fp_symbolic_#2_symbolic_o:ww }	21904
21905	}	21905
21906	\tl_map_inline:nn { + - * / ^ & } %	21906
21907	{ \exp_args:Nc __fp_tmp:w { __fp_#1_o:ww } {#1} }	21907
21908	\cs_new:Npn __fp_symbolic_unary_o:NNw #1#2#3__fp_sep: @	21908
21909	{	21909
21910	__fp_exp_after_symbolic_f:nw { \exp_after:wN \exp_stop_f: }	21910
21911	\s__fp_symbolic __fp_symbolic_chk:w	21911
21912	__fp_types_unary:NNw #1#2 , { #3__fp_sep: } __fp_sep:	21912
21913	}	21913
21914	\tl_map_inline:nn	21914
21915	{	21915
21916	{acos} {acsc} {asec} {asin} {cos} {cot} {csc} {exp} {fact} {ln}	21916
21917	{not} {sec} {set_sign} {sin} {sign} {sqrt} {tan}	21917
21918	}	21918
21919	{	21919
21920	\cs_new:cpe { __fp_symbolic_#1_o:w }	21920
21921	{	21921
21922	\exp_not:N __fp_symbolic_unary_o:NNw	21922
21923	\exp_not:c { __fp_#1_o:w }	21923
21924	}	21924
21925	}	21925
21926	\cs_set_protected:Npn __fp_tmp:w #1#2#3	21926
21927	{	21927
21928	\cs_new:cpn { __fp_symbolic_to_#1:w }	21928

```

21929         {
21930             \exp_after:wN \__fp_symbolic_convert:wnnN
21931             \exp:w \exp_end_continue_f:w
21932             \__fp_exp_after_symbolic_f:nw { { #2 } { fp_to_#1 } #3 }
21933         }
21934     }
21935     \__fp_tmp:w { decimal } { 0 } \__fp_to_decimal_dispatch:w
21936     \__fp_tmp:w { int } { 0 } \__fp_to_int_dispatch:w
21937     \__fp_tmp:w { scientific } { nan } \__fp_to_scientific_dispatch:w
21938     \cs_new:Npn \__fp_symbolic_convert:wnnN #1#2\__fp_sep: #3#4#5
21939     {
21940         \str_if_eq:nnTF {#1} { \s__fp_symbolic }
21941         { \__fp_invalid_operation:nnw {#3} {#4} #1#2\__fp_sep: }
21942         { #5 #1#2\__fp_sep: }
21943     }
21944     \cs_new:Npn \__fp_symbolic_cs_arg_to_fn:NN #1
21945     {
21946         \exp_args:Nf \__fp_symbolic_op_arg_to_fn:nN
21947         { \__fp_types_cs_to_op:N #1 }
21948     }
21949     \cs_new:Npn \__fp_symbolic_op_arg_to_fn:nN #1#2
21950     {
21951         \str_case:nnF { #1 #2 }
21952         {
21953             { not ? } { ! }
21954             { set_sign 0 } { abs }
21955             { set_sign 2 } { - }
21956         }
21957         {
21958             \token_if_eq_meaning:NNTF #2 \use_ii:nn
21959             { #1 d } {#1}
21960         }
21961     }
21962     \cs_new:Npn \__fp_symbolic_to_tl:w
21963     \s__fp_symbolic \__fp_symbolic_chk:w #1#2, #3#4\__fp_sep:
21964     {
21965         \str_case:nnTF {#1}
21966         {
21967             { \__fp_types_unary:NNw } { \__fp_symbolic_unary_to_tl:NNw }
21968             { \__fp_types_binary:Nww } { \__fp_symbolic_binary_to_tl:Nww }
21969             { \__fp_function_o:w } { \__fp_symbolic_function_to_tl:Nw }
21970         }
21971         { #2, #3 @ }
21972         { \tl_to_str:n {#2} }
21973     }
21974     \cs_new:Npn \__fp_symbolic_unary_to_tl:NNw #1#2 , #3 @

```

```

21975 {
21976     \use:e
21977     {
21978         \__fp_symbolic_cs_arg_to_fn:NN #1#2
21979         ( \__fp_to_tl_dispatch:w #3 )
21980     }
21981 }
21982 \cs_new:Npn \__fp_symbolic_binary_to_tl:Nww #1, #2\__fp_sep: #3\__fp_sep: @
21983 {
21984     \use:e
21985     {
21986         ( \__fp_to_tl_dispatch:w #2\__fp_sep: )
21987         \__fp_types_cs_to_op:N #1
21988         ( \__fp_to_tl_dispatch:w #3\__fp_sep: )
21989     }
21990 }
21991 \cs_new:Npn \__fp_symbolic_function_to_tl:Nw #1, #2@
21992 {
21993     \use:e
21994     {
21995         \__fp_types_cs_to_op:N #1
21996         ( \__fp_array_to_clist:n {#2} )
21997     }
21998 }
21999 \prg_new_protected_conditional:Npnn
22000     \__fp_id_if_invalid:n #1 { T , F , TF }
22001 {
22002     \tl_if_empty:nTF {#1}
22003     { \prg_return_true: }
22004     {
22005         \tl_if_in:nnTF { #1 } { ~ }
22006         { \prg_return_true: }
22007         {
22008             \__fp_id_if_invalid_aux:N #1
22009             { ? \prg_break:n \prg_return_false: }
22010             \prg_break_point:
22011         }
22012     }
22013 }
22014 \cs_new:Npn \__fp_id_if_invalid_aux:N #1
22015 {
22016     \use_none:n #1
22017     \int_compare:nF { `a <= `#1 <= `z }
22018     {
22019         \int_compare:nF { `A <= `#1 <= `Z }
22020         { \prg_break:n \prg_return_true: }

```

```
22021 }
22022 \__fp_id_if_invalid_aux:N
22023 }
22024 \cs_new:Npn \__fp_variable_o:w #1 @ #2
22025 {
22026   \fp_if_exist:cTF { l__fp_variable_#1_fp }
22027   {
22028     \exp_last_unbraced:Nf \__fp_exp_after_array_f:w
22029     { \use:c { l__fp_variable_#1_fp } } \s__fp_expr_stop
22030     \exp_after:wN \exp_stop_f: #2
22031   }
22032   {
22033     \token_if_eq_meaning:NNTF #2 \prg_do_nothing:
22034     {
22035       \s__fp_symbolic \__fp_symbolic_chk:w
22036       \__fp_variable_o:w #1 , { } \__fp_sep:
22037     }
22038     {
22039       \exp_after:wN \s__fp_symbolic
22040       \exp_after:wN \__fp_symbolic_chk:w
22041       \exp_after:wN \__fp_variable_o:w
22042       \exp:w
22043       \__fp_exp_after_symbolic_loop:N #1
22044       { , \exp:w \use_none:nn }
22045       \exp_after:wN \exp_end:
22046       \exp_after:wN { \exp_after:wN } \exp_after:wN \__fp_sep:
22047       #2
22048     }
22049   }
22050 }
22051 \cs_new_protected:Npn \__fp_variable_set_parsing:Nn #1#2
22052 {
22053   \cs_set:Npn \__fp_tmp:w
22054   {
22055     \__fp_exp_after_symbolic_f:nw { \__fp_parse_infix:NN }
22056     \s__fp_symbolic \__fp_symbolic_chk:w
22057     \__fp_variable_o:w #2 , { } \__fp_sep:
22058   }
22059   \exp_args:NNc \__fp_variable_set_parsing_aux:NNn #1
22060   { __fp_parse_word_#2:N } {#2}
22061 }
22062 \cs_new_protected:Npn \__fp_variable_set_parsing_aux:NNn #1#2#3
22063 {
22064   \cs_if_eq:NNF #2 \__fp_tmp:w
22065   {
22066     \cs_if_exist:NTF #2
```

```

22067         {
22068             \msg_warning:nnnn
22069             { fp } { id-used-elsewhere } {#3} { variable }
22070             #1 #2 \__fp_tmp:w
22071         }
22072     {
22073         \cs_new_eq:NN #2 \scan_stop: % to declare the function
22074         #1 #2 \__fp_tmp:w
22075     }
22076 }
22077 }
22078 \cs_new_protected:Npn \fp_clear_variable:n #1
22079 {
22080     \exp_args:No \__fp_clear_variable:n { \tl_to_str:n {#1} }
22081 }
22082 \cs_new_protected:Npn \__fp_clear_variable:n #1
22083 {
22084     \__fp_id_if_invalid:nTF {#1}
22085     { \msg_error:nnn { fp } { id-invalid } {#1} }
22086     { \__fp_clear_variable_aux:n {#1} }
22087 }
22088 \cs_new_protected:Npn \__fp_clear_variable_aux:n #1
22089 {
22090     \cs_set_eq:cN { l__fp_variable_#1_fp } \tex_undefined:D
22091     \__fp_variable_set_parsing:Nn \cs_set_eq:NN {#1}
22092 }
22093 \cs_new_protected:Npn \fp_new_variable:n #1
22094 {
22095     \exp_args:No \__fp_new_variable:n { \tl_to_str:n {#1} }
22096 }
22097 \cs_new_protected:Npn \__fp_new_variable:n #1
22098 {
22099     \__fp_id_if_invalid:nTF {#1}
22100     { \msg_error:nnn { fp } { id-invalid } {#1} }
22101     {
22102         \cs_if_exist:cT { __fp_parse_word_#1:N }
22103         {
22104             \msg_error:nnn
22105             { fp } { id-already-defined } {#1}
22106             \cs_undefine:c { __fp_parse_word_#1:N }
22107             \cs_set_eq:cN { l__fp_variable_#1_fp } \tex_undefined:D
22108         }
22109         \__fp_variable_set_parsing:Nn \cs_gset_eq:NN {#1}
22110     }
22111 }
22112 \flag_new:N \l__fp_symbolic_flag

```

```

22113 \cs_new_protected:Npn \fp_set_variable:nn #1 22113
22114 { 22114
22115     \exp_args:No \__fp_set_variable:nn { \tl_to_str:n {#1} } 22115
22116 } 22116
22117 \cs_new_protected:Npn \__fp_set_variable:nn #1#2 22117
22118 { 22118
22119     \__fp_id_if_invalid:nTF {#1} 22119
22120     { \msg_error:nnn { fp } { id-invalid } {#1} } 22120
22121     { 22121
22122         \cs_if_exist:cF { __fp_parse_word_#1:N } 22122
22123         { 22123
22124             \msg_error:nnn {fp} { id-undefined } {#1} 22124
22125             \__fp_variable_set_parsing:Nn \cs_set_eq:NN {#1} 22125
22126         } 22126
22127         \fp_set:Nn \l__fp_symbolic_fp {#2} 22127
22128         \cs_set_nopar:cpn { l__fp_variable_#1_fp } 22128
22129         { \flag_ensure_raised:N \l__fp_symbolic_flag \c_nan_fp } 22129
22130         \flag_clear:N \l__fp_symbolic_flag 22130
22131         \fp_set:cn { l__fp_variable_#1_fp } { \l__fp_symbolic_fp } 22131
22132         \flag_if_raised:NT \l__fp_symbolic_flag 22132
22133         { 22133
22134             \msg_error:nneee { fp } { id-loop } 22134
22135             { #1 } 22135
22136             { \tl_to_str:n {#2} } 22136
22137             { \fp_to_tl:N \l__fp_symbolic_fp } 22137
22138         } 22138
22139     } 22139
22140 } 22140
22141 \msg_new:nnnn { fp } { id-invalid } 22141
22142 { Floating~point~identifier~'#1'~invalid. } 22142
22143 { 22143
22144     LaTeX~has~been~asked~to~create~a~new~floating~point~identifier~'#1'~ 22144
22145     but~this~may~only~contain~ASCII~letters. 22145
22146 } 22146
22147 \msg_new:nnnn { fp } { id-already-defined } 22147
22148 { Floating~point~identifier~'#1'~already~defined. } 22148
22149 { 22149
22150     LaTeX~has~been~asked~to~create~a~new~floating~point~identifier~'#1'~ 22150
22151     but~this~name~has~already~been~used~elsewhere. 22151
22152 } 22152
22153 \msg_new:nnnn { fp } { id-undefined } 22153
22154 { Floating~point~identifier~'#1'~is~undefined. } 22154
22155 { 22155
22156     LaTeX~has~been~asked~to~set~a~floating~point~identifier~'#1'~ 22156
22157     but~this~name~has~not~been~declared. 22157
22158 } 22158

```



```

22159 \msg_new:nnnn { fp } { id-used-elsewhere } 22159
22160 { Floating-point~identifier~'#1'~already~used~for~something~else. } 22160
22161 { 22161
22162 LaTeX~has~been~asked~to~create~a~new~floating~point~identifier~'#1'~ 22162
22163 but~this~name~is~used,~and~is~not~a~user~defined~#2. 22163
22164 } 22164
22165 \msg_new:nnnn { fp } { id-loop } 22165
22166 { Variable~'#1'~used~in~the~definition~of~'#1'. } 22166
22167 { 22167
22168 LaTeX~has~been~asked~to~set~the~floating~point~identifier~'#1'~ 22168
22169 to~the~expression~'#2'.~Evaluating~this~expression~yields~'#3',~ 22169
22170 which~contains~'#1'~itself. 22170
22171 } 22171
22172 %% File l3fp-functions.dtx (C) Copyright 2012-2018,2020,2021,2023 The LaTeX Project 22172
22173 \cs_new_protected:Npn \fp_new_function:n #1 22173
22174 { \exp_args:No \__fp_new_function:n { \tl_to_str:n {#1} } } 22174
22175 \cs_new_protected:Npn \__fp_new_function:n #1 22175
22176 { 22176
22177 \__fp_id_if_invalid:nTF {#1} 22177
22178 { \msg_error:nnn { fp } { id-invalid } {#1} } 22178
22179 { 22179
22180 \cs_if_exist:cT { __fp_parse_word_#1:N } 22180
22181 { 22181
22182 \msg_error:nnn 22182
22183 { fp } { id-already-defined } {#1} 22183
22184 \cs_undefine:c { __fp_parse_word_#1:N } 22184
22185 \cs_undefine:c { __fp_#1_o:w } 22185
22186 } 22186
22187 \__fp_function_set_parsing:Nn \cs_gset_eq:NN {#1} 22187
22188 } 22188
22189 } 22189
22190 \cs_new_protected:Npn \__fp_function_set_parsing:Nn #1#2 22190
22191 { 22191
22192 \exp_args:NNc \__fp_function_set_parsing_aux:NNn #1 22192
22193 { __fp_parse_word_#2:N } {#2} 22193
22194 } 22194
22195 \cs_new_protected:Npn \__fp_function_set_parsing_aux:NNn #1#2#3 22195
22196 { 22196
22197 \cs_set:Npe \__fp_tmp:w 22197
22198 { 22198
22199 \exp_not:N \__fp_parse_function:NNN 22199
22200 \exp_not:N \__fp_function_o:w 22200
22201 \exp_not:c { __fp_#3_o:w } 22201
22202 } 22202
22203 \cs_if_eq:NNF #2 \__fp_tmp:w 22203
22204 { 22204

```

```
22205 \cs_if_exist:NTF #2
22206 {
22207     \msg_warning:nnnn
22208     { fp } { id-used-elsewhere } {#3} { function }
22209     #1 #2 \__fp_tmp:w
22210 }
22211 {
22212     \cs_new_eq:NN #2 \scan_stop: % to declare the function
22213     #1 #2 \__fp_tmp:w
22214 }
22215 }
22216 }
22217 \cs_new:Npn \__fp_function_o:w #1#2 @
22218 {
22219     \cs_if_exist:NTF #1
22220     { #1 #2 @ }
22221     {
22222         \exp_after:wN \s__fp_symbolic
22223         \exp_after:wN \__fp_symbolic_chk:w
22224         \exp_after:wN \__fp_function_o:w
22225         \exp_after:wN #1
22226         \exp_after:wN ,
22227         \exp_after:wN {
22228             \exp:w \exp_end_continue_f:w
22229             \__fp_exp_after_array_f:w #2 \s__fp_expr_stop
22230             \exp_after:wN
22231         }
22232         \exp_after:wN \__fp_sep:
22233     }
22234 }
22235 \int_new:N \l__fp_function_arg_int
22236 \cs_new_protected:Npn \fp_set_function:nnn #1
22237 {
22238     \exp_args:NNo \__fp_set_function:Nnnn \cs_set_eq:cN
22239     { \tl_to_str:n {#1} }
22240 }
22241 \cs_new_protected:Npn \__fp_set_function:Nnnn #1#2#3#4
22242 {
22243     \__fp_id_if_invalid:nTF {#2}
22244     { \msg_error:nnn { fp } { id-invalid } {#2} }
22245     {
22246         \cs_if_exist:cF { __fp_parse_word_#2:N }
22247         {
22248             \msg_error:nnn {fp} { id-undefined } {#2}
22249             \__fp_function_set_parsing:Nn \cs_set_eq:NN {#2}
22250         }
22251     }
```

```

22251 \group_begin:
22252 \int_zero:N \l__fp_function_arg_int
22253 \exp_args:No \clist_map_inline:nn { \tl_to_str:n {#3} }
22254 {
22255 \int_incr:N \l__fp_function_arg_int
22256 \exp_args:Ne \__fp_clear_variable_aux:n
22257 {
22258 \c_underscore_str \tex_romannumeral:D \l__fp_function_arg_int
22259 }
22260 \fp_clear_variable:n {##1}
22261 \cs_set_nopar:cpe { l__fp_variable_##1_fp }
22262 {
22263 \exp_not:N \s__fp_symbolic
22264 \exp_not:N \__fp_symbolic_chk:w
22265 \exp_not:N \__fp_function_arg_o:w
22266 \int_use:N \l__fp_function_arg_int
22267 #####1 , { } \__fp_sep:
22268 }
22269 }
22270 \cs_set:Npn \__fp_function_arg_o:w ##1 @
22271 {
22272 \exp_after:wN \s__fp_symbolic
22273 \exp_after:wN \__fp_symbolic_chk:w
22274 \exp_after:wN \__fp_function_arg_o:w
22275 \tex_romannumeral:D
22276 \__fp_exp_after_symbolic_loop:N ##1
22277 { , \tex_romannumeral:D \use_none:nn }
22278 \exp_after:wN \c_zero_int
22279 \exp_after:wN { \exp_after:wN } \exp_after:wN \__fp_sep:
22280 }
22281 \fp_set:Nn \l__fp_symbolic_fp {#4}
22282 \use:e
22283 {
22284 \exp_not:n { \cs_gset:Npn \__fp_tmp:w ##1 }
22285 { \exp_not:o { \l__fp_symbolic_fp } }
22286 }
22287 \use:e
22288 {
22289 \exp_not:n { \cs_gset:Npn \__fp_tmp:w ##1 @ }
22290 {
22291 \exp_not:N \__fp_exp_after_symbolic_f:nw
22292 \exp_not:n { { \exp_after:wN \exp_stop_f: } }
22293 \exp_not:o { \__fp_tmp:w { . , {##1} } }
22294 }
22295 }
22296 \group_end:

```

```

22297 #1 { __fp_#2_o:w } \__fp_tmp:w 22297
22298 } 22298
22299 } 22299
22300 \cs_new:Npn \__fp_function_arg_o:w #1. #2 22300
22301 { 22301
22302 \if_meaning:w @ #2 22302
22303 \exp_after:wN \__fp_function_arg_few:w 22303
22304 \fi: 22304
22305 \if_int_compare:w #1 = \c_one_int 22305
22306 \exp_after:wN \__fp_function_arg_get:w 22306
22307 \fi: 22307
22308 \__fp_use_i_until_s:nw 22308
22309 { 22309
22310 \exp_after:wN \__fp_function_arg_o:w 22310
22311 \int_value:w \int_eval:n { #1 - 1 } . 22311
22312 } 22312
22313 #2 22313
22314 } 22314
22315 \cs_new:Npn \__fp_function_arg_few:w #1 @ { \exp_after:wN \c_nan_fp } 22315
22316 \cs_new:Npn \__fp_function_arg_get:w #1#2#3\__fp_sep: #4 @ 22316
22317 { 22317
22318 \__fp_exp_after_array_f:w #3\__fp_sep: \s__fp_expr_stop 22318
22319 \exp_after:wN \exp_stop_f: 22319
22320 } 22320
22321 \cs_new_protected:Npn \fp_clear_function:n #1 22321
22322 { \exp_args:No \__fp_clear_function:n { \tl_to_str:n {#1} } } 22322
22323 \cs_new_protected:Npn \__fp_clear_function:n #1 22323
22324 { 22324
22325 \__fp_id_if_invalid:nTF {#1} 22325
22326 { \msg_error:nnn { fp } { id-invalid } {#1} } 22326
22327 { 22327
22328 \cs_set_eq:cN { __fp_#1_o:w } \tex_undefine:D 22328
22329 \__fp_function_set_parsing:Nn \cs_set_eq:NN {#1} 22329
22330 } 22330
22331 } 22331
22332 %% File: l3fparray.dtx 22332
22333 \int_new:N \g__fp_array_int 22333
22334 \int_new:N \l__fp_array_loop_int 22334
22335 \cs_new_protected:Npn \fparray_new:Nn #1#2 22335
22336 { 22336
22337 \tl_new:N #1 22337
22338 \prg_replicate:nn { 3 } 22338
22339 { 22339
22340 \int_gincr:N \g__fp_array_int 22340
22341 \exp_args:NNc \tl_gput_right:Nn #1 22341
22342 { g__fp_array_ \__fp_int_to_roman:w \g__fp_array_int _intarray } 22342

```

```
22343     }
22344     \exp_last_unbraced:Nfo \__fp_array_new:nNNNN
22345     { \int_eval:n {#2} } #1 #1
22346 }
22347 \cs_generate_variant:Nn \fparray_new:Nn { c }
22348 \cs_new_protected:Npn \__fp_array_new:nNNNN #1#2#3#4#5
22349 {
22350     \int_compare:nNnTF {#1} < 0
22351     {
22352         \msg_error:nnn { kernel } { negative-array-size } {#1}
22353         \cs_undefine:N #1
22354         \int_gsub:Nn \g__fp_array_int { 3 }
22355     }
22356     {
22357         \intarray_new:Nn #2 {#1}
22358         \intarray_new:Nn #3 {#1}
22359         \intarray_new:Nn #4 {#1}
22360     }
22361 }
22362 \cs_new:Npn \fparray_count:N #1
22363 {
22364     \exp_after:wN \use_i:nnn
22365     \exp_after:wN \intarray_count:N #1
22366 }
22367 \cs_generate_variant:Nn \fparray_count:N { c }
22368 \cs_new:Npn \__fp_array_bounds:NNnTF #1#2#3#4#5
22369 {
22370     \if_int_compare:w 1 > #3 \exp_stop_f:
22371         \__fp_array_bounds_error:NNn #1 #2 {#3}
22372         #5
22373     \else:
22374         \if_int_compare:w #3 > \fparray_count:N #2 \exp_stop_f:
22375             \__fp_array_bounds_error:NNn #1 #2 {#3}
22376             #5
22377         \else:
22378             #4
22379         \fi:
22380     \fi:
22381 }
22382 \cs_new:Npn \__fp_array_bounds_error:NNn #1#2#3
22383 {
22384     #1 { kernel } { out-of-bounds }
22385     { \token_to_str:N #2 } {#3} { \fparray_count:N #2 }
22386 }
22387 \cs_new_protected:Npn \fparray_gset:Nnn #1#2#3
22388 {
```

```
22389 \exp_after:wN \exp_after:wN 22389
22390 \exp_after:wN \__fp_array_gset:NNNNww 22390
22391 \exp_after:wN #1 22391
22392 \exp_after:wN #1 22392
22393 \int_value:w \int_eval:n {#2} \exp_after:wN \__fp_sep: 22393
22394 \exp:w \exp_end_continue_f:w \__fp_parse:n {#3} 22394
22395 } 22395
22396 \cs_generate_variant:Nn \fparray_gset:Nnn { c } 22396
22397 \cs_new_protected:Npn \__fp_array_gset:NNNNww #1#2#3#4#5 \__fp_sep: #6 \__fp_sep: 22397
22398 { 22398
22399 \__fp_array_bounds:NNnTF \msg_error:nneee #4 {#5} 22399
22400 { 22400
22401 \exp_after:wN \__fp_change_func_type:NNN 22401
22402 \__fp_use_i_until_s:nw #6 \__fp_sep: 22402
22403 \__fp_array_gset:w 22403
22404 \__fp_array_gset_recover:Nw 22404
22405 #6 \__fp_sep: {#5} #1 #2 #3 22405
22406 } 22406
22407 { } 22407
22408 } 22408
22409 \cs_new_protected:Npn \__fp_array_gset_recover:Nw #1#2 \__fp_sep: 22409
22410 { 22410
22411 \__fp_error:nffn { unknown-type } { \tl_to_str:n { #2 \__fp_sep: } } { } { } { } 22411
22412 \exp_after:wN #1 \c_nan_fp 22412
22413 } 22413
22414 \cs_new_protected:Npn \__fp_array_gset:w \s__fp \__fp_chk:w #1#2 22414
22415 { 22415
22416 \if_case:w #1 \exp_stop_f: 22416
22417 \__fp_case_return:nw { \__fp_array_gset_special:nnNNN {#2} } 22417
22418 \or: \exp_after:wN \__fp_array_gset_normal:w 22418
22419 \or: \__fp_case_return:nw { \__fp_array_gset_special:nnNNN { #2 3 } } 22419
22420 \or: \__fp_case_return:nw { \__fp_array_gset_special:nnNNN { 1 } } 22420
22421 \fi: 22421
22422 \s__fp \__fp_chk:w #1 #2 22422
22423 } 22423
22424 \cs_new_protected:Npn \__fp_array_gset_normal:w 22424
22425 \s__fp \__fp_chk:w 1 #1 #2 #3#4#5 \__fp_sep: #6#7#8#9 22425
22426 { 22426
22427 \__kernel_intarray_gset:Nnn #7 {#6} {#2} 22427
22428 \__kernel_intarray_gset:Nnn #8 {#6} 22428
22429 { \if_meaning:w 2 #1 3 \else: 1 \fi: #3#4 } 22429
22430 \__kernel_intarray_gset:Nnn #9 {#6} { 1 \use:nn #5 } 22430
22431 } 22431
22432 \cs_new_protected:Npn \__fp_array_gset_special:nnNNN #1#2#3#4#5 22432
22433 { 22433
22434 \__kernel_intarray_gset:Nnn #3 {#2} {#1} 22434
```

```
22435 \__kernel_intarray_gset:Nnn #4 {#2} {0} 22435
22436 \__kernel_intarray_gset:Nnn #5 {#2} {0} 22436
22437 } 22437
22438 \cs_new_protected:Npn \fparray_gzero:N #1 22438
22439 { 22439
22440 \int_zero:N \l__fp_array_loop_int 22440
22441 \prg_replicate:nn { \fparray_count:N #1 } 22441
22442 { 22442
22443 \int_incr:N \l__fp_array_loop_int 22443
22444 \exp_after:wN \__fp_array_gset_special:nnNNN 22444
22445 \exp_after:wN 0 22445
22446 \exp_after:wN \l__fp_array_loop_int 22446
22447 #1 22447
22448 } 22448
22449 } 22449
22450 \cs_generate_variant:Nn \fparray_gzero:N { c } 22450
22451 \cs_new:Npn \fparray_item:Nn #1#2 22451
22452 { 22452
22453 \exp_after:wN \__fp_array_item:NwN 22453
22454 \exp_after:wN #1 22454
22455 \int_value:w \int_eval:n {#2} \__fp_sep: 22455
22456 \__fp_to_decimal:w 22456
22457 } 22457
22458 \cs_generate_variant:Nn \fparray_item:Nn { c } 22458
22459 \cs_new:Npn \fparray_item_to_tl:Nn #1#2 22459
22460 { 22460
22461 \exp_after:wN \__fp_array_item:NwN 22461
22462 \exp_after:wN #1 22462
22463 \int_value:w \int_eval:n {#2} \__fp_sep: 22463
22464 \__fp_to_tl:w 22464
22465 } 22465
22466 \cs_generate_variant:Nn \fparray_item_to_tl:Nn { c } 22466
22467 \cs_new:Npn \__fp_array_item:NwN #1#2 \__fp_sep: #3 22467
22468 { 22468
22469 \__fp_array_bounds:NNnTF \msg_expandable_error:nnfff #1 {#2} 22469
22470 { \exp_after:wN \__fp_array_item:NNNnN #1 {#2} #3 } 22470
22471 { \exp_after:wN #3 \c_nan_fp } 22471
22472 } 22472
22473 \cs_new:Npn \__fp_array_item:NNNnN #1#2#3#4 22473
22474 { 22474
22475 \exp_after:wN \__fp_array_item:N 22475
22476 \int_value:w \__kernel_intarray_item:Nn #2 {#4} \exp_after:wN \__fp_sep: 22476
22477 \int_value:w \__kernel_intarray_item:Nn #3 {#4} \exp_after:wN \__fp_sep: 22477
22478 \int_value:w \__kernel_intarray_item:Nn #1 {#4} \__fp_sep: 22478
22479 } 22479
22480 \cs_new:Npn \__fp_array_item:N #1 22480
```



```

22481 {
22482     \if_meaning:w 0 #1 \exp_after:wN \__fp_array_item_special:w \fi:
22483     \__fp_array_item:w #1
22484 }
22485 \cs_new:Npn \__fp_array_item:w #1 #2#3#4#5 #6 \__fp_sep: 1 #7 \__fp_sep:
22486 {
22487     \exp_after:wN \__fp_array_item_normal:w
22488     \int_value:w \if_meaning:w #1 1 0 \else: 2 \fi: \exp_stop_f:
22489     #7 \__fp_sep: {#2#3#4#5} {#6} \__fp_sep:
22490 }
22491 \cs_new:Npn \__fp_array_item_special:w #1 \__fp_sep: #2 \__fp_sep: #3 \__fp_sep: #4
22492 {
22493     \exp_after:wN #4
22494     \exp:w \exp_end_continue_f:w
22495     \if_case:w #3 \exp_stop_f:
22496         \exp_after:wN \c_zero_fp
22497     \or: \exp_after:wN \c_nan_fp
22498     \or: \exp_after:wN \c_minus_zero_fp
22499     \or: \exp_after:wN \c_inf_fp
22500     \else: \exp_after:wN \c_minus_inf_fp
22501     \fi:
22502 }
22503 \cs_new:Npn \__fp_array_item_normal:w
22504     #1 #2#3#4#5 #6 \__fp_sep: #7 \__fp_sep: #8 \__fp_sep: #9
22505     { #9 \s__fp \__fp_chk:w 1 #1 {#8} #7 {#2#3#4#5} {#6} \__fp_sep: }
22506 \prg_new_eq_conditional:NNn \fpararray_if_exist:N \cs_if_exist:N
22507 { TF , T , F , p }
22508 \prg_new_eq_conditional:NNn \fpararray_if_exist:c \cs_if_exist:c
22509 { TF , T , F , p }
22510 %% File: l3bitset.dtx
22511 \cs_if_exist:NT \@expl@finalise@setup@@
22512 {
22513     \tl_gput_right:Nn \@expl@finalise@setup@@
22514     { \declare@file@substitution { l3bitset.sty } { null.tex } }
22515 }
22516 \cs_new_protected:Npn \bitset_new:N #1
22517 {
22518     \__kernel_chk_if_free_cs:N #1
22519     \cs_gset_eq:NN #1 \c_zero_str
22520     \prop_new:c { g__bitset_ \cs_to_str:N #1 _name_prop }
22521 }
22522 \cs_new_protected:Npn \bitset_new:Nn #1 #2
22523 {
22524     \__kernel_chk_if_free_cs:N #1
22525     \cs_gset_eq:NN #1 \c_zero_str
22526     \prop_new:c { g__bitset_ \cs_to_str:N #1 _name_prop }

```

```

22527     \prop_gset_from_keyval:cn
22528     { g__bitset_ \cs_to_str:N #1 _name_prop }
22529     {#2}
22530 }
22531 \cs_generate_variant:Nn \bitset_new:N { c }
22532 \cs_generate_variant:Nn \bitset_new:Nn { c }
22533 \cs_new_protected:Npn \bitset_addto_named_index:Nn #1#2
22534 {
22535     \prop_gput_from_keyval:cn
22536     { g__bitset_ \cs_to_str:N #1 _name_prop } { #2 }
22537 }
22538 \prg_new_eq_conditional:NNn
22539     \bitset_if_exist:N \str_if_exist:N { p , T , F , TF }
22540 \prg_new_eq_conditional:NNn
22541     \bitset_if_exist:c \str_if_exist:c { p , T , F , TF }
22542 \cs_new_protected:Npn \__bitset_set_true:Nn #1#2
22543 { \__bitset_set:NNnN \str_set:Ne #1 {#2} 1 }
22544 \cs_new_protected:Npn \__bitset_gset_true:Nn #1#2
22545 { \__bitset_set:NNnN \str_gset:Ne #1 {#2} 1 }
22546 \cs_new_protected:Npn \__bitset_set_false:Nn #1#2
22547 { \__bitset_set:NNnN \str_set:Ne #1 {#2} 0 }
22548 \cs_new_protected:Npn \__bitset_gset_false:Nn #1#2
22549 { \__bitset_set:NNnN \str_gset:Ne #1 {#2} 0 }
22550 \cs_new_protected:Npn \__bitset_set:NNnN #1#2#3#4
22551 {
22552     \int_compare:nNnT {#3} > { 0 }
22553     {
22554         \int_compare:nNnTF { \str_count:N #2 } < {#3}
22555         {
22556             #1 #2
22557             {
22558                 #4
22559                 \prg_replicate:nn { #3 - \str_count:N #2 - 1 } { 0 }
22560                 #2
22561             }
22562         }
22563         {
22564             #1 #2
22565             {
22566                 \str_range:Nnn #2 { 1 } { -1 - (#3) }
22567                 #4
22568                 \str_range:Nnn #2 { 1 - (#3) } { -1 }
22569             }
22570         }
22571     }
22572 }

```

```

22573 \int_new:N \l__bitset_internal_int
22574 \prg_new_protected_conditional:Npnn \__bitset_test_digits:n #1 { TF }
22575 {
22576   \tex_afterassignment:D \__bitset_test_digits:w
22577   \l__bitset_internal_int = 0 \tl_trim_spaces_apply:nN {#1} \tl_to_str:n
22578   \__bitset_test_digits_end:
22579   \use_i:nnn \if_false:
22580   \__bitset_test_digits_end:
22581   \if_int_compare:w \c_zero_int < \l__bitset_internal_int
22582     \prg_return_true:
22583   \else:
22584     \prg_return_false:
22585   \fi:
22586 }
22587 \cs_new_eq:NN \__bitset_test_digits_end: \exp_stop_f:
22588 \cs_new_protected:Npn \__bitset_test_digits:w #1 \__bitset_test_digits_end: { }
22589 \cs_new_protected:Npn \bitset_set_true:Nn #1#2
22590 { \__bitset_set:NNn \__bitset_set_true:Nn #1 {#2} }
22591 \cs_new_protected:Npn \bitset_gset_true:Nn #1#2
22592 { \__bitset_set:NNn \__bitset_gset_true:Nn #1 {#2} }
22593 \cs_new_protected:Npn \bitset_set_false:Nn #1#2
22594 { \__bitset_set:NNn \__bitset_set_false:Nn #1 {#2} }
22595 \cs_new_protected:Npn \bitset_gset_false:Nn #1#2
22596 { \__bitset_set:NNn \__bitset_gset_false:Nn #1 {#2} }
22597 \cs_new_protected:Npn \__bitset_set:NNn #1#2#3
22598 {
22599   \prop_if_in:cnTF { g__bitset_ \cs_to_str:N #2 _name_prop } {#3}
22600   {
22601     #1 #2
22602     {
22603       \prop_item:cn { g__bitset_ \cs_to_str:N #2 _name_prop } {#3}
22604     }
22605   }
22606   {
22607     \__bitset_test_digits:nTF {#3}
22608     {
22609       #1 #2 {#3}
22610       \prop_gput:cnn { g__bitset_ \cs_to_str:N #2 _name_prop } {#3} {#3}
22611     }
22612     {
22613       \msg_warning:nnee { bitset } { unknown-name }
22614       { \token_to_str:N #2 }
22615       { \tl_to_str:n {#3} }
22616     }
22617   }
22618 }

```

```
22619 \cs_generate_variant:Nn \bitset_set_true:Nn { c } 22619
22620 \cs_generate_variant:Nn \bitset_gset_true:Nn { c } 22620
22621 \cs_generate_variant:Nn \bitset_set_false:Nn { c } 22621
22622 \cs_generate_variant:Nn \bitset_gset_false:Nn { c } 22622
22623 \cs_new_protected:Npn \bitset_clear:N #1 22623
22624 { 22624
22625     \str_set_eq:NN #1 \c_zero_str 22625
22626 } 22626
22627 \cs_new_protected:Npn \bitset_gclear:N #1 22627
22628 { 22628
22629     \str_gset_eq:NN #1 \c_zero_str 22629
22630 } 22630
22631 \cs_generate_variant:Nn \bitset_clear:N { c } 22631
22632 \cs_generate_variant:Nn \bitset_gclear:N { c } 22632
22633 \cs_new:Npn \bitset_to_arabic:N #1 22633
22634 { 22634
22635     \int_compare:nNnTF { \str_count:N #1 } < { 32 } 22635
22636     { \exp_args:No \int_from_bin:n {#1} } 22636
22637     { 22637
22638         \exp_after:wN \__bitset_to_int:nN \exp_after:wN 0 22638
22639         #1 \q_recursion_tail \q_recursion_stop 22639
22640     } 22640
22641 } 22641
22642 \cs_new:Npn \__bitset_to_int:nN #1#2 22642
22643 { 22643
22644     \quark_if_recursion_tail_stop_do:Nn #2 {#1} 22644
22645     \exp_args:Nf \__bitset_to_int:nN { \fp_eval:n { #1 * 2 + #2 } } 22645
22646 } 22646
22647 \cs_new:Npn \bitset_to_bin:N #1 22647
22648 { 22648
22649     #1 22649
22650 } 22650
22651 \cs_generate_variant:Nn \bitset_to_arabic:N { c } 22651
22652 \cs_generate_variant:Nn \bitset_to_bin:N { c } 22652
22653 \cs_new_eq:NN \bitset_use:N \tl_use:N 22653
22654 \cs_generate_variant:Nn \bitset_use:N { c } 22654
22655 \cs_new:Npn \bitset_item:Nn #1#2 22655
22656 { 22656
22657     \prop_if_in:cnTF { g__bitset_ \cs_to_str:N #1 _name_prop } {#2} 22657
22658     { 22658
22659         \int_eval:n 22659
22660         { 22660
22661             \str_item:Nn #1 22661
22662             { 0 - ( \prop_item:cn { g__bitset_ \cs_to_str:N #1 _name_prop } {#2} ) } 22662
22663             +0 22663
22664         } 22664
```

```
22665     } 22665
22666     { 22666
22667         0 22667
22668     } 22668
22669 } 22669
22670 \cs_generate_variant:Nn \bitset_item:Nn { c } 22670
22671 \cs_new_protected:Npn \bitset_show:N { \__bitset_show:NN \msg_show:nneeee } 22671
22672 \cs_generate_variant:Nn \bitset_show:N { c } 22672
22673 \cs_new_protected:Npn \bitset_log:N { \__bitset_show:NN \msg_log:nneeee } 22673
22674 \cs_generate_variant:Nn \bitset_log:N { c } 22674
22675 \cs_new_protected:Npn \__bitset_show:NN #1#2 22675
22676 { 22676
22677     \__kernel_chk_defined:NT #2 22677
22678     { 22678
22679         #1 { bitset } { show } 22679
22680         { \token_to_str:N #2 } 22680
22681         { \bitset_to_bin:N #2 } 22681
22682         { \bitset_to_arabic:N #2 } 22682
22683         { } 22683
22684     } 22684
22685 } 22685
22686 \cs_new_protected:Npn \bitset_show_named_index:N 22686
22687 { \__bitset_show_named_index:NN \msg_show:nneeee } 22687
22688 \cs_generate_variant:Nn \bitset_show_named_index:N { c } 22688
22689 \cs_new_protected:Npn \bitset_log_named_index:N 22689
22690 { \__bitset_show_named_index:NN \msg_log:nneeee } 22690
22691 \cs_generate_variant:Nn \bitset_log_named_index:N { c } 22691
22692 \cs_new_protected:Npn \__bitset_show_named_index:NN #1#2 22692
22693 { 22693
22694     \__kernel_chk_defined:NT #2 22694
22695     { 22695
22696         #1 { bitset } { show-names } 22696
22697         { \token_to_str:N #2 } 22697
22698         { \prop_map_function:cN { g__bitset_ \cs_to_str:N #2 _name_prop } 22698
22699             \msg_show_item:nn } 22699
22700         { } { } 22700
22701     } 22701
22702 \msg_new:nnn { bitset } { show } 22702
22703 { 22703
22704     The~bitset~#1~has~the~representation: \ 22704
22705     >~binary:~#2 \ 22705
22706     >~arabic:~#3 . 22706
22707 } 22707
22708 \msg_new:nnn { bitset } { show-names } 22708
22709 { 22709
```

```

22710 The~bitset~#1~
22711 \tl_if_empty:nTF {#2}
22712 { knows~no~names~yet \>~ . }
22713 { knows~the~name/index~pairs~(without~outer~braces): #2 . }
22714 }
22715 \msg_new:nnn { bitset } { unknown-name }
22716 { The~name~'#2'~is~unknown~for~bitset~\tl_to_str:n {#1} }
22717 \prop_gput:Nnn \g_msg_module_name_prop { bitset } { LaTeX }
22718 \prop_gput:Nnn \g_msg_module_type_prop { bitset } { }
22719 %% File: l3cctab.dtx
22720 \seq_new:N \g__cctab_stack_seq
22721 \seq_new:N \g__cctab_unused_seq
22722 \seq_new:N \g__cctab_group_seq
22723 \int_new:N \g__cctab_allocate_int
22724 \tl_new:N \l__cctab_internal_a_tl
22725 \tl_new:N \l__cctab_internal_b_tl
22726 \prop_new:N \g__cctab_endlinechar_prop
22727 \sys_if_engine luatex:TF
22728 {
22729 \cs_new_protected:Npn \cctab_new:N #1
22730 {
22731 \__kernel_chk_if_free_cs:N #1
22732 \__cctab_new:N #1
22733 }
22734 \cs_new_protected:Npn \__cctab_new:N #1
22735 {
22736 \newcatcodetable #1
22737 \tex_initcatcodetable:D #1
22738 }
22739 }
22740 {
22741 \cs_new_protected:Npn \__cctab_new:N #1
22742 {
22743 \debug_suspend:
22744 \intarray_new:Nn #1 { 257 }
22745 \debug_resume:
22746 }
22747 \cs_new_protected:Npn \__cctab_gstore:Nnn #1#2#3
22748 { \intarray_gset:Nnn #1 { #2 + 1 } {#3} }
22749 \cs_new_protected:Npn \cctab_new:N #1
22750 {
22751 \__kernel_chk_if_free_cs:N #1
22752 \__cctab_new:N #1
22753 \int_step_inline:nn { 256 }
22754 { \__kernel_intarray_gset:Nnn #1 {##1} { 12 } }
22755 \__kernel_intarray_gset:Nnn #1 { 257 } { 13 }

```

```

22756 \__cctab_gstore:Nnn #1 { 0 } { 9 }
22757 \__cctab_gstore:Nnn #1 { 13 } { 5 }
22758 \__cctab_gstore:Nnn #1 { 32 } { 10 }
22759 \__cctab_gstore:Nnn #1 { 37 } { 14 }
22760 \int_step_inline:nnn { 65 } { 90 }
22761 { \__cctab_gstore:Nnn #1 {##1} { 11 } }
22762 \__cctab_gstore:Nnn #1 { 92 } { 0 }
22763 \int_step_inline:nnn { 97 } { 122 }
22764 { \__cctab_gstore:Nnn #1 {##1} { 11 } }
22765 \__cctab_gstore:Nnn #1 { 127 } { 15 }
22766 }
22767 }
22768 \cs_generate_variant:Nn \cctab_new:N { c }
22769 \sys_if_engine luatex:TF
22770 {
22771 \cs_new_protected:Npn \__cctab_gset:n #1
22772 { \exp_args:Nf \__cctab_gset_aux:n { \int_eval:n {#1} } }
22773 \cs_new_protected:Npn \__cctab_gset_aux:n #1
22774 {
22775 \tex_savecatcodetable:D #1 \scan_stop:
22776 \int_compare:nNnTF { \tex_endlinechar:D } = { 13 }
22777 { \prop_gremove:Nn \g__cctab_endlinechar_prop {#1} }
22778 {
22779 \prop_gput:NnV \g__cctab_endlinechar_prop {#1}
22780 \tex_endlinechar:D
22781 }
22782 }
22783 }
22784 {
22785 \cs_new_protected:Npn \__cctab_gset:n #1
22786 {
22787 \int_step_inline:nn { 256 }
22788 {
22789 \__kernel_intarray_gset:Nnn #1 {##1}
22790 { \char_value_catcode:n { ##1 - 1 } }
22791 }
22792 \__kernel_intarray_gset:Nnn #1 { 257 }
22793 { \tex_endlinechar:D }
22794 }
22795 }
22796 \cs_new_protected:Npn \cctab_gset:Nn #1#2
22797 {
22798 \__cctab_chk_if_valid:NT #1
22799 {
22800 \group_begin:
22801 \cctab_select:N \c_initex_cctab

```



```

22802         #2 \scan_stop:                                22802
22803         \__cctab_gset:n {#1}                          22803
22804     \group_end:                                        22804
22805 }                                                       22805
22806 }                                                       22806
22807 \cs_generate_variant:Nn \cctab_gset:Nn { c }          22807
22808 \cs_new_protected:Npn \cctab_gsave_current:N #1       22808
22809 {                                                       22809
22810     \__cctab_chk_if_valid:NT #1                        22810
22811     { \__cctab_gset:n {#1} }                          22811
22812 }                                                       22812
22813 \cs_generate_variant:Nn \cctab_gsave_current:N { c }  22813
22814 \sys_if_engine luatex:T                               22814
22815 {                                                       22815
22816     \__cctab_new:N \g__cctab_internal_cctab           22816
22817     \cs_new:Npn \__cctab_internal_cctab_name:         22817
22818     {                                                   22818
22819         g__cctab_internal                             22819
22820         \tex_romannumeral:D \tex_currentgrouplevel:D  22820
22821         _cctab                                         22821
22822     }                                                  22822
22823 }                                                       22823
22824 \cs_new_protected:Npn \cctab_select:N #1              22824
22825 { \__cctab_chk_if_valid:NT #1 { \__cctab_select:N #1 } } 22825
22826 \cs_generate_variant:Nn \cctab_select:N { c }         22826
22827 \sys_if_engine luatex:TF                              22827
22828 {                                                       22828
22829     \cs_new_protected:Npn \__cctab_select:N #1         22829
22830     {                                                   22830
22831         \tex_catcodetable:D #1                        22831
22832         \prop_get:NVNTF \g__cctab_endlinechar_prop #1 \l__cctab_internal_a_tl 22832
22833         { \int_set:Nn \tex_endlinechar:D { \l__cctab_internal_a_tl } } 22833
22834         { \int_set:Nn \tex_endlinechar:D { 13 } }     22834
22835         \cs_if_exist:cF { \__cctab_internal_cctab_name: } 22835
22836         { \exp_args:Nc \__cctab_new:N { \__cctab_internal_cctab_name: } } 22836
22837         \exp_args:Nc \tex_savecatcodetable:D { \__cctab_internal_cctab_name: } 22837
22838         \exp_args:Nc \tex_catcodetable:D { \__cctab_internal_cctab_name: } 22838
22839     }                                                  22839
22840 }                                                       22840
22841 {                                                       22841
22842     \cs_new_protected:Npn \__cctab_select:N #1         22842
22843     {                                                   22843
22844         \int_step_inline:nn { 256 }                   22844
22845         {                                               22845
22846             \char_set_catcode:nn { ##1 - 1 }          22846
22847             { \__kernel_intarray_item:Nn #1 {##1} }   22847

```

```
22848     }
22849     \int_set:Nn \tex_endlinechar:D
22850     { \__kernel_intarray_item:Nn #1 { 257 } }
22851 }
22852 }
22853 \sys_if_engine luatex:TF
22854 {
22855   \cs_new_protected:Npn \__cctab_begin_aux:
22856   {
22857     \__cctab_new:N \g__cctab_next_cctab
22858     \tl_set:NV \l__cctab_internal_a_tl \g__cctab_next_cctab
22859     \cs_undefine:N \g__cctab_next_cctab
22860   }
22861 }
22862 {
22863   \cs_new_protected:Npn \__cctab_begin_aux:
22864   {
22865     \int_gincr:N \g__cctab_allocate_int
22866     \exp_args:Nc \__cctab_new:N
22867     { g__cctab_ \int_use:N \g__cctab_allocate_int _cctab }
22868     \exp_args:NNc \tl_set:Nn \l__cctab_internal_a_tl
22869     { g__cctab_ \int_use:N \g__cctab_allocate_int _cctab }
22870   }
22871 }
22872 \cs_new_protected:Npn \cctab_begin:N #1
22873 {
22874   \__cctab_chk_if_valid:NT #1
22875   {
22876     \seq_gpop:N NF \g__cctab_unused_seq \l__cctab_internal_a_tl
22877     { \__cctab_begin_aux: }
22878     \__cctab_chk_group_begin:e
22879     { \__cctab_nesting_number:N \l__cctab_internal_a_tl }
22880     \seq_gpush:N V \g__cctab_stack_seq \l__cctab_internal_a_tl
22881     \exp_args:N V \__cctab_gset:n \l__cctab_internal_a_tl
22882     \__cctab_select:N #1
22883   }
22884 }
22885 \cs_generate_variant:Nn \cctab_begin:N { c }
22886 \cs_new_protected:Npn \cctab_end:
22887 {
22888   \seq_gpop:N NTF \g__cctab_stack_seq \l__cctab_internal_a_tl
22889   {
22890     \seq_gpush:N V \g__cctab_unused_seq \l__cctab_internal_a_tl
22891     \exp_args:Ne \__cctab_chk_group_end:n
22892     { \__cctab_nesting_number:N \l__cctab_internal_a_tl }
22893     \__cctab_select:N \l__cctab_internal_a_tl
```

```

22894     }
22895     { \msg_error:nn { cctab } { extra-end } }
22896 }
22897 \cs_new_protected:Npn \__cctab_chk_group_begin:n #1
22898 {
22899     \seq_gpush:Ne \g__cctab_group_seq
22900     { \int_use:N \tex_currentgrouplevel:D }
22901     \cs_set_eq:cN { __cctab_group_ #1 _chk: } \prg_do_nothing:
22902 }
22903 \cs_generate_variant:Nn \__cctab_chk_group_begin:n { e }
22904 \cs_new_protected:Npn \__cctab_chk_group_end:n #1
22905 {
22906     \seq_gpop:NN \g__cctab_group_seq \l__cctab_internal_b_tl
22907     \bool_lazy_and:nnF
22908     {
22909         \int_compare_p:nNn
22910             { \tex_currentgrouplevel:D } = { \l__cctab_internal_b_tl }
22911     }
22912     { \cs_if_exist_p:c { __cctab_group_ #1 _chk: } }
22913     {
22914         \msg_error:nne { cctab } { group-mismatch }
22915         {
22916             \int_sign:n
22917                 { \tex_currentgrouplevel:D - \l__cctab_internal_b_tl }
22918         }
22919     }
22920     \cs_undefine:c { __cctab_group_ #1 _chk: }
22921 }
22922 \sys_if_engine luatex:TF
22923 { \cs_new:Npn \__cctab_nesting_number:N #1 {#1} }
22924 {
22925     \cs_new:Npn \__cctab_nesting_number:N #1
22926     {
22927         \exp_after:wN \exp_after:wN \exp_after:wN \__cctab_nesting_number:w
22928         \exp_after:wN \token_to_str:N #1
22929     }
22930     \use:e
22931     {
22932         \cs_new:Npn \exp_not:N \__cctab_nesting_number:w
22933             #1 \tl_to_str:n { g__cctab_ } #2 \tl_to_str:n { _cctab } {#2}
22934     }
22935 }
22936 \cs_if_exist:NT \hook_gput_code:nnn
22937 {
22938     \hook_gput_code:nnn { enddocument/end } { cctab }
22939     {

```

```

22940 \seq_if_empty:NF \g__cctab_stack_seq
22941 { \msg_error:nn { cctab } { missing-end } }
22942 }
22943 }
22944 \cs_new:Npn \cctab_item:Nn #1#2
22945 { \exp_args:Nf \__cctab_item:nN { \int_eval:n {#2} } #1 }
22946 \sys_if_engine luatex:TF
22947 {
22948 \cs_new:Npn \__cctab_item:nN #1#2
22949 { \lua_now:e { tex.print(-2, tex.getcatcode(\int_use:N #2, #1)) } }
22950 }
22951 {
22952 \cs_new:Npn \__cctab_item:nN #1#2
22953 {
22954 \int_compare:nNnTF {#1} < { 256 }
22955 { \intarray_item:Nn #2 { #1 + 1 } }
22956 { \char_value_catcode:n {#1} }
22957 }
22958 }
22959 \cs_generate_variant:Nn \cctab_item:Nn { c }
22960 \prg_new_eq_conditional:NNn \cctab_if_exist:N \cs_if_exist:N
22961 { TF , T , F , p }
22962 \prg_new_eq_conditional:NNn \cctab_if_exist:c \cs_if_exist:c
22963 { TF , T , F , p }
22964 \prg_new_protected_conditional:Npnn \__cctab_chk_if_valid:N #1
22965 { TF , T , F }
22966 {
22967 \cctab_if_exist:NTF #1
22968 {
22969 \__cctab_chk_if_valid_aux:NTF #1
22970 { \prg_return_true: }
22971 {
22972 \msg_error:nne { cctab } { invalid-cctab }
22973 { \token_to_str:N #1 }
22974 \prg_return_false:
22975 }
22976 }
22977 {
22978 \msg_error:nne { kernel } { command-not-defined }
22979 { \token_to_str:N #1 }
22980 \prg_return_false:
22981 }
22982 }
22983 \sys_if_engine luatex:TF
22984 {
22985 \cs_new_protected:Npn \__cctab_chk_if_valid_aux:NTF #1

```

```

22986 {
22987     \int_compare:nNnTF {#1-1} < { \e@alloc@ccodetable@count }
22988 }
22989 \cs_if_exist:NT \c_syst_catcodes_n
22990 {
22991     \cs_gset_protected:Npn \__cctab_chk_if_valid_aux:NTF #1
22992     {
22993         \int_compare:nTF { #1 <= \c_syst_catcodes_n }
22994     }
22995 }
22996 }
22997 {
22998     \cs_new_protected:Npn \__cctab_chk_if_valid_aux:NTF #1
22999     {
23000         \exp_args:Nf \str_if_in:nnTF
23001         { \cs_meaning:N #1 }
23002         { select~font~cmr10~at~ }
23003     }
23004 }
23005 \cs_new_protected:Npn \cctab_const:Nn #1#2
23006 {
23007     \__kernel_chk_if_free_cs:N #1
23008     \__cctab_new:N #1
23009     \group_begin:
23010         \cctab_select:N \c_initex_cctab
23011         #2 \scan_stop:
23012         \__cctab_gset:n {#1}
23013     \group_end:
23014 }
23015 \cs_generate_variant:Nn \cctab_const:Nn { c }
23016 \cctab_new:N \c_initex_cctab
23017 \cctab_const:Nn \c_other_cctab
23018 {
23019     \cctab_select:N \c_initex_cctab
23020     \int_set:Nn \tex_endlinechar:D { -1 }
23021     \int_step_inline:nnn { 0 } { 127 }
23022     { \char_set_catcode_other:n {#1} }
23023 }
23024 \cctab_const:Nn \c_str_cctab
23025 {
23026     \cctab_select:N \c_other_cctab
23027     \char_set_catcode_space:n { 32 }
23028 }
23029 \cs_if_exist:NTF \@expl@finalise@setup@@
23030 { \tl_gput_right:Nn \@expl@finalise@setup@@ }
23031 { \use:n }

```

```

23032 {
23033   \__cctab_new:N \c_code_cctab
23034   \group_begin:
23035     \int_set:Nn \tex_endlinechar:D { 32 }
23036     \char_set_catcode_invalid:n { 0 }
23037     \sys_if_engine_opentype:TF
23038       { \int_step_function:nnN { 31 } { 31 } \char_set_catcode_invalid:n }
23039       { \int_step_function:nnN { 31 } { 31 } \char_set_catcode_active:n }
23040     \int_step_function:nnN { 33 } { 64 } \char_set_catcode_other:n
23041     \int_step_function:nnN { 65 } { 90 } \char_set_catcode_letter:n
23042     \int_step_function:nnN { 91 } { 96 } \char_set_catcode_other:n
23043     \int_step_function:nnN { 97 } { 122 } \char_set_catcode_letter:n
23044     \char_set_catcode_ignore:n { 9 } % tab
23045     \char_set_catcode_other:n { 10 } % lf
23046     \char_set_catcode_active:n { 12 } % ff
23047     \char_set_catcode_end_line:n { 13 } % cr
23048     \char_set_catcode_ignore:n { 32 } % space
23049     \char_set_catcode_parameter:n { 35 } % hash
23050     \char_set_catcode_math_toggle:n { 36 } % dollar
23051     \char_set_catcode_comment:n { 37 } % percent
23052     \char_set_catcode_alignment:n { 38 } % ampersand
23053     \char_set_catcode_letter:n { 58 } % colon
23054     \char_set_catcode_escape:n { 92 } % backslash
23055     \char_set_catcode_math_superscript:n { 94 } % circumflex
23056     \char_set_catcode_letter:n { 95 } % underscore
23057     \char_set_catcode_group_begin:n { 123 } % left brace
23058     \char_set_catcode_other:n { 124 } % pipe
23059     \char_set_catcode_group_end:n { 125 } % right brace
23060     \char_set_catcode_space:n { 126 } % tilde
23061     \char_set_catcode_invalid:n { 127 } % ^^?
23062     \sys_if_engine_opentype:F
23063       { \int_step_function:nnN { 128 } { 255 } \char_set_catcode_active:n }
23064     \__cctab_gset:n { \c_code_cctab }
23065   \group_end:
23066   \cctab_const:Nn \c_document_cctab
23067   {
23068     \cctab_select:N \c_code_cctab
23069     \int_set:Nn \tex_endlinechar:D { 13 }
23070     \char_set_catcode_space:n { 9 }
23071     \char_set_catcode_space:n { 32 }
23072     \char_set_catcode_other:n { 58 }
23073     \char_set_catcode_math_subscript:n { 95 }
23074     \char_set_catcode_active:n { 126 }
23075   }
23076 }
23077 \cctab_new:N \g_tmpa_cctab

```

```

23078 \cctab_new:N \g_tmpb_cctab 23078
23079 \msg_new:nnnn { cctab } { stack-full } 23079
23080 { The~category~code~table~stack~is~exhausted. } 23080
23081 { 23081
23082 LaTeX~has~been~asked~to~switch~to~a~new~category~code~table,~ 23082
23083 but~there~is~no~more~space~to~do~this! 23083
23084 } 23084
23085 \msg_new:nnnn { cctab } { extra-end } 23085
23086 { Extra~\iow_char:N\\cctab_end:~ignored~\msg_line_context:. } 23086
23087 { 23087
23088 LaTeX~came~across~a~\iow_char:N\\cctab_end:~without~a~matching~ 23088
23089 \iow_char:N\\cctab_begin:N.~This~command~will~be~ignored. 23089
23090 } 23090
23091 \msg_new:nnnn { cctab } { missing-end } 23091
23092 { Missing~\iow_char:N\\cctab_end:~before~end~of~TeX~run. } 23092
23093 { 23093
23094 LaTeX~came~across~more~\iow_char:N\\cctab_begin:N~than~ 23094
23095 \iow_char:N\\cctab_end:. 23095
23096 } 23096
23097 \msg_new:nnnn { cctab } { invalid-cctab } 23097
23098 { Invalid~\iow_char:N\\catcode~table. } 23098
23099 { 23099
23100 You~can~only~switch~to~a~\iow_char:N\\catcode~table~that~is~ 23100
23101 initialized~using~\iow_char:N\\cctab_new:N~or~ 23101
23102 \iow_char:N\\cctab_const:Nn. 23102
23103 } 23103
23104 \msg_new:nnnn { cctab } { group-mismatch } 23104
23105 { 23105
23106 \iow_char:N\\cctab_end:~occurred~in~a~ 23106
23107 \int_case:nn {#1} 23107
23108 { 23108
23109 { 0 } { different~group } 23109
23110 { 1 } { higher~group~level } 23110
23111 { -1 } { lower~group~level } 23111
23112 } ~than~ 23112
23113 the~matching~\iow_char:N\\cctab_begin:N. 23113
23114 } 23114
23115 { 23115
23116 Catcode~tables~and~groups~must~be~properly~nested,~but~ 23116
23117 you~tried~to~interleave~them.~LaTeX~will~try~to~proceed,~ 23117
23118 but~results~may~be~unexpected. 23118
23119 } 23119
23120 \prop_gput:Nnn \g_msg_module_name_prop { cctab } { LaTeX } 23120
23121 \prop_gput:Nnn \g_msg_module_type_prop { cctab } { } 23121
23122 %% File l3sort.dtx 23122
23123 \cs_new_eq:NN \__sort_sep: \__kernel_int_sep: 23123

```



```

23124 \seq_new:N \g__sort_internal_seq 23124
23125 \tl_new:N \g__sort_internal_tl 23125
23126 \int_new:N \l__sort_length_int 23126
23127 \int_new:N \l__sort_min_int 23127
23128 \int_new:N \l__sort_top_int 23128
23129 \int_new:N \l__sort_max_int 23129
23130 \int_new:N \l__sort_true_max_int 23130
23131 \int_new:N \l__sort_block_int 23131
23132 \int_new:N \l__sort_begin_int 23132
23133 \int_new:N \l__sort_end_int 23133
23134 \int_new:N \l__sort_A_int 23134
23135 \int_new:N \l__sort_B_int 23135
23136 \int_new:N \l__sort_C_int 23136
23137 \scan_new:N \s__sort_mark 23137
23138 \scan_new:N \s__sort_stop 23138
23139 \cs_new_protected:Npn \__sort_shrink_range: 23139
23140 { 23140
23141 \int_set:Nn \l__sort_A_int 23141
23142 { \l__sort_true_max_int - \l__sort_min_int + 1 } 23142
23143 \int_set:Nn \l__sort_block_int { \c_max_register_int / 2 } 23143
23144 \__sort_shrink_range_loop: 23144
23145 \int_set:Nn \l__sort_max_int 23145
23146 { 23146
23147 \int_compare:nNnTF 23147
23148 { \l__sort_block_int * 3 / 2 } > \l__sort_A_int 23148
23149 { 23149
23150 \l__sort_min_int 23150
23151 + ( \l__sort_A_int - 1 ) / 2 23151
23152 + \l__sort_block_int / 4 23152
23153 - 1 23153
23154 } 23154
23155 { \l__sort_true_max_int - \l__sort_block_int / 2 } 23155
23156 } 23156
23157 } 23157
23158 \cs_new_protected:Npn \__sort_shrink_range_loop: 23158
23159 { 23159
23160 \if_int_compare:w \l__sort_A_int < \l__sort_block_int 23160
23161 \tex_divide:D \l__sort_block_int 2 \exp_stop_f: 23161
23162 \exp_after:wN \__sort_shrink_range_loop: 23162
23163 \fi: 23163
23164 } 23164
23165 \cs_new_protected:Npn \__sort_compute_range: 23165
23166 { 23166
23167 \int_set:Nn \l__sort_min_int { \tex_count:D 15 + 1 } 23167
23168 \int_set:Nn \l__sort_true_max_int { \c_max_register_int + 1 } 23168
23169 \__sort_shrink_range: 23169

```

```

23170 \if_meaning:w \loctoks \tex_undefined:D \else:
23171 \if_meaning:w \loctoks \scan_stop: \else:
23172 \__sort_redefine_compute_range:
23173 \__sort_compute_range:
23174 \fi:
23175 \fi:
23176 }
23177 \cs_new_protected:Npn \__sort_redefine_compute_range:
23178 {
23179 \cs_if_exist:cTF { ver@elocalloc.sty }
23180 {
23181 \cs_gset_protected:Npn \__sort_compute_range:
23182 {
23183 \int_set:Nn \l__sort_min_int { \tex_count:D 265 }
23184 \int_set_eq:NN \l__sort_true_max_int \e@alloc@top
23185 \__sort_shrink_range:
23186 }
23187 }
23188 {
23189 \cs_gset_protected:Npn \__sort_compute_range:
23190 {
23191 \int_set:Nn \l__sort_min_int { \tex_count:D 265 }
23192 \int_set:Nn \l__sort_true_max_int { \tex_count:D 275 }
23193 \__sort_shrink_range:
23194 }
23195 }
23196 }
23197 \cs_if_exist:NT \loctoks { \__sort_redefine_compute_range: }
23198 \tl_map_inline:nn { \lastallocatedtoks \c_syst_last_allocated_toks }
23199 {
23200 \cs_if_exist:NT #1
23201 {
23202 \cs_gset_protected:Npn \__sort_compute_range:
23203 {
23204 \int_set:Nn \l__sort_min_int { #1 + 1 }
23205 \int_set:Nn \l__sort_true_max_int { \c_max_register_int + 1 }
23206 \__sort_shrink_range:
23207 }
23208 }
23209 }
23210 \cs_new_protected:Npn \__sort_main:NNNn #1#2#3#4
23211 {
23212 \__sort_disable_toksdef:
23213 \__sort_compute_range:
23214 \int_set_eq:NN \l__sort_top_int \l__sort_min_int
23215 #1 #3

```

```

23216 {
23217     \if_int_compare:w \l__sort_top_int = \l__sort_max_int
23218         \__sort_too_long_error:NNw #2 #3
23219     \fi:
23220     \tex_toks:D \l__sort_top_int {##1}
23221     \int_incr:N \l__sort_top_int
23222 }
23223 \int_set:Nn \l__sort_length_int
23224 { \l__sort_top_int - \l__sort_min_int }
23225 \cs_set:Npn \__sort_compare:nn ##1 ##2 {#4}
23226 \int_set:Nn \l__sort_block_int { 1 }
23227 \__sort_level:
23228 }
23229 \cs_new_protected:Npn \tl_sort:Nn { \__sort_tl:NNn \tl_set_eq:NN }
23230 \cs_generate_variant:Nn \tl_sort:Nn { c }
23231 \cs_new_protected:Npn \tl_gsort:Nn { \__sort_tl:NNn \tl_gset_eq:NN }
23232 \cs_generate_variant:Nn \tl_gsort:Nn { c }
23233 \cs_new_protected:Npn \__sort_tl:NNn #1#2#3
23234 {
23235     \group_begin:
23236         \__sort_main:NNNn \tl_map_inline:Nn \tl_map_break:n #2 {#3}
23237         \__kernel_tl_gset:Nx \g__sort_internal_tl
23238         { \__sort_tl_toks:w \l__sort_min_int \__sort_sep: }
23239     \group_end:
23240     #1 #2 \g__sort_internal_tl
23241     \tl_gclear:N \g__sort_internal_tl
23242     \prg_break_point:
23243 }
23244 \cs_new:Npn \__sort_tl_toks:w #1 \__sort_sep:
23245 {
23246     \if_int_compare:w #1 < \l__sort_top_int
23247         { \tex_the:D \tex_toks:D #1 }
23248         \exp_after:wN \__sort_tl_toks:w
23249         \int_value:w \int_eval:n { #1 + 1 } \exp_after:wN \__sort_sep:
23250     \fi:
23251 }
23252 \cs_new_protected:Npn \seq_sort:Nn
23253 { \__sort_seq:NNNNn \seq_map_inline:Nn \seq_map_break:n \seq_set_eq:NN }
23254 \cs_generate_variant:Nn \seq_sort:Nn { c }
23255 \cs_new_protected:Npn \seq_gsort:Nn
23256 { \__sort_seq:NNNNn \seq_map_inline:Nn \seq_map_break:n \seq_gset_eq:NN }
23257 \cs_generate_variant:Nn \seq_gsort:Nn { c }
23258 \cs_new_protected:Npn \clist_sort:Nn
23259 {
23260     \__sort_seq:NNNNn \clist_map_inline:Nn \clist_map_break:n
23261     \clist_set_from_seq:NN

```

```

23262 }
23263 \cs_generate_variant:Nn \clist_sort:Nn { c }
23264 \cs_new_protected:Npn \clist_gsort:Nn
23265 {
23266     \__sort_seq:NNNNn \clist_map_inline:Nn \clist_map_break:n
23267     \clist_gset_from_seq:NN
23268 }
23269 \cs_generate_variant:Nn \clist_gsort:Nn { c }
23270 \cs_new_protected:Npn \__sort_seq:NNNNn #1#2#3#4#5
23271 {
23272     \group_begin:
23273     \__sort_main:NNNn #1 #2 #4 {#5}
23274     \seq_gclear:N \g__sort_internal_seq
23275     \int_step_inline:nnn
23276     \l__sort_min_int { \l__sort_top_int - 1 }
23277     {
23278         \seq_gput_right:Ne \g__sort_internal_seq
23279         { \tex_the:D \tex_toks:D ##1 }
23280     }
23281     \group_end:
23282     #3 #4 \g__sort_internal_seq
23283     \seq_gclear:N \g__sort_internal_seq
23284     \prg_break_point:
23285 }
23286 \cs_new_protected:Npn \__sort_level:
23287 {
23288     \if_int_compare:w \l__sort_block_int < \l__sort_length_int
23289     \l__sort_end_int \l__sort_min_int
23290     \__sort_merge_blocks:
23291     \tex_advance:D \l__sort_block_int \l__sort_block_int
23292     \exp_after:wN \__sort_level:
23293     \fi:
23294 }
23295 \cs_new_protected:Npn \__sort_merge_blocks:
23296 {
23297     \l__sort_begin_int \l__sort_end_int
23298     \tex_advance:D \l__sort_end_int \l__sort_block_int
23299     \if_int_compare:w \l__sort_end_int < \l__sort_top_int
23300     \l__sort_A_int \l__sort_end_int
23301     \tex_advance:D \l__sort_end_int \l__sort_block_int
23302     \if_int_compare:w \l__sort_end_int > \l__sort_top_int
23303     \l__sort_end_int \l__sort_top_int
23304     \fi:
23305     \l__sort_B_int \l__sort_A_int
23306     \l__sort_C_int \l__sort_top_int
23307     \__sort_copy_block:

```

```

23308      \int_decr:N \l__sort_A_int
23309      \int_decr:N \l__sort_B_int
23310      \int_decr:N \l__sort_C_int
23311      \exp_after:wN \__sort_merge_blocks_aux:
23312      \exp_after:wN \__sort_merge_blocks:
23313  \fi:
23314 }
23315 \cs_new_protected:Npn \__sort_copy_block:
23316 {
23317   \tex_toks:D \l__sort_C_int \tex_toks:D \l__sort_B_int
23318   \int_incr:N \l__sort_C_int
23319   \int_incr:N \l__sort_B_int
23320   \if_int_compare:w \l__sort_B_int = \l__sort_end_int
23321     \use_i:nn
23322   \fi:
23323   \__sort_copy_block:
23324 }
23325 \cs_new_protected:Npn \__sort_merge_blocks_aux:
23326 {
23327   \exp_after:wN \__sort_compare:nn \exp_after:wN
23328     { \tex_the:D \tex_toks:D \exp_after:wN \l__sort_A_int \exp_after:wN }
23329     \exp_after:wN { \tex_the:D \tex_toks:D \l__sort_C_int }
23330   \prg_do_nothing:
23331   \__sort_return_mark:w
23332   \__sort_return_mark:w
23333   \s__sort_mark
23334   \__sort_return_none_error:
23335 }
23336 \cs_new_protected:Npn \sort_return_same:
23337   #1 \__sort_return_mark:w #2 \s__sort_mark
23338 {
23339   #1
23340   #2
23341   \__sort_return_two_error:
23342   \__sort_return_mark:w
23343   \s__sort_mark
23344   \__sort_return_same:w
23345 }
23346 \cs_new_protected:Npn \sort_return_swapped:
23347   #1 \__sort_return_mark:w #2 \s__sort_mark
23348 {
23349   #1
23350   #2
23351   \__sort_return_two_error:
23352   \__sort_return_mark:w
23353   \s__sort_mark

```

```

23354 \__sort_return_swapped:w 23354
23355 } 23355
23356 \cs_new_protected:Npn \__sort_return_mark:w #1 \s__sort_mark { } 23356
23357 \cs_new_protected:Npn \__sort_return_none_error: 23357
23358 { 23358
23359 \msg_error:nnee { sort } { return-none } 23359
23360 { \tex_the:D \tex_toks:D \l__sort_A_int } 23360
23361 { \tex_the:D \tex_toks:D \l__sort_C_int } 23361
23362 \__sort_return_same:w \__sort_return_none_error: 23362
23363 } 23363
23364 \cs_new_protected:Npn \__sort_return_two_error: 23364
23365 { 23365
23366 \msg_error:nnee { sort } { return-two } 23366
23367 { \tex_the:D \tex_toks:D \l__sort_A_int } 23367
23368 { \tex_the:D \tex_toks:D \l__sort_C_int } 23368
23369 } 23369
23370 \cs_new_protected:Npn \__sort_return_same:w #1 \__sort_return_none_error: 23370
23371 { 23371
23372 \tex_toks:D \l__sort_B_int \tex_toks:D \l__sort_C_int 23372
23373 \int_decr:N \l__sort_B_int 23373
23374 \int_decr:N \l__sort_C_int 23374
23375 \if_int_compare:w \l__sort_C_int < \l__sort_top_int 23375
23376 \use_i:nn 23376
23377 \fi: 23377
23378 \__sort_merge_blocks_aux: 23378
23379 } 23379
23380 \cs_new_protected:Npn \__sort_return_swapped:w #1 \__sort_return_none_error: 23380
23381 { 23381
23382 \tex_toks:D \l__sort_B_int \tex_toks:D \l__sort_A_int 23382
23383 \int_decr:N \l__sort_B_int 23383
23384 \int_decr:N \l__sort_A_int 23384
23385 \if_int_compare:w \l__sort_A_int < \l__sort_begin_int 23385
23386 \__sort_merge_blocks_end: \use_i:nn 23386
23387 \fi: 23387
23388 \__sort_merge_blocks_aux: 23388
23389 } 23389
23390 \cs_new_protected:Npn \__sort_merge_blocks_end: 23390
23391 { 23391
23392 \tex_toks:D \l__sort_B_int \tex_toks:D \l__sort_C_int 23392
23393 \int_decr:N \l__sort_B_int 23393
23394 \int_decr:N \l__sort_C_int 23394
23395 \if_int_compare:w \l__sort_B_int < \l__sort_begin_int 23395
23396 \use_i:nn 23396
23397 \fi: 23397
23398 \__sort_merge_blocks_end: 23398
23399 } 23399

```

23400	\cs_new:Npn \tl_sort:nN #1#2	23400
23401	{	23401
23402	\exp_not:f	23402
23403	{	23403
23404	\tl_if_blank:nF {#1}	23404
23405	{	23405
23406	__sort_quick_prepare:Nnnn #2 { } { }	23406
23407	#1	23407
23408	{ \prg_break_point: __sort_quick_prepare_end:NNNnw }	23408
23409	\s__sort_stop	23409
23410	}	23410
23411	}	23411
23412	}	23412
23413	\cs_new:Npn __sort_quick_prepare:Nnnn #1#2#3#4	23413
23414	{	23414
23415	\prg_break: #4 \prg_break_point:	23415
23416	__sort_quick_prepare:Nnnn #1 { #2 #3 } { #1 {#4} }	23416
23417	}	23417
23418	\cs_new:Npn __sort_quick_prepare_end:NNNnw #1#2#3#4#5 \s__sort_stop	23418
23419	{	23419
23420	__sort_quick_split:NnNn #4 __sort_quick_end:nnTFNn { }	23420
23421	\s__sort_mark { __sort_quick_cleanup:w \exp_stop_f: }	23421
23422	\s__sort_mark \s__sort_stop	23422
23423	}	23423
23424	\cs_new:Npn __sort_quick_cleanup:w #1 \s__sort_mark \s__sort_stop {#1}	23424
23425	\cs_new:Npn __sort_quick_split:NnNn #1#2#3#4	23425
23426	{	23426
23427	#3 {#2} {#4} __sort_quick_only_ii:NnnnnNn	23427
23428	__sort_quick_only_i:NnnnnNn	23428
23429	__sort_quick_single_end:nnwnw	23429
23430	{ #3 {#4} } { } { } {#2}	23430
23431	}	23431
23432	\cs_new:Npn __sort_quick_only_i:NnnnnNn #1#2#3#4#5#6#7	23432
23433	{	23433
23434	#6 {#5} {#7} __sort_quick_split_ii:NnnnnNn	23434
23435	__sort_quick_only_i:NnnnnNn	23435
23436	__sort_quick_only_i_end:nnwnw	23436
23437	{ #6 {#7} } { #3 #2 } { } {#5}	23437
23438	}	23438
23439	\cs_new:Npn __sort_quick_only_ii:NnnnnNn #1#2#3#4#5#6#7	23439
23440	{	23440
23441	#6 {#5} {#7} __sort_quick_only_ii:NnnnnNn	23441
23442	__sort_quick_split_i:NnnnnNn	23442
23443	__sort_quick_only_ii_end:nnwnw	23443
23444	{ #6 {#7} } { } { #4 #2 } {#5}	23444
23445	}	23445


```
23446 \cs_new:Npn \__sort_quick_split_i:NnnnnNn #1#2#3#4#5#6#7 23446
23447 { 23447
23448     #6 {#5} {#7} \__sort_quick_split_ii:NnnnnNn 23448
23449     \__sort_quick_split_i:NnnnnNn 23449
23450     \__sort_quick_split_end:nnnwnw 23450
23451     { #6 {#7} } { #3 #2 } {#4} {#5} 23451
23452 } 23452
23453 \cs_new:Npn \__sort_quick_split_ii:NnnnnNn #1#2#3#4#5#6#7 23453
23454 { 23454
23455     #6 {#5} {#7} \__sort_quick_split_ii:NnnnnNn 23455
23456     \__sort_quick_split_i:NnnnnNn 23456
23457     \__sort_quick_split_end:nnnwnw 23457
23458     { #6 {#7} } {#3} { #4 #2 } {#5} 23458
23459 } 23459
23460 \cs_new:Npn \__sort_quick_end:nnTFNn #1#2#3#4#5#6 {#5} 23460
23461 \cs_new:Npn \__sort_quick_single_end:nnnwnw #1#2#3#4 \s__sort_mark #5#6 \s__sort_stop 23461
23462 { #5 {#3} #6 \s__sort_stop } 23462
23463 \cs_new:Npn \__sort_quick_only_i_end:nnnwnw #1#2#3#4 \s__sort_mark #5#6 \s__sort_stop 23463
23464 { 23464
23465     \__sort_quick_split:NnNn #1 23465
23466     \__sort_quick_end:nnTFNn { } \s__sort_mark {#5} 23466
23467     {#3} 23467
23468     #6 \s__sort_stop 23468
23469 } 23469
23470 \cs_new:Npn \__sort_quick_only_ii_end:nnnwnw #1#2#3#4 \s__sort_mark #5#6 \s__sort_stop 23470
23471 { 23471
23472     \__sort_quick_split:NnNn #2 23472
23473     \__sort_quick_end:nnTFNn { } \s__sort_mark { #5 {#3} } 23473
23474     #6 \s__sort_stop 23474
23475 } 23475
23476 \cs_new:Npn \__sort_quick_split_end:nnnwnw #1#2#3#4 \s__sort_mark #5#6 \s__sort_stop 23476
23477 { 23477
23478     \__sort_quick_split:NnNn #2 \__sort_quick_end:nnTFNn { } \s__sort_mark 23478
23479     { 23479
23480         \__sort_quick_split:NnNn #1 23480
23481         \__sort_quick_end:nnTFNn { } \s__sort_mark {#5} 23481
23482         {#3} 23482
23483     } 23483
23484     #6 \s__sort_stop 23484
23485 } 23485
23486 \cs_new_protected:Npn \__sort_error: 23486
23487 { 23487
23488     \cs_set_eq:NN \__sort_merge_blocks_aux: \prg_do_nothing: 23488
23489     \cs_set_eq:NN \__sort_merge_blocks: \prg_do_nothing: 23489
23490     \cs_set_protected:Npn \__sort_level: { \group_end: \prg_break: } 23490
23491 } 23491
```

```

23492 \cs_new_protected:Npn \__sort_disable_toksdef: 23492
23493 { \cs_set_eq:NN \toksdef \__sort_disabled_toksdef:n } 23493
23494 \cs_new_protected:Npn \__sort_disabled_toksdef:n #1 23494
23495 { 23495
23496 \msg_error:nne { sort } { toksdef } 23496
23497 { \token_to_str:N #1 } 23497
23498 \__sort_error: 23498
23499 \tex_toksdef:D #1 23499
23500 } 23500
23501 \msg_new:nnnn { sort } { toksdef } 23501
23502 { Allocation~of~\iow_char:N\\toks~registers~impossible~while~sorting. } 23502
23503 { 23503
23504 The~comparison~code~used~for~sorting~a~list~has~attempted~to~ 23504
23505 define~#1~as~a~new~\iow_char:N\\toks~register~using~ 23505
23506 \iow_char:N\\newtoks~ 23506
23507 or~a~similar~command.~The~list~will~not~be~sorted. 23507
23508 } 23508
23509 \cs_new_protected:Npn \__sort_too_long_error:NNw #1#2 \fi: 23509
23510 { 23510
23511 \fi: 23511
23512 \msg_error:nneee { sort } { too-large } 23512
23513 { \token_to_str:N #2 } 23513
23514 { \int_eval:n { \l__sort_true_max_int - \l__sort_min_int } } 23514
23515 { \int_eval:n { \l__sort_top_int - \l__sort_min_int } } 23515
23516 #1 \__sort_error: 23516
23517 } 23517
23518 \msg_new:nnnn { sort } { too-large } 23518
23519 { The~list~#1~is~too~long~to~be~sorted~by~TeX. } 23519
23520 { 23520
23521 TeX~has~#2~toks~registers~still~available:~ 23521
23522 this~only~allows~to~sort~with~up~to~#3~ 23522
23523 items.~The~list~will~not~be~sorted. 23523
23524 } 23524
23525 \msg_new:nnnn { sort } { return-none } 23525
23526 { The~comparison~code~did~not~return. } 23526
23527 { 23527
23528 When~sorting~a~list,~the~code~to~compare~items~#1~and~#2~ 23528
23529 did~not~call~ 23529
23530 \iow_char:N\\sort_return_same: ~nor~ 23530
23531 \iow_char:N\\sort_return_swapped: .~ 23531
23532 Exactly~one~of~these~should~be~called. 23532
23533 } 23533
23534 \msg_new:nnnn { sort } { return-two } 23534
23535 { The~comparison~code~returned~multiple~times. } 23535
23536 { 23536
23537 When~sorting~a~list,~the~code~to~compare~items~#1~and~#2~called~ 23537

```

```

23538 \iow_char:N\sort_return_same: ~or~
23539 \iow_char:N\sort_return_swapped: ~multiple~times.~
23540 Exactly~one~of~these~should~be~called.
23541 }
23542 \prop_gput:Nnn \g_msg_module_name_prop { sort } { LaTeX }
23543 \prop_gput:Nnn \g_msg_module_type_prop { sort } { }
23544 %% File: l3str-convert.dtx
23545 \cs_new_protected:Npn \__str_tmp:w { }
23546 \tl_new:N \l__str_internal_tl
23547 \tl_new:N \g__str_result_tl
23548 \int_const:Nn \c__str_replacement_char_int { "FFFD }
23549 \int_const:Nn \c__str_max_byte_int { 255 }
23550 \scan_new:N \s__str
23551 \quark_new:N \q__str_nil
23552 \prop_new:N \g__str_alias_prop
23553 \prop_gput:Nnn \g__str_alias_prop { latin1 } { iso88591 }
23554 \prop_gput:Nnn \g__str_alias_prop { latin2 } { iso88592 }
23555 \prop_gput:Nnn \g__str_alias_prop { latin3 } { iso88593 }
23556 \prop_gput:Nnn \g__str_alias_prop { latin4 } { iso88594 }
23557 \prop_gput:Nnn \g__str_alias_prop { latin5 } { iso88599 }
23558 \prop_gput:Nnn \g__str_alias_prop { latin6 } { iso885910 }
23559 \prop_gput:Nnn \g__str_alias_prop { latin7 } { iso885913 }
23560 \prop_gput:Nnn \g__str_alias_prop { latin8 } { iso885914 }
23561 \prop_gput:Nnn \g__str_alias_prop { latin9 } { iso885915 }
23562 \prop_gput:Nnn \g__str_alias_prop { latin10 } { iso885916 }
23563 \prop_gput:Nnn \g__str_alias_prop { utf16le } { utf16 }
23564 \prop_gput:Nnn \g__str_alias_prop { utf16be } { utf16 }
23565 \prop_gput:Nnn \g__str_alias_prop { utf32le } { utf32 }
23566 \prop_gput:Nnn \g__str_alias_prop { utf32be } { utf32 }
23567 \prop_gput:Nnn \g__str_alias_prop { hexadecimal } { hex }
23568 \bool_lazy_any:nTF
23569 {
23570 \sys_if_engine luatex_p:
23571 \sys_if_engine xetex_p:
23572 }
23573 {
23574 \prop_gput:Nnn \g__str_alias_prop { default } { }
23575 }
23576 {
23577 \prop_gput:Nnn \g__str_alias_prop { default } { utf8 }
23578 }
23579 \bool_new:N \g__str_error_bool
23580 \flag_new:N \l__str_byte_flag
23581 \flag_new:N \l__str_error_flag
23582 \prg_new_conditional:Npnn \__str_if_contains_char:Nn #1#2 { T , TF }
23583 {

```

```
23584 \exp_after:wN \__str_if_contains_char_aux:nn \exp_after:wN {#1} {#2} 23584
23585 { \prg_break:n { ? \fi: } } 23585
23586 \prg_break_point: 23586
23587 \prg_return_false: 23587
23588 } 23588
23589 \cs_new:Npn \__str_if_contains_char_aux:nn #1#2 23589
23590 { \__str_if_contains_char_auxi:nN {#2} #1 } 23590
23591 \prg_new_conditional:Npnn \__str_if_contains_char:nn #1#2 { TF } 23591
23592 { 23592
23593 \__str_if_contains_char_auxi:nN {#2} #1 { \prg_break:n { ? \fi: } } 23593
23594 \prg_break_point: 23594
23595 \prg_return_false: 23595
23596 } 23596
23597 \cs_new:Npn \__str_if_contains_char_auxi:nN #1#2 23597
23598 { 23598
23599 \if_charcode:w #1 #2 23599
23600 \exp_after:wN \__str_if_contains_char_true: 23600
23601 \fi: 23601
23602 \__str_if_contains_char_auxi:nN {#1} 23602
23603 } 23603
23604 \cs_new:Npn \__str_if_contains_char_true: 23604
23605 { \prg_break:n { \prg_return_true: \use_none:n } } 23605
23606 \prg_new_conditional:Npnn \__str_octal_use:N #1 { TF } 23606
23607 { 23607
23608 \if_int_compare:w 1 < '1 \token_to_str:N #1 \exp_stop_f: 23608
23609 #1 \prg_return_true: 23609
23610 \else: 23610
23611 \prg_return_false: 23611
23612 \fi: 23612
23613 } 23613
23614 \prg_new_conditional:Npnn \__str_hexadecimal_use:N #1 { TF } 23614
23615 { 23615
23616 \if_int_compare:w 1 < "1 \token_to_str:N #1 \exp_stop_f: 23616
23617 #1 \prg_return_true: 23617
23618 \else: 23618
23619 \if_case:w \int_eval:n { \exp_after:wN ` \token_to_str:N #1 - `a } 23619
23620 A 23620
23621 \or: B 23621
23622 \or: C 23622
23623 \or: D 23623
23624 \or: E 23624
23625 \or: F 23625
23626 \else: 23626
23627 \prg_return_false: 23627
23628 \exp_after:wN \use_none:n 23628
23629 \fi: 23629
```

23630	\prg_return_true:	23630
23631	\fi:	23631
23632	}	23632
23633	\group_begin:	23633
23634	__kernel_tl_set:Nx \l__str_internal_tl { \tl_to_str:n { 0123456789ABCDEF } }	23634
23635	\tl_map_inline:Nn \l__str_internal_tl	23635
23636	{	23636
23637	\tl_map_inline:Nn \l__str_internal_tl	23637
23638	{	23638
23639	\tl_const:ce { c__str_byte_ \int_eval:n {"#1##1} _tl }	23639
23640	{ \char_generate:nn { "#1##1 } { 12 } #1 ##1 }	23640
23641	}	23641
23642	}	23642
23643	\group_end:	23643
23644	\tl_const:cn { c__str_byte_-1_tl } { { } \use_none:n { } }	23644
23645	\cs_new:Npn __str_output_byte:n #1	23645
23646	{ __str_output_byte:w #1 __str_output_end: }	23646
23647	\cs_new:Npn __str_output_byte:w	23647
23648	{	23648
23649	\exp_after:wN \exp_after:wN	23649
23650	\exp_after:wN \use_i:nnn	23650
23651	\cs:w c__str_byte_ \int_eval:w	23651
23652	}	23652
23653	\cs_new:Npn __str_output_hexadecimal:n #1	23653
23654	{	23654
23655	\exp_after:wN \exp_after:wN	23655
23656	\exp_after:wN \use_none:n	23656
23657	\cs:w c__str_byte_ \int_eval:n {#1} _tl \cs_end:	23657
23658	}	23658
23659	\cs_new:Npn __str_output_end:	23659
23660	{ \scan_stop: _tl \cs_end: }	23660
23661	\cs_new:Npn __str_output_byte_pair_be:n #1	23661
23662	{	23662
23663	\exp_args:Nf __str_output_byte_pair:nnN	23663
23664	{ \int_div_truncate:nn { #1 } { "100 } } {#1} \use:nn	23664
23665	}	23665
23666	\cs_new:Npn __str_output_byte_pair_le:n #1	23666
23667	{	23667
23668	\exp_args:Nf __str_output_byte_pair:nnN	23668
23669	{ \int_div_truncate:nn { #1 } { "100 } } {#1} \use_ii_i:nn	23669
23670	}	23670
23671	\cs_new:Npn __str_output_byte_pair:nnN #1#2#3	23671
23672	{	23672
23673	#3	23673
23674	{ __str_output_byte:n { #1 } }	23674
23675	{ __str_output_byte:n { #2 - #1 * "100 } }	23675

```
23676 }
23677 \cs_new_protected:Npn \__str_convert_gmap:N #1
23678 {
23679   \__kernel_tl_gset:Nx \g__str_result_tl
23680   {
23681     \exp_after:wN \__str_convert_gmap_loop:NN
23682     \exp_after:wN #1
23683     \g__str_result_tl { ? \prg_break: }
23684     \prg_break_point:
23685   }
23686 }
23687 \cs_new:Npn \__str_convert_gmap_loop:NN #1#2
23688 {
23689   \use_none:n #2
23690   #1#2
23691   \__str_convert_gmap_loop:NN #1
23692 }
23693 \cs_new_protected:Npn \__str_convert_gmap_internal:N #1
23694 {
23695   \__kernel_tl_gset:Nx \g__str_result_tl
23696   {
23697     \exp_after:wN \__str_convert_gmap_internal_loop:Nww
23698     \exp_after:wN #1
23699     \g__str_result_tl \s__str \s__str_stop \prg_break: \s__str
23700     \prg_break_point:
23701   }
23702 }
23703 \cs_new:Npn \__str_convert_gmap_internal_loop:Nww #1 #2 \s__str #3 \s__str
23704 {
23705   \__str_use_none_delimit_by_s_stop:w #3 \s__str_stop
23706   #1 {#3}
23707   \__str_convert_gmap_internal_loop:Nww #1
23708 }
23709 \cs_new_protected:Npn \__str_if_flag_error:Nne #1
23710 {
23711   \flag_if_raised:NTF #1
23712   { \msg_error:nne { str } }
23713   { \use_none:nn }
23714 }
23715 \cs_new_protected:Npn \__str_if_flag_no_error:Nne #1#2#3
23716 { \flag_if_raised:NT #1 { \bool_gset_true:N \g__str_error_bool } }
23717 \cs_new:Npn \__str_if_flag_times:NT #1#2
23718 { \flag_if_raised:NT #1 { #2~(x \flag_height:N #1 ) } }
23719 \cs_new_protected:Npn \str_set_convert:Nnnn
23720 { \__str_convert:nNNnnn { } \tl_set_eq:NN }
23721 \cs_new_protected:Npn \str_gset_convert:Nnnn
```

```
23722 { \__str_convert:nNNnnn { } \tl_gset_eq:NN } 23722
23723 \prg_new_protected_conditional:Npnn 23723
23724 \str_set_convert:Nnnn #1#2#3#4 { T , F , TF } 23724
23725 { 23725
23726 \bool_gset_false:N \g__str_error_bool 23726
23727 \__str_convert:nNNnnn 23727
23728 { \cs_set_eq:NN \__str_if_flag_error:Nne \__str_if_flag_no_error:Nne } 23728
23729 \tl_set_eq:NN #1 {#2} {#3} {#4} 23729
23730 \bool_if:NTF \g__str_error_bool \prg_return_false: \prg_return_true: 23730
23731 } 23731
23732 \prg_new_protected_conditional:Npnn 23732
23733 \str_gset_convert:Nnnn #1#2#3#4 { T , F , TF } 23733
23734 { 23734
23735 \bool_gset_false:N \g__str_error_bool 23735
23736 \__str_convert:nNNnnn 23736
23737 { \cs_set_eq:NN \__str_if_flag_error:Nne \__str_if_flag_no_error:Nne } 23737
23738 \tl_gset_eq:NN #1 {#2} {#3} {#4} 23738
23739 \bool_if:NTF \g__str_error_bool \prg_return_false: \prg_return_true: 23739
23740 } 23740
23741 \cs_new_protected:Npn \__str_convert:nNNnnn #1#2#3#4#5#6 23741
23742 { 23742
23743 \group_begin: 23743
23744 #1 23744
23745 \__kernel_tl_gset:Nx \g__str_result_tl { \__kernel_str_to_other_fast:n {#4} } 23745
23746 \exp_after:wN \__str_convert:wwwnn 23746
23747 \tl_to_str:n {#5} /// \s__str_stop 23747
23748 { decode } { unescape } 23748
23749 \prg_do_nothing: 23749
23750 \__str_convert_decode_: 23750
23751 \exp_after:wN \__str_convert:wwwnn 23751
23752 \tl_to_str:n {#6} /// \s__str_stop 23752
23753 { encode } { escape } 23753
23754 \use_i_i:nn 23754
23755 \__str_convert_encode_: 23755
23756 \__kernel_tl_gset:Nx \g__str_result_tl 23756
23757 { \tl_to_str:V \g__str_result_tl } 23757
23758 \group_end: 23758
23759 #2 #3 \g__str_result_tl 23759
23760 } 23760
23761 \cs_new_protected:Npn \__str_convert:wwwnn 23761
23762 #1 / #2 // #3 \s__str_stop #4#5 23762
23763 { 23763
23764 \__str_convert:nnn {enc} {#4} {#1} 23764
23765 \__str_convert:nnn {esc} {#5} {#2} 23765
23766 \exp_args:Ncc \__str_convert:NNnNN 23766
23767 { __str_convert #4 #1: } { __str_convert #5 #2: } {#2} 23767
```



```

23768 } 23768
23769 \cs_new_protected:Npn \__str_convert:NNnNN #1#2#3#4#5 23769
23770 { 23770
23771 \if_meaning:w #1 #5 23771
23772 \tl_if_empty:nF {#3} 23772
23773 { \msg_error:nne { str } { native-escaping } {#3} } 23773
23774 #1 23774
23775 \else: 23775
23776 #4 #2 #1 23776
23777 \fi: 23777
23778 } 23778
23779 \cs_new_protected:Npn \__str_convert:nnn #1#2#3 23779
23780 { 23780
23781 \cs_if_exist:cF { __str_convert_#2_#3: } 23781
23782 { 23782
23783 \exp_args:Ne \__str_convert:nnnn 23783
23784 { \__str_convert_lowercase_alphanum:n {#3} } 23784
23785 {#1} {#2} {#3} 23785
23786 } 23786
23787 } 23787
23788 \cs_new_protected:Npn \__str_convert:nnnn #1#2#3#4 23788
23789 { 23789
23790 \cs_if_exist:cF { __str_convert_#3_#1: } 23790
23791 { 23791
23792 \prop_get:NnNF \g__str_alias_prop {#1} \l__str_internal_tl 23792
23793 { \tl_set:Nn \l__str_internal_tl {#1} } 23793
23794 \cs_if_exist:cF { __str_convert_#3_ \l__str_internal_tl : } 23794
23795 { 23795
23796 \file_if_exist:nTF { l3str-#2- \l__str_internal_tl .def } 23796
23797 { 23797
23798 \group_begin: 23798
23799 \cctab_select:N \c_code_cctab 23799
23800 \file_input:n { l3str-#2- \l__str_internal_tl .def } 23800
23801 \group_end: 23801
23802 } 23802
23803 { 23803
23804 \tl_clear:N \l__str_internal_tl 23804
23805 \msg_error:nnee { str } { unknown-#2 } {#4} {#1} 23805
23806 } 23806
23807 } 23807
23808 \cs_if_exist:cF { __str_convert_#3_#1: } 23808
23809 { 23809
23810 \cs_gset_eq:cc { __str_convert_#3_#1: } 23810
23811 { __str_convert_#3_ \l__str_internal_tl : } 23811
23812 } 23812
23813 } 23813

```

23814	\cs_gset_eq:cc {__str_convert_#3_#4: } {__str_convert_#3_#1: }	23814
23815	}	23815
23816	\cs_new:Npn __str_convert_lowercase_alphanum:n #1	23816
23817	{	23817
23818	\exp_after:wN __str_convert_lowercase_alphanum_loop:N	23818
23819	\tl_to_str:n {#1} { ? \prg_break: }	23819
23820	\prg_break_point:	23820
23821	}	23821
23822	\cs_new:Npn __str_convert_lowercase_alphanum_loop:N #1	23822
23823	{	23823
23824	\use_none:n #1	23824
23825	\if_int_compare:w `#1 > `Z \exp_stop_f:	23825
23826	\if_int_compare:w `#1 > `z \exp_stop_f: \else:	23826
23827	\if_int_compare:w `#1 < `a \exp_stop_f: \else:	23827
23828	#1	23828
23829	\fi:	23829
23830	\fi:	23830
23831	\else:	23831
23832	\if_int_compare:w `#1 < `A \exp_stop_f:	23832
23833	\if_int_compare:w 1 < 1#1 \exp_stop_f:	23833
23834	#1	23834
23835	\fi:	23835
23836	\else:	23836
23837	__str_output_byte:n { `#1 + `a - `A }	23837
23838	\fi:	23838
23839	\fi:	23839
23840	__str_convert_lowercase_alphanum_loop:N	23840
23841	}	23841
23842	\bool_lazy_any:nTF	23842
23843	{	23843
23844	\sys_if_engine luatex_p:	23844
23845	\sys_if_engine xetex_p:	23845
23846	}	23846
23847	{	23847
23848	\cs_new:Npn __str_filter_bytes:n #1	23848
23849	{	23849
23850	__str_filter_bytes_aux:N #1	23850
23851	{ ? \prg_break: }	23851
23852	\prg_break_point:	23852
23853	}	23853
23854	\cs_new:Npn __str_filter_bytes_aux:N #1	23854
23855	{	23855
23856	\use_none:n #1	23856
23857	\if_int_compare:w `#1 < 256 \exp_stop_f:	23857
23858	#1	23858
23859	\else:	23859

23860	\flag_raise:N \l__str_byte_flag	23860
23861	\fi:	23861
23862	__str_filter_bytes_aux:N	23862
23863	}	23863
23864	}	23864
23865	{ \cs_new_eq:NN __str_filter_bytes:n \use:n }	23865
23866	\bool_lazy_any:nTF	23866
23867	{	23867
23868	\sys_if_engine luatex_p:	23868
23869	\sys_if_engine xetex_p:	23869
23870	}	23870
23871	{	23871
23872	\cs_new_protected:Npn __str_convert_unescape_:	23872
23873	{	23873
23874	\flag_clear:N \l__str_byte_flag	23874
23875	__kernel_tl_gset:Nx \g__str_result_tl	23875
23876	{ \exp_args:No __str_filter_bytes:n \g__str_result_tl }	23876
23877	__str_if_flag_error:Nne \l__str_byte_flag { non-byte } { bytes }	23877
23878	}	23878
23879	}	23879
23880	{ \cs_new_protected:Npn __str_convert_unescape_: { } }	23880
23881	\cs_new_eq:NN __str_convert_unescape_bytes: __str_convert_unescape_:	23881
23882	\cs_new_protected:Npn __str_convert_escape_: { }	23882
23883	\cs_new_eq:NN __str_convert_escape_bytes: __str_convert_escape_:	23883
23884	\cs_new_protected:Npn __str_convert_decode_:	23884
23885	{ __str_convert_gmap:N __str_decode_native_char:N }	23885
23886	\cs_new:Npn __str_decode_native_char:N #1	23886
23887	{ #1 \s_str \int_value:w `#1 \s_str }	23887
23888	\bool_lazy_any:nTF	23888
23889	{	23889
23890	\sys_if_engine luatex_p:	23890
23891	\sys_if_engine xetex_p:	23891
23892	}	23892
23893	{	23893
23894	\cs_new_protected:Npn __str_convert_encode_:	23894
23895	{ __str_convert_gmap_internal:N __str_encode_native_char:n }	23895
23896	\cs_new:Npn __str_encode_native_char:n #1	23896
23897	{ \char_generate:nn {#1} {12} }	23897
23898	}	23898
23899	{	23899
23900	\cs_new_protected:Npn __str_convert_encode_:	23900
23901	{	23901
23902	\flag_clear:N \l__str_error_flag	23902
23903	__str_convert_gmap_internal:N __str_encode_native_char:n	23903
23904	__str_if_flag_error:Nne \l__str_error_flag	23904
23905	{ native-overflow } { }	23905

```

23906 }
23907 \cs_new:Npn \__str_encode_native_char:n #1
23908 {
23909     \if_int_compare:w #1 > \c__str_max_byte_int
23910         \flag_raise:N \l__str_error_flag
23911         ?
23912     \else:
23913         \char_generate:nn {#1} {12}
23914     \fi:
23915 }
23916 \msg_new:nnnn { str } { native-overflow }
23917 { Character~code~too~large~for~this~engine. }
23918 {
23919     This~engine~only~support~8-bit~characters:~
23920     valid~character~codes~are~in~the~range~[0,255].~
23921     To~manipulate~arbitrary~Unicode,~use~LuaTeX~or~XeTeX.
23922 }
23923 }
23924 \cs_new_protected:Npn \__str_convert_decode_clist:
23925 {
23926     \clist_gset:No \g__str_result_tl \g__str_result_tl
23927     \__kernel_tl_gset:Nx \g__str_result_tl
23928     {
23929         \exp_args:No \clist_map_function:nN
23930         \g__str_result_tl \__str_decode_clist_char:n
23931     }
23932 }
23933 \cs_new:Npn \__str_decode_clist_char:n #1
23934 { #1 \s__str \int_eval:n {#1} \s__str }
23935 \cs_new_protected:Npn \__str_convert_encode_clist:
23936 {
23937     \__str_convert_gmap_internal:N \__str_encode_clist_char:n
23938     \__kernel_tl_gset:Nx \g__str_result_tl { \tl_tail:N \g__str_result_tl }
23939 }
23940 \cs_new:Npn \__str_encode_clist_char:n #1 { , #1 }
23941 \cs_new_protected:Npn \__str_declare_eight_bit_encoding:nnnn #1
23942 {
23943     \tl_set:Nn \l__str_internal_tl {#1}
23944     \cs_new_protected:cpn { __str_convert_decode_#1: }
23945     { \__str_convert_decode_eight_bit:n {#1} }
23946     \cs_new_protected:cpn { __str_convert_encode_#1: }
23947     { \__str_convert_encode_eight_bit:n {#1} }
23948     \exp_args:Ncc \__str_declare_eight_bit_aux:NNnnn
23949     { g__str_decode_#1_intarray } { g__str_encode_#1_intarray }
23950 }
23951 \cs_new_protected:Npn \__str_declare_eight_bit_aux:NNnnn #1#2#3#4#5

```

```

23952 {
23953     \intarray_new:Nn #1 { 256 }
23954     \int_step_inline:nnn { 0 } { 255 }
23955     { \intarray_gset:Nnn #1 { 1 + ##1 } {##1} }
23956     \__str_declare_eight_bit_loop:Nnn #1
23957     #4 { \s__str_stop \prg_break: } { }
23958     \prg_break_point:
23959     \__str_declare_eight_bit_loop:Nn #1
23960     #5 { \s__str_stop \prg_break: }
23961     \prg_break_point:
23962     \intarray_new:Nn #2 {#3}
23963     \int_step_inline:nnn { 0 } { 255 }
23964     {
23965         \int_compare:nNnF { \intarray_item:Nn #1 { 1 + ##1 } } = { -1 }
23966         {
23967             \intarray_gset:Nnn #2
23968             {
23969                 1 +
23970                 \int_mod:nn { \intarray_item:Nn #1 { 1 + ##1 } }
23971                 { \intarray_count:N #2 }
23972             }
23973             {##1}
23974         }
23975     }
23976 }
23977 \cs_new_protected:Npn \__str_declare_eight_bit_loop:Nnn #1#2#3
23978 {
23979     \__str_use_none_delimit_by_s_stop:w #2 \s__str_stop
23980     \intarray_gset:Nnn #1 { 1 + "#2 } { "#3 }
23981     \__str_declare_eight_bit_loop:Nnn #1
23982 }
23983 \cs_new_protected:Npn \__str_declare_eight_bit_loop:Nn #1#2
23984 {
23985     \__str_use_none_delimit_by_s_stop:w #2 \s__str_stop
23986     \intarray_gset:Nnn #1 { 1 + "#2 } { -1 }
23987     \__str_declare_eight_bit_loop:Nn #1
23988 }
23989 \cs_new_protected:Npn \__str_convert_decode_eight_bit:n #1
23990 {
23991     \cs_set:Npe \__str_tmp:w
23992     {
23993         \exp_not:N \__str_decode_eight_bit_aux:Nn
23994         \exp_not:c { g__str_decode_#1_intarray }
23995     }
23996     \flag_clear:N \l__str_error_flag
23997     \__str_convert_gmap:N \__str_tmp:w

```

```
23998 \__str_if_flag_error:Nne \l__str_error_flag { decode-8-bit } {#1} 23998
23999 } 23999
24000 \cs_new:Npn \__str_decode_eight_bit_aux:Nn #1#2 24000
24001 { 24001
24002 #2 \s__str 24002
24003 \exp_args:Nf \__str_decode_eight_bit_aux:n 24003
24004 { \intarray_item:Nn #1 { 1 + `#2 } } 24004
24005 \s__str 24005
24006 } 24006
24007 \cs_new:Npn \__str_decode_eight_bit_aux:n #1 24007
24008 { 24008
24009 \if_int_compare:w #1 < \c_zero_int 24009
24010 \flag_raise:N \l__str_error_flag 24010
24011 \int_value:w \c__str_replacement_char_int 24011
24012 \else: 24012
24013 #1 24013
24014 \fi: 24014
24015 } 24015
24016 \int_new:N \l__str_modulo_int 24016
24017 \cs_new_protected:Npn \__str_convert_encode_eight_bit:n #1 24017
24018 { 24018
24019 \cs_set:Npe \__str_tmp:w 24019
24020 { 24020
24021 \exp_not:N \__str_encode_eight_bit_aux:NNn 24021
24022 \exp_not:c { g__str_encode_#1_intarray } 24022
24023 \exp_not:c { g__str_decode_#1_intarray } 24023
24024 } 24024
24025 \flag_clear:N \l__str_error_flag 24025
24026 \__str_convert_gmap_internal:N \__str_tmp:w 24026
24027 \__str_if_flag_error:Nne \l__str_error_flag { encode-8-bit } {#1} 24027
24028 } 24028
24029 \cs_new:Npn \__str_encode_eight_bit_aux:NNn #1#2#3 24029
24030 { 24030
24031 \exp_args:Nf \__str_encode_eight_bit_aux:nnN 24031
24032 { 24032
24033 \intarray_item:Nn #1 24033
24034 { 1 + \int_mod:nn {#3} { \intarray_count:N #1 } } 24034
24035 } 24035
24036 {#3} 24036
24037 #2 24037
24038 } 24038
24039 \cs_new:Npn \__str_encode_eight_bit_aux:nnN #1#2#3 24039
24040 { 24040
24041 \int_compare:nnnTF { \intarray_item:Nn #3 { 1 + #1 } } = {#2} 24041
24042 { \__str_output_byte:n {#1} } 24042
24043 { \flag_raise:N \l__str_error_flag } 24043
```

```

24044 }
24045 \msg_new:nnn { str } { unknown-esc }
24046 { Escaping~scheme~'#1'~(filtered:~'#2')~unknown. }
24047 \msg_new:nnn { str } { unknown-enc }
24048 { Encoding~scheme~'#1'~(filtered:~'#2')~unknown. }
24049 \msg_new:nnnn { str } { native-escaping }
24050 { The~'native'~encoding~scheme~does~not~support~any~escaping. }
24051 {
24052     Since~native~strings~do~not~consist~in~bytes,~
24053     none~of~the~escaping~methods~make~sense.~
24054     The~specified~escaping,~'#1',~will~be~ignored.
24055 }
24056 \msg_new:nnn { str } { file-not-found }
24057 { File~'l3str-#1.def'~not~found. }
24058 \bool_lazy_any:nT
24059 {
24060     \sys_if_engine luatex_p:
24061     \sys_if_engine xetex_p:
24062 }
24063 {
24064     \msg_new:nnnn { str } { non-byte }
24065     { String~invalid~in~escaping~'#1':~it~may~only~contain~bytes. }
24066     {
24067         Some~characters~in~the~string~you~asked~to~convert~are~not~
24068         8-bit~characters.~Perhaps~the~string~is~a~'native'~Unicode~string?~
24069         If~it~is,~try~using\\
24070         \\
24071         \iow_indent:n
24072         {
24073             \iow_char:N\\str_set_convert:Nnnn \\
24074             \_\_<str~var>~\{~<string>~\}~\{~native~\}~\{~<target~encoding>~\}
24075         }
24076     }
24077 }
24078 \msg_new:nnnn { str } { decode-8-bit }
24079 { Invalid~string~in~encoding~'#1'. }
24080 {
24081     LaTeX~came~across~a~byte~which~is~not~defined~to~represent~
24082     any~character~in~the~encoding~'#1'.
24083 }
24084 \msg_new:nnnn { str } { encode-8-bit }
24085 { Unicode~string~cannot~be~converted~to~encoding~'#1'. }
24086 {
24087     The~encoding~'#1'~only~contains~a~subset~of~all~Unicode~characters.~
24088     LaTeX~was~asked~to~convert~a~string~to~that~encoding,~but~that~
24089     string~contains~a~character~that~'#1'~does~not~support.

```



```

24090 }
24091 \cs_new_protected:Npn \__str_convert_unescape_hex:
24092 {
24093   \group_begin:
24094     \flag_clear:N \l__str_error_flag
24095     \int_set:Nn \tex_escapechar:D { 92 }
24096     \__kernel_tl_gset:Nx \g__str_result_tl
24097     {
24098       \__str_output_byte:w "
24099       \exp_last_unbraced:Nf \__str_unescape_hex_auxi:N
24100         { \tl_to_str:N \g__str_result_tl }
24101       0 { ? 0 - 1 \prg_break: }
24102       \prg_break_point:
24103       \__str_output_end:
24104     }
24105     \__str_if_flag_error:Nne \l__str_error_flag { unescape-hex } { }
24106   \group_end:
24107 }
24108 \cs_new:Npn \__str_unescape_hex_auxi:N #1
24109 {
24110   \use_none:n #1
24111   \__str_hexadecimal_use:NTF #1
24112   { \__str_unescape_hex_auxii:N }
24113   {
24114     \flag_raise:N \l__str_error_flag
24115     \__str_unescape_hex_auxi:N
24116   }
24117 }
24118 \cs_new:Npn \__str_unescape_hex_auxii:N #1
24119 {
24120   \use_none:n #1
24121   \__str_hexadecimal_use:NTF #1
24122   {
24123     \__str_output_end:
24124     \__str_output_byte:w " \__str_unescape_hex_auxi:N
24125   }
24126   {
24127     \flag_raise:N \l__str_error_flag
24128     \__str_unescape_hex_auxii:N
24129   }
24130 }
24131 \msg_new:nnnn { str } { unescape-hex }
24132 { String~invalid~in~escaping~'hex':~only~hexadecimal~digits~allowed. }
24133 {
24134   Some~characters~in~the~string~you~asked~to~convert~are~not~
24135   hexadecimal~digits~(0-9,~A-F,~a-f)~nor~spaces.

```

```

24136 }
24137 \cs_set_protected:Npn \__str_tmp:w #1#2#3
24138 {
24139   \cs_new_protected:cpn { __str_convert_unescape_#2: }
24140   {
24141     \group_begin:
24142       \flag_clear:N \l__str_byte_flag
24143       \flag_clear:N \l__str_error_flag
24144       \int_set:Nn \tex_escapechar:D { 92 }
24145       \__kernel_tl_gset:Nx \g__str_result_tl
24146       {
24147         \exp_after:wN #3 \g__str_result_tl
24148         #1 ? { ? \prg_break: }
24149         \prg_break_point:
24150       }
24151       \__str_if_flag_error:Nne \l__str_byte_flag { non-byte } { #2 }
24152       \__str_if_flag_error:Nne \l__str_error_flag { unescape-#2 } { }
24153     \group_end:
24154   }
24155   \cs_new:Npn #3 ##1#1##2##3
24156   {
24157     \__str_filter_bytes:n {##1}
24158     \use_none:n ##3
24159     \__str_output_byte:w "
24160     \__str_hexadecimal_use:NTF ##2
24161     {
24162       \__str_hexadecimal_use:NTF ##3
24163       { }
24164       {
24165         \flag_raise:N \l__str_error_flag
24166         * 0 + `#1 \use_i:nn
24167       }
24168     }
24169     {
24170       \flag_raise:N \l__str_error_flag
24171       0 + `#1 \use_i:nn
24172     }
24173     \__str_output_end:
24174     \use_i:nnn #3 ##2##3
24175   }
24176   \msg_new:nnnn { str } { unescape-#2 }
24177   { String~invalid~in~escaping~'#2'. }
24178   {
24179     LaTeX~came~across~the~escape~character~'#1'~not~followed~by~
24180     two~hexadecimal~digits.~This~is~invalid~in~the~escaping~'#2'.
24181   }

```

```
24182 } 24182
24183 \exp_after:wN \__str_tmp:w \c_hash_str { name } 24183
24184 \__str_unescape_name_loop:wNN 24184
24185 \exp_after:wN \__str_tmp:w \c_percent_str { url } 24185
24186 \__str_unescape_url_loop:wNN 24186
24187 \group_begin: 24187
24188 \char_set_catcode_other:N \^^J 24188
24189 \char_set_catcode_other:N \^^M 24189
24190 \cs_set_protected:Npn \__str_tmp:w #1 24190
24191 { 24191
24192 \cs_new_protected:Npn \__str_convert_unescape_string: 24192
24193 { 24193
24194 \group_begin: 24194
24195 \flag_clear:N \l__str_byte_flag 24195
24196 \flag_clear:N \l__str_error_flag 24196
24197 \int_set:Nn \tex_escapechar:D { 92 } 24197
24198 \__kernel_tl_gset:Nx \g__str_result_tl 24198
24199 { 24199
24200 \exp_after:wN \__str_unescape_string_newlines:wN 24200
24201 \g__str_result_tl \prg_break: ^^M ? 24201
24202 \prg_break_point: 24202
24203 } 24203
24204 \__kernel_tl_gset:Nx \g__str_result_tl 24204
24205 { 24205
24206 \exp_after:wN \__str_unescape_string_loop:wNNN 24206
24207 \g__str_result_tl #1 ?? { ? \prg_break: } 24207
24208 \prg_break_point: 24208
24209 } 24209
24210 \__str_if_flag_error:Nne \l__str_byte_flag { non-byte } { string } 24210
24211 \__str_if_flag_error:Nne \l__str_error_flag { unescape-string } { } 24211
24212 \group_end: 24212
24213 } 24213
24214 } 24214
24215 \exp_args:No \__str_tmp:w { \c_backslash_str } 24215
24216 \exp_last_unbraced:NNNNNo 24216
24217 \cs_new:Npn \__str_unescape_string_loop:wNNN #1 \c_backslash_str #2#3#4 24217
24218 { 24218
24219 \__str_filter_bytes:n {#1} 24219
24220 \use_none:n #4 24220
24221 \__str_output_byte:w ' 24221
24222 \__str_octal_use:NTF #2 24222
24223 { 24223
24224 \__str_octal_use:NTF #3 24224
24225 } 24225
24226 \__str_octal_use:NTF #4 24226
24227 { 24227
```

```
24228         \if_int_compare:w #2 > 3 \exp_stop_f: 24228
24229         - 256 24229
24230         \fi: 24230
24231         \__str_unescape_string_repeat:NNNNNN 24231
24232     } 24232
24233     { \__str_unescape_string_repeat:NNNNNN ? } 24233
24234 } 24234
24235 { \__str_unescape_string_repeat:NNNNNN ?? } 24235
24236 } 24236
24237 { 24237
24238     \str_case_e:nnF {#2} 24238
24239     { 24239
24240         { \c_backslash_str } { 134 } 24240
24241         { ( } { 50 } 24241
24242         { ) } { 51 } 24242
24243         { r } { 15 } 24243
24244         { f } { 14 } 24244
24245         { n } { 12 } 24245
24246         { t } { 11 } 24246
24247         { b } { 10 } 24247
24248         { ^^J } { 0 - 1 } 24248
24249     } 24249
24250 } 24250
24251     \flag_raise:N \l__str_error_flag 24251
24252     0 - 1 \use_i:nn 24252
24253 } 24253
24254 } 24254
24255     \__str_output_end: 24255
24256     \use_i:nn \__str_unescape_string_loop:wNNN #2#3#4 24256
24257 } 24257
24258 \cs_new:Npn \__str_unescape_string_repeat:NNNNNN #1#2#3#4#5#6 24258
24259 { \__str_output_end: \__str_unescape_string_loop:wNNN } 24259
24260 \cs_new:Npn \__str_unescape_string_newlines:wN #1 ^^M #2 24260
24261 { 24261
24262     #1 24262
24263     \if_charcode:w ^^J #2 \else: ^^J \fi: 24263
24264     \__str_unescape_string_newlines:wN #2 24264
24265 } 24265
24266 \msg_new:nnnn { str } { unescape-string } 24266
24267 { String~invalid~in~escaping~'string'. } 24267
24268 { 24268
24269     LaTeX~came~across~an~escape~character~'\c_backslash_str'~ 24269
24270     not~followed~by~any~of:~'n',~'r',~'t',~'b',~'f',~'(',~')',~ 24270
24271     '\c_backslash_str',~one~to~three~octal~digits,~or~the~end~ 24271
24272     of~a~line. 24272
24273 } 24273
```

24274	\group_end:	24274
24275	\cs_new_protected:Npn __str_convert_escape_hex:	24275
24276	{ __str_convert_gmap:N __str_escape_hex_char:N }	24276
24277	\cs_new:Npn __str_escape_hex_char:N #1	24277
24278	{ __str_output_hexadecimal:n {`#1} }	24278
24279	\str_const:Nn \c__str_escape_name_not_str { ! " \$ & ' } % \$ }	24279
24280	\str_const:Nn \c__str_escape_name_str { {} / < > [] }	24280
24281	\cs_new_protected:Npn __str_convert_escape_name:	24281
24282	{ __str_convert_gmap:N __str_escape_name_char:n }	24282
24283	\cs_new:Npn __str_escape_name_char:n #1	24283
24284	{	24284
24285	__str_if_escape_name:nTF {#1} {#1}	24285
24286	{ \c_hash_str __str_output_hexadecimal:n {`#1} }	24286
24287	}	24287
24288	\prg_new_conditional:Npnn __str_if_escape_name:n #1 { TF }	24288
24289	{	24289
24290	\if_int_compare:w `#1 < "2A \exp_stop_f:	24290
24291	__str_if_contains_char:NnTF \c__str_escape_name_not_str {#1}	24291
24292	\prg_return_true: \prg_return_false:	24292
24293	\else:	24293
24294	\if_int_compare:w `#1 > "7E \exp_stop_f:	24294
24295	\prg_return_false:	24295
24296	\else:	24296
24297	__str_if_contains_char:NnTF \c__str_escape_name_str {#1}	24297
24298	\prg_return_false: \prg_return_true:	24298
24299	\fi:	24299
24300	\fi:	24300
24301	}	24301
24302	\str_const:Ne \c__str_escape_string_str	24302
24303	{ \c_backslash_str () }	24303
24304	\cs_new_protected:Npn __str_convert_escape_string:	24304
24305	{ __str_convert_gmap:N __str_escape_string_char:N }	24305
24306	\cs_new:Npn __str_escape_string_char:N #1	24306
24307	{	24307
24308	__str_if_escape_string:NTF #1	24308
24309	{	24309
24310	__str_if_contains_char:NnT	24310
24311	\c__str_escape_string_str {#1}	24311
24312	{ \c_backslash_str }	24312
24313	#1	24313
24314	}	24314
24315	{	24315
24316	\c_backslash_str	24316
24317	\int_div_truncate:nn {`#1} {64}	24317
24318	\int_mod:nn { \int_div_truncate:nn {`#1} { 8 } } { 8 }	24318
24319	\int_mod:nn {`#1} { 8 }	24319

```
24320     }
24321 }
24322 \prg_new_conditional:Npnn \__str_if_escape_string:N #1 { TF }
24323 {
24324     \if_int_compare:w `#1 < "27 \exp_stop_f:
24325     \prg_return_false:
24326 \else:
24327     \if_int_compare:w `#1 > "7A \exp_stop_f:
24328     \prg_return_false:
24329 \else:
24330     \prg_return_true:
24331 \fi:
24332 \fi:
24333 }
24334 \cs_new_protected:Npn \__str_convert_escape_url:
24335 { \__str_convert_gmap:N \__str_escape_url_char:n }
24336 \cs_new:Npn \__str_escape_url_char:n #1
24337 {
24338     \__str_if_escape_url:nTF {#1} {#1}
24339     { \c_percent_str \__str_output_hexadecimal:n { `#1 } }
24340 }
24341 \prg_new_conditional:Npnn \__str_if_escape_url:n #1 { TF }
24342 {
24343     \if_int_compare:w `#1 < "30 \exp_stop_f:
24344     \__str_if_contains_char:nnTF { "-. } {#1}
24345     \prg_return_true: \prg_return_false:
24346 \else:
24347     \if_int_compare:w `#1 > "7E \exp_stop_f:
24348     \prg_return_false:
24349 \else:
24350     \__str_if_contains_char:nnTF { : ; = ? @ [ ] } {#1}
24351     \prg_return_false: \prg_return_true:
24352 \fi:
24353 \fi:
24354 }
24355 \cs_new_protected:cpn { __str_convert_encode_utf8: }
24356 { \__str_convert_gmap_internal:N \__str_encode_utf_viii_char:n }
24357 \cs_new:Npn \__str_encode_utf_viii_char:n #1
24358 {
24359     \__str_encode_utf_viii_loop:wnnw #1 \__str_sep: - 1 + 0 * \__str_sep:
24360     { 128 } { 0 }
24361     { 32 } { 192 }
24362     { 16 } { 224 }
24363     { 8 } { 240 }
24364 \s__str_stop
24365 }
```

```
24366 \cs_new:Npn \__str_encode_utf_viii_loop:wwnnw 24366
24367 #1 \__str_sep: #2 \__str_sep: #3#4 #5 \s__str_stop 24367
24368 { 24368
24369 \if_int_compare:w #1 < #3 \exp_stop_f: 24369
24370 \__str_output_byte:n { #1 + #4 } 24370
24371 \exp_after:wN \__str_use_none_delimit_by_s_stop:w 24371
24372 \fi: 24372
24373 \exp_after:wN \__str_encode_utf_viii_loop:wwnnw 24373
24374 \int_value:w \int_div_truncate:nn {#1} {64} \__str_sep: #1 \__str_sep: 24374
24375 #5 \s__str_stop 24375
24376 \__str_output_byte:n { #2 - 64 * ( #1 - 2 ) } 24376
24377 } 24377
24378 \flag_clear_new:N \l__str_missing_flag 24378
24379 \flag_clear_new:N \l__str_extra_flag 24379
24380 \flag_clear_new:N \l__str_overlong_flag 24380
24381 \flag_clear_new:N \l__str_overflow_flag 24381
24382 \msg_new:nnnn { str } { utf8-decode } 24382
24383 { 24383
24384 Invalid~UTF-8~string: 24384
24385 \exp_last_unbraced:Nf \use_none:n 24385
24386 { 24386
24387 \__str_if_flag_times:NT \l__str_missing_flag { ,~missing~continuation~byte } 24387
24388 \__str_if_flag_times:NT \l__str_extra_flag { ,~extra~continuation~byte } 24388
24389 \__str_if_flag_times:NT \l__str_overlong_flag { ,~overlong~form } 24389
24390 \__str_if_flag_times:NT \l__str_overflow_flag { ,~code~point~too~large } 24390
24391 } 24391
24392 . 24392
24393 } 24393
24394 { 24394
24395 In~the~UTF-8~encoding,~each~Unicode~character~consists~in~ 24395
24396 1~to~4~bytes,~with~the~following~bit~pattern: \ 24396
24397 \iow_indent:n 24397
24398 { 24398
24399 Code~point~\ 24399
24400 Code~point~\ 24400
24401 Code~point~\ 24401
24402 Code~point~ 24402
24403 } 24403
24404 Bytes~of~the~form~10xxxxxx~are~called~continuation~bytes. 24404
24405 \flag_if_raised:NT \l__str_missing_flag 24405
24406 { 24406
24407 \ 24407
24408 A~leading~byte~(in~the~range~[192,255])~was~not~followed~by~ 24408
24409 the~appropriate~number~of~continuation~bytes. 24409
24410 } 24410
24411 \flag_if_raised:NT \l__str_extra_flag 24411
```



```

24412 {
24413     \\\
24414     LaTeX~came~across~a~continuation~byte~when~it~was~not~expected.
24415 }
24416 \flag_if_raised:NT \l__str_overlong_flag
24417 {
24418     \\\
24419     Every~Unicode~code~point~must~be~expressed~in~the~shortest~
24420     possible~form.~For~instance,~'0xC0'~'0x83'~is~not~a~valid~
24421     representation~for~the~code~point~3.
24422 }
24423 \flag_if_raised:NT \l__str_overflow_flag
24424 {
24425     \\\
24426     Unicode~limits~code~points~to~the~range~[0,1114111].
24427 }
24428 }
24429 \prop_gput:Nnn \g_msg_module_name_prop { str } { LaTeX }
24430 \prop_gput:Nnn \g_msg_module_type_prop { str } { }
24431 \cs_new_protected:cpn { __str_convert_decode_utf8: }
24432 {
24433     \flag_clear:N \l__str_error_flag
24434     \flag_clear:N \l__str_missing_flag
24435     \flag_clear:N \l__str_extra_flag
24436     \flag_clear:N \l__str_overlong_flag
24437     \flag_clear:N \l__str_overflow_flag
24438     \__kernel_tl_gset:Nx \g__str_result_tl
24439     {
24440         \exp_after:wN \__str_decode_utf_viii_start:N \g__str_result_tl
24441         { \prg_break: \__str_decode_utf_viii_end: }
24442         \prg_break_point:
24443     }
24444     \__str_if_flag_error:Nne \l__str_error_flag { utf8-decode } { }
24445 }
24446 \cs_new:Npn \__str_decode_utf_viii_start:N #1
24447 {
24448     #1
24449     \if_int_compare:w `#1 < "C0 \exp_stop_f:
24450         \s__str
24451         \if_int_compare:w `#1 < "80 \exp_stop_f:
24452             \int_value:w `#1
24453         \else:
24454             \flag_raise:N \l__str_extra_flag
24455             \flag_raise:N \l__str_error_flag
24456             \int_use:N \c__str_replacement_char_int
24457             \fi:

```

```
24458 \else: 24458
24459 \exp_after:wN \__str_decode_utf_viii_continuation:wwN 24459
24460 \int_value:w \int_eval:n { `#1 - "C0 } \exp_after:wN 24460
24461 \fi: 24461
24462 \s__str 24462
24463 \__str_use_none_delimit_by_s_stop:w {"80} {"800} {"10000} {"110000} \s__str_stop 24463
24464 \__str_decode_utf_viii_start:N 24464
24465 } 24465
24466 \cs_new:Npn \__str_decode_utf_viii_continuation:wwN 24466
24467 #1 \s__str #2 \__str_decode_utf_viii_start:N #3 24467
24468 { 24468
24469 \use_none:n #3 24469
24470 \if_int_compare:w `#3 < 24470
24471 \if_int_compare:w `#3 < "80 \exp_stop_f: - \fi: 24471
24472 "C0 \exp_stop_f: 24472
24473 #3 24473
24474 \exp_after:wN \__str_decode_utf_viii_aux:wNnnwN 24474
24475 \int_value:w \int_eval:n { #1 * "40 + `#3 - "80 } \exp_after:wN 24475
24476 \else: 24476
24477 \s__str 24477
24478 \flag_raise:N \l__str_missing_flag 24478
24479 \flag_raise:N \l__str_error_flag 24479
24480 \int_use:N \c__str_replacement_char_int 24480
24481 \fi: 24481
24482 \s__str 24482
24483 #2 24483
24484 \__str_decode_utf_viii_start:N #3 24484
24485 } 24485
24486 \cs_new:Npn \__str_decode_utf_viii_aux:wNnnwN 24486
24487 #1 \s__str #2#3#4 #5 \__str_decode_utf_viii_start:N #6 24487
24488 { 24488
24489 \if_int_compare:w #1 < #4 \exp_stop_f: 24489
24490 \s__str 24490
24491 \if_int_compare:w #1 < #3 \exp_stop_f: 24491
24492 \flag_raise:N \l__str_overlong_flag 24492
24493 \flag_raise:N \l__str_error_flag 24493
24494 \int_use:N \c__str_replacement_char_int 24494
24495 \else: 24495
24496 #1 24496
24497 \fi: 24497
24498 \else: 24498
24499 \if_meaning:w \s__str_stop #5 24499
24500 \__str_decode_utf_viii_overflow:w #1 24500
24501 \fi: 24501
24502 \exp_after:wN \__str_decode_utf_viii_continuation:wwN 24502
24503 \int_value:w \int_eval:n { #1 - #4 } \exp_after:wN 24503
```

24504	\fi:	24504
24505	\s__str	24505
24506	#2 {#4} #5	24506
24507	__str_decode_utf_viii_start:N	24507
24508	}	24508
24509	\cs_new:Npn __str_decode_utf_viii_overflow:w #1 \fi: #2 \fi:	24509
24510	{	24510
24511	\fi: \fi:	24511
24512	\flag_raise:N \l__str_overflow_flag	24512
24513	\flag_raise:N \l__str_error_flag	24513
24514	\int_use:N \c__str_replacement_char_int	24514
24515	}	24515
24516	\cs_new:Npn __str_decode_utf_viii_end:	24516
24517	{	24517
24518	\s__str	24518
24519	\flag_raise:N \l__str_missing_flag	24519
24520	\flag_raise:N \l__str_error_flag	24520
24521	\int_use:N \c__str_replacement_char_int \s__str	24521
24522	\prg_break:	24522
24523	}	24523
24524	\group_begin:	24524
24525	\char_set_catcode_other:N \^^fe	24525
24526	\char_set_catcode_other:N \^^ff	24526
24527	\cs_new_protected:cpn { __str_convert_encode_utf16: }	24527
24528	{	24528
24529	__str_encode_utf_xvi_aux:N __str_output_byte_pair_be:n	24529
24530	\tl_gput_left:Ne \g__str_result_tl { ^^fe ^^ff }	24530
24531	}	24531
24532	\cs_new_protected:cpn { __str_convert_encode_utf16be: }	24532
24533	{ __str_encode_utf_xvi_aux:N __str_output_byte_pair_be:n }	24533
24534	\cs_new_protected:cpn { __str_convert_encode_utf16le: }	24534
24535	{ __str_encode_utf_xvi_aux:N __str_output_byte_pair_le:n }	24535
24536	\cs_new_protected:Npn __str_encode_utf_xvi_aux:N #1	24536
24537	{	24537
24538	\flag_clear:N \l__str_error_flag	24538
24539	\cs_set_eq:NN __str_tmp:w #1	24539
24540	__str_convert_gmap_internal:N __str_encode_utf_xvi_char:n	24540
24541	__str_if_flag_error:Nne \l__str_error_flag { utf16-encode } { }	24541
24542	}	24542
24543	\cs_new:Npn __str_encode_utf_xvi_char:n #1	24543
24544	{	24544
24545	\if_int_compare:w #1 < "D800 \exp_stop_f:	24545
24546	__str_tmp:w {#1}	24546
24547	\else:	24547
24548	\if_int_compare:w #1 < "10000 \exp_stop_f:	24548
24549	\if_int_compare:w #1 < "E000 \exp_stop_f:	24549

```
24550         \flag_raise:N \l__str_error_flag
24551         \__str_tmp:w { \c__str_replacement_char_int }
24552     \else:
24553         \__str_tmp:w {#1}
24554     \fi:
24555 \else:
24556     \exp_args:Nf \__str_tmp:w { \int_div_truncate:nn {#1} {"400} + "D7C0 }
24557     \exp_args:Nf \__str_tmp:w { \int_mod:nn {#1} {"400} + "DC00 }
24558 \fi:
24559 \fi:
24560 }
24561 \flag_clear_new:N \l__str_missing_flag
24562 \flag_clear_new:N \l__str_extra_flag
24563 \flag_clear_new:N \l__str_end_flag
24564 \msg_new:nnnn { str } { utf16-encode }
24565 { Unicode~string~cannot~be~expressed~in~UTF-16:~surrogate. }
24566 {
24567     Surrogate~code~points~(in~the~range~[U+D800,~U+DFFF])~
24568     can~be~expressed~in~the~UTF-8~and~UTF-32~encodings,~
24569     but~not~in~the~UTF-16~encoding.
24570 }
24571 \msg_new:nnnn { str } { utf16-decode }
24572 {
24573     Invalid~UTF-16~string:
24574     \exp_last_unbraced:Nf \use_none:n
24575     {
24576         \__str_if_flag_times:NT \l__str_missing_flag { ,~missing~trail~surrogate }
24577         \__str_if_flag_times:NT \l__str_extra_flag { ,~extra~trail~surrogate }
24578         \__str_if_flag_times:NT \l__str_end_flag { ,~odd~number~of~bytes }
24579     }
24580     .
24581 }
24582 {
24583     In~the~UTF-16~encoding,~each~Unicode~character~is~encoded~as~
24584     2~or~4~bytes: \\
24585     \iow_indent:n
24586     {
24587         Code~point~in~[U+0000,~U+D7FF]:~two~bytes \\
24588         Code~point~in~[U+D800,~U+DFFF]:~illegal \\
24589         Code~point~in~[U+E000,~U+FFFF]:~two~bytes \\
24590         Code~point~in~[U+10000,~U+10FFFF]:~
24591         a~lead~surrogate~and~a~trail~surrogate \\
24592     }
24593     Lead~surrogates~are~pairs~of~bytes~in~the~range~[0xD800,~0xDBFF]~,~
24594     and~trail~surrogates~are~in~the~range~[0xDC00,~0xDFFF]~.
24595     \flag_if_raised:NT \l__str_missing_flag
```

```
24596 { 24596
24597     \\\ 24597
24598     A~lead~surrogate~was~not~followed~by~a~trail~surrogate. 24598
24599 } 24599
24600 \flag_if_raised:NT \l__str_extra_flag 24600
24601 { 24601
24602     \\\ 24602
24603     LaTeX~came~across~a~trail~surrogate~when~it~was~not~expected. 24603
24604 } 24604
24605 \flag_if_raised:NT \l__str_end_flag 24605
24606 { 24606
24607     \\\ 24607
24608     The~string~contained~an~odd~number~of~bytes.~This~is~invalid:~ 24608
24609     the~basic~code~unit~for~UTF-16~is~16~bits~(2~bytes). 24609
24610 } 24610
24611 } 24611
24612 \cs_new_protected:cpn { __str_convert_decode_utf16be: } 24612
24613 { \__str_decode_utf_xvi:Nw 1 \g__str_result_tl \s__str_stop } 24613
24614 \cs_new_protected:cpn { __str_convert_decode_utf16le: } 24614
24615 { \__str_decode_utf_xvi:Nw 2 \g__str_result_tl \s__str_stop } 24615
24616 \cs_new_protected:cpn { __str_convert_decode_utf16: } 24616
24617 { 24617
24618     \exp_after:wN \__str_decode_utf_xvi_bom:NN 24618
24619     \g__str_result_tl \s__str_stop \s__str_stop \s__str_stop 24619
24620 } 24620
24621 \cs_new_protected:Npn \__str_decode_utf_xvi_bom:NN #1#2 24621
24622 { 24622
24623     \str_if_eq:nnTF { #1#2 } { ^^ff ^^fe } 24623
24624     { \__str_decode_utf_xvi:Nw 2 } 24624
24625     { 24625
24626         \str_if_eq:nnTF { #1#2 } { ^^fe ^^ff } 24626
24627         { \__str_decode_utf_xvi:Nw 1 } 24627
24628         { \__str_decode_utf_xvi:Nw 1 #1#2 } 24628
24629     } 24629
24630 } 24630
24631 \cs_new_protected:Npn \__str_decode_utf_xvi:Nw #1#2 \s__str_stop 24631
24632 { 24632
24633     \flag_clear:N \l__str_error_flag 24633
24634     \flag_clear:N \l__str_missing_flag 24634
24635     \flag_clear:N \l__str_extra_flag 24635
24636     \flag_clear:N \l__str_end_flag 24636
24637     \cs_set:Npn \__str_tmp:w ##1 ##2 { ` ## #1 } 24637
24638     \__kernel_tl_gset:Nx \g__str_result_tl 24638
24639     { 24639
24640         \exp_after:wN \__str_decode_utf_xvi_pair:NN 24640
24641         #2 \q__str_nil \q__str_nil 24641
```

```
24642 \prg_break_point:
24643 }
24644 \__str_if_flag_error:Nne \l__str_error_flag { utf16-decode } { }
24645 }
24646 \cs_new:Npn \__str_decode_utf_xvi_pair:NN #1#2
24647 {
24648   \if_meaning:w \q__str_nil #2
24649     \__str_decode_utf_xvi_pair_end:Nw #1
24650   \fi:
24651   \if_case:w
24652     \int_eval:n { ( \__str_tmp:w #1#2 - "D6 ) / 4 } \scan_stop:
24653   \or: \exp_after:wN \__str_decode_utf_xvi_quad:NNwNN
24654   \or: \exp_after:wN \__str_decode_utf_xvi_extra:NNw
24655   \fi:
24656   #1#2 \s__str
24657   \int_eval:n { "100 * \__str_tmp:w #1#2 + \__str_tmp:w #2#1 } \s__str
24658   \__str_decode_utf_xvi_pair:NN
24659 }
24660 \cs_new:Npn \__str_decode_utf_xvi_quad:NNwNN
24661   #1#2 #3 \__str_decode_utf_xvi_pair:NN #4#5
24662 {
24663   \if_meaning:w \q__str_nil #5
24664     \__str_decode_utf_xvi_error:nNN { missing } #1#2
24665     \__str_decode_utf_xvi_pair_end:Nw #4
24666   \fi:
24667   \if_int_compare:w
24668     \if_int_compare:w \__str_tmp:w #4#5 < "DC \exp_stop_f:
24669       0 = 1
24670     \else:
24671       \__str_tmp:w #4#5 < "E0
24672     \fi:
24673     \exp_stop_f:
24674     #1 #2 #4 #5 \s__str
24675     \int_eval:n
24676       {
24677         ( "100 * \__str_tmp:w #1#2 + \__str_tmp:w #2#1 - "D7F7 ) * "400
24678         + "100 * \__str_tmp:w #4#5 + \__str_tmp:w #5#4
24679       }
24680     \s__str
24681     \exp_after:wN \use_i:nnn
24682   \else:
24683     \__str_decode_utf_xvi_error:nNN { missing } #1#2
24684   \fi:
24685   \__str_decode_utf_xvi_pair:NN #4#5
24686 }
24687 \cs_new:Npn \__str_decode_utf_xvi_pair_end:Nw #1 \fi:
```

```

24688 {
24689     \fi:
24690     \if_meaning:w \q__str_nil #1
24691     \else:
24692         \__str_decode_utf_xvi_error:nNN { end } #1 \prg_do_nothing:
24693     \fi:
24694     \prg_break:
24695 }
24696 \cs_new:Npn \__str_decode_utf_xvi_extra:NNw #1#2 \s__str #3 \s__str
24697 { \__str_decode_utf_xvi_error:nNN { extra } #1#2 }
24698 \cs_new:Npn \__str_decode_utf_xvi_error:nNN #1#2#3
24699 {
24700     \flag_raise:N \l__str_error_flag
24701     \flag_raise:c { l__str_#1_flag }
24702     #2 #3 \s__str
24703     \int_use:N \c__str_replacement_char_int \s__str
24704 }
24705 \group_end:
24706 \group_begin:
24707     \char_set_catcode_other:N \^^00
24708     \char_set_catcode_other:N \^^fe
24709     \char_set_catcode_other:N \^^ff
24710     \cs_new_protected:cpn { __str_convert_encode_utf32: }
24711     {
24712         \__str_convert_gmap_internal:N \__str_encode_utf_xxxii_be:n
24713         \tl_gput_left:Ne \g__str_result_tl { ^^00 ^^00 ^^fe ^^ff }
24714     }
24715     \cs_new_protected:cpn { __str_convert_encode_utf32be: }
24716     { \__str_convert_gmap_internal:N \__str_encode_utf_xxxii_be:n }
24717     \cs_new_protected:cpn { __str_convert_encode_utf32le: }
24718     { \__str_convert_gmap_internal:N \__str_encode_utf_xxxii_le:n }
24719     \cs_new:Npn \__str_encode_utf_xxxii_be:n #1
24720     {
24721         \exp_args:Nf \__str_encode_utf_xxxii_be_aux:nn
24722         { \int_div_truncate:nn {#1} { "100 } } {#1}
24723     }
24724     \cs_new:Npn \__str_encode_utf_xxxii_be_aux:nn #1#2
24725     {
24726         ^^00
24727         \__str_output_byte_pair_be:n {#1}
24728         \__str_output_byte:n { #2 - #1 * "100 }
24729     }
24730     \cs_new:Npn \__str_encode_utf_xxxii_le:n #1
24731     {
24732         \exp_args:Nf \__str_encode_utf_xxxii_le_aux:nn
24733         { \int_div_truncate:nn {#1} { "100 } } {#1}

```



```

24734 } 24734
24735 \cs_new:Npn \__str_encode_utf_xxxii_le_aux:nn #1#2 24735
24736 { 24736
24737 \__str_output_byte:n { #2 - #1 * "100 } 24737
24738 \__str_output_byte_pair_le:n {#1} 24738
24739 ^^00 24739
24740 } 24740
24741 \flag_clear_new:N \l__str_overflow_flag 24741
24742 \flag_clear_new:N \l__str_end_flag 24742
24743 \msg_new:nnnn { str } { utf32-decode } 24743
24744 { 24744
24745 Invalid~UTF-32~string: 24745
24746 \exp_last_unbraced:Nf \use_none:n 24746
24747 { 24747
24748 \__str_if_flag_times:NT \l__str_overflow_flag { ,~code~point~too~large } 24748
24749 \__str_if_flag_times:NT \l__str_end_flag { ,~truncated~string } 24749
24750 } 24750
24751 . 24751
24752 } 24752
24753 { 24753
24754 In~the~UTF-32~encoding,~every~Unicode~character~ 24754
24755 (in~the~range~[U+0000,~U+10FFFF])~is~encoded~as~4~bytes. 24755
24756 \flag_if_raised:NT \l__str_overflow_flag 24756
24757 { 24757
24758 \\\ 24758
24759 LaTeX~came~across~a~code~point~larger~than~1114111,~ 24759
24760 the~maximum~code~point~defined~by~Unicode.~ 24760
24761 Perhaps~the~string~was~not~encoded~in~the~UTF-32~encoding? 24761
24762 } 24762
24763 \flag_if_raised:NT \l__str_end_flag 24763
24764 { 24764
24765 \\\ 24765
24766 The~length~of~the~string~is~not~a~multiple~of~4.~ 24766
24767 Perhaps~the~string~was~truncated? 24767
24768 } 24768
24769 } 24769
24770 \cs_new_protected:cpn { __str_convert_decode_utf32be: } 24770
24771 { \__str_decode_utf_xxxii:Nw 1 \g__str_result_tl \s__str_stop } 24771
24772 \cs_new_protected:cpn { __str_convert_decode_utf32le: } 24772
24773 { \__str_decode_utf_xxxii:Nw 2 \g__str_result_tl \s__str_stop } 24773
24774 \cs_new_protected:cpn { __str_convert_decode_utf32: } 24774
24775 { 24775
24776 \exp_after:wN \__str_decode_utf_xxxii_bom:NNNN \g__str_result_tl 24776
24777 \s__str_stop \s__str_stop \s__str_stop \s__str_stop \s__str_stop 24777
24778 } 24778
24779 \cs_new_protected:Npn \__str_decode_utf_xxxii_bom:NNNN #1#2#3#4 24779

```

```
24780 { 24780
24781 \str_if_eq:nnTF { #1#2#3#4 } { ^^ff ^^fe ^^00 ^^00 } 24781
24782 { \__str_decode_utf_xxxii:Nw 2 } 24782
24783 { 24783
24784 \str_if_eq:nnTF { #1#2#3#4 } { ^^00 ^^00 ^^fe ^^ff } 24784
24785 { \__str_decode_utf_xxxii:Nw 1 } 24785
24786 { \__str_decode_utf_xxxii:Nw 1 #1#2#3#4 } 24786
24787 } 24787
24788 } 24788
24789 \cs_new_protected:Npn \__str_decode_utf_xxxii:Nw #1#2 \s__str_stop 24789
24790 { 24790
24791 \flag_clear:N \l__str_overflow_flag 24791
24792 \flag_clear:N \l__str_end_flag 24792
24793 \flag_clear:N \l__str_error_flag 24793
24794 \cs_set:Npn \__str_tmp:w ##1 ##2 { ` ## #1 } 24794
24795 \__kernel_tl_gset:Nx \g__str_result_tl 24795
24796 { 24796
24797 \exp_after:wN \__str_decode_utf_xxxii_loop:NNNN 24797
24798 #2 \s__str_stop \s__str_stop \s__str_stop \s__str_stop 24798
24799 \prg_break_point: 24799
24800 } 24800
24801 \__str_if_flag_error:Nne \l__str_error_flag { utf32-decode } { } 24801
24802 } 24802
24803 \cs_new:Npn \__str_decode_utf_xxxii_loop:NNNN #1#2#3#4 24803
24804 { 24804
24805 \if_meaning:w \s__str_stop #4 24805
24806 \exp_after:wN \__str_decode_utf_xxxii_end:w 24806
24807 \fi: 24807
24808 #1#2#3#4 \s__str 24808
24809 \if_int_compare:w \__str_tmp:w #1#4 > \c_zero_int 24809
24810 \flag_raise:N \l__str_overflow_flag 24810
24811 \flag_raise:N \l__str_error_flag 24811
24812 \int_use:N \c__str_replacement_char_int 24812
24813 \else: 24813
24814 \if_int_compare:w \__str_tmp:w #2#3 > 16 \exp_stop_f: 24814
24815 \flag_raise:N \l__str_overflow_flag 24815
24816 \flag_raise:N \l__str_error_flag 24816
24817 \int_use:N \c__str_replacement_char_int 24817
24818 \else: 24818
24819 \int_eval:n 24819
24820 { \__str_tmp:w #2#3*"10000 + \__str_tmp:w #3#2*"100 + \__str_tmp:w #4#1 } 24820
24821 \fi: 24821
24822 \fi: 24822
24823 \s__str 24823
24824 \__str_decode_utf_xxxii_loop:NNNN 24824
24825 } 24825
```

```
24826 \cs_new:Npn \__str_decode_utf_xxxii_end:w #1 \s__str_stop 24826
24827 { 24827
24828 \tl_if_empty:nF {#1} 24828
24829 { 24829
24830 \flag_raise:N \l__str_end_flag 24830
24831 \flag_raise:N \l__str_error_flag 24831
24832 #1 \s__str 24832
24833 \int_use:N \c__str_replacement_char_int \s__str 24833
24834 } 24834
24835 \prg_break: 24835
24836 } 24836
24837 \group_end: 24837
24838 \cs_new:Npn \str_convert_pdfname:n #1 24838
24839 { 24839
24840 \exp_args:Ne \tl_to_str:n 24840
24841 { \str_map_function:nN {#1} \__str_convert_pdfname:n } 24841
24842 } 24842
24843 \sys_if_engine_opentype:TF 24843
24844 { 24844
24845 \cs_new:Npn \__str_convert_pdfname:n #1 24845
24846 { 24846
24847 \int_compare:nNnTF { `#1 } > { "7F } 24847
24848 { \__str_convert_pdfname_bytes:n {#1} } 24848
24849 { \__str_escape_name_char:n {#1} } 24849
24850 } 24850
24851 \cs_new:Npn \__str_convert_pdfname_bytes:n #1 24851
24852 { 24852
24853 \exp_args:Ne \__str_convert_pdfname_bytes_aux:n 24853
24854 { \__kernel_codepoint_to_bytes:n {`#1} } 24854
24855 } 24855
24856 \cs_new:Npn \__str_convert_pdfname_bytes_aux:n #1 24856
24857 { \__str_convert_pdfname_bytes_aux:nnnn #1 } 24857
24858 \cs_new:Npe \__str_convert_pdfname_bytes_aux:nnnn #1#2#3#4 24858
24859 { 24859
24860 \c_hash_str \exp_not:N \__str_output_hexadecimal:n {#1} 24860
24861 \c_hash_str \exp_not:N \__str_output_hexadecimal:n {#2} 24861
24862 \exp_not:N \tl_if_blank:nF {#3} 24862
24863 { 24863
24864 \c_hash_str \exp_not:N \__str_output_hexadecimal:n {#3} 24864
24865 \exp_not:N \tl_if_blank:nF {#4} 24865
24866 { 24866
24867 \c_hash_str \exp_not:N \__str_output_hexadecimal:n {#4} 24867
24868 } 24868
24869 } 24869
24870 } 24870
24871 } 24871
```

24872	{ \cs_new_eq:NN __str_convert_pdfname:n __str_escape_name_char:n }	24872
24873	%% File: l3tl-analysis.dtx	24873
24874	\scan_new:N \s__tl	24874
24875	\cs_new_eq:NN \l__tl_analysis_token ?	24875
24876	\cs_new_eq:NN \l__tl_analysis_char_token ?	24876
24877	\tl_new:N \l__tl_peek_code_tl	24877
24878	\group_begin:	24878
24879	\char_set_active_eq:NN _ \scan_stop:	24879
24880	\tl_const:Ne \c__tl_peek_catcodes_tl	24880
24881	{	24881
24882	\char_generate:nn { 32 } { 3 } 3	24882
24883	\char_generate:nn { 32 } { 4 } 4	24883
24884	\char_generate:nn { 32 } { 7 } 7	24884
24885	\char_generate:nn { 32 } { 8 } 8	24885
24886	\c_space_tl \token_to_str:N A	24886
24887	\char_generate:nn { 32 } { 11 } \token_to_str:N B	24887
24888	\char_generate:nn { 32 } { 12 } \token_to_str:N C	24888
24889	\char_generate:nn { 32 } { 13 } \token_to_str:N D	24889
24890	}	24890
24891	\group_end:	24891
24892	\int_new:N \l__tl_analysis_normal_int	24892
24893	\int_new:N \l__tl_analysis_index_int	24893
24894	\int_new:N \l__tl_analysis_nesting_int	24894
24895	\int_new:N \l__tl_analysis_type_int	24895
24896	\tl_new:N \g__tl_analysis_result_tl	24896
24897	\cs_new:Npn __tl_analysis_extract_charcode:	24897
24898	{	24898
24899	\exp_after:wN __tl_analysis_extract_charcode_aux:w	24899
24900	\token_to_meaning:N \l__tl_analysis_token	24900
24901	}	24901
24902	\cs_new:Npn __tl_analysis_extract_charcode_aux:w #1 ~ #2 ~ { ` }	24902
24903	\bool_lazy_and:nnT	24903
24904	{ \cs_if_exist_p:N \tex_luatexversion:D }	24904
24905	{ \int_compare_p:nNn { \int_div_truncate:nn { \tex_luatexversion:D } { 100 } } > 1 }	24905
24906	{	24906
24907	\cs_gset:Npn __tl_analysis_extract_charcode_aux:w #1 + #2 ~ ' #3 ' {"#2}	24907
24908	}	24908
24909	\cs_new:Npe __tl_analysis_cs_space_count:NN #1 #2	24909
24910	{	24910
24911	\exp_not:N \exp_after:wN #1	24911
24912	\exp_not:N \int_value:w \exp_not:N \int_eval:w 0	24912
24913	\exp_not:N \exp_after:wN \exp_not:N __tl_analysis_cs_space_count:w	24913
24914	\exp_not:N \token_to_str:N #2	24914
24915	\exp_not:N \fi: \exp_not:N __tl_analysis_cs_space_count_end:w	24915
24916	\exp_not:N __tl_sep: \c_space_tl !	24916
24917	}	24917

```
24918 \cs_new:Npn \__tl_analysis_cs_space_count:w #1 ~
24919 {
24920   \if_false: #1 #1 \fi:
24921   + 1
24922   \__tl_analysis_cs_space_count:w
24923 }
24924 \cs_new:Npn \__tl_analysis_cs_space_count_end:w \__tl_sep: #1 \fi: #2 !
24925 {
24926   \exp_after:wN \__tl_sep: \int_value:w
24927   \str_count_ignore_spaces:n {#1} \__tl_sep:
24928 }
24929 \cs_new_protected:Npn \__tl_analysis:n #1
24930 {
24931   \group_begin:
24932   \group_align_safe_begin:
24933     \__tl_analysis_a:n {#1}
24934     \__tl_analysis_b:n {#1}
24935   \group_align_safe_end:
24936   \group_end:
24937 }
24938 \group_begin:
24939   \char_set_catcode_active:N \^^@
24940   \cs_new_protected:Npn \__tl_analysis_disable:n #1
24941   {
24942     \tex_lccode:D 0 = #1 \exp_stop_f:
24943     \tex_lowercase:D { \tex_let:D \^^@ } \tex_undefined:D
24944   }
24945   \bool_lazy_or:nnT
24946   { \sys_if_engine_ptex_p: }
24947   { \sys_if_engine_uptex_p: }
24948   {
24949     \cs_gset_protected:Npn \__tl_analysis_disable:n #1
24950     {
24951       \if_int_compare:w 256 > #1 \exp_stop_f:
24952       \tex_lccode:D 0 = #1 \exp_stop_f:
24953       \tex_lowercase:D { \tex_let:D \^^@ } \tex_undefined:D
24954       \fi:
24955     }
24956   }
24957 \group_end:
24958 \group_begin:
24959   \char_set_catcode_active:N \^^@
24960   \cs_new_protected:Npn \__tl_analysis_disable_char:N #1
24961   {
24962     \tex_lccode:D `#1 = 32 \exp_stop_f:
24963     \tex_lowercase:D { \if_meaning:w #1 } \tex_undefined:D
```

```

24964 \tex_let:D #1 \tex_undefined:D
24965 \fi:
24966 }
24967 \bool_lazy_or:nnT
24968 { \sys_if_engine_ptex_p: }
24969 { \sys_if_engine_uptex_p: }
24970 {
24971 \cs_gset_protected:Npn \__tl_analysis_disable_char:N #1
24972 {
24973 \if_int_compare:w 256 > `#1 \exp_stop_f:
24974 \tex_lccode:D `#1 = 32 \exp_stop_f:
24975 \tex_lowercase:D { \if_meaning:w #1 } \tex_undefined:D
24976 \tex_let:D #1 \tex_undefined:D
24977 \fi:
24978 \fi:
24979 }
24980 }
24981 \group_end:
24982 \cs_new_protected:Npn \__tl_analysis_a:n #1
24983 {
24984 \__tl_analysis_disable:n { 32 }
24985 \int_set:Nn \tex_escapechar:D { 92 }
24986 \int_zero:N \l__tl_analysis_normal_int
24987 \int_zero:N \l__tl_analysis_index_int
24988 \int_zero:N \l__tl_analysis_nesting_int
24989 \if_false: { \fi: \__tl_analysis_a_loop:w #1 }
24990 \int_decr:N \l__tl_analysis_index_int
24991 }
24992 \cs_new_protected:Npn \__tl_analysis_a_loop:w
24993 { \tex_futurelet:D \l__tl_analysis_token \__tl_analysis_a_type:w }
24994 \cs_new_protected:Npn \__tl_analysis_a_type:w
24995 {
24996 \l__tl_analysis_type_int =
24997 \if_meaning:w \l__tl_analysis_token \c_space_token
24998 0
24999 \else:
25000 \if_catcode:w \exp_not:N \l__tl_analysis_token \c_group_begin_token
25001 1
25002 \else:
25003 \if_catcode:w \exp_not:N \l__tl_analysis_token \c_group_end_token
25004 - 1
25005 \else:
25006 2
25007 \fi:
25008 \fi:
25009 \fi:

```

```

25010 \exp_stop_f:
25011 \if_case:w \l__tl_analysis_type_int
25012 \exp_after:wN \__tl_analysis_a_space:w
25013 \or: \exp_after:wN \__tl_analysis_a_bgroup:w
25014 \or: \exp_after:wN \__tl_analysis_a_safe:N
25015 \else: \exp_after:wN \__tl_analysis_a_egroup:w
25016 \fi:
25017 }
25018 \cs_new_protected:Npn \__tl_analysis_a_space:w
25019 {
25020 \tex_afterassignment:D \__tl_analysis_a_space_test:w
25021 \exp_after:wN \cs_set_eq:NN
25022 \exp_after:wN \l__tl_analysis_char_token
25023 \token_to_str:N
25024 }
25025 \cs_new_protected:Npn \__tl_analysis_a_space_test:w
25026 {
25027 \if_meaning:w \l__tl_analysis_char_token \c_space_token
25028 \tex_toks:D \l__tl_analysis_index_int { \exp_not:n { ~ } }
25029 \__tl_analysis_a_store:
25030 \else:
25031 \int_incr:N \l__tl_analysis_normal_int
25032 \fi:
25033 \__tl_analysis_a_loop:w
25034 }
25035 \group_begin:
25036 \char_set_catcode_group_begin:N ^^@ % {
25037 \cs_new_protected:Npn \__tl_analysis_a_bgroup:w
25038 { \__tl_analysis_a_group:nw { \exp_after:wN ^^@ \if_false: } \fi: } }
25039 \char_set_catcode_group_end:N ^^@
25040 \cs_new_protected:Npn \__tl_analysis_a_egroup:w
25041 { \__tl_analysis_a_group:nw { \if_false: { \fi: ^^@ } } % }
25042 \group_end:
25043 \cs_new_protected:Npn \__tl_analysis_a_group:nw #1
25044 {
25045 \tex_lccode:D 0 = \__tl_analysis_extract_charcode: \scan_stop:
25046 \tex_lowercase:D { \tex_toks:D \l__tl_analysis_index_int {#1} }
25047 \if_int_compare:w \tex_lccode:D 0 = \tex_escapechar:D
25048 \int_set:Nn \tex_escapechar:D { 139 - \tex_escapechar:D }
25049 \fi:
25050 \__tl_analysis_disable:n { \tex_lccode:D 0 }
25051 \tex_futurelet:D \l__tl_analysis_token \__tl_analysis_a_group_aux:w
25052 }
25053 \cs_new_protected:Npn \__tl_analysis_a_group_aux:w
25054 {
25055 \if_meaning:w \l__tl_analysis_token \tex_undefined:D

```



```

25056 \exp_after:wN \__tl_analysis_a_safe:N 25056
25057 \else: 25057
25058 \exp_after:wN \__tl_analysis_a_group_auxii:w 25058
25059 \fi: 25059
25060 } 25060
25061 \cs_new_protected:Npn \__tl_analysis_a_group_auxii:w 25061
25062 { 25062
25063 \tex_afterassignment:D \__tl_analysis_a_group_test:w 25063
25064 \exp_after:wN \cs_set_eq:NN 25064
25065 \exp_after:wN \l__tl_analysis_char_token 25065
25066 \token_to_str:N 25066
25067 } 25067
25068 \cs_new_protected:Npn \__tl_analysis_a_group_test:w 25068
25069 { 25069
25070 \if_charcode:w \l__tl_analysis_token \l__tl_analysis_char_token 25070
25071 \__tl_analysis_a_store: 25071
25072 \else: 25072
25073 \int_incr:N \l__tl_analysis_normal_int 25073
25074 \fi: 25074
25075 \__tl_analysis_a_loop:w 25075
25076 } 25076
25077 \cs_new_protected:Npn \__tl_analysis_a_store: 25077
25078 { 25078
25079 \tex_advance:D \l__tl_analysis_nesting_int \l__tl_analysis_type_int 25079
25080 \if_int_compare:w \tex_lccode:D 0 = `\_ \exp_stop_f: 25080
25081 \tex_advance:D \l__tl_analysis_type_int \l__tl_analysis_type_int 25081
25082 \fi: 25082
25083 \tex_skip:D \l__tl_analysis_index_int 25083
25084 = \l__tl_analysis_normal_int sp 25084
25085 plus \l__tl_analysis_type_int sp \scan_stop: 25085
25086 \int_incr:N \l__tl_analysis_index_int 25086
25087 \int_zero:N \l__tl_analysis_normal_int 25087
25088 \if_int_compare:w \l__tl_analysis_nesting_int = - \c_one_int 25088
25089 \cs_set_eq:NN \__tl_analysis_a_loop:w \scan_stop: 25089
25090 \fi: 25090
25091 } 25091
25092 \cs_new_protected:Npn \__tl_analysis_a_safe:N #1 25092
25093 { 25093
25094 \if_charcode:w 25094
25095 \scan_stop: 25095
25096 \exp_after:wN \use_none:n \token_to_str:N #1 \prg_do_nothing: 25096
25097 \scan_stop: 25097
25098 \exp_after:wN \use_i:nn 25098
25099 \else: 25099
25100 \exp_after:wN \use_ii:nn 25100
25101 \fi: 25101

```

```

25102 {
25103     \_tl_analysis_disable_char:N #1
25104     \int_incr:N \l__tl_analysis_normal_int
25105 }
25106 { \_tl_analysis_cs_space_count:NN \_tl_analysis_a_cs:ww #1 }
25107 \_tl_analysis_a_loop:w
25108 }
25109 \cs_new_protected:Npn \_tl_analysis_a_cs:ww #1 \_tl_sep: #2 \_tl_sep:
25110 {
25111     \if_int_compare:w #1 > \c_zero_int
25112         \tex_skip:D \l__tl_analysis_index_int
25113         = \int_eval:n { \l__tl_analysis_normal_int + 1 } sp \exp_stop_f:
25114         \tex_advance:D \l__tl_analysis_index_int #1 \exp_stop_f:
25115     \else:
25116         \tex_advance:D
25117     \fi:
25118     \l__tl_analysis_normal_int #2 \exp_stop_f:
25119 }
25120 \cs_new_protected:Npn \_tl_analysis_b:n #1
25121 {
25122     \__kernel_tl_gset:Nx \g__tl_analysis_result_tl
25123     {
25124         \_tl_analysis_b_loop:w 0 \_tl_sep: #1
25125         \prg_break_point:
25126     }
25127 }
25128 \cs_new:Npn \_tl_analysis_b_loop:w #1 \_tl_sep:
25129 {
25130     \exp_after:wN \_tl_analysis_b_normals:ww
25131     \int_value:w \tex_skip:D #1 \_tl_sep: #1 \_tl_sep:
25132 }
25133 \cs_new:Npn \_tl_analysis_b_normals:ww #1 \_tl_sep:
25134 {
25135     \if_int_compare:w #1 = \c_zero_int
25136         \_tl_analysis_b_special:w
25137     \fi:
25138     \_tl_analysis_b_normal:wwN #1 \_tl_sep:
25139 }
25140 \cs_new:Npn \_tl_analysis_b_normal:wwN #1 \_tl_sep: #2 \_tl_sep: #3
25141 {
25142     \exp_not:n { \exp_not:n { #3 } } \s__tl
25143     \if_charcode:w
25144         \scan_stop:
25145         \exp_after:wN \use_none:n \token_to_str:N #3 \prg_do_nothing:
25146         \scan_stop:
25147         \exp_after:wN \_tl_analysis_b_char:Nn

```

```

25148 \exp_after:wN \__tl_analysis_b_char_aux:nww 25148
25149 \else: 25149
25150 \exp_after:wN \__tl_analysis_b_cs:Nww 25150
25151 \fi: 25151
25152 #3 #1 \__tl_sep: #2 \__tl_sep: 25152
25153 } 25153
25154 \cs_new:Npe \__tl_analysis_b_char:Nn #1#2 25154
25155 { 25155
25156 \exp_not:N \if_meaning:w #2 \exp_not:N \tex_undefined:D 25156
25157 \token_to_str:N D \exp_not:N \else: 25157
25158 \exp_not:N \if_catcode:w #2 \c_catcode_other_token 25158
25159 \token_to_str:N C \exp_not:N \else: 25159
25160 \exp_not:N \if_catcode:w #2 \c_catcode_letter_token 25160
25161 \token_to_str:N B \exp_not:N \else: 25161
25162 \exp_not:N \if_catcode:w #2 \c_math_toggle_token 3 25162
25163 \exp_not:N \else: 25163
25164 \exp_not:N \if_catcode:w #2 \c_alignment_token 4 25164
25165 \exp_not:N \else: 25165
25166 \exp_not:N \if_catcode:w #2 \c_math_superscript_token 7 25166
25167 \exp_not:N \else: 25167
25168 \exp_not:N \if_catcode:w #2 \c_math_subscript_token 8 25168
25169 \exp_not:N \else: 25169
25170 \exp_not:N \if_catcode:w #2 \c_space_token 25170
25171 \token_to_str:N A \exp_not:N \else: 25171
25172 6 25172
25173 \exp_not:n { \fi: \fi: \fi: \fi: \fi: \fi: \fi: \fi: } 25173
25174 #1 {#2} 25174
25175 } 25175
25176 \cs_new:Npn \__tl_analysis_b_char_aux:nww #1 25176
25177 { 25177
25178 \int_value:w `#1 \s__tl 25178
25179 \exp_after:wN \__tl_analysis_b_normals:ww 25179
25180 \int_value:w \int_eval:w - 1 + 25180
25181 } 25181
25182 \cs_new:Npn \__tl_analysis_b_cs:Nww #1 25182
25183 { 25183
25184 0 -1 \s__tl 25184
25185 \__tl_analysis_cs_space_count:NN \__tl_analysis_b_cs_test:ww #1 25185
25186 } 25186
25187 \cs_new:Npn \__tl_analysis_b_cs_test:ww 25187
25188 #1 \__tl_sep: #2 \__tl_sep: #3 \__tl_sep: #4 \__tl_sep: 25188
25189 { 25189
25190 \exp_after:wN \__tl_analysis_b_normals:ww 25190
25191 \int_value:w \int_eval:w 25191
25192 \if_int_compare:w #1 = \c_zero_int 25192
25193 #3 25193

```

```

25194 \else: 25194
25195 \tex_skip:D \int_eval:n { #4 + #1 } \exp_stop_f: 25195
25196 \fi: 25196
25197 - #2 25197
25198 \exp_after:wN \__tl_sep: 25198
25199 \int_value:w \int_eval:n { #4 + #1 } \__tl_sep: 25199
25200 } 25200
25201 \group_begin: 25201
25202 \char_set_catcode_other:N A 25202
25203 \cs_new:Npn \__tl_analysis_b_special:w 25203
25204 \fi: \__tl_analysis_b_normal:wwN 0 \__tl_sep: #1 \__tl_sep: 25204
25205 { 25205
25206 \fi: 25206
25207 \if_int_compare:w #1 = \l__tl_analysis_index_int 25207
25208 \exp_after:wN \prg_break: 25208
25209 \fi: 25209
25210 \tex_the:D \tex_toks:D #1 \s__tl 25210
25211 \if_case:w \tex_gluestretch:D \tex_skip:D #1 \exp_stop_f: 25211
25212 \token_to_str:N A 25212
25213 \or: 1 25213
25214 \or: 1 25214
25215 \else: 2 25215
25216 \fi: 25216
25217 \if_int_odd:w \tex_gluestretch:D \tex_skip:D #1 \exp_stop_f: 25217
25218 \exp_after:wN \__tl_analysis_b_special_char:wN \int_value:w 25218
25219 \else: 25219
25220 \exp_after:wN \__tl_analysis_b_special_space:w \int_value:w 25220
25221 \fi: 25221
25222 \int_eval:n { 1 + #1 } \exp_after:wN \__tl_sep: 25222
25223 \token_to_str:N 25223
25224 } 25224
25225 \group_end: 25225
25226 \cs_new:Npn \__tl_analysis_b_special_char:wN #1 \__tl_sep: #2 25226
25227 { 25227
25228 \int_value:w `#2 \s__tl 25228
25229 \__tl_analysis_b_loop:w #1 \__tl_sep: 25229
25230 } 25230
25231 \use:e 25231
25232 { 25232
25233 \cs_new:Npn \exp_not:N \__tl_analysis_b_special_space:w 25233
25234 #1 \exp_not:N \__tl_sep: \c_space_tl 25234
25235 } 25235
25236 { 25236
25237 32 \s__tl 25237
25238 \__tl_analysis_b_loop:w #1 \__tl_sep: 25238
25239 } 25239

```

```
25240 \cs_new_protected:Npn \tl_analysis_map_inline:Nn #1
25241 { \exp_args:No \tl_analysis_map_inline:nn #1 }
25242 \cs_new_protected:Npn \tl_analysis_map_inline:nn #1
25243 {
25244   \__tl_analysis:n {#1}
25245   \int_gincr:N \g__kernel_prg_map_int
25246   \exp_args:Nc \__tl_analysis_map:Nn
25247     { \__tl_analysis_map_inline_ \int_use:N \g__kernel_prg_map_int :wNw }
25248 }
25249 \cs_new_protected:Npn \__tl_analysis_map:Nn #1#2
25250 {
25251   \cs_gset_protected:Npn #1 ##1##2##3 {#2}
25252   \exp_after:wN \__tl_analysis_map:NwNw \exp_after:wN #1
25253   \g__tl_analysis_result_tl
25254   \s__tl { ? \tl_map_break: } \s__tl
25255   \prg_break_point:Nn \tl_map_break:
25256   { \int_gdecr:N \g__kernel_prg_map_int }
25257 }
25258 \cs_new_protected:Npn \__tl_analysis_map:NwNw #1 #2 \s__tl #3 #4 \s__tl
25259 {
25260   \use_none:n #3
25261   #1 {#2} {#4} {#3}
25262   \__tl_analysis_map:NwNw #1
25263 }
25264 \cs_new_protected:Npn \tl_analysis_show:N
25265 { \__tl_analysis_show:NNnnN \msg_show:nneeee \tl_show:N {} {} }
25266 \cs_new_protected:Npn \tl_analysis_log:N
25267 {
25268   \__tl_analysis_show:NNnnN \msg_log:nneeee \tl_log:N
25269   { \iow_newline: >~ . } { . }
25270 }
25271 \cs_new_protected:Npn \__tl_analysis_show:NNnnN #1#2#3#4#5
25272 {
25273   \tl_if_exist:NTF #5
25274   {
25275     \exp_args:No \__tl_analysis:n {#5}
25276     #1 { tl } { show-analysis }
25277     { \token_to_str:N #5 } { \__tl_analysis_show: } {#3} {#4}
25278   }
25279   { #2 #3 }
25280 }
25281 \cs_new_protected:Npn \tl_analysis_show:n
25282 { \__tl_analysis_show:Nnnn \msg_show:nneeee {} {} }
25283 \cs_new_protected:Npn \tl_analysis_log:n
25284 { \__tl_analysis_show:Nnnn \msg_log:nneeee { \iow_newline: >~ . } { . } }
25285 \cs_new_protected:Npn \__tl_analysis_show:Nnnn #1#2#3#4
```

```

25286 {
25287     \__tl_analysis:n {#4}
25288     #1 { tl } { show-analysis } { } { \__tl_analysis_show: } {#2} {#3}
25289 }
25290 \cs_new:Npn \__tl_analysis_show:
25291 {
25292     \exp_after:wN \__tl_analysis_show_loop:wNw \g__tl_analysis_result_tl
25293     \s__tl { ? \prg_break: } \s__tl
25294     \prg_break_point:
25295 }
25296 \cs_new:Npn \__tl_analysis_show_loop:wNw #1 \s__tl #2 #3 \s__tl
25297 {
25298     \use_none:n #2
25299     \iow_newline: > \use:nn { ~ } { ~ }
25300     \if_int_compare:w "#2 = \c_zero_int
25301         \exp_after:wN \__tl_analysis_show_cs:n
25302     \else:
25303         \if_int_compare:w "#2 = 13 \exp_stop_f:
25304             \exp_after:wN \exp_after:wN
25305             \exp_after:wN \__tl_analysis_show_active:n
25306         \else:
25307             \exp_after:wN \exp_after:wN
25308             \exp_after:wN \__tl_analysis_show_normal:n
25309         \fi:
25310     \fi:
25311     {#1}
25312     \__tl_analysis_show_loop:wNw
25313 }
25314 \cs_new:Npn \__tl_analysis_show_normal:n #1
25315 {
25316     \exp_after:wN \token_to_str:N #1 ~
25317     ( \exp_after:wN \token_to_meaning:N #1 )
25318 }
25319 \cs_new:Npn \__tl_analysis_show_value:N #1
25320 {
25321     \token_if_expandable:NF #1
25322     {
25323         \token_if_chardef:NTF #1 \prg_break: { }
25324         \token_if_mathchardef:NTF #1 \prg_break: { }
25325         \token_if_dim_register:NTF #1 \prg_break: { }
25326         \token_if_int_register:NTF #1 \prg_break: { }
25327         \token_if_skip_register:NTF #1 \prg_break: { }
25328         \token_if_toks_register:NTF #1 \prg_break: { }
25329         \use_none:nnn
25330         \prg_break_point:
25331         \use:n { \exp_after:wN = \tex_the:D #1 }

```

```

25332     }
25333 }
25334 \cs_new:Npn \__tl_analysis_show_cs:n #1
25335 { \exp_args:No \__tl_analysis_show_long:nn {#1} { control~sequence= } }
25336 \cs_new:Npn \__tl_analysis_show_active:n #1
25337 { \exp_args:No \__tl_analysis_show_long:nn {#1} { active~character= } }
25338 \cs_new:Npn \__tl_analysis_show_long:nn #1
25339 {
25340     \__tl_analysis_show_long_aux:oofn
25341     { \token_to_str:N #1 }
25342     { \token_to_meaning:N #1 }
25343     { \__tl_analysis_show_value:N #1 }
25344 }
25345 \cs_new:Npn \__tl_analysis_show_long_aux:nnnn #1#2#3#4
25346 {
25347     \int_compare:nNnTF
25348     { \str_count:n { #1 ~ ( #4 #2 #3 ) } }
25349     > { \l_iow_line_count_int - 3 }
25350     {
25351         \str_range:nnn { #1 ~ ( #4 #2 #3 ) } { 1 }
25352         {
25353             \l_iow_line_count_int - 3
25354             - \str_count:N \c__tl_analysis_show_etc_str
25355         }
25356         \c__tl_analysis_show_etc_str
25357     }
25358     { #1 ~ ( #4 #2 #3 ) }
25359 }
25360 \cs_generate_variant:Nn \__tl_analysis_show_long_aux:nnnn { oof }
25361 \cs_new:Npn \peek_analysis_map_break:
25362 { \prg_map_break:Nn \peek_analysis_map_break: { } }
25363 \cs_new:Npn \peek_analysis_map_break:n
25364 { \prg_map_break:Nn \peek_analysis_map_break: }
25365 \int_new:N \l__tl_peek_charcode_int
25366 \cs_new:Npn \__tl_analysis_char_arg:Nw
25367 {
25368     \if_meaning:w \l__tl_analysis_token \c_space_token
25369     \exp_after:wN \__tl_analysis_char_arg_aux:Nw
25370     \fi:
25371 }
25372 \cs_new:Npn \__tl_analysis_char_arg_aux:Nw #1 ~ { #1 { ~ } }
25373 \cs_new_protected:Npn \peek_analysis_map_inline:n #1
25374 {
25375     \group_align_safe_begin:
25376     \int_gincr:N \g__kernel_prg_map_int
25377     \cs_set_protected:cpn

```



```

25378 { __tl_analysis_map_ \int_use:N \g__kernel_prg_map_int :nnN } 25378
25379 ##1##2##3 25379
25380 { 25380
25381 \group_end: 25381
25382 #1 25382
25383 \__tl_peek_analysis_loop:NNn 25383
25384 \prg_break_point:Nn \peek_analysis_map_break: 25384
25385 { 25385
25386 \int_gdecr:N \g__kernel_prg_map_int 25386
25387 \group_align_safe_end: 25387
25388 } 25388
25389 } 25389
25390 \__tl_peek_analysis_loop:NNn ? ? ? 25390
25391 } 25391
25392 \cs_new_protected:Npn \__tl_peek_analysis_loop:NNn #1#2#3 25392
25393 { 25393
25394 \group_begin: 25394
25395 \tl_set:Ne \l__tl_peek_code_tl 25395
25396 { 25396
25397 \exp_not:c 25397
25398 { __tl_analysis_map_ \int_use:N \g__kernel_prg_map_int :nnN } 25398
25399 } 25399
25400 \int_set:Nn \tex_escapechar:D { ``\ } 25400
25401 \peek_after:Nw \__tl_peek_analysis_test: 25401
25402 } 25402
25403 \cs_new_protected:Npn \__tl_peek_analysis_test: 25403
25404 { 25404
25405 \if_case:w 25405
25406 \if_catcode:w \exp_not:N \l_peek_token { \c_max_int \fi: 25406
25407 \if_catcode:w \exp_not:N \l_peek_token } \c_max_int \fi: 25407
25408 \if_meaning:w \l_peek_token \c_space_token \c_max_int \fi: 25408
25409 \exp_after:wN \if_meaning:w \exp_not:N \l_peek_token \l_peek_token 25409
25410 \c_one_int 25410
25411 \fi: 25411
25412 \c_zero_int 25412
25413 \exp_after:wN \exp_after:wN 25413
25414 \exp_after:wN \__tl_peek_analysis_exp:N 25414
25415 \exp_after:wN \exp_not:N 25415
25416 \or: 25416
25417 \exp_after:wN \__tl_peek_analysis_nonexp:N 25417
25418 \else: 25418
25419 \exp_after:wN \__tl_peek_analysis_special: 25419
25420 \fi: 25420
25421 } 25421
25422 \cs_new_protected:Npn \__tl_peek_analysis_exp:N #1 25422
25423 { 25423

```

```
25424 \cs_set_nopar:Npe \l__tl_peek_code_tl 25424
25425 { 25425
25426 \tex_let:D \exp_not:N #1 \l__tl_peek_code_tl 25426
25427 \l__tl_peek_code_tl 25427
25428 { 25428
25429 \exp_not:n { \__kernel_exp_not:w \exp_after:wN } 25429
25430 { \exp_not:N \exp_not:N \exp_not:N #1 } 25430
25431 } 25431
25432 \exp_after:wN \__tl_peek_analysis_exp_aux:Nw 25432
25433 \token_to_str:N #1 { } \s__tl 25433
25434 } 25434
25435 \l__tl_peek_code_tl 25435
25436 } 25436
25437 \cs_new:Npe \__tl_peek_analysis_exp_aux:Nw #1#2 \s__tl 25437
25438 { 25438
25439 \exp_not:N \if_meaning:w \scan_stop: #2 \scan_stop: 25439
25440 { \exp_not:N \int_value:w `#1 ~ } \token_to_str:N D 25440
25441 \exp_not:N \else: 25441
25442 { -1 } 0 25442
25443 \exp_not:N \fi: 25443
25444 } 25444
25445 \cs_new_protected:Npn \__tl_peek_analysis_nonexp:N #1 25445
25446 { 25446
25447 \if_charcode:w 25447
25448 \scan_stop: 25448
25449 \exp_after:wN \use_none:n \token_to_str:N #1 \prg_do_nothing: 25449
25450 \scan_stop: 25450
25451 \exp_after:wN \__tl_peek_analysis_char:N 25451
25452 \else: 25452
25453 \exp_after:wN \__tl_peek_analysis_cs:N 25453
25454 \fi: 25454
25455 #1 25455
25456 } 25456
25457 \cs_new_protected:Npn \__tl_peek_analysis_cs:N #1 25457
25458 { \l__tl_peek_code_tl { \exp_not:n {#1} } { -1 } 0 } 25458
25459 \group_begin: 25459
25460 \char_set_active_eq:NN \__scan_stop: 25460
25461 \cs_new_protected:Npe \__tl_peek_analysis_char:N #1 25461
25462 { 25462
25463 \cs_set_eq:NN 25463
25464 \char_generate:nn { 32 } { 13 } 25464
25465 \exp_not:N \tex_undefined:D 25465
25466 \tex_lccode:D `#1 = 32 \exp_stop_f: 25466
25467 \tex_lowercase:D 25467
25468 { 25468
25469 \tl_put_right:Ne \exp_not:N \l__tl_peek_code_tl 25469
```

```

25470         { \exp_not:n { \__tl_analysis_b_char:Nn \use_none:n } {#1} }
25471     }
25472 \exp_not:n
25473 {
25474     \exp_after:wN \__tl_peek_analysis_char:w
25475     \int_value:w
25476 }
25477 `#1
25478 \exp_not:n { \exp_after:wN \s__tl \l__tl_peek_code_tl }
25479 #1
25480 }
25481 \group_end:
25482 \cs_new_protected:Npn \__tl_peek_analysis_char:w #1 \s__tl #2#3#4
25483 {
25484     \if_charcode:w 6 #3
25485     \else:
25486         \if_charcode:w D #3
25487         \else:
25488             \exp_args:NNNo
25489             \fi:
25490         \fi:
25491         #2 { \exp_not:n {#4} } {#1} #3
25492     }
25493 \cs_new_protected:Npn \__tl_peek_analysis_special:
25494 {
25495     \tex_let:D \l__tl_analysis_token = ~ \l_peek_token
25496     \int_set:Nn \l__tl_peek_charcode_int
25497     { \__tl_analysis_extract_charcode: }
25498     \if_int_compare:w \l__tl_peek_charcode_int = \tex_escapechar:D
25499     \int_set:Nn \tex_escapechar:D { ` \_ }
25500     \fi:
25501     \char_set_active_eq:nN { \l__tl_peek_charcode_int } \scan_stop:
25502     \char_set_active_eq:nN { \tex_escapechar:D } \scan_stop:
25503     \cs_set_eq:cN { } \scan_stop:
25504     \tex_futurelet:D \l__tl_analysis_token
25505     \__tl_peek_analysis_retest:
25506 }
25507 \cs_new_protected:Npn \__tl_peek_analysis_retest:
25508 {
25509     \if_meaning:w \l__tl_analysis_token \scan_stop:
25510         \exp_after:wN \__tl_peek_analysis_nonexp:N
25511     \else:
25512         \exp_after:wN \__tl_peek_analysis_str:
25513     \fi:
25514 }
25515 \cs_new_protected:Npn \__tl_peek_analysis_str:

```

```

25516 {
25517     \exp_after:wN \tex_futurelet:D
25518     \exp_after:wN \l__tl_analysis_token
25519     \exp_after:wN \__tl_peek_analysis_str:w
25520     \token_to_str:N
25521 }
25522 \cs_new_protected:Npn \__tl_peek_analysis_str:w
25523 { \__tl_analysis_char_arg:Nw \__tl_peek_analysis_str:n }
25524 \cs_new_protected:Npn \__tl_peek_analysis_str:n #1
25525 {
25526     \int_case:nnF { `#1 }
25527     {
25528         { \l__tl_peek_charcode_int }
25529         { \__tl_peek_analysis_explicit:n {#1} }
25530         { \tex_escapechar:D } { \__tl_peek_analysis_escape: }
25531     }
25532     { \__tl_peek_analysis_active_str:n {#1} }
25533 }
25534 \cs_new_protected:Npn \__tl_peek_analysis_active_str:n #1
25535 {
25536     \tl_put_right:Ne \l__tl_peek_code_tl
25537     {
25538         { \char_generate:nn { `#1 } { 13 } }
25539         { \int_value:w `#1 }
25540         \token_to_str:N D
25541     }
25542     \l__tl_peek_code_tl
25543 }
25544 \cs_new_protected:Npn \__tl_peek_analysis_explicit:n #1
25545 {
25546     \tl_put_right:Ne \l__tl_peek_code_tl
25547     {
25548         \if_meaning:w \l_peek_token \c_space_token
25549             { ~ } { 32 } \token_to_str:N A
25550         \else:
25551             \if_catcode:w \l_peek_token \c_group_begin_token
25552             {
25553                 \exp_not:N \exp_after:wN
25554                 \char_generate:nn { `#1 } { 1 }
25555                 \exp_not:N \if_false:
25556                 \if_false: { \fi: }
25557                 \exp_not:N \fi:
25558             }
25559             { \int_value:w `#1 }
25560             1
25561         \else:

```

```

25562         {
25563             \exp_not:N \if_false:
25564             { \if_false: } \fi:
25565             \exp_not:N \fi:
25566             \char_generate:nn { `#1 } { 2 }
25567         }
25568         { \int_value:w `#1 }
25569         2
25570         \fi:
25571         \fi:
25572     }
25573     \l__tl_peek_code_tl
25574 }
25575 \cs_new_protected:Npn \__tl_peek_analysis_escape:
25576 {
25577     \tl_clear:N \l__tl_internal_a_tl
25578     \tex_futurelet:D \l__tl_analysis_token
25579     \__tl_peek_analysis_collect:w
25580 }
25581 \cs_new_protected:Npn \__tl_peek_analysis_collect:w
25582 { \__tl_analysis_char_arg:Nw \__tl_peek_analysis_collect:n }
25583 \cs_new_protected:Npn \__tl_peek_analysis_collect:n #1
25584 {
25585     \tl_put_right:Nn \l__tl_internal_a_tl {#1}
25586     \__tl_peek_analysis_collect_loop:
25587 }
25588 \cs_new_protected:Npn \__tl_peek_analysis_collect_loop:
25589 {
25590     \exp_after:wN \if_meaning:w
25591     \cs:w
25592     \if_cs_exist:w \l__tl_internal_a_tl \cs_end:
25593     \l__tl_internal_a_tl
25594     \else:
25595         c_one % anything short
25596     \fi:
25597     \cs_end:
25598     \l_peek_token
25599     \__tl_peek_analysis_collect_end:NNNN
25600     \fi:
25601     \tex_futurelet:D \l__tl_analysis_token
25602     \__tl_peek_analysis_collect:w
25603 }
25604 \cs_new_protected:Npn \__tl_peek_analysis_collect_end:NNNN #1#2#3#4
25605 {
25606     #1
25607     \tl_put_right:Ne \l__tl_peek_code_tl

```

```
25608     { 25608
25609         { \exp_not:N \exp_not:n { \exp_not:c { \l__tl_internal_a_tl } } } 25609
25610         { -1 } 25610
25611         0 25611
25612     } 25612
25613     \l__tl_peek_code_tl 25613
25614 } 25614
25615 \tl_const:Ne \c__tl_analysis_show_etc_str % ( 25615
25616     { \token_to_str:N \ETC.) } 25616
25617 \msg_new:nnn { tl } { show-analysis } 25617
25618 { 25618
25619     The~token~list~ \tl_if_empty:nF {#1} { #1 ~ } 25619
25620     \tl_if_empty:nTF {#2} 25620
25621     { is~empty #3 } 25621
25622     { contains~the~tokens: #2 #4 } 25622
25623 } 25623
25624 %% File: l3benchmark.dtx 25624
25625 \fp_new:N \g_benchmark_duration_target_fp 25625
25626 \fp_gset:Nn \g_benchmark_duration_target_fp { 1 } 25626
25627 \int_new:N \g__benchmark_nesting_int 25627
25628 \cs_new_protected:Npn \__benchmark_raw:nN #1 25628
25629 { 25629
25630     \int_gincr:N \g__benchmark_nesting_int 25630
25631     \exp_args:Nc \__benchmark_raw_aux:N 25631
25632     { g__benchmark_ \int_use:N \g__benchmark_nesting_int _int } 25632
25633     \__benchmark_raw_aux: 25633
25634     #1 25634
25635     \__benchmark_raw_end:N 25635
25636 } 25636
25637 \cs_new_protected:Npn \__benchmark_raw_aux:N #1 25637
25638 { 25638
25639     \int_gzero_new:N #1 25639
25640     \cs_gset_protected:Npn \__benchmark_raw_aux: { \int_gset:Nn #1 { \sys_timer: } } 25640
25641 } 25641
25642 \cs_new_protected:Npn \__benchmark_raw_end:N #1 25642
25643 { 25643
25644     \int_gset:Nn #1 25644
25645     { 25645
25646         \sys_timer: - 25646
25647         \int_use:c { g__benchmark_ \int_use:N \g__benchmark_nesting_int _int } 25647
25648     } 25648
25649     \int_gdecr:N \g__benchmark_nesting_int 25649
25650 } 25650
25651 \cs_new_eq:NN \__benchmark_tmp:w ? 25651
25652 \cs_new_protected:Npn \__benchmark_raw_replicate:nnN #1 25652
25653 { 25653
```

25654	\int_compare:nNnTF {#1} > { 500000 }	25654
25655	{ __benchmark_raw_replicate_large:nnN {#1} }	25655
25656	{ __benchmark_raw_replicate_small:nnN {#1} }	25656
25657	}	25657
25658	\cs_new_protected:Npn __benchmark_raw_replicate_large:nnN #1#2	25658
25659	{	25659
25660	\cs_set:Npe __benchmark_tmp:w ##1 { \prg_replicate:nn { 5000 } {##1} }	25660
25661	\exp_args:Nno __benchmark_raw_replicate:nnN { #1 / 5000 }	25661
25662	{ __benchmark_tmp:w {#2} }	25662
25663	}	25663
25664	\cs_new_protected:Npn __benchmark_raw_replicate_small:nnN #1#2	25664
25665	{	25665
25666	\cs_set:Npe __benchmark_tmp:w ##1##2 { \prg_replicate:nn {#1} {##1} }	25666
25667	__benchmark_raw:nN { __benchmark_tmp:w {#2} { } } \g__benchmark_time_int	25667
25668	\exp_args:No __benchmark_raw_replicate_aux:nnN	25668
25669	{ \int_use:N \g__benchmark_time_int } {#2}	25669
25670	}	25670
25671	\cs_new_protected:Npn __benchmark_raw_replicate_aux:nnN #1#2#3	25671
25672	{	25672
25673	__benchmark_raw:nN { __benchmark_tmp:w { } {#2} } \g__benchmark_time_int	25673
25674	\int_gset:Nn #3 { #1 - \g__benchmark_time_int }	25674
25675	\cs_set_eq:NN __benchmark_tmp:w \prg_do_nothing:	25675
25676	}	25676
25677	\fp_new:N \g_benchmark_time_fp	25677
25678	\fp_new:N \g_benchmark_ops_fp	25678
25679	\int_new:N \g__benchmark_duration_int	25679
25680	\int_new:N \g__benchmark_time_int	25680
25681	\int_new:N \g__benchmark_time_a_int	25681
25682	\int_new:N \g__benchmark_time_b_int	25682
25683	\int_new:N \g__benchmark_time_c_int	25683
25684	\int_new:N \g__benchmark_time_d_int	25684
25685	\int_new:N \g__benchmark_repeat_int	25685
25686	\tl_new:N \g__benchmark_code_tl	25686
25687	\cs_new_protected:Npn \benchmark_once:n #1	25687
25688	{	25688
25689	\benchmark_once_silent:n {#1}	25689
25690	__benchmark_display:	25690
25691	}	25691
25692	\cs_new_protected:Npn \benchmark_once_silent:n #1	25692
25693	{	25693
25694	__benchmark_measure_op:	25694
25695	__benchmark_raw:nN {#1} \g__benchmark_time_int	25695
25696	\fp_gset:Nn \g_benchmark_time_fp { \g__benchmark_time_int / 65536 }	25696
25697	\fp_gset:Nn \g_benchmark_ops_fp { \g_benchmark_time_fp / \g__benchmark_one_op_fp }	25697
25698	}	25698
25699	\cs_new_protected:Npn \benchmark:n #1	25699


```
25700 {
25701     \benchmark_silent:n {#1}
25702     \__benchmark_display:
25703 }
25704 \cs_new_protected:Npn \benchmark_silent:n #1
25705 {
25706     \__benchmark_measure_op:
25707     \tl_gset:Nn \g__benchmark_code_tl {#1}
25708     \__benchmark_aux:
25709     \fp_gset:Nn \g_benchmark_ops_fp { \g_benchmark_time_fp / \g__benchmark_one_op_fp }
25710 }
25711 \cs_new_protected:Npn \__benchmark_aux:
25712 {
25713     \int_gset:Nn \g__benchmark_repeat_int { 1 }
25714     \__benchmark_raw:nN { \g__benchmark_code_tl } \g__benchmark_time_int
25715     \int_compare:nNnF \g__benchmark_time_int < { \g__benchmark_duration_int / 2 }
25716         { \prg_break: }
25717     \bool_until_do:nn
25718     {
25719         \int_compare_p:nNn \g__benchmark_time_int > { \g__benchmark_duration_int / 32 }
25720         || \int_compare_p:nNn \g__benchmark_repeat_int > { \c_max_int / 4 }
25721     }
25722     {
25723         \int_gset:Nn \g__benchmark_repeat_int { 4 * \g__benchmark_repeat_int }
25724         \__benchmark_run:N \g__benchmark_time_int
25725     }
25726     \int_gset:Nn \g__benchmark_repeat_int
25727     {
25728         \fp_to_int:n
25729         {
25730             max ( 1 , min ( \c_max_int ,
25731                 \g__benchmark_duration_int * \g__benchmark_repeat_int /
25732                 \int_eval:n { 4 * \g__benchmark_time_int } ) )
25733         }
25734     }
25735     \int_compare:nNnTF \g__benchmark_repeat_int = 1
25736     { \int_gset_eq:NN \g__benchmark_time_a_int \g__benchmark_time_int }
25737     { \__benchmark_run:N \g__benchmark_time_a_int }
25738     \__benchmark_run:N \g__benchmark_time_b_int
25739     \__benchmark_run:N \g__benchmark_time_c_int
25740     \__benchmark_run:N \g__benchmark_time_d_int
25741     \int_gset:Nn \g__benchmark_time_int
25742     {
25743         \int_min:nn
25744         { \int_min:nn \g__benchmark_time_a_int \g__benchmark_time_b_int }
25745         { \int_min:nn \g__benchmark_time_c_int \g__benchmark_time_d_int }
```

```
25746 } 25746
25747 \prg_break_point: 25747
25748 \int_compare:nNnT \g__benchmark_time_int < 3 { \int_gzero:N \g__benchmark_time_int } 25748
25749 \fp_gset:Nn \g_benchmark_time_fp 25749
25750 { \g__benchmark_time_int / \g__benchmark_repeat_int / 65536 } 25750
25751 } 25751
25752 \cs_new_protected:Npn \__benchmark_run:N 25752
25753 { \exp_args:NNo \__benchmark_raw_replicate:nnN \g__benchmark_repeat_int { 25753
25754 \g__benchmark_code_tl } } 25754
25755 \fp_new:N \g__benchmark_one_op_fp 25755
25756 \cs_new_protected:Npn \__benchmark_measure_op: 25756
25757 { 25757
25758 \int_gset:Nn \g__benchmark_duration_int 25758
25759 { \fp_to_int:n { 65536 * \g_benchmark_duration_target_fp } / 4 } 25759
25760 \tl_gset:Nn \g__benchmark_code_tl 25760
25761 { \int_gadd:Nn \g__benchmark_duration_int { 0 } } 25761
25762 \__benchmark_aux: 25762
25763 \fp_gset:Nn \g__benchmark_one_op_fp { max(\g_benchmark_time_fp, 1e-8) } 25763
25764 \int_gset:Nn \g__benchmark_duration_int 25764
25765 { \fp_to_int:n { 65536 * \g_benchmark_duration_target_fp } } 25765
25766 } 25766
25767 \cs_new:Npn \__benchmark_fp_to_tl:N #1 25767
25768 { 25768
25769 \fp_compare:nTF { abs(#1) < 1000 } 25769
25770 { \fp_to_tl:n { round(#1, 2 - logb(#1)) } } 25770
25771 { 25771
25772 \exp_args:Nf \__benchmark_fp_to_tl_aux:nN 25772
25773 { \fp_to_int:n { logb(#1) } } #1 25773
25774 } 25774
25775 } 25775
25776 \cs_new:Npn \__benchmark_fp_to_tl_aux:nN #1#2 25776
25777 { \fp_to_tl:n { round(#2 * 1e-#1, 2) } e#1 } 25777
25778 \cs_new_protected:Npn \__benchmark_display: 25778
25779 { 25779
25780 \iow_term:e 25780
25781 { 25781
25782 \__benchmark_fp_to_tl:N \g_benchmark_time_fp \c_space_tl seconds \c_space_tl 25782
25783 ( \__benchmark_fp_to_tl:N \g_benchmark_ops_fp \c_space_tl ops) 25783
25784 } 25784
25785 } 25785
25786 \int_new:N \g__benchmark_tictoc_int 25786
25787 \seq_new:N \g__benchmark_tictoc_seq 25787
25788 \tl_new:N \l__benchmark_tictoc_pop_tl 25788
25789 \cs_new:Npn \__benchmark_tictoc_prefix: 25789
25790 { 25790
25791 (l3benchmark) \c_space_tl 25791
```

```

25791 + \prg_replicate:nn { \g__benchmark_tictoc_int } { --+ } \c_space_tl 25791
25792 } 25792
25793 \cs_new_protected:Npn \benchmark_tic: 25793
25794 { 25794
25795     \iow_term:e { \__benchmark_tictoc_prefix: TIC } 25795
25796     \exp_args:NNf \seq_gput_right:Nn \g__benchmark_tictoc_seq { \sys_timer: } 25796
25797     \int_gincr:N \g__benchmark_tictoc_int 25797
25798 } 25798
25799 \cs_new:Npn \benchmark_toc: 25799
25800 { 25800
25801     \seq_gpop_right:NNTF \g__benchmark_tictoc_seq \l__benchmark_tictoc_pop_tl 25801
25802     { \__benchmark_toc: } 25802
25803     { \msg_error:nn { benchmark } { toc-first } } 25803
25804 } 25804
25805 \cs_new_protected:Npn \__benchmark_toc: 25805
25806 { 25806
25807     \int_gdecr:N \g__benchmark_tictoc_int 25807
25808     \fp_gset:Nn \g_benchmark_time_fp 25808
25809     { ( \sys_timer: - \l__benchmark_tictoc_pop_tl ) / 65536 } 25809
25810     \iow_term:e 25810
25811     { 25811
25812         \__benchmark_tictoc_prefix: 25812
25813         TOC: \c_space_tl 25813
25814         \__benchmark_fp_to_tl:N \g_benchmark_time_fp \c_space_tl s 25814
25815     } 25815
25816 } 25816
25817 \msg_new:nnn { benchmark } { toc-first } 25817
25818 { 25818
25819     \token_to_str:N \benchmark_toc: \c_space_tl without~ 25819
25820     \token_to_str:N \benchmark_tic: \c_space_tl ! 25820
25821 } 25821
25822 %% File: l3regex.dtx 25822
25823 \cs_new_eq:NN \__regex_int_eval:w \tex_numexpr:D 25823
25824 \cs_new_eq:NN \__regex_sep: \__kernel_int_sep: 25824
25825 \cs_new_protected:Npn \__regex_standard_escapechar: 25825
25826 { \int_set:Nn \tex_escapechar:D { ``\ } } 25826
25827 \cs_new:Npn \__regex_toks_use:w { \tex_the:D \tex_toks:D } 25827
25828 \cs_new_protected:Npn \__regex_toks_clear:N #1 25828
25829 { \tex_toks:D #1 = { } } 25829
25830 \cs_new_eq:NN \__regex_toks_set:Nn \tex_toks:D 25830
25831 \cs_new_protected:Npn \__regex_toks_set:No #1 25831
25832 { \tex_toks:D #1 = \exp_after:wN } 25832
25833 \cs_new_protected:Npn \__regex_toks_memcpy:Nn #1#2#3 25833
25834 { 25834
25835     \prg_replicate:nn {#3} 25835
25836     { 25836

```

```
25837         \tex_toks:D #1 = \tex_toks:D #2
25838         \int_incr:N #1
25839         \int_incr:N #2
25840     }
25841 }
25842 \cs_if_exist:NTF \tex_etokspre:D
25843 { \cs_new_eq:NN \__regex_toks_put_left:Ne \tex_etokspre:D }
25844 {
25845     \cs_new_protected:Npn \__regex_toks_put_left:Ne #1#2
25846     { \tex_toks:D #1 = \tex_expanded:D {{ #2 \tex_the:D \tex_toks:D #1 }} }
25847 }
25848 \cs_if_exist:NTF \tex_etoksapp:D
25849 { \cs_new_eq:NN \__regex_toks_put_right:Ne \tex_etoksapp:D }
25850 {
25851     \cs_new_protected:Npn \__regex_toks_put_right:Ne #1#2
25852     { \tex_toks:D #1 = \tex_expanded:D {{ \tex_the:D \tex_toks:D #1 #2 }} }
25853 }
25854 \cs_if_exist:NTF \tex_toksapp:D
25855 { \cs_new_eq:NN \__regex_toks_put_right:Nn \tex_toksapp:D }
25856 {
25857     \cs_new_protected:Npn \__regex_toks_put_right:Nn #1#2
25858     { \tex_toks:D #1 = \exp_after:wN { \tex_the:D \tex_toks:D #1 #2 } }
25859 }
25860 \cs_new:Npn \__regex_curr_cs_to_str:
25861 {
25862     \exp_after:wN \exp_after:wN \exp_after:wN \cs_to_str:N
25863     \l__regex_curr_token_tl
25864 }
25865 \cs_new:Npn \__regex_intarray_item:NnF #1#2
25866 { \exp_args:No \__regex_intarray_item_aux:nNF { \tex_the:D \__regex_int_eval:w #2 }
25867 #1 }
25867 \cs_new:Npn \__regex_intarray_item_aux:nNF #1#2
25868 {
25869     \if_int_compare:w #1 > \c_zero_int
25870     \exp_after:wN \use_ii:nnn
25871     \fi:
25872     \use_ii:nn { \__kernel_intarray_item:Nn #2 {#1} }
25873 }
25874 \cs_new:Npn \__regex_maplike_break:
25875 { \prg_map_break:Nn \__regex_maplike_break: { } }
25876 \cs_new:Npn \__regex_tl_odd_items:n #1 { \__regex_tl_even_items:n { ? #1 } }
25877 \cs_new:Npn \__regex_tl_even_items:n #1
25878 {
25879     \__regex_tl_even_items_loop:nn #1 \q__regex_nil \q__regex_nil
25880     \prg_break_point:
25881 }
```

25882	\cs_new:Npn __regex_tl_even_items_loop:nn #1#2	25882
25883	{	25883
25884	__regex_use_none_delimit_by_q_nil:w #2 \prg_break: \q__regex_nil	25884
25885	{ \exp_not:n {#2} }	25885
25886	__regex_tl_even_items_loop:nn	25886
25887	}	25887
25888	\cs_new:Npn __regex_tmp:w { }	25888
25889	\tl_new:N \l__regex_internal_a_tl	25889
25890	\tl_new:N \l__regex_internal_b_tl	25890
25891	\int_new:N \l__regex_internal_a_int	25891
25892	\int_new:N \l__regex_internal_b_int	25892
25893	\int_new:N \l__regex_internal_c_int	25893
25894	\bool_new:N \l__regex_internal_bool	25894
25895	\seq_new:N \l__regex_internal_seq	25895
25896	\tl_new:N \g__regex_internal_tl	25896
25897	\tl_new:N \l__regex_build_tl	25897
25898	\tl_const:Nn \c__regex_no_match_regex	25898
25899	{	25899
25900	__regex_branch:n	25900
25901	{ __regex_class:NnnnN \c_true_bool { } { 1 } { 0 } \c_true_bool }	25901
25902	}	25902
25903	\int_new:N \l__regex_balance_int	25903
25904	\int_const:Nn \c__regex_ascii_min_int { 0 }	25904
25905	\int_const:Nn \c__regex_ascii_max_control_int { 31 }	25905
25906	\int_const:Nn \c__regex_ascii_max_int { 127 }	25906
25907	\int_const:Nn \c__regex_ascii_lower_int { `a - `A }	25907
25908	\quark_new:N \q__regex_recursion_stop	25908
25909	\quark_new:N \q__regex_nil	25909
25910	\cs_new:Npn __regex_use_none_delimit_by_q_recursion_stop:w	25910
25911	#1 \q__regex_recursion_stop { }	25911
25912	\cs_new:Npn __regex_use_i_delimit_by_q_recursion_stop:nw	25912
25913	#1 #2 \q__regex_recursion_stop {#1}	25913
25914	\cs_new:Npn __regex_use_none_delimit_by_q_nil:w #1 \q__regex_nil { }	25914
25915	__kernel_quark_new_conditional:Nn __regex_quark_if_nil:N { F }	25915
25916	\cs_new_protected:Npn __regex_break_true:w	25916
25917	#1 __regex_break_point:TF #2 #3 {#2}	25917
25918	\cs_new_protected:Npn __regex_break_point:TF #1 #2 { #2 }	25918
25919	\cs_new_protected:Npn __regex_item_reverse:n #1	25919
25920	{	25920
25921	#1	25921
25922	__regex_break_point:TF { } __regex_break_true:w	25922
25923	}	25923
25924	\cs_new_protected:Npn __regex_item_caseful_equal:n #1	25924
25925	{	25925
25926	\if_int_compare:w #1 = \l__regex_curr_char_int	25926
25927	\exp_after:wN __regex_break_true:w	25927

```
25928 \fi: 25928
25929 } 25929
25930 \cs_new_protected:Npn \__regex_item_caseful_range:nn #1 #2 25930
25931 { 25931
25932 \reverse_if:N \if_int_compare:w #1 > \l__regex_curr_char_int 25932
25933 \reverse_if:N \if_int_compare:w #2 < \l__regex_curr_char_int 25933
25934 \exp_after:wN \exp_after:wN \exp_after:wN \__regex_break_true:w 25934
25935 \fi: 25935
25936 \fi: 25936
25937 } 25937
25938 \cs_new_protected:Npn \__regex_item_caseless_equal:n #1 25938
25939 { 25939
25940 \if_int_compare:w #1 = \l__regex_curr_char_int 25940
25941 \exp_after:wN \__regex_break_true:w 25941
25942 \fi: 25942
25943 \__regex_maybe_compute_ccc: 25943
25944 \if_int_compare:w #1 = \l__regex_case_changed_char_int 25944
25945 \exp_after:wN \__regex_break_true:w 25945
25946 \fi: 25946
25947 } 25947
25948 \cs_new_protected:Npn \__regex_item_caseless_range:nn #1 #2 25948
25949 { 25949
25950 \reverse_if:N \if_int_compare:w #1 > \l__regex_curr_char_int 25950
25951 \reverse_if:N \if_int_compare:w #2 < \l__regex_curr_char_int 25951
25952 \exp_after:wN \exp_after:wN \exp_after:wN \__regex_break_true:w 25952
25953 \fi: 25953
25954 \fi: 25954
25955 \__regex_maybe_compute_ccc: 25955
25956 \reverse_if:N \if_int_compare:w #1 > \l__regex_case_changed_char_int 25956
25957 \reverse_if:N \if_int_compare:w #2 < \l__regex_case_changed_char_int 25957
25958 \exp_after:wN \exp_after:wN \exp_after:wN \__regex_break_true:w 25958
25959 \fi: 25959
25960 \fi: 25960
25961 } 25961
25962 \cs_new_protected:Npn \__regex_compute_case_changed_char: 25962
25963 { 25963
25964 \int_set_eq:NN \l__regex_case_changed_char_int \l__regex_curr_char_int 25964
25965 \if_int_compare:w \l__regex_curr_char_int > `Z \exp_stop_f: 25965
25966 \if_int_compare:w \l__regex_curr_char_int > `z \exp_stop_f: \else: 25966
25967 \if_int_compare:w \l__regex_curr_char_int < `a \exp_stop_f: \else: 25967
25968 \int_sub:Nn \l__regex_case_changed_char_int \c__regex_ascii_lower_int 25968
25969 \fi: 25969
25970 \fi: 25970
25971 \else: 25971
25972 \if_int_compare:w \l__regex_curr_char_int < `A \exp_stop_f: \else: 25972
25973 \int_add:Nn \l__regex_case_changed_char_int \c__regex_ascii_lower_int 25973
```



```

25974 \fi:
25975 \fi:
25976 \cs_set_eq:NN \__regex_maybe_compute_ccc: \prg_do_nothing:
25977 }
25978 \cs_new_eq:NN \__regex_maybe_compute_ccc: \__regex_compute_case_changed_char:
25979 \cs_new_eq:NN \__regex_item_equal:n ?
25980 \cs_new_eq:NN \__regex_item_range:nn ?
25981 \cs_new_protected:Npn \__regex_item_catcode:
25982 {
25983 "
25984 \if_case:w \l__regex_curr_catcode_int
25985 1 \or: 4 \or: 10 \or: 40
25986 \or: 100 \or: \or: 1000 \or: 4000
25987 \or: 10000 \or: \or: 100000 \or: 400000
25988 \or: 1000000 \or: 4000000 \else: 1*0
25989 \fi:
25990 }
25991 \prg_new_protected_conditional:Npnn \__regex_item_catcode:n #1 { T }
25992 {
25993 \if_int_odd:w \__regex_int_eval:w #1 / \__regex_item_catcode: \scan_stop:
25994 \prg_return_true:
25995 \else:
25996 \prg_return_false:
25997 \fi:
25998 }
25999 \cs_new_protected:Npn \__regex_item_catcode_reverse:nT #1#2
26000 { \__regex_item_catcode:nT {#1} { \__regex_item_reverse:n {#2} } }
26001 \cs_new_protected:Npn \__regex_item_exact:nn #1#2
26002 {
26003 \if_int_compare:w #1 = \l__regex_curr_catcode_int
26004 \if_int_compare:w #2 = \l__regex_curr_char_int
26005 \exp_after:wN \exp_after:wN \exp_after:wN \__regex_break_true:w
26006 \fi:
26007 \fi:
26008 }
26009 \cs_new_protected:Npn \__regex_item_exact_cs:n #1
26010 {
26011 \int_compare:nNnTF \l__regex_curr_catcode_int = \c_zero_int
26012 {
26013 \__kernel_tl_set:Nx \l__regex_internal_a_tl
26014 { \scan_stop: \__regex_curr_cs_to_str: \scan_stop: }
26015 \tl_if_in:noTF { \scan_stop: #1 \scan_stop: }
26016 \l__regex_internal_a_tl
26017 { \__regex_break_true:w } { }
26018 }
26019 { }

```



```
26020 }
26021 \cs_new_protected:Npn \__regex_item_cs:n #1
26022 {
26023   \int_compare:nNnT \l__regex_curr_catcode_int = \c_zero_int
26024   {
26025     \group_begin:
26026       \__regex_single_match:
26027       \__regex_disable_submatches:
26028       \__regex_build_for_cs:n {#1}
26029       \bool_set_eq:NN \l__regex_saved_success_bool
26030       \g__regex_success_bool
26031       \exp_args:Ne \__regex_match_cs:n { \__regex_curr_cs_to_str: }
26032       \if_meaning:w \c_true_bool \g__regex_success_bool
26033       \group_insert_after:N \__regex_break_true:w
26034       \fi:
26035       \bool_gset_eq:NN \g__regex_success_bool
26036       \l__regex_saved_success_bool
26037     \group_end:
26038   }
26039 }
26040 \cs_new_protected:Npn \__regex_prop_d:
26041 { \__regex_item_caseful_range:nn { `0 } { `9 } }
26042 \cs_new_protected:Npn \__regex_prop_h:
26043 {
26044   \__regex_item_caseful_equal:n { `\_ }
26045   \__regex_item_caseful_equal:n { `^~I }
26046 }
26047 \cs_new_protected:Npn \__regex_prop_s:
26048 {
26049   \__regex_item_caseful_equal:n { `\_ }
26050   \__regex_item_caseful_equal:n { `^~I }
26051   \__regex_item_caseful_equal:n { `^~J }
26052   \__regex_item_caseful_equal:n { `^~L }
26053   \__regex_item_caseful_equal:n { `^~M }
26054 }
26055 \cs_new_protected:Npn \__regex_prop_v:
26056 { \__regex_item_caseful_range:nn { `^~J } { `^~M } } % lf, vtab, ff, cr
26057 \cs_new_protected:Npn \__regex_prop_w:
26058 {
26059   \__regex_item_caseful_range:nn { `a } { `z }
26060   \__regex_item_caseful_range:nn { `A } { `Z }
26061   \__regex_item_caseful_range:nn { `0 } { `9 }
26062   \__regex_item_caseful_equal:n { `_ }
26063 }
26064 \cs_new_protected:Npn \__regex_prop_N:
26065 {
```

```
26066 \__regex_item_reverse:n
26067 { \__regex_item_caseful_equal:n { `^^J } }
26068 }
26069 \cs_new_protected:Npn \__regex_posix_alnum:
26070 { \__regex_posix_alpha: \__regex_posix_digit: }
26071 \cs_new_protected:Npn \__regex_posix_alpha:
26072 { \__regex_posix_lower: \__regex_posix_upper: }
26073 \cs_new_protected:Npn \__regex_posix_ascii:
26074 {
26075 \__regex_item_caseful_range:nn
26076 \c__regex_ascii_min_int
26077 \c__regex_ascii_max_int
26078 }
26079 \cs_new_eq:NN \__regex_posix_blank: \__regex_prop_h:
26080 \cs_new_protected:Npn \__regex_posix_cntrl:
26081 {
26082 \__regex_item_caseful_range:nn
26083 \c__regex_ascii_min_int
26084 \c__regex_ascii_max_control_int
26085 \__regex_item_caseful_equal:n \c__regex_ascii_max_int
26086 }
26087 \cs_new_eq:NN \__regex_posix_digit: \__regex_prop_d:
26088 \cs_new_protected:Npn \__regex_posix_graph:
26089 { \__regex_item_caseful_range:nn { `! } { `~ } }
26090 \cs_new_protected:Npn \__regex_posix_lower:
26091 { \__regex_item_caseful_range:nn { `a } { `z } }
26092 \cs_new_protected:Npn \__regex_posix_print:
26093 { \__regex_item_caseful_range:nn { `\_ } { `~ } }
26094 \cs_new_protected:Npn \__regex_posix_punct:
26095 {
26096 \__regex_item_caseful_range:nn { `! } { `/ }
26097 \__regex_item_caseful_range:nn { `: } { `@ }
26098 \__regex_item_caseful_range:nn { `[ } { `` }
26099 \__regex_item_caseful_range:nn { `{ } { `~ }
26100 }
26101 \cs_new_protected:Npn \__regex_posix_space:
26102 {
26103 \__regex_item_caseful_equal:n { `\_ }
26104 \__regex_item_caseful_range:nn { `^^I } { `^^M }
26105 }
26106 \cs_new_protected:Npn \__regex_posix_upper:
26107 { \__regex_item_caseful_range:nn { `A } { `Z } }
26108 \cs_new_eq:NN \__regex_posix_word: \__regex_prop_w:
26109 \cs_new_protected:Npn \__regex_posix_xdigit:
26110 {
26111 \__regex_posix_digit:
```

```
26112 \__regex_item_caseful_range:nn { `A } { `F } 26112
26113 \__regex_item_caseful_range:nn { `a } { `f } 26113
26114 } 26114
26115 \cs_new_protected:Npn \__regex_escape_use:nnnn #1#2#3#4 26115
26116 { 26116
26117 \group_begin: 26117
26118 \tl_clear:N \l__regex_internal_a_tl 26118
26119 \cs_set:Npn \__regex_escape_unescaped:N ##1 { #1 } 26119
26120 \cs_set:Npn \__regex_escape_escaped:N ##1 { #2 } 26120
26121 \cs_set:Npn \__regex_escape_raw:N ##1 { #3 } 26121
26122 \__regex_standard_escapechar: 26122
26123 \__kernel_tl_gset:Nx \g__regex_internal_tl 26123
26124 { \__kernel_str_to_other_fast:n {#4} } 26124
26125 \tl_put_right:Ne \l__regex_internal_a_tl 26125
26126 { 26126
26127 \exp_after:wN \__regex_escape_loop:N \g__regex_internal_tl 26127
26128 \scan_stop: \prg_break_point: 26128
26129 } 26129
26130 \exp_after:wN 26130
26131 \group_end: 26131
26132 \l__regex_internal_a_tl 26132
26133 } 26133
26134 \cs_new:Npn \__regex_escape_loop:N #1 26134
26135 { 26135
26136 \cs_if_exist_use:cF { __regex_escape_\token_to_str:N #1:w } 26136
26137 { \__regex_escape_unescaped:N #1 } 26137
26138 \__regex_escape_loop:N 26138
26139 } 26139
26140 \cs_new:cpn { __regex_escape_ \c_backslash_str :w } 26140
26141 \__regex_escape_loop:N #1 26141
26142 { 26142
26143 \cs_if_exist_use:cF { __regex_escape_\token_to_str:N #1:w } 26143
26144 { \__regex_escape_escaped:N #1 } 26144
26145 \__regex_escape_loop:N 26145
26146 } 26146
26147 \cs_new_eq:NN \__regex_escape_unescaped:N ? 26147
26148 \cs_new_eq:NN \__regex_escape_escaped:N ? 26148
26149 \cs_new_eq:NN \__regex_escape_raw:N ? 26149
26150 \cs_new_eq:cN { __regex_escape_ \iow_char:N\scan_stop: :w } \prg_break: 26150
26151 \cs_new:cpn { __regex_escape_/ \iow_char:N\scan_stop: :w } 26151
26152 { 26152
26153 \msg_expandable_error:nn { regex } { trailing-backslash } 26153
26154 \prg_break: 26154
26155 } 26155
26156 \cs_new:cpn { __regex_escape_~:w } { } 26156
26157 \cs_new:cpe { __regex_escape_/a:w } 26157
```

```
26158 { \exp_not:N \__regex_escape_raw:N \iow_char:N \^^G } 26158
26159 \cs_new:cpe { __regex_escape_/t:w } 26159
26160 { \exp_not:N \__regex_escape_raw:N \iow_char:N \^^I } 26160
26161 \cs_new:cpe { __regex_escape_/n:w } 26161
26162 { \exp_not:N \__regex_escape_raw:N \iow_char:N \^^J } 26162
26163 \cs_new:cpe { __regex_escape_/f:w } 26163
26164 { \exp_not:N \__regex_escape_raw:N \iow_char:N \^^L } 26164
26165 \cs_new:cpe { __regex_escape_/r:w } 26165
26166 { \exp_not:N \__regex_escape_raw:N \iow_char:N \^^M } 26166
26167 \cs_new:cpe { __regex_escape_/e:w } 26167
26168 { \exp_not:N \__regex_escape_raw:N \iow_char:N \^^[ } 26168
26169 \cs_new:cpn { __regex_escape_/x:w } \__regex_escape_loop:N 26169
26170 { 26170
26171 \exp_after:wN \__regex_escape_x_end:w 26171
26172 \int_value:w "0 \__regex_escape_x_test:N 26172
26173 } 26173
26174 \cs_new:Npn \__regex_escape_x_end:w #1 \__regex_sep: 26174
26175 { 26175
26176 \int_compare:nNnTF {#1} > \c_max_char_int 26176
26177 { 26177
26178 \msg_expandable_error:nnff { regex } { x-overflow } 26178
26179 {#1} { \int_to_Hex:n {#1} } 26179
26180 } 26180
26181 { 26181
26182 \exp_last_unbraced:Nf \__regex_escape_raw:N 26182
26183 { \char_generate:nn {#1} { 12 } } 26183
26184 } 26184
26185 } 26185
26186 \cs_new:Npn \__regex_escape_x_test:N #1 26186
26187 { 26187
26188 \if_meaning:w \scan_stop: #1 26188
26189 \exp_after:wN \use_i:nnn \exp_after:wN \__regex_sep: 26189
26190 \fi: 26190
26191 \use:n 26191
26192 { 26192
26193 \if_charcode:w \c_space_token #1 26193
26194 \exp_after:wN \__regex_escape_x_test:N 26194
26195 \else: 26195
26196 \exp_after:wN \__regex_escape_x_testii:N 26196
26197 \exp_after:wN #1 26197
26198 \fi: 26198
26199 } 26199
26200 } 26200
26201 \cs_new:Npn \__regex_escape_x_testii:N #1 26201
26202 { 26202
26203 \if_charcode:w \c_left_brace_str #1 26203
```

```
26204 \exp_after:wN \__regex_escape_x_loop:N 26204
26205 \else: 26205
26206 \__regex_hexadecimal_use:NTF #1 26206
26207 { \exp_after:wN \__regex_escape_x:N } 26207
26208 { \__regex_sep: \exp_after:wN \__regex_escape_loop:N \exp_after:wN #1 } 26208
26209 \fi: 26209
26210 } 26210
26211 \cs_new:Npn \__regex_escape_x:N #1 26211
26212 { 26212
26213 \if_meaning:w \scan_stop: #1 26213
26214 \exp_after:wN \use_i:nnn \exp_after:wN \__regex_sep: 26214
26215 \fi: 26215
26216 \use:n 26216
26217 { 26217
26218 \__regex_hexadecimal_use:NTF #1 26218
26219 { \__regex_sep: \__regex_escape_loop:N } 26219
26220 { \__regex_sep: \__regex_escape_loop:N #1 } 26220
26221 } 26221
26222 } 26222
26223 \cs_new:Npn \__regex_escape_x_loop:N #1 26223
26224 { 26224
26225 \if_meaning:w \scan_stop: #1 26225
26226 \exp_after:wN \use_ii:nnn 26226
26227 \fi: 26227
26228 \use_ii:nn 26228
26229 { \__regex_sep: \__regex_escape_x_loop_error:n { } {#1} } 26229
26230 { 26230
26231 \__regex_hexadecimal_use:NTF #1 26231
26232 { \__regex_escape_x_loop:N } 26232
26233 { 26233
26234 \token_if_eq_charcode:NNTF \c_space_token #1 26234
26235 { \__regex_escape_x_loop:N } 26235
26236 { 26236
26237 \__regex_sep: 26237
26238 \exp_after:wN 26238
26239 \token_if_eq_charcode:NNTF \c_right_brace_str #1 26239
26240 { \__regex_escape_loop:N } 26240
26241 { \__regex_escape_x_loop_error:n {#1} } 26241
26242 } 26242
26243 } 26243
26244 } 26244
26245 } 26245
26246 \cs_new:Npn \__regex_escape_x_loop_error:n #1 26246
26247 { 26247
26248 \msg_expandable_error:nnn { regex } { x-missing-rbrace } {#1} 26248
26249 \__regex_escape_loop:N #1 26249
```

```
26250 } 26250
26251 \cs_new:Npn \__regex_hexadecimal_use:NTF #1 26251
26252 { 26252
26253 \if_int_compare:w \c_one_int < "1 \token_to_str:N #1 \exp_stop_f: 26253
26254 #1 26254
26255 \else: 26255
26256 \if_case:w 26256
26257 \__regex_int_eval:w \exp_after:wN ` \token_to_str:N #1 - `a \scan_stop: 26257
26258 A 26258
26259 \or: B 26259
26260 \or: C 26260
26261 \or: D 26261
26262 \or: E 26262
26263 \or: F 26263
26264 \else: 26264
26265 \exp_after:wN \exp_after:wN \exp_after:wN \use_iii:nnn 26265
26266 \fi: 26266
26267 \fi: 26267
26268 \use_i:nn 26268
26269 } 26269
26270 \prg_new_conditional:Npnn \__regex_char_if_special:N #1 { TF } 26270
26271 { 26271
26272 \if:w 26272
26273 T 26273
26274 \if_int_compare:w `#1 > `Z \exp_stop_f: 26274
26275 \if_int_compare:w `#1 > `z \exp_stop_f: 26275
26276 \if_int_compare:w `#1 < \c__regex_ascii_max_int 26276
26277 \else: F \fi: 26277
26278 \else: 26278
26279 \if_int_compare:w `#1 < `a \exp_stop_f: 26279
26280 \else: F \fi: 26280
26281 \fi: 26281
26282 \else: 26282
26283 \if_int_compare:w `#1 > `9 \exp_stop_f: 26283
26284 \if_int_compare:w `#1 < `A \exp_stop_f: 26284
26285 \else: F \fi: 26285
26286 \else: 26286
26287 \if_int_compare:w `#1 < `0 \exp_stop_f: 26287
26288 \if_int_compare:w `#1 < `_\exp_stop_f: 26288
26289 F \fi: 26289
26290 \else: F \fi: 26290
26291 \fi: 26291
26292 \fi: 26292
26293 T 26293
26294 \prg_return_true: 26294
26295 \else: 26295
```

26296	\prg_return_false:	26296
26297	\fi:	26297
26298	}	26298
26299	\prg_new_conditional:Npnn __regex_char_if_alphanumeric:N #1 { TF }	26299
26300	{	26300
26301	\if:w	26301
26302	T	26302
26303	\if_int_compare:w `#1 > `Z \exp_stop_f:	26303
26304	\if_int_compare:w `#1 > `z \exp_stop_f:	26304
26305	F	26305
26306	\else:	26306
26307	\if_int_compare:w `#1 < `a \exp_stop_f:	26307
26308	F \fi:	26308
26309	\fi:	26309
26310	\else:	26310
26311	\if_int_compare:w `#1 > `9 \exp_stop_f:	26311
26312	\if_int_compare:w `#1 < `A \exp_stop_f:	26312
26313	F \fi:	26313
26314	\else:	26314
26315	\if_int_compare:w `#1 < `0 \exp_stop_f:	26315
26316	F \fi:	26316
26317	\fi:	26317
26318	\fi:	26318
26319	T	26319
26320	\prg_return_true:	26320
26321	\else:	26321
26322	\prg_return_false:	26322
26323	\fi:	26323
26324	}	26324
26325	\int_new:N \l__regex_group_level_int	26325
26326	\int_new:N \l__regex_mode_int	26326
26327	\int_const:Nn \c__regex_cs_in_class_mode_int { -6 }	26327
26328	\int_const:Nn \c__regex_cs_mode_int { -2 }	26328
26329	\int_const:Nn \c__regex_outer_mode_int { 0 }	26329
26330	\int_const:Nn \c__regex_catcode_mode_int { 2 }	26330
26331	\int_const:Nn \c__regex_class_mode_int { 3 }	26331
26332	\int_const:Nn \c__regex_catcode_in_class_mode_int { 6 }	26332
26333	\int_new:N \l__regex_catcodes_int	26333
26334	\int_new:N \l__regex_default_catcodes_int	26334
26335	\bool_new:N \l__regex_catcodes_bool	26335
26336	\int_const:Nn \c__regex_catcode_C_int { "1 }	26336
26337	\int_const:Nn \c__regex_catcode_B_int { "4 }	26337
26338	\int_const:Nn \c__regex_catcode_E_int { "10 }	26338
26339	\int_const:Nn \c__regex_catcode_M_int { "40 }	26339
26340	\int_const:Nn \c__regex_catcode_T_int { "100 }	26340
26341	\int_const:Nn \c__regex_catcode_P_int { "1000 }	26341

26342	\int_const:Nn \c__regex_catcode_U_int { "4000 }	26342
26343	\int_const:Nn \c__regex_catcode_D_int { "10000 }	26343
26344	\int_const:Nn \c__regex_catcode_S_int { "100000 }	26344
26345	\int_const:Nn \c__regex_catcode_L_int { "400000 }	26345
26346	\int_const:Nn \c__regex_catcode_O_int { "1000000 }	26346
26347	\int_const:Nn \c__regex_catcode_A_int { "4000000 }	26347
26348	\int_const:Nn \c__regex_all_catcodes_int { "5515155 }	26348
26349	\cs_new_eq:NN \l__regex_internal_regex \c__regex_no_match_regex	26349
26350	\seq_new:N \l__regex_show_prefix_seq	26350
26351	\int_new:N \l__regex_show_lines_int	26351
26352	\cs_new:Npn __regex_two_if_eq:NNNTF #1#2#3#4	26352
26353	{	26353
26354	\if_meaning:w #1 #3	26354
26355	\if:w #2 #4	26355
26356	\exp_after:wN \exp_after:wN \exp_after:wN \use_ii:nnn	26356
26357	\fi:	26357
26358	\fi:	26358
26359	\use_ii:nn	26359
26360	}	26360
26361	\cs_new_protected:Npn __regex_get_digits:NTFw #1#2#3#4#5	26361
26362	{	26362
26363	__regex_if_raw_digit:NNTF #4 #5	26363
26364	{ #1 = #5 __regex_get_digits_loop:nw {#2} }	26364
26365	{ #3 #4 #5 }	26365
26366	}	26366
26367	\cs_new:Npn __regex_get_digits_loop:nw #1#2#3	26367
26368	{	26368
26369	__regex_if_raw_digit:NNTF #2 #3	26369
26370	{ #3 __regex_get_digits_loop:nw {#1} }	26370
26371	{ \scan_stop: #1 #2 #3 }	26371
26372	}	26372
26373	\cs_new:Npn __regex_if_raw_digit:NNTF #1#2	26373
26374	{	26374
26375	\if_meaning:w __regex_compile_raw:N #1	26375
26376	\if_int_compare:w \c_one_int < 1 #2 \exp_stop_f:	26376
26377	\exp_after:wN \exp_after:wN \exp_after:wN \use_ii:nnn	26377
26378	\fi:	26378
26379	\fi:	26379
26380	\use_ii:nn	26380
26381	}	26381
26382	\prg_new_conditional:Npnn __regex_if_in_class: { TF }	26382
26383	{	26383
26384	\if_int_odd:w \l__regex_mode_int	26384
26385	\prg_return_true:	26385
26386	\else:	26386
26387	\prg_return_false:	26387

26388	\fi:	26388
26389	}	26389
26390	\cs_new:Npn __regex_if_in_cs:TF	26390
26391	{	26391
26392	\if_int_odd:w \l__regex_mode_int	26392
26393	\else:	26393
26394	\if_int_compare:w \l__regex_mode_int < \c__regex_outer_mode_int	26394
26395	\exp_after:wN \exp_after:wN \exp_after:wN \use_ii:nnn	26395
26396	\fi:	26396
26397	\fi:	26397
26398	\use_ii:nn	26398
26399	}	26399
26400	\cs_new:Npn __regex_if_in_class_or_catcode:TF	26400
26401	{	26401
26402	\if_int_odd:w \l__regex_mode_int	26402
26403	\else:	26403
26404	\if_int_compare:w \l__regex_mode_int > \c__regex_outer_mode_int	26404
26405	\else:	26405
26406	\exp_after:wN \exp_after:wN \exp_after:wN \use_iii:nnn	26406
26407	\fi:	26407
26408	\fi:	26408
26409	\use_i:nn	26409
26410	}	26410
26411	\prg_new_conditional:Npnn __regex_if_within_catcode: { TF }	26411
26412	{	26412
26413	\if_int_compare:w \l__regex_mode_int > \c__regex_outer_mode_int	26413
26414	\prg_return_true:	26414
26415	\else:	26415
26416	\prg_return_false:	26416
26417	\fi:	26417
26418	}	26418
26419	\cs_new_protected:Npn __regex_chk_c_allowed:T	26419
26420	{	26420
26421	\if_int_compare:w \l__regex_mode_int = \c__regex_outer_mode_int	26421
26422	\else:	26422
26423	\if_int_compare:w \l__regex_mode_int = \c__regex_class_mode_int	26423
26424	\else:	26424
26425	\msg_error:nn { regex } { c-bad-mode }	26425
26426	\exp_after:wN \use_i:nnn	26426
26427	\fi:	26427
26428	\fi:	26428
26429	\use:n	26429
26430	}	26430
26431	\cs_new_protected:Npn __regex_mode_quit_c:	26431
26432	{	26432
26433	\if_int_compare:w \l__regex_mode_int = \c__regex_catcode_mode_int	26433

```

26434 \int_set_eq:NN \l__regex_mode_int \c__regex_outer_mode_int 26434
26435 \else: 26435
26436 \if_int_compare:w \l__regex_mode_int = 26436
26437 \c__regex_catcode_in_class_mode_int 26437
26438 \int_set_eq:NN \l__regex_mode_int \c__regex_class_mode_int 26438
26439 \fi: 26439
26440 \fi: 26440
26441 } 26441
26442 \cs_new_protected:Npn \__regex_compile:w 26442
26443 { 26443
26444 \group_begin: 26444
26445 \tl_build_begin:N \l__regex_build_tl 26445
26446 \int_zero:N \l__regex_group_level_int 26446
26447 \int_set_eq:NN \l__regex_default_catcodes_int 26447
26448 \c__regex_all_catcodes_int 26448
26449 \int_set_eq:NN \l__regex_catcodes_int \l__regex_default_catcodes_int 26449
26450 \cs_set:Npn \__regex_item_equal:n { \__regex_item_caseful_equal:n } 26450
26451 \cs_set:Npn \__regex_item_range:nn { \__regex_item_caseful_range:nn } 26451
26452 \tl_build_put_right:Nn \l__regex_build_tl 26452
26453 { \__regex_branch:n { \if_false: } \fi: } 26453
26454 } 26454
26455 \cs_new_protected:Npn \__regex_compile_end: 26455
26456 { 26456
26457 \__regex_if_in_class:TF 26457
26458 { 26458
26459 \msg_error:nn { regex } { missing-rbrack } 26459
26460 \use:c { __regex_compile_]: } 26460
26461 \prg_do_nothing: \prg_do_nothing: 26461
26462 } 26462
26463 { } 26463
26464 \if_int_compare:w \l__regex_group_level_int > \c_zero_int 26464
26465 \msg_error:nne { regex } { missing-rparen } 26465
26466 { \int_use:N \l__regex_group_level_int } 26466
26467 \prg_replicate:nn 26467
26468 \l__regex_group_level_int 26468
26469 { 26469
26470 \tl_build_put_right:Nn \l__regex_build_tl 26470
26471 { 26471
26472 \if_false: { \fi: } 26472
26473 \if_false: { \fi: } { 1 } { 0 } \c_true_bool 26473
26474 } 26474
26475 \tl_build_end:N \l__regex_build_tl 26475
26476 \exp_args:NNNo 26476
26477 \group_end: 26477
26478 \tl_build_put_right:Nn \l__regex_build_tl 26478
26479 { \l__regex_build_tl } 26479

```

```
26480     }
26481     \fi:
26482     \tl_build_put_right:Nn \l__regex_build_tl { \if_false: { \fi: } }
26483     \tl_build_end:N \l__regex_build_tl
26484     \exp_args:NNNe
26485     \group_end:
26486     \tl_set:Nn \l__regex_internal_regex { \l__regex_build_tl }
26487 }
26488 \cs_new_protected:Npn \__regex_compile:n #1
26489 {
26490     \__regex_compile:w
26491     \__regex_standard_escapechar:
26492     \int_set_eq:NN \l__regex_mode_int \c__regex_outer_mode_int
26493     \__regex_escape_use:nnnn
26494     {
26495         \__regex_char_if_special:NTF ##1
26496         \__regex_compile_special:N \__regex_compile_raw:N ##1
26497     }
26498     {
26499         \__regex_char_if_alphanumeric:NTF ##1
26500         \__regex_compile_escaped:N \__regex_compile_raw:N ##1
26501     }
26502     { \__regex_compile_raw:N ##1 }
26503     { #1 }
26504     \prg_do_nothing: \prg_do_nothing:
26505     \prg_do_nothing: \prg_do_nothing:
26506     \int_compare:nNnT \l__regex_mode_int = \c__regex_catcode_mode_int
26507     { \msg_error:nn { regex } { c-trailing } }
26508     \int_compare:nNnT \l__regex_mode_int < \c__regex_outer_mode_int
26509     {
26510         \msg_error:nn { regex } { c-missing-rbrace }
26511         \__regex_compile_end_cs:
26512         \prg_do_nothing: \prg_do_nothing:
26513         \prg_do_nothing: \prg_do_nothing:
26514     }
26515     \__regex_compile_end:
26516 }
26517 \cs_new_protected:Npn \__regex_compile_use:n #1
26518 {
26519     \tl_if_single_token:nT {#1}
26520     {
26521         \exp_after:wN \__regex_compile_use_aux:w
26522         \token_to_meaning:N #1 ~ \q__regex_nil
26523     }
26524     \__regex_compile:n {#1} \l__regex_internal_regex
26525 }
```

```
26526 \cs_new_protected:Npn \__regex_compile_use_aux:w #1 ~ #2 \q__regex_nil 26526
26527 { 26527
26528 \str_if_eq:nnT { #1 ~ } { macro:->\__regex_branch:n } 26528
26529 { \use_ii:nnn } 26529
26530 } 26530
26531 \cs_new_protected:Npn \__regex_compile_special:N #1 26531
26532 { 26532
26533 \cs_if_exist_use:cF { __regex_compile_#1: } 26533
26534 { \__regex_compile_raw:N #1 } 26534
26535 } 26535
26536 \cs_new_protected:Npn \__regex_compile_escaped:N #1 26536
26537 { 26537
26538 \cs_if_exist_use:cF { __regex_compile_/#1: } 26538
26539 { \__regex_compile_raw:N #1 } 26539
26540 } 26540
26541 \cs_new_protected:Npn \__regex_compile_one:n #1 26541
26542 { 26542
26543 \__regex_mode_quit_c: 26543
26544 \__regex_if_in_class:TF { } 26544
26545 { 26545
26546 \tl_build_put_right:Nn \l__regex_build_tl 26546
26547 { \__regex_class:NnnnN \c_true_bool { \if_false: } \fi: } 26547
26548 } 26548
26549 \tl_build_put_right:Ne \l__regex_build_tl 26549
26550 { 26550
26551 \if_int_compare:w \l__regex_catcodes_int < 26551
26552 \c__regex_all_catcodes_int 26552
26553 \__regex_item_catcode:nT { \int_use:N \l__regex_catcodes_int } 26553
26554 { \exp_not:N \exp_not:n {#1} } 26554
26555 \else: 26555
26556 \exp_not:N \exp_not:n {#1} 26556
26557 \fi: 26557
26558 } 26558
26559 \int_set_eq:NN \l__regex_catcodes_int \l__regex_default_catcodes_int 26559
26560 \__regex_if_in_class:TF { } { \__regex_compile_quantifier:w } 26560
26561 } 26561
26562 \cs_new_protected:Npn \__regex_compile_abort_tokens:n #1 26562
26563 { 26563
26564 \use:e 26564
26565 { 26565
26566 \exp_args:No \tl_map_function:nN { \tl_to_str:n {#1} } 26566
26567 \__regex_compile_raw:N 26567
26568 } 26568
26569 } 26569
26570 \cs_generate_variant:Nn \__regex_compile_abort_tokens:n { e } 26570
26571 \cs_new_protected:Npn \__regex_compile_if_quantifier:TFw #1#2#3#4 26571
```

```
26572 { 26572
26573 \token_if_eq_meaning:NNTF #3 \__regex_compile_special:N 26573
26574 { \cs_if_exist:cTF { __regex_compile_quantifier_#4:w } } 26574
26575 { \use_ii:nn } 26575
26576 {#1} {#2} #3 #4 26576
26577 } 26577
26578 \cs_new_protected:Npn \__regex_compile_quantifier:w #1#2 26578
26579 { 26579
26580 \token_if_eq_meaning:NNTF #1 \__regex_compile_special:N 26580
26581 { 26581
26582 \cs_if_exist_use:cF { __regex_compile_quantifier_#2:w } 26582
26583 { \__regex_compile_quantifier_none: #1 #2 } 26583
26584 } 26584
26585 { \__regex_compile_quantifier_none: #1 #2 } 26585
26586 } 26586
26587 \cs_new_protected:Npn \__regex_compile_quantifier_none: 26587
26588 { 26588
26589 \tl_build_put_right:Nn \l__regex_build_tl 26589
26590 { \if_false: { \fi: } { 1 } { 0 } \c_false_bool } 26590
26591 } 26591
26592 \cs_new_protected:Npn \__regex_compile_quantifier_abort:eNN #1#2#3 26592
26593 { 26593
26594 \__regex_compile_quantifier_none: 26594
26595 \msg_warning:nnee { regex } { invalid-quantifier } {#1} {#3} 26595
26596 \__regex_compile_abort_tokens:e {#1} 26596
26597 #2 #3 26597
26598 } 26598
26599 \cs_new_protected:Npn \__regex_compile_quantifier_laziness:nnNN #1#2#3#4 26599
26600 { 26600
26601 \__regex_two_if_eq:NNNTF #3 #4 \__regex_compile_special:N ? 26601
26602 { 26602
26603 \tl_build_put_right:Nn \l__regex_build_tl 26603
26604 { \if_false: { \fi: } { #1 } { #2 } \c_true_bool } 26604
26605 } 26605
26606 { 26606
26607 \tl_build_put_right:Nn \l__regex_build_tl 26607
26608 { \if_false: { \fi: } { #1 } { #2 } \c_false_bool } 26608
26609 #3 #4 26609
26610 } 26610
26611 } 26611
26612 \cs_new_protected:cpn { __regex_compile_quantifier_?:w } 26612
26613 { \__regex_compile_quantifier_laziness:nnNN { 0 } { 1 } } 26613
26614 \cs_new_protected:cpn { __regex_compile_quantifier_*:w } 26614
26615 { \__regex_compile_quantifier_laziness:nnNN { 0 } { -1 } } 26615
26616 \cs_new_protected:cpn { __regex_compile_quantifier_+:w } 26616
26617 { \__regex_compile_quantifier_laziness:nnNN { 1 } { -1 } } 26617
```

26618	\cs_new_protected:cpn { __regex_compile_quantifier_ \c_left_brace_str :w }	26618
26619	{	26619
26620	__regex_get_digits:NTFw \l__regex_internal_a_int	26620
26621	{ __regex_compile_quantifier_braced_auxi:w }	26621
26622	{ __regex_compile_quantifier_abort:eNN { \c_left_brace_str } }	26622
26623	}	26623
26624	\cs_new_protected:Npn __regex_compile_quantifier_braced_auxi:w #1#2	26624
26625	{	26625
26626	\str_case:e:nnF { #1 #2 }	26626
26627	{	26627
26628	{ __regex_compile_special:N \c_right_brace_str }	26628
26629	{	26629
26630	\exp_args:No __regex_compile_quantifier_laziness:nnNN	26630
26631	{ \int_use:N \l__regex_internal_a_int } 0	26631
26632	}	26632
26633	{ __regex_compile_special:N , }	26633
26634	{	26634
26635	__regex_get_digits:NTFw \l__regex_internal_b_int	26635
26636	{ __regex_compile_quantifier_braced_auxiii:w }	26636
26637	{ __regex_compile_quantifier_braced_auxii:w }	26637
26638	}	26638
26639	}	26639
26640	{	26640
26641	__regex_compile_quantifier_abort:eNN	26641
26642	{ \c_left_brace_str \int_use:N \l__regex_internal_a_int }	26642
26643	#1 #2	26643
26644	}	26644
26645	}	26645
26646	\cs_new_protected:Npn __regex_compile_quantifier_braced_auxii:w #1#2	26646
26647	{	26647
26648	__regex_two_if_eq:NNNNTF #1 #2 __regex_compile_special:N \c_right_brace_str	26648
26649	{	26649
26650	\exp_args:No __regex_compile_quantifier_laziness:nnNN	26650
26651	{ \int_use:N \l__regex_internal_a_int } { -1 }	26651
26652	}	26652
26653	{	26653
26654	__regex_compile_quantifier_abort:eNN	26654
26655	{ \c_left_brace_str \int_use:N \l__regex_internal_a_int , }	26655
26656	#1 #2	26656
26657	}	26657
26658	}	26658
26659	\cs_new_protected:Npn __regex_compile_quantifier_braced_auxiii:w #1#2	26659
26660	{	26660
26661	__regex_two_if_eq:NNNNTF #1 #2 __regex_compile_special:N \c_right_brace_str	26661
26662	{	26662
26663	\if_int_compare:w \l__regex_internal_a_int >	26663


```
26664         \l__regex_internal_b_int                                26664
26665         \msg_error:nnee { regex } { backwards-quantifier }    26665
26666             { \int_use:N \l__regex_internal_a_int }           26666
26667             { \int_use:N \l__regex_internal_b_int }           26667
26668         \int_zero:N \l__regex_internal_b_int                   26668
26669     \else:                                                       26669
26670         \int_sub:Nn \l__regex_internal_b_int \l__regex_internal_a_int 26670
26671     \fi:                                                         26671
26672     \exp_args:Noo \__regex_compile_quantifier_laziness:nnNN    26672
26673         { \int_use:N \l__regex_internal_a_int }               26673
26674         { \int_use:N \l__regex_internal_b_int }               26674
26675     }                                                            26675
26676     {                                                            26676
26677         \__regex_compile_quantifier_abort:eNN                  26677
26678         {                                                       26678
26679             \c_left_brace_str                                  26679
26680             \int_use:N \l__regex_internal_a_int ,             26680
26681             \int_use:N \l__regex_internal_b_int               26681
26682         }                                                       26682
26683         #1 #2                                                  26683
26684     }                                                            26684
26685 }                                                                26685
26686 \cs_new_protected:Npn \__regex_compile_raw_error:N #1          26686
26687 {                                                                26687
26688     \msg_error:nne { regex } { bad-escape } {#1}             26688
26689     \__regex_compile_raw:N #1                                  26689
26690 }                                                                26690
26691 \cs_new_protected:Npn \__regex_compile_raw:N #1#2#3            26691
26692 {                                                                26692
26693     \__regex_if_in_class:TF                                    26693
26694     {                                                            26694
26695         \__regex_two_if_eq:NNNNTF #2 #3 \__regex_compile_special:N - 26695
26696         { \__regex_compile_range:Nw #1 }                       26696
26697         {                                                        26697
26698             \__regex_compile_one:n                             26698
26699             { \__regex_item_equal:n { \int_value:w `#1 } }    26699
26700             #2 #3                                              26700
26701         }                                                       26701
26702     }                                                            26702
26703     {                                                            26703
26704         \__regex_compile_one:n                                 26704
26705         { \__regex_item_equal:n { \int_value:w `#1 } }        26705
26706         #2 #3                                                  26706
26707     }                                                            26707
26708 }                                                                26708
26709 \cs_new_protected:Npn \__regex_if_end_range:NNTF #1#2          26709
```

```
26710 { 26710
26711 \if_meaning:w \__regex_compile_raw:N #1 26711
26712 \else: 26712
26713 \if_meaning:w \__regex_compile_special:N #1 26713
26714 \if_charcode:w ] #2 26714
26715 \use_i:nn 26715
26716 \fi: 26716
26717 \else: 26717
26718 \exp_after:wN \exp_after:wN \exp_after:wN \use_iii:nnn 26718
26719 \fi: 26719
26720 \fi: 26720
26721 \use_i:nn 26721
26722 } 26722
26723 \cs_new_protected:Npn \__regex_compile_range:Nw #1#2#3 26723
26724 { 26724
26725 \__regex_if_end_range:NNTF #2 #3 26725
26726 { 26726
26727 \if_int_compare:w `#1 > `#3 \exp_stop_f: 26727
26728 \msg_error:nnee { regex } { range-backwards } {#1} {#3} 26728
26729 \else: 26729
26730 \tl_build_put_right:Ne \l__regex_build_tl 26730
26731 { 26731
26732 \if_int_compare:w `#1 = `#3 \exp_stop_f: 26732
26733 \__regex_item_equal:n 26733
26734 \else: 26734
26735 \__regex_item_range:nn { \int_value:w `#1 } 26735
26736 \fi: 26736
26737 { \int_value:w `#3 } 26737
26738 } 26738
26739 \fi: 26739
26740 } 26740
26741 { 26741
26742 \msg_warning:nnee { regex } { range-missing-end } 26742
26743 {#1} { \c_backslash_str #3 } 26743
26744 \tl_build_put_right:Ne \l__regex_build_tl 26744
26745 { 26745
26746 \__regex_item_equal:n { \int_value:w `#1 \exp_stop_f: } 26746
26747 \__regex_item_equal:n { \int_value:w `~ \exp_stop_f: } 26747
26748 } 26748
26749 #2#3 26749
26750 } 26750
26751 } 26751
26752 \cs_new_protected:cpe { __regex_compile_.: } 26752
26753 { 26753
26754 \exp_not:N \__regex_if_in_class:TF 26754
26755 { \__regex_compile_raw:N . } 26755
```

```
26756 { \__regex_compile_one:n \exp_not:c { __regex_prop_.: } } 26756
26757 } 26757
26758 \cs_new_protected:cpn { __regex_prop_.: } 26758
26759 { 26759
26760 \if_int_compare:w \l__regex_curr_char_int > - 2 \exp_stop_f: 26760
26761 \exp_after:wN \__regex_break_true:w 26761
26762 \fi: 26762
26763 } 26763
26764 \cs_set_protected:Npn \__regex_tmp:w #1#2 26764
26765 { 26765
26766 \cs_new_protected:cpe { __regex_compile_/#1: } 26766
26767 { \__regex_compile_one:n \exp_not:c { __regex_prop_#1: } } 26767
26768 \cs_new_protected:cpe { __regex_compile_/#2: } 26768
26769 { 26769
26770 \__regex_compile_one:n 26770
26771 { \__regex_item_reverse:n { \exp_not:c { __regex_prop_#1: } } } 26771
26772 } 26772
26773 } 26773
26774 \__regex_tmp:w d D 26774
26775 \__regex_tmp:w h H 26775
26776 \__regex_tmp:w s S 26776
26777 \__regex_tmp:w v V 26777
26778 \__regex_tmp:w w W 26778
26779 \cs_new_protected:cpn { __regex_compile_/N: } 26779
26780 { \__regex_compile_one:n \__regex_prop_N: } 26780
26781 \cs_new_protected:Npn \__regex_compile_anchor_letter:NNN #1#2#3 26781
26782 { 26782
26783 \__regex_if_in_class_or_catcode:TF { \__regex_compile_raw_error:N #1 } 26783
26784 { 26784
26785 \tl_build_put_right:Nn \l__regex_build_tl 26785
26786 { \__regex_assertion:Nn #2 {#3} } 26786
26787 } 26787
26788 } 26788
26789 \cs_new_protected:cpn { __regex_compile_/A: } 26789
26790 { \__regex_compile_anchor_letter:NNN A \c_true_bool \__regex_A_test: } 26790
26791 \cs_new_protected:cpn { __regex_compile_/G: } 26791
26792 { \__regex_compile_anchor_letter:NNN G \c_true_bool \__regex_G_test: } 26792
26793 \cs_new_protected:cpn { __regex_compile_/Z: } 26793
26794 { \__regex_compile_anchor_letter:NNN Z \c_true_bool \__regex_Z_test: } 26794
26795 \cs_new_protected:cpn { __regex_compile_/z: } 26795
26796 { \__regex_compile_anchor_letter:NNN z \c_true_bool \__regex_Z_test: } 26796
26797 \cs_new_protected:cpn { __regex_compile_/b: } 26797
26798 { \__regex_compile_anchor_letter:NNN b \c_true_bool \__regex_b_test: } 26798
26799 \cs_new_protected:cpn { __regex_compile_/B: } 26799
26800 { \__regex_compile_anchor_letter:NNN B \c_false_bool \__regex_b_test: } 26800
26801 \cs_set_protected:Npn \__regex_tmp:w #1#2 26801
```

```

26802 {
26803   \cs_new_protected:cpn { __regex_compile_#1: }
26804   {
26805     \__regex_if_in_class_or_catcode:TF { \__regex_compile_raw:N #1 }
26806     {
26807       \tl_build_put_right:Nn \l__regex_build_tl
26808       { \__regex_assertion:Nn \c_true_bool {#2} }
26809     }
26810   }
26811 }
26812 \exp_args:Ne \__regex_tmp:w { \iow_char:N \^ } { \__regex_A_test: }
26813 \exp_args:Ne \__regex_tmp:w { \iow_char:N \$ } { \__regex_Z_test: }
26814 \cs_new_protected:cpn { __regex_compile_]: }
26815 {
26816   \__regex_if_in_class:TF
26817   {
26818     \if_int_compare:w \l__regex_mode_int >
26819     \c__regex_catcode_in_class_mode_int
26820     \tl_build_put_right:Nn \l__regex_build_tl { \if_false: { \fi: } }
26821     \fi:
26822     \tex_advance:D \l__regex_mode_int - 15 \exp_stop_f:
26823     \tex_divide:D \l__regex_mode_int 13 \exp_stop_f:
26824     \if_int_odd:w \l__regex_mode_int \else:
26825       \exp_after:wN \__regex_compile_quantifier:w
26826     \fi:
26827   }
26828   { \__regex_compile_raw:N ] }
26829 }
26830 \cs_new_protected:cpn { __regex_compile[: }
26831 {
26832   \__regex_if_in_class:TF
26833   { \__regex_compile_class_posix_test:w }
26834   {
26835     \__regex_if_within_catcode:TF
26836     {
26837       \exp_after:wN \__regex_compile_class_catcode:w
26838       \int_use:N \l__regex_catcodes_int \__regex_sep:
26839     }
26840     { \__regex_compile_class_normal:w }
26841   }
26842 }
26843 \cs_new_protected:Npn \__regex_compile_class_normal:w
26844 {
26845   \__regex_compile_class:TFNN
26846   { \__regex_class:NnnnN \c_true_bool }
26847   { \__regex_class:NnnnN \c_false_bool }

```

```
26848 } 26848
26849 \cs_new_protected:Npn \__regex_compile_class_catcode:w #1 \__regex_sep: 26849
26850 { 26850
26851 \if_int_compare:w \l__regex_mode_int = \c__regex_catcode_mode_int 26851
26852 \tl_build_put_right:Nn \l__regex_build_tl 26852
26853 { \__regex_class:NnnnN \c_true_bool { \if_false: } \fi: } 26853
26854 \fi: 26854
26855 \int_set_eq:NN \l__regex_catcodes_int \l__regex_default_catcodes_int 26855
26856 \__regex_compile_class:TFNN 26856
26857 { \__regex_item_catcode:nT {#1} } 26857
26858 { \__regex_item_catcode_reverse:nT {#1} } 26858
26859 } 26859
26860 \cs_new_protected:Npn \__regex_compile_class:TFNN #1#2#3#4 26860
26861 { 26861
26862 \l__regex_mode_int = \int_value:w \l__regex_mode_int 3 \exp_stop_f: 26862
26863 \__regex_two_if_eq:NNNNTF #3 #4 \__regex_compile_special:N ^ 26863
26864 { 26864
26865 \tl_build_put_right:Nn \l__regex_build_tl { #2 { \if_false: } \fi: } 26865
26866 \__regex_compile_class:NN 26866
26867 } 26867
26868 { 26868
26869 \tl_build_put_right:Nn \l__regex_build_tl { #1 { \if_false: } \fi: } 26869
26870 \__regex_compile_class:NN #3 #4 26870
26871 } 26871
26872 } 26872
26873 \cs_new_protected:Npn \__regex_compile_class:NN #1#2 26873
26874 { 26874
26875 \token_if_eq_charcode:NNTF #2 ] 26875
26876 { \__regex_compile_raw:N #2 } 26876
26877 { #1 #2 } 26877
26878 } 26878
26879 \cs_new_protected:Npn \__regex_compile_class_posix_test:w #1#2 26879
26880 { 26880
26881 \token_if_eq_meaning:NNT \__regex_compile_special:N #1 26881
26882 { 26882
26883 \str_case:nn { #2 } 26883
26884 { 26884
26885 : { \__regex_compile_class_posix:NNNNw } 26885
26886 = { 26886
26887 \msg_warning:nne { regex } 26887
26888 { posix-unsupported } { = } 26888
26889 } 26889
26890 . { 26890
26891 \msg_warning:nne { regex } 26891
26892 { posix-unsupported } { . } 26892
26893 } 26893
```

```
26894     }
26895 }
26896 \__regex_compile_raw:N [ #1 #2
26897 }
26898 \cs_new_protected:Npn \__regex_compile_class_posix:NNNNw #1#2#3#4#5#6
26899 {
26900 \__regex_two_if_eq:NNNNTF #5 #6 \__regex_compile_special:N ^
26901 {
26902 \bool_set_false:N \l__regex_internal_bool
26903 \__kernel_tl_set:Nx \l__regex_internal_a_tl { \if_false: } \fi:
26904 \__regex_compile_class_posix_loop:w
26905 }
26906 {
26907 \bool_set_true:N \l__regex_internal_bool
26908 \__kernel_tl_set:Nx \l__regex_internal_a_tl { \if_false: } \fi:
26909 \__regex_compile_class_posix_loop:w #5 #6
26910 }
26911 }
26912 \cs_new:Npn \__regex_compile_class_posix_loop:w #1#2
26913 {
26914 \token_if_eq_meaning:NNTF \__regex_compile_raw:N #1
26915 { #2 \__regex_compile_class_posix_loop:w }
26916 { \if_false: { \fi: } \__regex_compile_class_posix_end:w #1 #2 }
26917 }
26918 \cs_new_protected:Npn \__regex_compile_class_posix_end:w #1#2#3#4
26919 {
26920 \__regex_two_if_eq:NNNNTF #1 #2 \__regex_compile_special:N :
26921 { \__regex_two_if_eq:NNNNTF #3 #4 \__regex_compile_special:N ] }
26922 { \use_i:nn }
26923 {
26924 \cs_if_exist:cTF { __regex_posix_ \l__regex_internal_a_tl : }
26925 {
26926 \__regex_compile_one:n
26927 {
26928 \bool_if:NTF \l__regex_internal_bool \use:n \__regex_item_reverse:n
26929 { \exp_not:c { __regex_posix_ \l__regex_internal_a_tl : } }
26930 }
26931 }
26932 {
26933 \msg_warning:nne { regex } { posix-unknown }
26934 { \l__regex_internal_a_tl }
26935 \__regex_compile_abort_tokens:e
26936 {
26937 [: \bool_if:NF \l__regex_internal_bool { ^ }
26938 \l__regex_internal_a_tl :]
26939 }
```

```

26940     }
26941 }
26942 {
26943     \msg_error:nnee { regex } { posix-missing-close }
26944     { [: \l__regex_internal_a_tl ] { #2 #4 }
26945     \__regex_compile_abort_tokens:e { [: \l__regex_internal_a_tl ]
26946     #1 #2 #3 #4
26947 }
26948 }
26949 \cs_new_protected:Npn \__regex_compile_group_begin:N #1
26950 {
26951     \tl_build_put_right:Nn \l__regex_build_tl { #1 { \if_false: } \fi: }
26952     \__regex_mode_quit_c:
26953     \group_begin:
26954         \tl_build_begin:N \l__regex_build_tl
26955         \int_set_eq:NN \l__regex_default_catcodes_int \l__regex_catcodes_int
26956         \int_incr:N \l__regex_group_level_int
26957         \tl_build_put_right:Nn \l__regex_build_tl
26958         { \__regex_branch:n { \if_false: } \fi: }
26959     }
26960 \cs_new_protected:Npn \__regex_compile_group_end:
26961 {
26962     \if_int_compare:w \l__regex_group_level_int > \c_zero_int
26963         \tl_build_put_right:Nn \l__regex_build_tl { \if_false: { \fi: } }
26964         \tl_build_end:N \l__regex_build_tl
26965         \exp_args:NNNe
26966         \group_end:
26967         \tl_build_put_right:Nn \l__regex_build_tl { \l__regex_build_tl }
26968         \int_set_eq:NN \l__regex_catcodes_int \l__regex_default_catcodes_int
26969         \exp_after:wN \__regex_compile_quantifier:w
26970     \else:
26971         \msg_warning:nn { regex } { extra-rparen }
26972         \exp_after:wN \__regex_compile_raw:N \exp_after:wN )
26973     \fi:
26974 }
26975 \cs_new_protected:cpn { __regex_compile(: }
26976 {
26977     \__regex_if_in_class:TF { \__regex_compile_raw:N ( }
26978     {
26979         \if_int_compare:w \l__regex_mode_int =
26980         \c__regex_catcode_in_class_mode_int
26981         \msg_error:nn { regex } { c-lparen-in-class }
26982         \exp_after:wN \__regex_compile_raw:N \exp_after:wN (
26983     \else:
26984         \exp_after:wN \__regex_compile_lparen:w
26985     \fi:

```



```
26986     }
26987 }
26988 \cs_new_protected:Npn \__regex_compile_lparen:w #1#2#3#4
26989 {
26990     \__regex_two_if_eq:NNNNTF #1 #2 \__regex_compile_special:N ?
26991     {
26992         \cs_if_exist_use:cF
26993         { \__regex_compile_special_group\_token_to_str:N #4 :w }
26994         {
26995             \msg_warning:nne { regex } { special-group-unknown }
26996             { (? #4 )
26997             \__regex_compile_group_begin:N \__regex_group:nnnN
26998             \__regex_compile_raw:N ? #3 #4
26999             }
27000     }
27001     {
27002         \__regex_compile_group_begin:N \__regex_group:nnnN
27003         #1 #2 #3 #4
27004     }
27005 }
27006 \cs_new_protected:cpn { \__regex_compile_|: }
27007 {
27008     \__regex_if_in_class:TF { \__regex_compile_raw:N | }
27009     {
27010         \tl_build_put_right:Nn \l__regex_build_tl
27011         { \if_false: { \fi: } \__regex_branch:n { \if_false: } \fi: }
27012     }
27013 }
27014 \cs_new_protected:cpn { \__regex_compile_): }
27015 {
27016     \__regex_if_in_class:TF { \__regex_compile_raw:N ) }
27017     { \__regex_compile_group_end: }
27018 }
27019 \cs_new_protected:cpn { \__regex_compile_special_group_:w }
27020 { \__regex_compile_group_begin:N \__regex_group_no_capture:nnnN }
27021 \cs_new_protected:cpn { \__regex_compile_special_group_|:w }
27022 { \__regex_compile_group_begin:N \__regex_group_resetting:nnnN }
27023 \cs_new_protected:Npn \__regex_compile_special_group_i:w #1#2
27024 {
27025     \__regex_two_if_eq:NNNNTF #1 #2 \__regex_compile_special:N )
27026     {
27027         \cs_set:Npn \__regex_item_equal:n
27028         { \__regex_item_caseless_equal:n }
27029         \cs_set:Npn \__regex_item_range:nn
27030         { \__regex_item_caseless_range:nn }
27031     }
```

```
27032 { 27032
27033 \msg_warning:nne { regex } { unknown-option } { (?i #2 } 27033
27034 \__regex_compile_raw:N ( 27034
27035 \__regex_compile_raw:N ? 27035
27036 \__regex_compile_raw:N i 27036
27037 #1 #2 27037
27038 } 27038
27039 } 27039
27040 \cs_new_protected:cpn { __regex_compile_special_group_:-:w } #1#2#3#4 27040
27041 { 27041
27042 \__regex_two_if_eq:NNNTF #1 #2 \__regex_compile_raw:N i 27042
27043 { \__regex_two_if_eq:NNNTF #3 #4 \__regex_compile_special:N ) } 27043
27044 { \use_ii:nn } 27044
27045 { 27045
27046 \cs_set:Npn \__regex_item_equal:n 27046
27047 { \__regex_item_caseful_equal:n } 27047
27048 \cs_set:Npn \__regex_item_range:nn 27048
27049 { \__regex_item_caseful_range:nn } 27049
27050 } 27050
27051 { 27051
27052 \msg_warning:nne { regex } { unknown-option } { (?-#2#4 } 27052
27053 \__regex_compile_raw:N ( 27053
27054 \__regex_compile_raw:N ? 27054
27055 \__regex_compile_raw:N - 27055
27056 #1 #2 #3 #4 27056
27057 } 27057
27058 } 27058
27059 \cs_new_protected:cpn { __regex_compile_/c: } 27059
27060 { \__regex_chk_c_allowed:T { \__regex_compile_c_test:NN } } 27060
27061 \cs_new_protected:Npn \__regex_compile_c_test:NN #1#2 27061
27062 { 27062
27063 \token_if_eq_meaning:NNTF #1 \__regex_compile_raw:N 27063
27064 { 27064
27065 \int_if_exist:cTF { c__regex_catcode_#2_int } 27065
27066 { 27066
27067 \int_set_eq:Nc \l__regex_catcodes_int 27067
27068 { c__regex_catcode_#2_int } 27068
27069 \l__regex_mode_int 27069
27070 = \if_case:w \l__regex_mode_int 27070
27071 \c__regex_catcode_mode_int 27071
27072 \else: 27072
27073 \c__regex_catcode_in_class_mode_int 27073
27074 \fi: 27074
27075 \token_if_eq_charcode:NNT C #2 { \__regex_compile_c_C:NN } 27075
27076 } 27076
27077 } 27077
```

```
27078 { \cs_if_exist_use:cF { __regex_compile_c_#2:w } } 27078
27079 { 27079
27080 \msg_error:nne { regex } { c-missing-category } {#2} 27080
27081 #1 #2 27081
27082 } 27082
27083 } 27083
27084 \cs_new_protected:Npn \__regex_compile_c_C:NN #1#2 27084
27085 { 27085
27086 \token_if_eq_meaning:NNTF #1 \__regex_compile_special:N 27086
27087 { 27087
27088 \token_if_eq_charcode:NNTF #2 . 27088
27089 { \use_none:n } 27089
27090 { \token_if_eq_charcode:NNF #2 ( } % ) 27090
27091 } 27091
27092 { \use:n } 27092
27093 { \msg_error:nnn { regex } { c-C-invalid } {#2} } 27093
27094 #1 #2 27094
27095 } 27095
27096 \cs_new_protected:cpn { __regex_compile_c_[:w } #1#2 27096
27097 { 27097
27098 \l__regex_mode_int 27098
27099 = \if_case:w \l__regex_mode_int 27099
27100 \c__regex_catcode_mode_int 27100
27101 \else: 27101
27102 \c__regex_catcode_in_class_mode_int 27102
27103 \fi: 27103
27104 \int_zero:N \l__regex_catcodes_int 27104
27105 \__regex_two_if_eq:NNNNTF #1 #2 \__regex_compile_special:N ^ 27105
27106 { 27106
27107 \bool_set_false:N \l__regex_catcodes_bool 27107
27108 \__regex_compile_c_lbrack_loop:NN 27108
27109 } 27109
27110 { 27110
27111 \bool_set_true:N \l__regex_catcodes_bool 27111
27112 \__regex_compile_c_lbrack_loop:NN 27112
27113 #1 #2 27113
27114 } 27114
27115 } 27115
27116 \cs_new_protected:Npn \__regex_compile_c_lbrack_loop:NN #1#2 27116
27117 { 27117
27118 \token_if_eq_meaning:NNTF #1 \__regex_compile_raw:N 27118
27119 { 27119
27120 \int_if_exist:cTF { c__regex_catcode_#2_int } 27120
27121 { 27121
27122 \exp_args:Nc \__regex_compile_c_lbrack_add:N 27122
27123 { c__regex_catcode_#2_int } 27123
```

```

27124 \__regex_compile_c_lbrack_loop:NN
27125 }
27126 }
27127 {
27128 \token_if_eq_charcode:NNTF #2 ]
27129 { \__regex_compile_c_lbrack_end: }
27130 }
27131 {
27132 \msg_error:nne { regex } { c-missing-rbrack } {#2}
27133 \__regex_compile_c_lbrack_end:
27134 #1 #2
27135 }
27136 }
27137 \cs_new_protected:Npn \__regex_compile_c_lbrack_add:N #1
27138 {
27139 \if_int_odd:w \__regex_int_eval:w \l__regex_catcodes_int / #1 \scan_stop:
27140 \else:
27141 \int_add:Nn \l__regex_catcodes_int {#1}
27142 \fi:
27143 }
27144 \cs_new_protected:Npn \__regex_compile_c_lbrack_end:
27145 {
27146 \if_meaning:w \c_false_bool \l__regex_catcodes_bool
27147 \int_set:Nn \l__regex_catcodes_int
27148 { \c__regex_all_catcodes_int - \l__regex_catcodes_int }
27149 \fi:
27150 }
27151 \cs_new_protected:cpn { __regex_compile_c_ \c_left_brace_str :w }
27152 {
27153 \__regex_compile:w
27154 \__regex_disable_submatches:
27155 \l__regex_mode_int
27156 = \if_case:w \l__regex_mode_int
27157 \c__regex_cs_mode_int
27158 \else:
27159 \c__regex_cs_in_class_mode_int
27160 \fi:
27161 }
27162 \cs_new_protected:cpn { __regex_compile_ \c_left_brace_str : }
27163 {
27164 \__regex_if_in_cs:TF
27165 { \msg_error:nnn { regex } { cu-lbrace } { c } }
27166 { \exp_after:wN \__regex_compile_raw:N \c_left_brace_str }
27167 }
27168 \flag_new:N \l__regex_cs_flag
27169 \cs_new_protected:cpn { __regex_compile_ \c_right_brace_str : }

```

```
27170 {
27171     \__regex_if_in_cs:TF
27172     { \__regex_compile_end_cs: }
27173     { \exp_after:wN \__regex_compile_raw:N \c_right_brace_str }
27174 }
27175 \cs_new_protected:Npn \__regex_compile_end_cs:
27176 {
27177     \__regex_compile_end:
27178     \flag_clear:N \l__regex_cs_flag
27179     \__kernel_tl_set:Nx \l__regex_internal_a_tl
27180     {
27181         \exp_after:wN \__regex_compile_cs_aux:Nn \l__regex_internal_regex
27182         \q__regex_nil \q__regex_nil \q__regex_recursion_stop
27183     }
27184     \exp_args:Ne \__regex_compile_one:n
27185     {
27186         \flag_if_raised:NTF \l__regex_cs_flag
27187         { \__regex_item_cs:n { \exp_not:o \l__regex_internal_regex } }
27188         {
27189             \__regex_item_exact_cs:n
27190             { \tl_tail:N \l__regex_internal_a_tl }
27191         }
27192     }
27193 }
27194 \cs_new:Npn \__regex_compile_cs_aux:Nn #1#2
27195 {
27196     \cs_if_eq:NNTF #1 \__regex_branch:n
27197     {
27198         \scan_stop:
27199         \__regex_compile_cs_aux:NNnnnN #2
27200         \q__regex_nil \q__regex_nil \q__regex_nil
27201         \q__regex_nil \q__regex_nil \q__regex_nil \q__regex_recursion_stop
27202         \__regex_compile_cs_aux:Nn
27203     }
27204     {
27205         \__regex_quark_if_nil:NF #1 { \flag_ensure_raised:N \l__regex_cs_flag }
27206         \__regex_use_none_delimit_by_q_recursion_stop:w
27207     }
27208 }
27209 \cs_new:Npn \__regex_compile_cs_aux:NNnnnN #1#2#3#4#5#6
27210 {
27211     \bool_lazy_all:nTF
27212     {
27213         { \cs_if_eq_p:NN #1 \__regex_class:NnnnN }
27214         {#2}
27215         { \tl_if_head_eq_meaning_p:nN {#3} \__regex_item_caseful_equal:n }
```

```
27216 { \int_compare_p:nNn { \tl_count:n {#3} } = { 2 } } 27216
27217 { \int_compare_p:nNn {#5} = \c_zero_int } 27217
27218 } 27218
27219 { 27219
27220 \prg_replicate:nn {#4} 27220
27221 { \char_generate:nn { \use_ii:nn #3 } {12} } 27221
27222 \__regex_compile_cs_aux:NNnnnN 27222
27223 } 27223
27224 { 27224
27225 \__regex_quark_if_nil:NF #1 27225
27226 { 27226
27227 \flag_ensure_raised:N \l__regex_cs_flag 27227
27228 \__regex_use_i_delimit_by_q_recursion_stop:nw 27228
27229 } 27229
27230 \__regex_use_none_delimit_by_q_recursion_stop:w 27230
27231 } 27231
27232 } 27232
27233 \cs_new_protected:cpn { __regex_compile_/u: } #1#2 27233
27234 { 27234
27235 \__regex_if_in_class_or_catcode:TF 27235
27236 { \__regex_compile_raw_error:N u #1 #2 } 27236
27237 { 27237
27238 \__regex_two_if_eq:NNNNTF #1 #2 \__regex_compile_raw:N r 27238
27239 { \__regex_compile_u_brace:NNN \__regex_compile_ur_end: } 27239
27240 { \__regex_compile_u_brace:NNN \__regex_compile_u_end: #1 #2 } 27240
27241 } 27241
27242 } 27242
27243 \cs_new:Npn \__regex_compile_u_brace:NNN #1#2#3 27243
27244 { 27244
27245 \__regex_two_if_eq:NNNNTF #2 #3 \__regex_compile_special:N \c_left_brace_str 27245
27246 { 27246
27247 \tl_set:Nn \l__regex_internal_b_tl {#1} 27247
27248 \__kernel_tl_set:Nx \l__regex_internal_a_tl { \if_false: } \fi: 27248
27249 \__regex_compile_u_loop:NN 27249
27250 } 27250
27251 { 27251
27252 \msg_error:nn { regex } { u-missing-lbrace } 27252
27253 \token_if_eq_meaning:NNTF #1 \__regex_compile_ur_end: 27253
27254 { \__regex_compile_raw:N u \__regex_compile_raw:N r } 27254
27255 { \__regex_compile_raw:N u } 27255
27256 #2 #3 27256
27257 } 27257
27258 } 27258
27259 \cs_new:Npn \__regex_compile_u_loop:NN #1#2 27259
27260 { 27260
27261 \token_if_eq_meaning:NNTF #1 \__regex_compile_raw:N 27261
```

```

27262 { #2 \__regex_compile_u_loop:NN } 27262
27263 { 27263
27264 \token_if_eq_meaning:NNTF #1 \__regex_compile_special:N 27264
27265 { 27265
27266 \exp_after:wN \token_if_eq_charcode:NNTF \c_right_brace_str #2 27266
27267 { \if_false: { \fi: } \l__regex_internal_b_tl } 27267
27268 { 27268
27269 \if_charcode:w \c_left_brace_str #2 27269
27270 \msg_expandable_error:nnn { regex } { cu-lbrace } { u } 27270
27271 \else: 27271
27272 #2 27272
27273 \fi: 27273
27274 \__regex_compile_u_loop:NN 27274
27275 } 27275
27276 } 27276
27277 { 27277
27278 \if_false: { \fi: } 27278
27279 \msg_error:nne { regex } { u-missing-rbrace } {#2} 27279
27280 \l__regex_internal_b_tl 27280
27281 #1 #2 27281
27282 } 27282
27283 } 27283
27284 } 27284
27285 \cs_new_protected:Npn \__regex_compile_ur_end: 27285
27286 { 27286
27287 \group_begin: 27287
27288 \cs_set:Npn \__regex_group:nnnN { \__regex_group_no_capture:nnnN } 27288
27289 \cs_set:Npn \__regex_group_resetting:nnnN { \__regex_group_no_capture:nnnN } 27289
27290 \exp_args:NNe 27290
27291 \group_end: 27291
27292 \__regex_compile_ur:n { \use:c { \l__regex_internal_a_tl } } 27292
27293 } 27293
27294 \cs_new_protected:Npn \__regex_compile_ur:n #1 27294
27295 { 27295
27296 \tl_if_empty:oTF { \__regex_compile_ur_aux:w #1 {} ? ? \q__regex_nil } 27296
27297 { \__regex_compile_if_quantifier:TFw } 27297
27298 { \use_i:nn } 27298
27299 { 27299
27300 \tl_build_put_right:Nn \l__regex_build_tl 27300
27301 { \__regex_group_no_capture:nnnN { \if_false: } \fi: #1 } 27301
27302 \__regex_compile_quantifier:w 27302
27303 } 27303
27304 { \tl_build_put_right:Nn \l__regex_build_tl { \use_ii:nn #1 } } 27304
27305 } 27305
27306 \cs_new:Npn \__regex_compile_ur_aux:w \__regex_branch:n #1#2#3 \q__regex_nil {#2} 27306
27307 \cs_new_protected:Npn \__regex_compile_u_end: 27307

```



```
27308 {
27309     \__regex_compile_if_quantifier:TFw
27310     {
27311         \tl_build_put_right:Nn \l__regex_build_tl
27312         {
27313             \__regex_group_no_capture:nnnN { \if_false: } \fi:
27314             \__regex_branch:n { \if_false: } \fi:
27315         }
27316         \__regex_compile_u_payload:
27317         \tl_build_put_right:Nn \l__regex_build_tl { \if_false: { \fi: } }
27318         \__regex_compile_quantifier:w
27319     }
27320     { \__regex_compile_u_payload: }
27321 }
27322 \cs_new_protected:Npn \__regex_compile_u_payload:
27323 {
27324     \tl_set:Nv \l__regex_internal_a_tl { \l__regex_internal_a_tl }
27325     \if_int_compare:w \l__regex_mode_int = \c__regex_outer_mode_int
27326         \__regex_compile_u_not_cs:
27327     \else:
27328         \__regex_compile_u_in_cs:
27329     \fi:
27330 }
27331 \cs_new_protected:Npn \__regex_compile_u_in_cs:
27332 {
27333     \__kernel_tl_gset:Nx \g__regex_internal_tl
27334     {
27335         \exp_args:No \__kernel_str_to_other_fast:n
27336         { \l__regex_internal_a_tl }
27337     }
27338     \tl_build_put_right:Ne \l__regex_build_tl
27339     {
27340         \tl_map_function:NN \g__regex_internal_tl
27341         \__regex_compile_u_in_cs_aux:n
27342     }
27343 }
27344 \cs_new:Npn \__regex_compile_u_in_cs_aux:n #1
27345 {
27346     \__regex_class:NnnnN \c_true_bool
27347     { \__regex_item_caseful_equal:n { \int_value:w `#1 } }
27348     { 1 } { 0 } \c_false_bool
27349 }
27350 \cs_new_protected:Npn \__regex_compile_u_not_cs:
27351 {
27352     \tl_analysis_map_inline:Nn \l__regex_internal_a_tl
27353     {
```

```
27354 \tl_build_put_right:Ne \l__regex_build_tl 27354
27355 { 27355
27356 \__regex_class:NnnnN \c_true_bool 27356
27357 { 27357
27358 \if_int_compare:w "##3 = \c_zero_int 27358
27359 \__regex_item_exact_cs:n 27359
27360 { \exp_after:wN \cs_to_str:N ##1 } 27360
27361 \else: 27361
27362 \__regex_item_exact:nn { \int_value:w "##3 } { ##2 } 27362
27363 \fi: 27363
27364 } 27364
27365 { 1 } { 0 } \c_false_bool 27365
27366 } 27366
27367 } 27367
27368 } 27368
27369 \cs_new_protected:cpn { __regex_compile_/K: } 27369
27370 { 27370
27371 \int_compare:nNnTF \l__regex_mode_int = \c__regex_outer_mode_int 27371
27372 { \tl_build_put_right:Nn \l__regex_build_tl { \__regex_command_K: } } 27372
27373 { \__regex_compile_raw_error:N K } 27373
27374 } 27374
27375 \cs_new:Npn \__regex_clean_bool:n #1 27375
27376 { 27376
27377 \tl_if_single:nTF {#1} 27377
27378 { \bool_if:NTF #1 \c_true_bool \c_false_bool } 27378
27379 { \c_true_bool } 27379
27380 } 27380
27381 \cs_new:Npn \__regex_clean_int:n #1 27381
27382 { 27382
27383 \tl_if_head_eq_meaning:nNTF {#1} - 27383
27384 { - \exp_args:No \__regex_clean_int:n { \use_none:n #1 } } 27384
27385 { \int_eval:n { 0 \str_map_function:nN {#1} \__regex_clean_int_aux:N } } 27385
27386 } 27386
27387 \cs_new:Npn \__regex_clean_int_aux:N #1 27387
27388 { 27388
27389 \if_int_compare:w \c_one_int < 1 #1 ~ 27389
27390 #1 27390
27391 \else: 27391
27392 \str_map_break:n 27392
27393 \fi: 27393
27394 } 27394
27395 \cs_new:Npn \__regex_clean_regex:n #1 27395
27396 { 27396
27397 \__regex_clean_regex_loop:w #1 27397
27398 \__regex_branch:n { \q_recursion_tail } \q_recursion_stop 27398
27399 } 27399
```

```
27400 \cs_new:Npn \__regex_clean_regex_loop:w #1 \__regex_branch:n #2
27401 {
27402     \quark_if_recursion_tail_stop:n {#2}
27403     \__regex_branch:n { \__regex_clean_branch:n {#2} }
27404     \__regex_clean_regex_loop:w
27405 }
27406 \cs_new:Npn \__regex_clean_branch:n #1
27407 {
27408     \__regex_clean_branch_loop:n #1
27409     ? ? ? ? ? ? \prg_break_point:
27410 }
27411 \cs_new:Npn \__regex_clean_branch_loop:n #1
27412 {
27413     \tl_if_single:nF {#1} \prg_break:
27414     \token_case_meaning:NnF #1
27415     {
27416         \__regex_command_K: { #1 \__regex_clean_branch_loop:n }
27417         \__regex_assertion:Nn { #1 \__regex_clean_assertion:Nn }
27418         \__regex_class:NnnnN { #1 \__regex_clean_class:NnnnN }
27419         \__regex_group:nnnN { #1 \__regex_clean_group:nnnN }
27420         \__regex_group_no_capture:nnnN { #1 \__regex_clean_group:nnnN }
27421         \__regex_group_resetting:nnnN { #1 \__regex_clean_group:nnnN }
27422     }
27423     \prg_break:
27424 }
27425 \cs_new:Npn \__regex_clean_assertion:Nn #1#2
27426 {
27427     \__regex_clean_bool:n {#1}
27428     \tl_if_single:nF {#2} { { \__regex_A_test: } \prg_break: }
27429     \token_case_meaning:NnTF #2
27430     {
27431         \__regex_A_test: { }
27432         \__regex_G_test: { }
27433         \__regex_Z_test: { }
27434         \__regex_b_test: { }
27435     }
27436     { {#2} }
27437     { { \__regex_A_test: } \prg_break: }
27438     \__regex_clean_branch_loop:n
27439 }
27440 \cs_new:Npn \__regex_clean_class:NnnnN #1#2#3#4#5
27441 {
27442     \__regex_clean_bool:n {#1}
27443     { \__regex_clean_class:n {#2} }
27444     { \int_max:nn \c_zero_int { \__regex_clean_int:n {#3} } }
27445     { \int_max:nn { -\c_one_int } { \__regex_clean_int:n {#4} } }
```

```
27446 \__regex_clean_bool:n {#5}
27447 \__regex_clean_branch_loop:n
27448 }
27449 \cs_new:Npn \__regex_clean_group:nnnN #1#2#3#4
27450 {
27451   { \__regex_clean_regex:n {#1} }
27452   { \int_max:nn \c_zero_int { \__regex_clean_int:n {#2} } }
27453   { \int_max:nn { -\c_one_int } { \__regex_clean_int:n {#3} } }
27454   \__regex_clean_bool:n {#4}
27455   \__regex_clean_branch_loop:n
27456 }
27457 \cs_new:Npn \__regex_clean_class:n #1
27458 { \__regex_clean_class_loop:nnn #1 ????? \prg_break_point: }
27459 \cs_new:Npn \__regex_clean_class_loop:nnn #1#2#3
27460 {
27461   \tl_if_single:nF {#1} \prg_break:
27462   \token_case_meaning:NnTF #1
27463   {
27464     \__regex_item_cs:n { #1 { \__regex_clean_regex:n {#2} } }
27465     \__regex_item_exact_cs:n { #1 { \__regex_clean_exact_cs:n {#2} } }
27466     \__regex_item_caseful_equal:n { #1 { \__regex_clean_int:n {#2} } }
27467     \__regex_item_caseless_equal:n { #1 { \__regex_clean_int:n {#2} } }
27468     \__regex_item_reverse:n { #1 { \__regex_clean_class:n {#2} } }
27469   }
27470   { \__regex_clean_class_loop:nnn {#3} }
27471   {
27472     \token_case_meaning:NnTF #1
27473     {
27474       \__regex_item_caseful_range:nn { }
27475       \__regex_item_caseless_range:nn { }
27476       \__regex_item_exact:nn { }
27477     }
27478     {
27479       #1 { \__regex_clean_int:n {#2} } { \__regex_clean_int:n {#3} }
27480       \__regex_clean_class_loop:nnn
27481     }
27482     {
27483       \token_case_meaning:NnTF #1
27484       {
27485         \__regex_item_catcode:nT { }
27486         \__regex_item_catcode_reverse:nT { }
27487       }
27488       {
27489         #1 { \__regex_clean_int:n {#2} } { \__regex_clean_class:n {#3} }
27490         \__regex_clean_class_loop:nnn
27491       }
27492     }
27493   }
27494 }
```

```

27492         {
27493             \exp_args:Ne \str_case:nnTF
27494             {
27495                 \exp_args:Ne \str_range:nnn
27496                 { \cs_to_str:N #1 } \c_one_int { 13 }
27497             }
27498             {
27499                 { __regex_prop_ } { }
27500                 { __regex_posix } { }
27501             }
27502             {
27503                 #1
27504                 \__regex_clean_class_loop:nnn {#2} {#3}
27505             }
27506             \prg_break:
27507         }
27508     }
27509 }
27510 }
27511 \cs_new:Npn \__regex_clean_exact_cs:n #1
27512 {
27513     \exp_last_unbraced:Nf \use_none:n
27514     {
27515         \__regex_clean_exact_cs:w #1
27516         \scan_stop: \q_recursion_tail \scan_stop:
27517         \q_recursion_stop
27518     }
27519 }
27520 \cs_new:Npn \__regex_clean_exact_cs:w #1 \scan_stop:
27521 {
27522     \quark_if_recursion_tail_stop:n {#1}
27523     \scan_stop: \tl_to_str:n {#1}
27524     \__regex_clean_exact_cs:w
27525 }
27526 \cs_new_protected:Npn \__regex_show:N #1
27527 {
27528     \group_begin:
27529     \tl_build_begin:N \l__regex_build_tl
27530     \cs_set_protected:Npn \__regex_branch:n
27531     {
27532         \seq_pop_right:NN \l__regex_show_prefix_seq
27533         \l__regex_internal_a_tl
27534         \__regex_show_one:n { +-branch }
27535         \seq_put_right:No \l__regex_show_prefix_seq
27536         \l__regex_internal_a_tl
27537         \use:n

```

```
27538 } 27538
27539 \cs_set_protected:Npn \__regex_group:nnnN 27539
27540 { \__regex_show_group_aux:nnnnN { } } 27540
27541 \cs_set_protected:Npn \__regex_group_no_capture:nnnN 27541
27542 { \__regex_show_group_aux:nnnnN { ~(no~capture) } } 27542
27543 \cs_set_protected:Npn \__regex_group_resetting:nnnN 27543
27544 { \__regex_show_group_aux:nnnnN { ~(resetting) } } 27544
27545 \cs_set_eq:NN \__regex_class:NnnnN \__regex_show_class:NnnnN 27545
27546 \cs_set_protected:Npn \__regex_command_K: 27546
27547 { \__regex_show_one:n { reset~match~start~(\iow_char:N\\K) } } 27547
27548 \cs_set_protected:Npn \__regex_assertion:Nn ##1##2 27548
27549 { 27549
27550 \__regex_show_one:n 27550
27551 { \bool_if:NF ##1 { negative~ } assertion:~##2 } 27551
27552 } 27552
27553 \cs_set:Npn \__regex_b_test: { word~boundary } 27553
27554 \cs_set:Npn \__regex_Z_test: { anchor~at~end~(\iow_char:N\\Z) } 27554
27555 \cs_set:Npn \__regex_A_test: { anchor~at~start~(\iow_char:N\\A) } 27555
27556 \cs_set:Npn \__regex_G_test: { anchor~at~start~of~match~(\iow_char:N\\G) } 27556
27557 \cs_set_protected:Npn \__regex_item_caseful_equal:n ##1 27557
27558 { \__regex_show_one:n { char~code~\__regex_show_char:n{##1} } } 27558
27559 \cs_set_protected:Npn \__regex_item_caseful_range:nn ##1##2 27559
27560 { 27560
27561 \__regex_show_one:n 27561
27562 { range~[\__regex_show_char:n{##1}, \__regex_show_char:n{##2}] } 27562
27563 } 27563
27564 \cs_set_protected:Npn \__regex_item_caseless_equal:n ##1 27564
27565 { \__regex_show_one:n { char~code~\__regex_show_char:n{##1}~(caseless) } } 27565
27566 \cs_set_protected:Npn \__regex_item_caseless_range:nn ##1##2 27566
27567 { 27567
27568 \__regex_show_one:n 27568
27569 { Range~[\__regex_show_char:n{##1}, \__regex_show_char:n{##2}]~(caseless) } 27569
27570 } 27570
27571 \cs_set_protected:Npn \__regex_item_catcode:nT 27571
27572 { \__regex_show_item_catcode:NnT \c_true_bool } 27572
27573 \cs_set_protected:Npn \__regex_item_catcode_reverse:nT 27573
27574 { \__regex_show_item_catcode:NnT \c_false_bool } 27574
27575 \cs_set_protected:Npn \__regex_item_reverse:n 27575
27576 { \__regex_show_scope:nn { Reversed~match } } 27576
27577 \cs_set_protected:Npn \__regex_item_exact:nn ##1##2 27577
27578 { \__regex_show_one:n { char~\__regex_show_char:n{##2},~catcode~##1 } } 27578
27579 \cs_set_eq:NN \__regex_item_exact_cs:n \__regex_show_item_exact_cs:n 27579
27580 \cs_set_protected:Npn \__regex_item_cs:n 27580
27581 { \__regex_show_scope:nn { control~sequence } } 27581
27582 \cs_set:cpn { __regex_prop_.: } { \__regex_show_one:n { any~token } } 27582
27583 \seq_clear:N \l__regex_show_prefix_seq 27583
```

```
27584 \__regex_show_push:n { ~ } 27584
27585 \cs_if_exist_use:N #1 27585
27586 \tl_build_end:N \l__regex_build_tl 27586
27587 \exp_args:NNNo 27587
27588 \group_end: 27588
27589 \tl_set:Nn \l__regex_internal_a_tl { \l__regex_build_tl } 27589
27590 } 27590
27591 \cs_new:Npn \__regex_show_char:n #1 27591
27592 { 27592
27593 \int_eval:n {#1} 27593
27594 \int_compare:nT { 32 <= #1 <= 126 } 27594
27595 { ~ ( \char_generate:nn {#1} {12} ) } 27595
27596 } 27596
27597 \cs_new_protected:Npn \__regex_show_one:n #1 27597
27598 { 27598
27599 \int_incr:N \l__regex_show_lines_int 27599
27600 \tl_build_put_right:Ne \l__regex_build_tl 27600
27601 { 27601
27602 \exp_not:N \iow_newline: 27602
27603 \seq_map_function:NN \l__regex_show_prefix_seq \use:n 27603
27604 #1 27604
27605 } 27605
27606 } 27606
27607 \cs_new_protected:Npn \__regex_show_push:n #1 27607
27608 { \seq_put_right:Ne \l__regex_show_prefix_seq { #1 ~ } } 27608
27609 \cs_new_protected:Npn \__regex_show_pop: 27609
27610 { \seq_pop_right:NN \l__regex_show_prefix_seq \l__regex_internal_a_tl } 27610
27611 \cs_new_protected:Npn \__regex_show_scope:nn #1#2 27611
27612 { 27612
27613 \__regex_show_one:n {#1} 27613
27614 \__regex_show_push:n { ~ } 27614
27615 #2 27615
27616 \__regex_show_pop: 27616
27617 } 27617
27618 \cs_new_protected:Npn \__regex_show_group_aux:nnnnN #1#2#3#4#5 27618
27619 { 27619
27620 \__regex_show_one:n { , -group~begin #1 } 27620
27621 \__regex_show_push:n { | } 27621
27622 \use_ii:nn #2 27622
27623 \__regex_show_pop: 27623
27624 \__regex_show_one:n 27624
27625 { ~-group~end \__regex_msg_repeated:nnN {#3} {#4} #5 } 27625
27626 } 27626
27627 \cs_new:Npn \__regex_show_class:NnnnN #1#2#3#4#5 27627
27628 { 27628
27629 \group_begin: 27629
```



```

27630 \tl_build_begin:N \l__regex_build_tl 27630
27631 \int_zero:N \l__regex_show_lines_int 27631
27632 \__regex_show_push:n {~} 27632
27633 #2 27633
27634 \int_compare:nTF { \l__regex_show_lines_int = \c_zero_int } 27634
27635 { 27635
27636 \group_end: 27636
27637 \__regex_show_one:n { \bool_if:NTF #1 { Fail } { Pass } } 27637
27638 } 27638
27639 { 27639
27640 \bool_if:NTF 27640
27641 { #1 && \int_compare_p:n { \l__regex_show_lines_int = \c_one_int } } 27641
27642 { 27642
27643 \group_end: 27643
27644 #2 27644
27645 \tl_build_put_right:Nn \l__regex_build_tl 27645
27646 { \__regex_msg_repeated:nnN {#3} {#4} #5 } 27646
27647 } 27647
27648 { 27648
27649 \tl_build_end:N \l__regex_build_tl 27649
27650 \exp_args:NNNo 27650
27651 \group_end: 27651
27652 \tl_set:Nn \l__regex_internal_a_tl \l__regex_build_tl 27652
27653 \__regex_show_one:n 27653
27654 { 27654
27655 \bool_if:NTF #1 { Match } { Don't~match } 27655
27656 \__regex_msg_repeated:nnN {#3} {#4} #5 27656
27657 } 27657
27658 \tl_build_put_right:Ne \l__regex_build_tl 27658
27659 { \exp_not:o \l__regex_internal_a_tl } 27659
27660 } 27660
27661 } 27661
27662 } 27662
27663 \cs_new_protected:Npn \__regex_show_item_catcode:NnT #1#2 27663
27664 { 27664
27665 \seq_set_split:Nnn \l__regex_internal_seq { } { CBEMTPUDSLOA } 27665
27666 \seq_set_filter:NNn \l__regex_internal_seq \l__regex_internal_seq 27666
27667 { \int_if_odd_p:n { #2 / \int_use:c { c__regex_catcode_##1_int } } } 27667
27668 \__regex_show_scope:nn 27668
27669 { 27669
27670 categories~ 27670
27671 \seq_map_function:NN \l__regex_internal_seq \use:n 27671
27672 , ~ 27672
27673 \bool_if:NF #1 { negative~ } class 27673
27674 } 27674
27675 } 27675

```

```
27676 \cs_new_protected:Npn \__regex_show_item_exact_cs:n #1
27677 {
27678   \seq_set_split:Nnn \l__regex_internal_seq { \scan_stop: } {#1}
27679   \seq_set_map_e:NNn \l__regex_internal_seq
27680     \l__regex_internal_seq { \iow_char:N\|#1 }
27681   \__regex_show_one:n
27682     { control~sequence~ \seq_use:Nn \l__regex_internal_seq { ~or~ } }
27683 }
27684 \int_new:N \l__regex_min_state_int
27685 \int_set:Nn \l__regex_min_state_int { 1 }
27686 \int_new:N \l__regex_max_state_int
27687 \int_new:N \l__regex_left_state_int
27688 \int_new:N \l__regex_right_state_int
27689 \seq_new:N \l__regex_left_state_seq
27690 \seq_new:N \l__regex_right_state_seq
27691 \int_new:N \l__regex_capturing_group_int
27692 \cs_new_protected:Npn \__regex_build:n
27693   { \__regex_build_aux:Nn \c_true_bool }
27694 \cs_new_protected:Npn \__regex_build:N
27695   { \__regex_build_aux:NN \c_true_bool }
27696 \cs_new_protected:Npn \__regex_build_aux:Nn #1#2
27697 {
27698   \__regex_compile:n {#2}
27699   \__regex_build_aux:NN #1 \l__regex_internal_regex
27700 }
27701 \cs_new_protected:Npn \__regex_build_aux:NN #1#2
27702 {
27703   \__regex_standard_escapechar:
27704   \int_zero:N \l__regex_capturing_group_int
27705   \int_set_eq:NN \l__regex_max_state_int \l__regex_min_state_int
27706   \__regex_build_new_state:
27707   \__regex_build_new_state:
27708   \__regex_toks_put_right:Nn \l__regex_left_state_int
27709     { \__regex_action_start_wildcard:N #1 }
27710   \__regex_group:nnnN {#2} { 1 } { 0 } \c_false_bool
27711   \__regex_toks_put_right:Nn \l__regex_right_state_int
27712     { \__regex_action_success: }
27713 }
27714 \int_new:N \g__regex_case_int
27715 \int_new:N \l__regex_case_max_group_int
27716 \cs_new_protected:Npn \__regex_case_build:n #1
27717 {
27718   \__regex_case_build_aux:Nn \c_true_bool {#1}
27719   \int_gzero:N \g__regex_case_int
27720 }
27721 \cs_generate_variant:Nn \__regex_case_build:n { e }
```

```
27722 \cs_new_protected:Npn \__regex_case_build_aux:Nn #1#2 27722
27723 { 27723
27724 \__regex_standard_escapechar: 27724
27725 \int_set_eq:NN \l__regex_max_state_int \l__regex_min_state_int 27725
27726 \__regex_build_new_state: 27726
27727 \__regex_build_new_state: 27727
27728 \__regex_toks_put_right:Nn \l__regex_left_state_int 27728
27729 { \__regex_action_start_wildcard:N #1 } 27729
27730 % 27730
27731 \__regex_build_new_state: 27731
27732 \__regex_toks_put_left:Ne \l__regex_left_state_int 27732
27733 { \__regex_action_submatch:nN \c_zero_int < } 27733
27734 \__regex_push_lr_states: 27734
27735 \int_zero:N \l__regex_case_max_group_int 27735
27736 \int_gzero:N \g__regex_case_int 27736
27737 \tl_map_inline:nn {#2} 27737
27738 { 27738
27739 \int_gincr:N \g__regex_case_int 27739
27740 \__regex_case_build_loop:n {##1} 27740
27741 } 27741
27742 \int_set_eq:NN \l__regex_capturing_group_int \l__regex_case_max_group_int 27742
27743 \__regex_pop_lr_states: 27743
27744 } 27744
27745 \cs_new_protected:Npn \__regex_case_build_loop:n #1 27745
27746 { 27746
27747 \int_set_eq:NN \l__regex_capturing_group_int \c_one_int 27747
27748 \__regex_compile_use:n {#1} 27748
27749 \int_set:Nn \l__regex_case_max_group_int 27749
27750 { \int_max:nn \l__regex_case_max_group_int \l__regex_capturing_group_int } 27750
27751 \seq_pop:NN \l__regex_right_state_seq \l__regex_internal_a_tl 27751
27752 \int_set:Nn \l__regex_right_state_int \l__regex_internal_a_tl 27752
27753 \__regex_toks_put_left:Ne \l__regex_right_state_int 27753
27754 { 27754
27755 \__regex_action_submatch:nN \c_zero_int > 27755
27756 \int_gset:Nn \g__regex_case_int 27756
27757 { \int_use:N \g__regex_case_int } 27757
27758 \__regex_action_success: 27758
27759 } 27759
27760 \__regex_toks_clear:N \l__regex_max_state_int 27760
27761 \seq_push:No \l__regex_right_state_seq 27761
27762 { \int_use:N \l__regex_max_state_int } 27762
27763 \int_incr:N \l__regex_max_state_int 27763
27764 } 27764
27765 \cs_new_protected:Npn \__regex_build_for_cs:n #1 27765
27766 { 27766
27767 \int_set_eq:NN \l__regex_min_state_int \l__regex_max_state_int 27767
```

```

27768 \__regex_build_new_state: 27768
27769 \__regex_build_new_state: 27769
27770 \__regex_push_lr_states: 27770
27771 #1 27771
27772 \__regex_pop_lr_states: 27772
27773 \__regex_toks_put_right:Nn \l__regex_right_state_int 27773
27774 { 27774
27775 \if_int_compare:w -2 = \l__regex_curr_char_int 27775
27776 \exp_after:wN \__regex_action_success: 27776
27777 \fi: 27777
27778 } 27778
27779 } 27779
27780 \cs_new_protected:Npn \__regex_push_lr_states: 27780
27781 { 27781
27782 \seq_push:No \l__regex_left_state_seq 27782
27783 { \int_use:N \l__regex_left_state_int } 27783
27784 \seq_push:No \l__regex_right_state_seq 27784
27785 { \int_use:N \l__regex_right_state_int } 27785
27786 } 27786
27787 \cs_new_protected:Npn \__regex_pop_lr_states: 27787
27788 { 27788
27789 \seq_pop:NN \l__regex_left_state_seq \l__regex_internal_a_tl 27789
27790 \int_set:Nn \l__regex_left_state_int \l__regex_internal_a_tl 27790
27791 \seq_pop:NN \l__regex_right_state_seq \l__regex_internal_a_tl 27791
27792 \int_set:Nn \l__regex_right_state_int \l__regex_internal_a_tl 27792
27793 } 27793
27794 \cs_new_protected:Npn \__regex_build_transition_left:NNN #1#2#3 27794
27795 { \__regex_toks_put_left:Ne #2 { #1 { \tex_the:D \__regex_int_eval:w #3 - #2 } } } 27795
27796 \cs_new_protected:Npn \__regex_build_transition_right:nNn #1#2#3 27796
27797 { \__regex_toks_put_right:Ne #2 { #1 { \tex_the:D \__regex_int_eval:w #3 - #2 } } } 27797
27798 \cs_new_protected:Npn \__regex_build_new_state: 27798
27799 { 27799
27800 \__regex_toks_clear:N \l__regex_max_state_int 27800
27801 \int_set_eq:NN \l__regex_left_state_int \l__regex_right_state_int 27801
27802 \int_set_eq:NN \l__regex_right_state_int \l__regex_max_state_int 27802
27803 \int_incr:N \l__regex_max_state_int 27803
27804 } 27804
27805 \cs_new_protected:Npn \__regex_build_transitions_laziness:NNNNN #1#2#3#4#5 27805
27806 { 27806
27807 \__regex_build_new_state: 27807
27808 \__regex_toks_put_right:Ne \l__regex_left_state_int 27808
27809 { 27809
27810 \if_meaning:w \c_true_bool #1 27810
27811 #2 { \tex_the:D \__regex_int_eval:w #3 - \l__regex_left_state_int } 27811
27812 #4 { \tex_the:D \__regex_int_eval:w #5 - \l__regex_left_state_int } 27812
27813 \else: 27813

```

```
27814         #4 { \tex_the:D \_regex_int_eval:w #5 - \l__regex_left_state_int } 27814
27815         #2 { \tex_the:D \_regex_int_eval:w #3 - \l__regex_left_state_int } 27815
27816     \fi: 27816
27817 } 27817
27818 } 27818
27819 \cs_new_protected:Npn \_regex_class:NnnnN #1#2#3#4#5 27819
27820 { 27820
27821     \cs_set:Npe \_regex_tests_action_cost:n ##1 27821
27822     { 27822
27823         \exp_not:n { \exp_not:n {#2} } 27823
27824         \bool_if:NTF #1 27824
27825         { \_regex_break_point:TF { \_regex_action_cost:n {##1} } { } } 27825
27826         { \_regex_break_point:TF { } { \_regex_action_cost:n {##1} } } 27826
27827     } 27827
27828     \if_case:w - #4 \exp_stop_f: 27828
27829         \_regex_class_repeat:n {#3} 27829
27830     \or: \_regex_class_repeat:nN {#3} #5 27830
27831     \else: \_regex_class_repeat:nnN {#3} {#4} #5 27831
27832     \fi: 27832
27833 } 27833
27834 \cs_new:Npn \_regex_tests_action_cost:n { \_regex_action_cost:n } 27834
27835 \cs_new_protected:Npn \_regex_class_repeat:n #1 27835
27836 { 27836
27837     \prg_replicate:nn {#1} 27837
27838     { 27838
27839         \_regex_build_new_state: 27839
27840         \_regex_build_transition_right:nNn \_regex_tests_action_cost:n 27840
27841         \l__regex_left_state_int \l__regex_right_state_int 27841
27842     } 27842
27843 } 27843
27844 \cs_new_protected:Npn \_regex_class_repeat:nN #1#2 27844
27845 { 27845
27846     \if_int_compare:w #1 = \c_zero_int 27846
27847         \_regex_build_transitions_laziness:NNNN #2 27847
27848         \_regex_action_free:n \l__regex_right_state_int 27848
27849         \_regex_tests_action_cost:n \l__regex_left_state_int 27849
27850     \else: 27850
27851         \_regex_class_repeat:n {#1} 27851
27852         \int_set_eq:NN \l__regex_internal_a_int \l__regex_left_state_int 27852
27853         \_regex_build_transitions_laziness:NNNN #2 27853
27854         \_regex_action_free:n \l__regex_right_state_int 27854
27855         \_regex_action_free:n \l__regex_internal_a_int 27855
27856     \fi: 27856
27857 } 27857
27858 \cs_new_protected:Npn \_regex_class_repeat:nnN #1#2#3 27858
27859 { 27859
```

```
27860 \__regex_class_repeat:n {#1} 27860
27861 \int_set:Nn \l__regex_internal_a_int 27861
27862 { \l__regex_max_state_int + #2 - \c_one_int } 27862
27863 \prg_replicate:nn { #2 } 27863
27864 { 27864
27865 \__regex_build_transitions_laziness:NNNN #3 27865
27866 \__regex_action_free:n \l__regex_internal_a_int 27866
27867 \__regex_tests_action_cost:n \l__regex_right_state_int 27867
27868 } 27868
27869 } 27869
27870 \cs_new_protected:Npn \__regex_group_aux:nnnnN #1#2#3#4#5 27870
27871 { 27871
27872 \if_int_compare:w #3 = \c_zero_int 27872
27873 \__regex_build_new_state: 27873
27874 \__regex_build_transition_right:nNn \__regex_action_free_group:n 27874
27875 \l__regex_left_state_int \l__regex_right_state_int 27875
27876 \fi: 27876
27877 \__regex_build_new_state: 27877
27878 \__regex_push_lr_states: 27878
27879 #2 27879
27880 \__regex_pop_lr_states: 27880
27881 \if_case:w - #4 \exp_stop_f: 27881
27882 \__regex_group_repeat:nn {#1} {#3} 27882
27883 \or: \__regex_group_repeat:nnN {#1} {#3} #5 27883
27884 \else: \__regex_group_repeat:nnnN {#1} {#3} {#4} #5 27884
27885 \fi: 27885
27886 } 27886
27887 \cs_new_protected:Npn \__regex_group:nnnN #1 27887
27888 { 27888
27889 \exp_args:No \__regex_group_aux:nnnnN 27889
27890 { \int_use:N \l__regex_capturing_group_int } 27890
27891 { 27891
27892 \int_incr:N \l__regex_capturing_group_int 27892
27893 #1 27893
27894 } 27894
27895 } 27895
27896 \cs_new_protected:Npn \__regex_group_no_capture:nnnN 27896
27897 { \__regex_group_aux:nnnnN { -1 } } 27897
27898 \cs_new_protected:Npn \__regex_group_resetting:nnnN #1 27898
27899 { 27899
27900 \__regex_group_aux:nnnnN { -1 } 27900
27901 { 27901
27902 \exp_args:Noo \__regex_group_resetting_loop:nnNn 27902
27903 { \int_use:N \l__regex_capturing_group_int } 27903
27904 { \int_use:N \l__regex_capturing_group_int } 27904
27905 #1 27905
```

```
27906 { ?? \prg_break:n } { } 27906
27907 \prg_break_point: 27907
27908 } 27908
27909 } 27909
27910 \cs_new_protected:Npn \__regex_group_resetting_loop:nnNn #1#2#3#4 27910
27911 { 27911
27912 \use_none:nn #3 { \int_set:Nn \l__regex_capturing_group_int {#1} } 27912
27913 \int_set:Nn \l__regex_capturing_group_int {#2} 27913
27914 #3 {#4} 27914
27915 \exp_args:Ne \__regex_group_resetting_loop:nnNn 27915
27916 { \int_max:nn {#1} \l__regex_capturing_group_int } 27916
27917 {#2} 27917
27918 } 27918
27919 \cs_new_protected:Npn \__regex_branch:n #1 27919
27920 { 27920
27921 \__regex_build_new_state: 27921
27922 \seq_get:NN \l__regex_left_state_seq \l__regex_internal_a_tl 27922
27923 \int_set:Nn \l__regex_left_state_int \l__regex_internal_a_tl 27923
27924 \__regex_build_transition_right:nNn \__regex_action_free:n 27924
27925 \l__regex_left_state_int \l__regex_right_state_int 27925
27926 #1 27926
27927 \seq_get:NN \l__regex_right_state_seq \l__regex_internal_a_tl 27927
27928 \__regex_build_transition_right:nNn \__regex_action_free:n 27928
27929 \l__regex_right_state_int \l__regex_internal_a_tl 27929
27930 } 27930
27931 \cs_new_protected:Npn \__regex_group_repeat:nn #1#2 27931
27932 { 27932
27933 \if_int_compare:w #2 = \c_zero_int 27933
27934 \int_set:Nn \l__regex_max_state_int 27934
27935 { \l__regex_left_state_int - \c_one_int } 27935
27936 \__regex_build_new_state: 27936
27937 \else: 27937
27938 \__regex_group_repeat_aux:n {#2} 27938
27939 \__regex_group_submatches:nNN {#1} 27939
27940 \l__regex_internal_a_int \l__regex_right_state_int 27940
27941 \__regex_build_new_state: 27941
27942 \fi: 27942
27943 } 27943
27944 \cs_new_protected:Npn \__regex_group_submatches:nNN #1#2#3 27944
27945 { 27945
27946 \if_int_compare:w #1 > - \c_one_int 27946
27947 \__regex_toks_put_left:Ne #2 { \__regex_action_submatch:nN {#1} < } 27947
27948 \__regex_toks_put_left:Ne #3 { \__regex_action_submatch:nN {#1} > } 27948
27949 \fi: 27949
27950 } 27950
27951 \cs_new_protected:Npn \__regex_group_repeat_aux:n #1 27951
```



```
27952 { 27952
27953 \__regex_build_transition_right:nNn \__regex_action_free:n 27953
27954 \l__regex_right_state_int \l__regex_max_state_int 27954
27955 \int_set_eq:NN \l__regex_internal_a_int \l__regex_left_state_int 27955
27956 \int_set_eq:NN \l__regex_internal_b_int \l__regex_max_state_int 27956
27957 \if_int_compare:w \__regex_int_eval:w #1 > \c_one_int 27957
27958 \int_set:Nn \l__regex_internal_c_int 27958
27959 { 27959
27960 ( #1 - \c_one_int ) 27960
27961 * ( \l__regex_internal_b_int - \l__regex_internal_a_int ) 27961
27962 } 27962
27963 \int_add:Nn \l__regex_right_state_int \l__regex_internal_c_int 27963
27964 \int_add:Nn \l__regex_max_state_int \l__regex_internal_c_int 27964
27965 \__regex_toks_memcpy:NNn 27965
27966 \l__regex_internal_b_int 27966
27967 \l__regex_internal_a_int 27967
27968 \l__regex_internal_c_int 27968
27969 \fi: 27969
27970 } 27970
27971 \cs_new_protected:Npn \__regex_group_repeat:nnN #1#2#3 27971
27972 { 27972
27973 \if_int_compare:w #2 = \c_zero_int 27973
27974 \__regex_group_submatches:nNN {#1} 27974
27975 \l__regex_left_state_int \l__regex_right_state_int 27975
27976 \int_set:Nn \l__regex_internal_a_int 27976
27977 { \l__regex_left_state_int - \c_one_int } 27977
27978 \__regex_build_transition_right:nNn \__regex_action_free:n 27978
27979 \l__regex_right_state_int \l__regex_internal_a_int 27979
27980 \__regex_build_new_state: 27980
27981 \if_meaning:w \c_true_bool #3 27981
27982 \__regex_build_transition_left:NNN \__regex_action_free:n 27982
27983 \l__regex_internal_a_int \l__regex_right_state_int 27983
27984 \else: 27984
27985 \__regex_build_transition_right:nNn \__regex_action_free:n 27985
27986 \l__regex_internal_a_int \l__regex_right_state_int 27986
27987 \fi: 27987
27988 \else: 27988
27989 \__regex_group_repeat_aux:n {#2} 27989
27990 \__regex_group_submatches:nNN {#1} 27990
27991 \l__regex_internal_a_int \l__regex_right_state_int 27991
27992 \if_meaning:w \c_true_bool #3 27992
27993 \__regex_build_transition_right:nNn \__regex_action_free_group:n 27993
27994 \l__regex_right_state_int \l__regex_internal_a_int 27994
27995 \else: 27995
27996 \__regex_build_transition_left:NNN \__regex_action_free_group:n 27996
27997 \l__regex_right_state_int \l__regex_internal_a_int 27997
```

```

27998     \fi:
27999     \__regex_build_new_state:
28000 \fi:
28001 }
28002 \cs_new_protected:Npn \__regex_group_repeat:nnnN #1#2#3#4
28003 {
28004     \__regex_group_submatches:nnN {#1}
28005     \l__regex_left_state_int \l__regex_right_state_int
28006     \__regex_group_repeat_aux:n { #2 + #3 }
28007     \if_meaning:w \c_true_bool #4
28008         \int_set_eq:NN \l__regex_left_state_int \l__regex_max_state_int
28009         \prg_replicate:nn { #3 }
28010         {
28011             \int_sub:Nn \l__regex_left_state_int
28012                 { \l__regex_internal_b_int - \l__regex_internal_a_int }
28013             \__regex_build_transition_left:NNN \__regex_action_free:n
28014                 \l__regex_left_state_int \l__regex_max_state_int
28015         }
28016     \else:
28017         \prg_replicate:nn { #3 - \c_one_int }
28018         {
28019             \int_sub:Nn \l__regex_right_state_int
28020                 { \l__regex_internal_b_int - \l__regex_internal_a_int }
28021             \__regex_build_transition_right:nNn \__regex_action_free:n
28022                 \l__regex_right_state_int \l__regex_max_state_int
28023         }
28024     \if_int_compare:w #2 = \c_zero_int
28025         \int_set:Nn \l__regex_right_state_int
28026             { \l__regex_left_state_int - \c_one_int }
28027     \else:
28028         \int_sub:Nn \l__regex_right_state_int
28029             { \l__regex_internal_b_int - \l__regex_internal_a_int }
28030     \fi:
28031     \__regex_build_transition_right:nNn \__regex_action_free:n
28032         \l__regex_right_state_int \l__regex_max_state_int
28033 \fi:
28034 \__regex_build_new_state:
28035 }
28036 \cs_new_protected:Npn \__regex_assertion:Nn #1#2
28037 {
28038     \__regex_build_new_state:
28039     \__regex_toks_put_right:Ne \l__regex_left_state_int
28040     {
28041         \exp_not:n {#2}
28042         \__regex_break_point:TF
28043         \bool_if:NF #1 { { } }

```

```

28044         {
28045             \__regex_action_free:n
28046         {
28047             \tex_the:D \__regex_int_eval:w
28048             \l__regex_right_state_int - \l__regex_left_state_int
28049         }
28050     }
28051     \bool_if:NT #1 { { } }
28052 }
28053 }
28054 \cs_new_protected:Npn \__regex_b_test:
28055 {
28056     \group_begin:
28057     \int_set_eq:NN \l__regex_curr_char_int \l__regex_last_char_int
28058     \__regex_prop_w:
28059     \__regex_break_point:TF
28060     { \group_end: \__regex_item_reverse:n { \__regex_prop_w: } }
28061     { \group_end: \__regex_prop_w: }
28062 }
28063 \cs_new_protected:Npn \__regex_Z_test:
28064 {
28065     \if_int_compare:w -2 = \l__regex_curr_char_int
28066     \exp_after:wN \__regex_break_true:w
28067     \fi:
28068 }
28069 \cs_new_protected:Npn \__regex_A_test:
28070 {
28071     \if_int_compare:w -2 = \l__regex_last_char_int
28072     \exp_after:wN \__regex_break_true:w
28073     \fi:
28074 }
28075 \cs_new_protected:Npn \__regex_G_test:
28076 {
28077     \if_int_compare:w \l__regex_curr_pos_int = \l__regex_start_pos_int
28078     \exp_after:wN \__regex_break_true:w
28079     \fi:
28080 }
28081 \cs_new_protected:Npn \__regex_command_K:
28082 {
28083     \__regex_build_new_state:
28084     \__regex_toks_put_right:Ne \l__regex_left_state_int
28085     {
28086         \__regex_action_submatch:nN \c_zero_int <
28087         \bool_set_true:N \l__regex_fresh_thread_bool
28088         \__regex_action_free:n
28089         {

```

```

28090         \tex_the:D \__regex_int_eval:w
28091         \l__regex_right_state_int - \l__regex_left_state_int
28092     }
28093     \bool_set_false:N \l__regex_fresh_thread_bool
28094 }
28095 }
28096 \int_new:N \l__regex_min_pos_int
28097 \int_new:N \l__regex_max_pos_int
28098 \int_new:N \l__regex_curr_pos_int
28099 \int_new:N \l__regex_start_pos_int
28100 \int_new:N \l__regex_success_pos_int
28101 \int_new:N \l__regex_curr_char_int
28102 \int_new:N \l__regex_curr_catcode_int
28103 \tl_new:N \l__regex_curr_token_tl
28104 \int_new:N \l__regex_last_char_int
28105 \int_new:N \l__regex_last_char_success_int
28106 \int_new:N \l__regex_case_changed_char_int
28107 \int_new:N \l__regex_curr_state_int
28108 \tl_new:N \l__regex_curr_submatches_tl
28109 \tl_new:N \l__regex_success_submatches_tl
28110 \int_new:N \l__regex_step_int
28111 \int_new:N \l__regex_min_thread_int
28112 \int_new:N \l__regex_max_thread_int
28113 \intarray_new:Nn \g__regex_state_active_intarray { 65536 }
28114 \intarray_new:Nn \g__regex_thread_info_intarray { 65536 }
28115 \tl_new:N \l__regex_matched_analysis_tl
28116 \tl_new:N \l__regex_curr_analysis_tl
28117 \tl_new:N \l__regex_every_match_tl
28118 \bool_new:N \l__regex_fresh_thread_bool
28119 \bool_new:N \l__regex_empty_success_bool
28120 \cs_new_eq:NN \__regex_if_two_empty_matches:F \use:n
28121 \bool_new:N \g__regex_success_bool
28122 \bool_new:N \l__regex_saved_success_bool
28123 \bool_new:N \l__regex_match_success_bool
28124 \cs_new_protected:Npn \__regex_match:n #1
28125 {
28126     \__regex_match_init:
28127     \__regex_match_once_init:
28128     \tl_analysis_map_inline:nn {#1}
28129         { \__regex_match_one_token:nnN {##1} {##2} ##3 }
28130     \__regex_match_one_token:nnN { } { -2 } F
28131     \prg_break_point:Nn \__regex_maplike_break: { }
28132 }
28133 \cs_new_protected:Npn \__regex_match_cs:n #1
28134 {
28135     \int_set_eq:NN \l__regex_min_thread_int \l__regex_max_thread_int

```

```

28136 \__regex_match_init:
28137 \__regex_match_once_init:
28138 \str_map_inline:nn {#1}
28139 {
28140     \tl_if_blank:nTF {##1}
28141         { \__regex_match_one_token:nnN {##1} {`##1} A }
28142         { \__regex_match_one_token:nnN {##1} {`##1} C }
28143     }
28144     \__regex_match_one_token:nnN { } { -2 } F
28145     \prg_break_point:Nn \__regex_maplike_break: { }
28146 }
28147 \cs_new_protected:Npn \__regex_match_init:
28148 {
28149     \bool_gset_false:N \g__regex_success_bool
28150     \int_step_inline:nnn
28151         \l__regex_min_state_int { \l__regex_max_state_int - \c_one_int }
28152     {
28153         \__kernel_intarray_gset:Nnn
28154         \g__regex_state_active_intarray {##1} \c_one_int
28155     }
28156     \int_zero:N \l__regex_step_int
28157     \int_set:Nn \l__regex_min_pos_int { 2 }
28158     \int_set_eq:NN \l__regex_success_pos_int \l__regex_min_pos_int
28159     \int_set:Nn \l__regex_last_char_success_int { -2 }
28160     \tl_build_begin:N \l__regex_matched_analysis_tl
28161     \tl_clear:N \l__regex_curr_analysis_tl
28162     \int_set_eq:NN \l__regex_min_submatch_int \c_one_int
28163     \int_set_eq:NN \l__regex_submatch_int \l__regex_min_submatch_int
28164     \bool_set_false:N \l__regex_empty_success_bool
28165 }
28166 \cs_new_protected:Npn \__regex_match_once_init:
28167 {
28168     \if_meaning:w \c_true_bool \l__regex_empty_success_bool
28169         \cs_set:Npn \__regex_if_two_empty_matches:F
28170         {
28171             \int_compare:nNnF
28172                 \l__regex_start_pos_int = \l__regex_curr_pos_int
28173             }
28174     \else:
28175         \cs_set_eq:NN \__regex_if_two_empty_matches:F \use:n
28176     \fi:
28177     \int_set_eq:NN \l__regex_start_pos_int \l__regex_success_pos_int
28178     \bool_set_false:N \l__regex_match_success_bool
28179     \tl_set:Ne \l__regex_curr_submatches_tl
28180         { \prg_replicate:nn { 2 * \l__regex_capturing_group_int } { 0 , } }
28181     \int_set_eq:NN \l__regex_max_thread_int \l__regex_min_thread_int

```

```
28182 \__regex_store_state:n { \l__regex_min_state_int } 28182
28183 \int_set:Nn \l__regex_curr_pos_int { \l__regex_start_pos_int - \c_one_int } 28183
28184 \int_set_eq:NN \l__regex_curr_char_int \l__regex_last_char_success_int 28184
28185 \tl_build_get_intermediate:NN \l__regex_matched_analysis_tl \l__regex_internal_a_tl 28185
28186 \exp_args:NNf \__regex_match_once_init_aux: 28186
28187 \tl_map_inline:nn 28187
28188 { \exp_after:wN \l__regex_internal_a_tl \l__regex_curr_analysis_tl } 28188
28189 { \__regex_match_one_token:nnN ##1 } 28189
28190 \prg_break_point:Nn \__regex_maplike_break: { } 28190
28191 } 28191
28192 \cs_new_protected:Npn \__regex_match_once_init_aux: 28192
28193 { 28193
28194 \tl_build_begin:N \l__regex_matched_analysis_tl 28194
28195 \tl_clear:N \l__regex_curr_analysis_tl 28195
28196 } 28196
28197 \cs_new_protected:Npn \__regex_single_match: 28197
28198 { 28198
28199 \tl_set:Nn \l__regex_every_match_tl 28199
28200 { 28200
28201 \bool_gset_eq:NN 28201
28202 \g__regex_success_bool 28202
28203 \l__regex_match_success_bool 28203
28204 \__regex_maplike_break: 28204
28205 } 28205
28206 } 28206
28207 \cs_new_protected:Npn \__regex_multi_match:n #1 28207
28208 { 28208
28209 \tl_set:Nn \l__regex_every_match_tl 28209
28210 { 28210
28211 \if_meaning:w \c_false_bool \l__regex_match_success_bool 28211
28212 \exp_after:wN \__regex_maplike_break: 28212
28213 \fi: 28213
28214 \bool_gset_true:N \g__regex_success_bool 28214
28215 #1 28215
28216 \__regex_match_once_init: 28216
28217 } 28217
28218 } 28218
28219 \cs_new_protected:Npn \__regex_match_one_token:nnN #1#2#3 28219
28220 { 28220
28221 \int_add:Nn \l__regex_step_int { 2 } 28221
28222 \int_incr:N \l__regex_curr_pos_int 28222
28223 \int_set_eq:NN \l__regex_last_char_int \l__regex_curr_char_int 28223
28224 \cs_set_eq:NN \__regex_maybe_compute_ccc: \__regex_compute_case_changed_char: 28224
28225 \tl_set:Nn \l__regex_curr_token_tl {#1} 28225
28226 \int_set:Nn \l__regex_curr_char_int {#2} 28226
28227 \int_set:Nn \l__regex_curr_catcode_int { "#3 } 28227
```



```

28228 \tl_build_put_right:Ne \l__regex_matched_analysis_tl 28228
28229 { \exp_not:o \l__regex_curr_analysis_tl } 28229
28230 \tl_set:Nn \l__regex_curr_analysis_tl { { {#1} {#2} #3 } } 28230
28231 \use:e 28231
28232 { 28232
28233 \int_set_eq:NN \l__regex_max_thread_int \l__regex_min_thread_int 28233
28234 \int_step_function:nnN 28234
28235 \l__regex_min_thread_int 28235
28236 { \l__regex_max_thread_int - \c_one_int } 28236
28237 \__regex_match_one_active:n 28237
28238 } 28238
28239 \prg_break_point: 28239
28240 \bool_set_false:N \l__regex_fresh_thread_bool 28240
28241 \if_int_compare:w \l__regex_max_thread_int > \l__regex_min_thread_int 28241
28242 \if_int_compare:w -2 < \l__regex_curr_char_int 28242
28243 \exp_after:wN \use_i:nn 28243
28244 \fi: 28244
28245 \fi: 28245
28246 \l__regex_every_match_tl 28246
28247 } 28247
28248 \cs_new:Npn \__regex_match_one_active:n #1 28248
28249 { 28249
28250 \__regex_use_state_and_submatches:w 28250
28251 \__kernel_intarray_range_to_clist:Nnn 28251
28252 \g__regex_thread_info_intarray 28252
28253 { \c_one_int + #1 * (\l__regex_capturing_group_int * 2 + \c_one_int) } 28253
28254 { (\c_one_int + #1) * (\l__regex_capturing_group_int * 2 + \c_one_int) } 28254
28255 \__regex_sep: 28255
28256 } 28256
28257 \cs_new_protected:Npn \__regex_use_state: 28257
28258 { 28258
28259 \__kernel_intarray_gset:Nnn \g__regex_state_active_intarray 28259
28260 \l__regex_curr_state_int \l__regex_step_int 28260
28261 \__regex_toks_use:w \l__regex_curr_state_int 28261
28262 \__kernel_intarray_gset:Nnn \g__regex_state_active_intarray 28262
28263 \l__regex_curr_state_int 28263
28264 { \__regex_int_eval:w \l__regex_step_int + \c_one_int \scan_stop: } 28264
28265 } 28265
28266 \cs_new_protected:Npn \__regex_use_state_and_submatches:w #1 , #2 \__regex_sep: 28266
28267 { 28267
28268 \int_set:Nn \l__regex_curr_state_int {#1} 28268
28269 \if_int_compare:w 28269
28270 \__kernel_intarray_item:Nn \g__regex_state_active_intarray 28270
28271 \l__regex_curr_state_int 28271
28272 < \l__regex_step_int 28272
28273 \tl_set:Nn \l__regex_curr_submatches_tl { #2 , } 28273

```



```
28274 \exp_after:wN \__regex_use_state: 28274
28275 \fi: 28275
28276 \scan_stop: 28276
28277 } 28277
28278 \cs_new_protected:Npn \__regex_action_start_wildcard:N #1 28278
28279 { 28279
28280 \bool_set_true:N \l__regex_fresh_thread_bool 28280
28281 \__regex_action_free:n {1} 28281
28282 \bool_set_false:N \l__regex_fresh_thread_bool 28282
28283 \bool_if:NT #1 { \__regex_action_cost:n {0} } 28283
28284 } 28284
28285 \cs_new_protected:Npn \__regex_action_free:n 28285
28286 { \__regex_action_free_aux:nn { > \l__regex_step_int \else: } } 28286
28287 \cs_new_protected:Npn \__regex_action_free_group:n 28287
28288 { \__regex_action_free_aux:nn { < \l__regex_step_int } } 28288
28289 \cs_new_protected:Npn \__regex_action_free_aux:nn #1#2 28289
28290 { 28290
28291 \use:e 28291
28292 { 28292
28293 \int_add:Nn \l__regex_curr_state_int {#2} 28293
28294 \exp_not:n 28294
28295 { 28295
28296 \if_int_compare:w 28296
28297 \__kernel_intarray_item:Nn \g__regex_state_active_intarray 28297
28298 \l__regex_curr_state_int 28298
28299 #1 28299
28300 \exp_after:wN \__regex_use_state: 28300
28301 \fi: 28301
28302 } 28302
28303 \int_set:Nn \l__regex_curr_state_int 28303
28304 { \int_use:N \l__regex_curr_state_int } 28304
28305 \tl_set:Nn \exp_not:N \l__regex_curr_submatches_tl 28305
28306 { \exp_not:o \l__regex_curr_submatches_tl } 28306
28307 } 28307
28308 } 28308
28309 \cs_new_protected:Npn \__regex_action_cost:n #1 28309
28310 { 28310
28311 \exp_args:No \__regex_store_state:n 28311
28312 { \tex_the:D \__regex_int_eval:w \l__regex_curr_state_int + #1 } 28312
28313 } 28313
28314 \cs_new_protected:Npn \__regex_store_state:n #1 28314
28315 { 28315
28316 \exp_args:No \__regex_store_submatches:nn 28316
28317 \l__regex_curr_submatches_tl {#1} 28317
28318 \int_incr:N \l__regex_max_thread_int 28318
28319 } 28319
```

```
28320 \cs_new_protected:Npn \__regex_store_submatches:nn #1#2 28320
28321 { 28321
28322 \__kernel_intarray_gset_range_from_clist:Nnn 28322
28323 \g__regex_thread_info_intarray 28323
28324 { 28324
28325 \__regex_int_eval:w 28325
28326 \c_one_int + \l__regex_max_thread_int * 28326
28327 (\l__regex_capturing_group_int * 2 + \c_one_int) 28327
28328 } 28328
28329 { #2 , #1 } 28329
28330 } 28330
28331 \cs_new_protected:Npn \__regex_disable_submatches: 28331
28332 { 28332
28333 \cs_set_protected:Npn \__regex_store_submatches:n ##1 { } 28333
28334 \cs_set_protected:Npn \__regex_action_submatch:nN ##1##2 { } 28334
28335 } 28335
28336 \cs_new_protected:Npn \__regex_action_submatch:nN #1#2 28336
28337 { 28337
28338 \exp_after:wN \__regex_action_submatch_aux:w 28338
28339 \l__regex_curr_submatches_tl \__regex_sep: {#1} #2 28339
28340 } 28340
28341 \cs_new_protected:Npn \__regex_action_submatch_aux:w #1 \__regex_sep: #2#3 28341
28342 { 28342
28343 \tl_set:Nc \l__regex_curr_submatches_tl 28343
28344 { 28344
28345 \prg_replicate:nn 28345
28346 { #2 \if_meaning:w > #3 + \l__regex_capturing_group_int \fi: } 28346
28347 { \__regex_action_submatch_auxii:w } 28347
28348 \__regex_action_submatch_auxiii:w 28348
28349 #1 28349
28350 } 28350
28351 } 28351
28352 \cs_new:Npn \__regex_action_submatch_auxii:w 28352
28353 #1 \__regex_action_submatch_auxiii:w #2 , 28353
28354 { #2 , #1 \__regex_action_submatch_auxiii:w } 28354
28355 \cs_new:Npn \__regex_action_submatch_auxiii:w #1 , 28355
28356 { \int_use:N \l__regex_curr_pos_int , } 28356
28357 \cs_new_protected:Npn \__regex_action_success: 28357
28358 { 28358
28359 \__regex_if_two_empty_matches:F 28359
28360 { 28360
28361 \bool_set_true:N \l__regex_match_success_bool 28361
28362 \bool_set_eq:NN \l__regex_empty_success_bool 28362
28363 \l__regex_fresh_thread_bool 28363
28364 \int_set_eq:NN \l__regex_success_pos_int \l__regex_curr_pos_int 28364
28365 \int_set_eq:NN \l__regex_last_char_success_int \l__regex_last_char_int 28365
```

```

28366         \tl_build_begin:N \l__regex_matched_analysis_tl          28366
28367         \tl_set_eq:NN \l__regex_success_submatches_tl          28367
28368         \l__regex_curr_submatches_tl          28368
28369     \prg_break:          28369
28370 }          28370
28371 }          28371
28372 \int_new:N \l__regex_replacement_csnames_int          28372
28373 \tl_new:N \l__regex_replacement_category_tl          28373
28374 \seq_new:N \l__regex_replacement_category_seq          28374
28375 \tl_new:N \g__regex_balance_tl          28375
28376 \cs_new:Npn \__regex_replacement_balance_one_match:n #1          28376
28377 { - \__regex_submatch_balance:n {#1} }          28377
28378 \cs_new:Npn \__regex_replacement_do_one_match:n #1          28378
28379 {          28379
28380     \__regex_query_range:nn          28380
28381     { \__kernel_intarray_item:Nn \g__regex_submatch_prev_intarray {#1} }          28381
28382     { \__kernel_intarray_item:Nn \g__regex_submatch_begin_intarray {#1} }          28382
28383 }          28383
28384 \cs_new:Npn \__regex_replacement_exp_not:N #1 { \exp_not:n {#1} }          28384
28385 \cs_new_eq:NN \__regex_replacement_exp_not:V \exp_not:V          28385
28386 \cs_new:Npn \__regex_query_range:nn #1#2          28386
28387 {          28387
28388     \exp_after:wN \__regex_query_range_loop:ww          28388
28389     \int_value:w \__regex_int_eval:w #1 \exp_after:wN \__regex_sep:          28389
28390     \int_value:w \__regex_int_eval:w #2 \__regex_sep:          28390
28391     \prg_break_point:          28391
28392 }          28392
28393 \cs_new:Npn \__regex_query_range_loop:ww #1 \__regex_sep: #2 \__regex_sep:          28393
28394 {          28394
28395     \if_int_compare:w #1 < #2 \exp_stop_f:          28395
28396     \else:          28396
28397         \prg_break:n          28397
28398     \fi:          28398
28399     \__regex_toks_use:w #1 \exp_stop_f:          28399
28400     \exp_after:wN \__regex_query_range_loop:ww          28400
28401     \int_value:w \__regex_int_eval:w #1 + \c_one_int \__regex_sep: #2 \__regex_sep:          28401
28402 }          28402
28403 \cs_new:Npn \__regex_query_submatch:n #1          28403
28404 {          28404
28405     \__regex_query_range:nn          28405
28406     { \__kernel_intarray_item:Nn \g__regex_submatch_begin_intarray {#1} }          28406
28407     { \__kernel_intarray_item:Nn \g__regex_submatch_end_intarray {#1} }          28407
28408 }          28408
28409 \cs_new_protected:Npn \__regex_submatch_balance:n #1          28409
28410 {          28410
28411     \tex_the:D \__regex_int_eval:w          28411

```

28412	__regex_intarray_item:NnF \g__regex_balance_intarray	28412
28413	{	28413
28414	__kernel_intarray_item:Nn	28414
28415	\g__regex_submatch_end_intarray {#1}	28415
28416	}	28416
28417	\c_zero_int	28417
28418	-	28418
28419	__regex_intarray_item:NnF \g__regex_balance_intarray	28419
28420	{	28420
28421	__kernel_intarray_item:Nn	28421
28422	\g__regex_submatch_begin_intarray {#1}	28422
28423	}	28423
28424	\c_zero_int	28424
28425	\scan_stop:	28425
28426	}	28426
28427	\cs_new_protected:Npn __regex_replacement:n	28427
28428	{ __regex_replacement_apply:Nn __regex_replacement_set:n }	28428
28429	\cs_new_protected:Npn __regex_replacement_apply:Nn #1#2	28429
28430	{	28430
28431	\group_begin:	28431
28432	\tl_build_begin:N \l__regex_build_tl	28432
28433	\int_zero:N \l__regex_balance_int	28433
28434	\tl_gclear:N \g__regex_balance_tl	28434
28435	__regex_escape_use:nnnn	28435
28436	{	28436
28437	\if_charcode:w \c_right_brace_str ##1	28437
28438	__regex_replacement_rbrace:N	28438
28439	\else:	28439
28440	\if_charcode:w \c_left_brace_str ##1	28440
28441	__regex_replacement_lbrace:N	28441
28442	\else:	28442
28443	__regex_replacement_normal:n	28443
28444	\fi:	28444
28445	\fi:	28445
28446	##1	28446
28447	}	28447
28448	{ __regex_replacement_escaped:N ##1 }	28448
28449	{ __regex_replacement_normal:n ##1 }	28449
28450	{#2}	28450
28451	\prg_do_nothing: \prg_do_nothing:	28451
28452	\if_int_compare:w \l__regex_replacement_csnames_int > \c_zero_int	28452
28453	\msg_error:nne { regex } { replacement-missing-rbrace }	28453
28454	{ \int_use:N \l__regex_replacement_csnames_int }	28454
28455	\tl_build_put_right:Ne \l__regex_build_tl	28455
28456	{ \prg_replicate:nn \l__regex_replacement_csnames_int \cs_end: }	28456
28457	\fi:	28457

```
28458 \seq_if_empty:NF \l__regex_replacement_category_seq 28458
28459 { 28459
28460 \msg_error:nne { regex } { replacement-missing-rparen } 28460
28461 { \seq_count:N \l__regex_replacement_category_seq } 28461
28462 \seq_clear:N \l__regex_replacement_category_seq 28462
28463 } 28463
28464 \tl_gput_right:Ne \g__regex_balance_tl 28464
28465 { + \int_use:N \l__regex_balance_int } 28465
28466 \tl_build_end:N \l__regex_build_tl 28466
28467 \exp_args:NNo 28467
28468 \group_end: 28468
28469 #1 \l__regex_build_tl 28469
28470 } 28470
28471 \cs_generate_variant:Nn \__regex_replacement:n { e } 28471
28472 \cs_new_protected:Npn \__regex_replacement_set:n #1 28472
28473 { 28473
28474 \cs_set:Npn \__regex_replacement_do_one_match:n ##1 28474
28475 { 28475
28476 \__regex_query_range:nn 28476
28477 { 28477
28478 \__kernel_intarray_item:Nn 28478
28479 \g__regex_submatch_prev_intarray {##1} 28479
28480 } 28480
28481 { 28481
28482 \__kernel_intarray_item:Nn 28482
28483 \g__regex_submatch_begin_intarray {##1} 28483
28484 } 28484
28485 #1 28485
28486 } 28486
28487 \exp_args:Nno \use:n 28487
28488 { \cs_gset:Npn \__regex_replacement_balance_one_match:n ##1 } 28488
28489 { 28489
28490 \g__regex_balance_tl 28490
28491 - \__regex_submatch_balance:n {##1} 28491
28492 } 28492
28493 } 28493
28494 \tl_new:N \g__regex_case_replacement_tl 28494
28495 \tl_new:N \g__regex_case_balance_tl 28495
28496 \cs_new_protected:Npn \__regex_case_replacement:n #1 28496
28497 { 28497
28498 \tl_gset:Nn \g__regex_case_balance_tl 28498
28499 { 28499
28500 \if_case:w 28500
28501 \__kernel_intarray_item:Nn 28501
28502 \g__regex_submatch_case_intarray {##1} 28502
28503 } 28503
```

```

28504 \tl_gset_eq:NN \g__regex_case_replacement_tl \g__regex_case_balance_tl 28504
28505 \tl_map_tokens:nn {#1} 28505
28506 { \__regex_replacement_apply:Nn \__regex_case_replacement_aux:n } 28506
28507 \tl_gset:No \g__regex_balance_tl 28507
28508 { \g__regex_case_balance_tl \fi: } 28508
28509 \exp_args:No \__regex_replacement_set:n 28509
28510 { \g__regex_case_replacement_tl \fi: } 28510
28511 } 28511
28512 \cs_generate_variant:Nn \__regex_case_replacement:n { e } 28512
28513 \cs_new_protected:Npn \__regex_case_replacement_aux:n #1 28513
28514 { 28514
28515 \tl_gput_right:Nn \g__regex_case_replacement_tl { \or: #1 } 28515
28516 \tl_gput_right:No \g__regex_case_balance_tl 28516
28517 { \exp_after:wN \or: \g__regex_balance_tl } 28517
28518 } 28518
28519 \cs_new_protected:Npn \__regex_replacement_put:n 28519
28520 { \tl_build_put_right:Nn \l__regex_build_tl } 28520
28521 \cs_new_protected:Npn \__regex_replacement_normal:n #1 28521
28522 { 28522
28523 \int_compare:nNnTF \l__regex_replacement_csnames_int > \c_zero_int 28523
28524 { \exp_args:No \__regex_replacement_put:n { \token_to_str:N #1 } } 28524
28525 { 28525
28526 \tl_if_empty:NTF \l__regex_replacement_category_tl 28526
28527 { \__regex_replacement_normal_aux:N #1 } 28527
28528 { % ( 28528
28529 \token_if_eq_charcode:NNTF #1 ) 28529
28530 { 28530
28531 \seq_pop:NN \l__regex_replacement_category_seq 28531
28532 \l__regex_replacement_category_tl 28532
28533 } 28533
28534 { 28534
28535 \use:c { __regex_replacement_c_ \l__regex_replacement_category_tl :w } 28535
28536 ? #1 28536
28537 } 28537
28538 } 28538
28539 } 28539
28540 } 28540
28541 \cs_new_protected:Npn \__regex_replacement_normal_aux:N #1 28541
28542 { 28542
28543 \token_if_eq_charcode:NNTF #1 \c_space_token 28543
28544 { \__regex_replacement_c_S:w } 28544
28545 { 28545
28546 \exp_after:wN \exp_after:wN 28546
28547 \if_case:w \tex_catcode:D `#1 \exp_stop_f: 28547
28548 \__regex_replacement_c_O:w 28548
28549 \or: \__regex_replacement_c_B:w 28549

```


28550	\\or: __regex_replacement_c_E:w	28550
28551	\\or: __regex_replacement_c_M:w	28551
28552	\\or: __regex_replacement_c_T:w	28552
28553	\\or: __regex_replacement_c_O:w	28553
28554	\\or: __regex_replacement_c_P:w	28554
28555	\\or: __regex_replacement_c_U:w	28555
28556	\\or: __regex_replacement_c_D:w	28556
28557	\\or: __regex_replacement_c_O:w	28557
28558	\\or: __regex_replacement_c_S:w	28558
28559	\\or: __regex_replacement_c_L:w	28559
28560	\\or: __regex_replacement_c_O:w	28560
28561	\\or: __regex_replacement_c_A:w	28561
28562	\\else: __regex_replacement_c_O:w	28562
28563	\\fi:	28563
28564	}	28564
28565	? #1	28565
28566	}	28566
28567	\\cs_new_protected:Npn __regex_replacement_escaped:N #1	28567
28568	{	28568
28569	\\cs_if_exist_use:cF { __regex_replacement_#1:w }	28569
28570	{	28570
28571	\\if_int_compare:w \\c_one_int < 1#1 \\exp_stop_f:	28571
28572	__regex_replacement_put_submatch:n {#1}	28572
28573	\\else:	28573
28574	__regex_replacement_normal:n {#1}	28574
28575	\\fi:	28575
28576	}	28576
28577	}	28577
28578	\\cs_new_protected:Npn __regex_replacement_put_submatch:n #1	28578
28579	{	28579
28580	\\if_int_compare:w #1 < \\l__regex_capturing_group_int	28580
28581	__regex_replacement_put_submatch_aux:n {#1}	28581
28582	\\else:	28582
28583	\\msg_expandable_error:nnff { regex } { submatch-too-big }	28583
28584	{#1} { \\int_eval:n { \\l__regex_capturing_group_int - \\c_one_int } }	28584
28585	\\fi:	28585
28586	}	28586
28587	\\cs_new_protected:Npn __regex_replacement_put_submatch_aux:n #1	28587
28588	{	28588
28589	\\tl_build_put_right:Nn \\l__regex_build_tl	28589
28590	{ __regex_query_submatch:n { __regex_int_eval:w #1 + ##1 \\scan_stop: } }	28590
28591	\\if_int_compare:w \\l__regex_replacement_csnames_int = \\c_zero_int	28591
28592	\\tl_gput_right:Nn \\g__regex_balance_tl	28592
28593	{ + __regex_submatch_balance:n { __regex_int_eval:w #1 + ##1 \\scan_stop: } }	28593
28594	\\fi:	28594
28595	}	28595


```
28596 \cs_new_protected:Npn \__regex_replacement_g:w #1#2 28596
28597 { 28597
28598 \token_if_eq_meaning:NNTF #1 \__regex_replacement_lbrace:N 28598
28599 { \l__regex_internal_a_int = \__regex_replacement_g_digits:NN } 28599
28600 { \__regex_replacement_error:NNN g #1 #2 } 28600
28601 } 28601
28602 \cs_new:Npn \__regex_replacement_g_digits:NN #1#2 28602
28603 { 28603
28604 \token_if_eq_meaning:NNTF #1 \__regex_replacement_normal:n 28604
28605 { 28605
28606 \if_int_compare:w \c_one_int < 1#2 \exp_stop_f: 28606
28607 #2 28607
28608 \exp_after:wN \use_i:nnn 28608
28609 \exp_after:wN \__regex_replacement_g_digits:NN 28609
28610 \else: 28610
28611 \exp_stop_f: 28611
28612 \exp_after:wN \__regex_replacement_error:NNN 28612
28613 \exp_after:wN g 28613
28614 \fi: 28614
28615 } 28615
28616 { 28616
28617 \exp_stop_f: 28617
28618 \if_meaning:w \__regex_replacement_rbrace:N #1 28618
28619 \exp_args:No \__regex_replacement_put_submatch:n 28619
28620 { \int_use:N \l__regex_internal_a_int } 28620
28621 \exp_after:wN \use_none:nn 28621
28622 \else: 28622
28623 \exp_after:wN \__regex_replacement_error:NNN 28623
28624 \exp_after:wN g 28624
28625 \fi: 28625
28626 } 28626
28627 #1 #2 28627
28628 } 28628
28629 \cs_new_protected:Npn \__regex_replacement_c:w #1#2 28629
28630 { 28630
28631 \token_if_eq_meaning:NNTF #1 \__regex_replacement_normal:n 28631
28632 { 28632
28633 \cs_if_exist:cTF { __regex_replacement_c_#2:w } 28633
28634 { \__regex_replacement_cat:NNN #2 } 28634
28635 { \__regex_replacement_error:NNN c #1#2 } 28635
28636 } 28636
28637 { 28637
28638 \token_if_eq_meaning:NNTF #1 \__regex_replacement_lbrace:N 28638
28639 { \__regex_replacement_cu_aux:Nw \__regex_replacement_exp_not:N } 28639
28640 { \__regex_replacement_error:NNN c #1#2 } 28640
28641 } 28641
```

```
28642 } 28642
28643 \cs_new_protected:Npn \__regex_replacement_cu_aux:Nw #1 28643
28644 { 28644
28645 \if_case:w \l__regex_replacement_csnames_int 28645
28646 \tl_build_put_right:Nn \l__regex_build_tl 28646
28647 { \exp_not:n { \exp_after:wN #1 \cs:w } } 28647
28648 \else: 28648
28649 \tl_build_put_right:Nn \l__regex_build_tl 28649
28650 { \exp_not:n { \exp_after:wN \tl_to_str:V \cs:w } } 28650
28651 \fi: 28651
28652 \int_incr:N \l__regex_replacement_csnames_int 28652
28653 } 28653
28654 \cs_new_protected:Npn \__regex_replacement_u:w #1#2 28654
28655 { 28655
28656 \token_if_eq_meaning:NNTF #1 \__regex_replacement_lbrace:N 28656
28657 { \__regex_replacement_cu_aux:Nw \__regex_replacement_exp_not:V } 28657
28658 { \__regex_replacement_error:NNN u #1#2 } 28658
28659 } 28659
28660 \cs_new_protected:Npn \__regex_replacement_rbrace:N #1 28660
28661 { 28661
28662 \if_int_compare:w \l__regex_replacement_csnames_int > \c_zero_int 28662
28663 \tl_build_put_right:Nn \l__regex_build_tl { \cs_end: } 28663
28664 \int_decr:N \l__regex_replacement_csnames_int 28664
28665 \else: 28665
28666 \__regex_replacement_normal:n {#1} 28666
28667 \fi: 28667
28668 } 28668
28669 \cs_new_protected:Npn \__regex_replacement_lbrace:N #1 28669
28670 { 28670
28671 \if_int_compare:w \l__regex_replacement_csnames_int > \c_zero_int 28671
28672 \msg_error:nnn { regex } { cu-lbrace } { u } 28672
28673 \else: 28673
28674 \__regex_replacement_normal:n {#1} 28674
28675 \fi: 28675
28676 } 28676
28677 \cs_new_protected:Npn \__regex_replacement_cat:NNN #1#2#3 28677
28678 { 28678
28679 \token_if_eq_meaning:NNTF \prg_do_nothing: #3 28679
28680 { \msg_error:nn { regex } { replacement-catcode-end } } 28680
28681 { 28681
28682 \int_compare:nNnTF \l__regex_replacement_csnames_int > \c_zero_int 28682
28683 { 28683
28684 \msg_error:nnnn 28684
28685 { regex } { replacement-catcode-in-cs } {#1} {#3} 28685
28686 #2 #3 28686
28687 } 28687
```

```

28688 {
28689 \__regex_two_if_eq:NNNTF #2 #3 \__regex_replacement_normal:n (
28690 {
28691 \seq_push:NV \l__regex_replacement_category_seq
28692 \l__regex_replacement_category_tl
28693 \tl_set:Nn \l__regex_replacement_category_tl {#1}
28694 }
28695 {
28696 \token_if_eq_meaning:NNT #2 \__regex_replacement_escaped:N
28697 {
28698 \__regex_char_if_alphanumeric:NTF #3
28699 {
28700 \msg_error:nnnn
28701 { regex } { replacement-catcode-escaped }
28702 {#1} {#3}
28703 }
28704 { }
28705 }
28706 \use:c { __regex_replacement_c_#1:w } #2 #3
28707 }
28708 }
28709 }
28710 }
28711 \group_begin:
28712 \cs_new_protected:Npn \__regex_replacement_char:nNN #1#2#3
28713 {
28714 \tex_lccode:D \c_zero_int = `#3 \scan_stop:
28715 \tex_lowercase:D { \__regex_replacement_put:n {#1} }
28716 }
28717 \char_set_catcode_active:N ^^@
28718 \cs_new_protected:Npn \__regex_replacement_c_A:w
28719 { \__regex_replacement_char:nNN { \exp_not:n { \exp_not:N ^^@ } } }
28720 \char_set_catcode_group_begin:N ^^@
28721 \cs_new_protected:Npn \__regex_replacement_c_B:w
28722 {
28723 \if_int_compare:w \l__regex_replacement_csnames_int = \c_zero_int
28724 \int_incr:N \l__regex_balance_int
28725 \fi:
28726 \__regex_replacement_char:nNN
28727 { \exp_not:n { \exp_after:wN ^^@ \if_false: } \fi: } }
28728 }
28729 \cs_new_protected:Npn \__regex_replacement_c_C:w #1#2
28730 {
28731 \tl_build_put_right:Nn \l__regex_build_tl
28732 { \exp_not:N \__regex_replacement_exp_not:N \exp_not:c {#2} }
28733 }

```

```

28734 \char_set_catcode_math_subscript:N \^^@ 28734
28735 \cs_new_protected:Npn \__regex_replacement_c_D:w 28735
28736 { \__regex_replacement_char:nNN { ^^@ } } 28736
28737 \char_set_catcode_group_end:N \^^@ 28737
28738 \cs_new_protected:Npn \__regex_replacement_c_E:w 28738
28739 { 28739
28740 \if_int_compare:w \l__regex_replacement_csnames_int = \c_zero_int 28740
28741 \int_decr:N \l__regex_balance_int 28741
28742 \fi: 28742
28743 \__regex_replacement_char:nNN 28743
28744 { \exp_not:n { \if_false: { \fi: ^^@ } } } 28744
28745 } 28745
28746 \char_set_catcode_letter:N \^^@ 28746
28747 \cs_new_protected:Npn \__regex_replacement_c_L:w 28747
28748 { \__regex_replacement_char:nNN { ^^@ } } 28748
28749 \char_set_catcode_math_toggle:N \^^@ 28749
28750 \cs_new_protected:Npn \__regex_replacement_c_M:w 28750
28751 { \__regex_replacement_char:nNN { ^^@ } } 28751
28752 \char_set_catcode_other:N \^^@ 28752
28753 \cs_new_protected:Npn \__regex_replacement_c_O:w 28753
28754 { \__regex_replacement_char:nNN { ^^@ } } 28754
28755 \char_set_catcode_parameter:N \^^@ 28755
28756 \cs_new_protected:Npn \__regex_replacement_c_P:w 28756
28757 { 28757
28758 \__regex_replacement_char:nNN 28758
28759 { \exp_not:n { \exp_not:n { ^^@^^@^^@^^@ } } } } 28759
28760 } 28760
28761 \cs_new_protected:Npn \__regex_replacement_c_S:w #1#2 28761
28762 { 28762
28763 \if_int_compare:w `#2 = \c_zero_int 28763
28764 \msg_error:nn { regex } { replacement-null-space } 28764
28765 \fi: 28765
28766 \tex_lccode:D \_ = `#2 \scan_stop: 28766
28767 \tex_lowercase:D { \__regex_replacement_put:n {~} } 28767
28768 } 28768
28769 \char_set_catcode_alignment:N \^^@ 28769
28770 \cs_new_protected:Npn \__regex_replacement_c_T:w 28770
28771 { \__regex_replacement_char:nNN { ^^@ } } 28771
28772 \char_set_catcode_math_superscript:N \^^@ 28772
28773 \cs_new_protected:Npn \__regex_replacement_c_U:w 28773
28774 { \__regex_replacement_char:nNN { ^^@ } } 28774
28775 \group_end: 28775
28776 \cs_new_protected:Npn \__regex_replacement_error:NNN #1#2#3 28776
28777 { 28777
28778 \msg_error:nne { regex } { replacement-#1 } {#3} 28778
28779 #2 #3 28779

```

28780	}	28780
28781	\cs_new_protected:Npn \regex_new:N #1	28781
28782	{ \cs_new_eq:NN #1 \c__regex_no_match_regex }	28782
28783	\regex_new:N \l_tmpa_regex	28783
28784	\regex_new:N \l_tmpb_regex	28784
28785	\regex_new:N \g_tmpa_regex	28785
28786	\regex_new:N \g_tmpb_regex	28786
28787	\cs_new_protected:Npn \regex_set:Nn #1#2	28787
28788	{	28788
28789	__regex_compile:n {#2}	28789
28790	\tl_set_eq:NN #1 \l__regex_internal_regex	28790
28791	}	28791
28792	\cs_new_protected:Npn \regex_gset:Nn #1#2	28792
28793	{	28793
28794	__regex_compile:n {#2}	28794
28795	\tl_gset_eq:NN #1 \l__regex_internal_regex	28795
28796	}	28796
28797	\cs_new_protected:Npn \regex_const:Nn #1#2	28797
28798	{	28798
28799	__regex_compile:n {#2}	28799
28800	\tl_const:Ne #1 { \exp_not:o \l__regex_internal_regex }	28800
28801	}	28801
28802	\cs_new_protected:Npn \regex_show:n { __regex_show:Nn \msg_show:nneeee }	28802
28803	\cs_new_protected:Npn \regex_log:n { __regex_show:Nn \msg_log:nneeee }	28803
28804	\cs_new_protected:Npn __regex_show:Nn #1#2	28804
28805	{	28805
28806	__regex_compile:n {#2}	28806
28807	__regex_show:N \l__regex_internal_regex	28807
28808	#1 { regex } { show }	28808
28809	{ \tl_to_str:n {#2} } { }	28809
28810	{ \l__regex_internal_a_tl } { }	28810
28811	}	28811
28812	\cs_new_protected:Npn \regex_show:N { __regex_show:NN \msg_show:nneeee }	28812
28813	\cs_new_protected:Npn \regex_log:N { __regex_show:NN \msg_log:nneeee }	28813
28814	\cs_new_protected:Npn __regex_show:NN #1#2	28814
28815	{	28815
28816	__kernel_chk_tl_type:NnnT #2 { regex }	28816
28817	{ \exp_args:No __regex_clean_regex:n {#2} }	28817
28818	{	28818
28819	__regex_show:N #2	28819
28820	#1 { regex } { show }	28820
28821	{ } { \token_to_str:N #2 }	28821
28822	{ \l__regex_internal_a_tl } { }	28822
28823	}	28823
28824	}	28824
28825	\prg_new_protected_conditional:Npnn \regex_match:nn #1#2 { T , F , TF }	28825

```
28826 { 28826
28827 \__regex_if_match:nn { \__regex_build:n {#1} } {#2} 28827
28828 \__regex_return: 28828
28829 } 28829
28830 \prg_generate_conditional_variant:Nnn \regex_match:nn { nV } { T , F , TF } 28830
28831 \prg_new_protected_conditional:Npnn \regex_match:Nn #1#2 { T , F , TF } 28831
28832 { 28832
28833 \__regex_if_match:nn { \__regex_build:N #1 } {#2} 28833
28834 \__regex_return: 28834
28835 } 28835
28836 \prg_generate_conditional_variant:Nnn \regex_match:Nn { NV } { T , F , TF } 28836
28837 \cs_new_protected:Npn \regex_count:nnN #1 28837
28838 { \__regex_count:nnN { \__regex_build:n {#1} } } 28838
28839 \cs_new_protected:Npn \regex_count:NnN #1 28839
28840 { \__regex_count:nnN { \__regex_build:N #1 } } 28840
28841 \cs_generate_variant:Nn \regex_count:nnN { nV } 28841
28842 \cs_generate_variant:Nn \regex_count:NnN { NV } 28842
28843 \cs_new_protected:Npn \regex_match_case:nnTF #1#2#3 28843
28844 { 28844
28845 \__regex_match_case:nnTF {#1} {#2} 28845
28846 { 28846
28847 \tl_item:nn {#1} { 2 * \g__regex_case_int } 28847
28848 #3 28848
28849 } 28849
28850 } 28850
28851 \cs_new_protected:Npn \regex_match_case:nn #1#2 28851
28852 { \regex_match_case:nnTF {#1} {#2} { } { } } 28852
28853 \cs_new_protected:Npn \regex_match_case:nnT #1#2#3 28853
28854 { \regex_match_case:nnTF {#1} {#2} {#3} { } } 28854
28855 \cs_new_protected:Npn \regex_match_case:nnF #1#2 28855
28856 { \regex_match_case:nnTF {#1} {#2} { } } 28856
28857 \cs_set_protected:Npn \__regex_tmp:w #1#2#3 28857
28858 { 28858
28859 \cs_new_protected:Npn #2 ##1 { #1 { \__regex_build:n {##1} } } 28859
28860 \cs_new_protected:Npn #3 ##1 { #1 { \__regex_build:N ##1 } } 28860
28861 \prg_new_protected_conditional:Npnn #2 ##1##2##3 { T , F , TF } 28861
28862 { #1 { \__regex_build:n {##1} } {##2} ##3 \__regex_return: } 28862
28863 \prg_new_protected_conditional:Npnn #3 ##1##2##3 { T , F , TF } 28863
28864 { #1 { \__regex_build:N ##1 } {##2} ##3 \__regex_return: } 28864
28865 \cs_generate_variant:Nn #2 { nV } 28865
28866 \prg_generate_conditional_variant:Nnn #2 { nV } { T , F , TF } 28866
28867 \cs_generate_variant:Nn #3 { NV } 28867
28868 \prg_generate_conditional_variant:Nnn #3 { NV } { T , F , TF } 28868
28869 } 28869
28870 \__regex_tmp:w \__regex_extract_once:nnN 28870
28871 \regex_extract_once:nnN \regex_extract_once:NnN 28871
```



```
28872 \__regex_tmp:w \__regex_extract_all:nnN
28873 \regex_extract_all:nnN \regex_extract_all:NnN
28874 \__regex_tmp:w \__regex_replace_once:nnN
28875 \regex_replace_once:nnN \regex_replace_once:NnN
28876 \__regex_tmp:w \__regex_replace_all:nnN
28877 \regex_replace_all:nnN \regex_replace_all:NnN
28878 \__regex_tmp:w \__regex_split:nnN \regex_split:nnN \regex_split:NnN
28879 \cs_new_protected:Npn \regex_replace_case_once:nNTF #1#2
28880 {
28881 \int_if_odd:nTF { \tl_count:n {#1} }
28882 {
28883 \msg_error:nneeee { regex } { case-odd }
28884 { \token_to_str:N \regex_replace_case_once:nN(TF) } { code }
28885 { \tl_count:n {#1} } { \tl_to_str:n {#1} }
28886 \use_ii:nn
28887 }
28888 {
28889 \__regex_replace_once_aux:nnN
28890 { \__regex_case_build:e { \__regex_tl_odd_items:n {#1} } }
28891 { \__regex_replacement:e { \tl_item:nn {#1} { 2 * \g__regex_case_int } } }
28892 #2
28893 \bool_if:NTF \g__regex_success_bool
28894 }
28895 }
28896 \cs_new_protected:Npn \regex_replace_case_once:nN #1#2
28897 { \regex_replace_case_once:nNTF {#1} {#2} { } { } }
28898 \cs_new_protected:Npn \regex_replace_case_once:nNT #1#2#3
28899 { \regex_replace_case_once:nNTF {#1} {#2} {#3} { } }
28900 \cs_new_protected:Npn \regex_replace_case_once:nNF #1#2
28901 { \regex_replace_case_once:nNTF {#1} {#2} { } }
28902 \cs_new_protected:Npn \regex_replace_case_all:nNTF #1#2
28903 {
28904 \int_if_odd:nTF { \tl_count:n {#1} }
28905 {
28906 \msg_error:nneeee { regex } { case-odd }
28907 { \token_to_str:N \regex_replace_case_all:nN(TF) } { code }
28908 { \tl_count:n {#1} } { \tl_to_str:n {#1} }
28909 \use_ii:nn
28910 }
28911 {
28912 \__regex_replace_all_aux:nnN
28913 { \__regex_case_build:e { \__regex_tl_odd_items:n {#1} } }
28914 { \__regex_case_replacement:e { \__regex_tl_even_items:n {#1} } }
28915 #2
28916 \bool_if:NTF \g__regex_success_bool
28917 }
```



```
28918 } 28918
28919 \cs_new_protected:Npn \regex_replace_case_all:nN #1#2 28919
28920 { \regex_replace_case_all:nNTF {#1} {#2} { } { } } 28920
28921 \cs_new_protected:Npn \regex_replace_case_all:nNT #1#2#3 28921
28922 { \regex_replace_case_all:nNTF {#1} {#2} {#3} { } } 28922
28923 \cs_new_protected:Npn \regex_replace_case_all:nNF #1#2 28923
28924 { \regex_replace_case_all:nNTF {#1} {#2} { } { } } 28924
28925 \int_new:N \l__regex_match_count_int 28925
28926 \flag_new:N \l__regex_begin_flag 28926
28927 \flag_new:N \l__regex_end_flag 28927
28928 \int_new:N \l__regex_min_submatch_int 28928
28929 \int_new:N \l__regex_submatch_int 28929
28930 \int_new:N \l__regex_zeroth_submatch_int 28930
28931 \intarray_new:Nn \g__regex_submatch_prev_intarray { 65536 } 28931
28932 \intarray_new:Nn \g__regex_submatch_begin_intarray { 65536 } 28932
28933 \intarray_new:Nn \g__regex_submatch_end_intarray { 65536 } 28933
28934 \intarray_new:Nn \g__regex_submatch_case_intarray { 65536 } 28934
28935 \intarray_new:Nn \g__regex_balance_intarray { 65536 } 28935
28936 \int_new:N \l__regex_added_begin_int 28936
28937 \int_new:N \l__regex_added_end_int 28937
28938 \cs_new_protected:Npn \__regex_return: 28938
28939 { 28939
28940 \if_meaning:w \c_true_bool \g__regex_success_bool 28940
28941 \prg_return_true: 28941
28942 \else: 28942
28943 \prg_return_false: 28943
28944 \fi: 28944
28945 } 28945
28946 \cs_new_protected:Npn \__regex_query_set:n #1 28946
28947 { 28947
28948 \int_zero:N \l__regex_balance_int 28948
28949 \int_zero:N \l__regex_curr_pos_int 28949
28950 \__regex_query_set_aux:nN { } F 28950
28951 \tl_analysis_map_inline:nn {#1} 28951
28952 { \__regex_query_set_aux:nN {##1} ##3 } 28952
28953 \__regex_query_set_aux:nN { } F 28953
28954 \int_set_eq:NN \l__regex_max_pos_int \l__regex_curr_pos_int 28954
28955 } 28955
28956 \cs_new_protected:Npn \__regex_query_set_aux:nN #1#2 28956
28957 { 28957
28958 \int_incr:N \l__regex_curr_pos_int 28958
28959 \__regex_toks_set:Nn \l__regex_curr_pos_int {#1} 28959
28960 \__kernel_intarray_gset:Nnn \g__regex_balance_intarray 28960
28961 \l__regex_curr_pos_int \l__regex_balance_int 28961
28962 \if_case:w "#2 \exp_stop_f: 28962
28963 \or: \int_incr:N \l__regex_balance_int 28963
```

```
28964 \or: \int_decr:N \l__regex_balance_int 28964
28965 \fi: 28965
28966 } 28966
28967 \cs_new_protected:Npn \__regex_if_match:nn #1#2 28967
28968 { 28968
28969 \group_begin: 28969
28970 \__regex_disable_submatches: 28970
28971 \__regex_single_match: 28971
28972 #1 28972
28973 \__regex_match:n {#2} 28973
28974 \group_end: 28974
28975 } 28975
28976 \cs_new_protected:Npn \__regex_match_case:nnTF #1#2 28976
28977 { 28977
28978 \int_if_odd:nTF { \tl_count:n {#1} } 28978
28979 { 28979
28980 \msg_error:nneeee { regex } { case-odd } 28980
28981 { \token_to_str:N \regex_match_case:nn(TF) } { code } 28981
28982 { \tl_count:n {#1} } { \tl_to_str:n {#1} } 28982
28983 \use_ii:nn 28983
28984 } 28984
28985 { 28985
28986 \__regex_if_match:nn 28986
28987 { \__regex_case_build:e { \__regex_tl_odd_items:n {#1} } } 28987
28988 {#2} 28988
28989 \bool_if:NTF \g__regex_success_bool 28989
28990 } 28990
28991 } 28991
28992 \cs_new:Npn \__regex_match_case_aux:nn #1#2 { \exp_not:n { {#1} } } 28992
28993 \cs_new_protected:Npn \__regex_count:nnN #1#2#3 28993
28994 { 28994
28995 \group_begin: 28995
28996 \__regex_disable_submatches: 28996
28997 \int_zero:N \l__regex_match_count_int 28997
28998 \__regex_multi_match:n { \int_incr:N \l__regex_match_count_int } 28998
28999 #1 28999
29000 \__regex_match:n {#2} 29000
29001 \exp_args:NNNo 29001
29002 \group_end: 29002
29003 \int_set:Nn #3 { \int_use:N \l__regex_match_count_int } 29003
29004 } 29004
29005 \cs_new_protected:Npn \__regex_extract_once:nnN #1#2#3 29005
29006 { 29006
29007 \group_begin: 29007
29008 \__regex_single_match: 29008
29009 #1 29009
```

```
29010     \__regex_match:n {#2}
29011     \__regex_extract:
29012     \__regex_query_set:n {#2}
29013     \__regex_group_end_extract_seq:N #3
29014 }
29015 \cs_new_protected:Npn \__regex_extract_all:nnN #1#2#3
29016 {
29017     \group_begin:
29018         \__regex_multi_match:n { \__regex_extract: }
29019         #1
29020         \__regex_match:n {#2}
29021         \__regex_query_set:n {#2}
29022         \__regex_group_end_extract_seq:N #3
29023     }
29024 \cs_new_protected:Npn \__regex_split:nnN #1#2#3
29025 {
29026     \group_begin:
29027         \__regex_multi_match:n
29028         {
29029             \if_int_compare:w
29030                 \l__regex_start_pos_int < \l__regex_success_pos_int
29031                 \__regex_extract:
29032                 \__kernel_intarray_gset:Nnn \g__regex_submatch_prev_intarray
29033                 \l__regex_zeroth_submatch_int \c_zero_int
29034                 \__kernel_intarray_gset:Nnn \g__regex_submatch_end_intarray
29035                 \l__regex_zeroth_submatch_int
29036                 {
29037                     \__kernel_intarray_item:Nn \g__regex_submatch_begin_intarray
29038                     \l__regex_zeroth_submatch_int
29039                 }
29040                 \__kernel_intarray_gset:Nnn \g__regex_submatch_begin_intarray
29041                 \l__regex_zeroth_submatch_int
29042                 \l__regex_start_pos_int
29043             \fi:
29044         }
29045         #1
29046         \__regex_match:n {#2}
29047         \__regex_query_set:n {#2}
29048         \__kernel_intarray_gset:Nnn \g__regex_submatch_prev_intarray
29049         \l__regex_submatch_int \c_zero_int
29050         \__kernel_intarray_gset:Nnn \g__regex_submatch_end_intarray
29051         \l__regex_submatch_int
29052         \l__regex_max_pos_int
29053         \__kernel_intarray_gset:Nnn \g__regex_submatch_begin_intarray
29054         \l__regex_submatch_int
29055         \l__regex_start_pos_int
```

29056	\int_incr:N \l__regex_submatch_int	29056
29057	\if_meaning:w \c_true_bool \l__regex_empty_success_bool	29057
29058	\if_int_compare:w \l__regex_start_pos_int = \l__regex_max_pos_int	29058
29059	\int_decr:N \l__regex_submatch_int	29059
29060	\fi:	29060
29061	\fi:	29061
29062	__regex_group_end_extract_seq:N #3	29062
29063	}	29063
29064	\cs_new_protected:Npn __regex_group_end_extract_seq:N #1	29064
29065	{	29065
29066	\flag_clear:N \l__regex_begin_flag	29066
29067	\flag_clear:N \l__regex_end_flag	29067
29068	\cs_set_eq:NN __regex_tmp:w \scan_stop:	29068
29069	__kernel_tl_gset:Nx \g__regex_internal_tl	29069
29070	{	29070
29071	\int_step_function:nnN \l__regex_min_submatch_int	29071
29072	{ \l__regex_submatch_int - \c_one_int } __regex_extract_seq_aux:n	29072
29073	__regex_tmp:w	29073
29074	}	29074
29075	\int_set:Nn \l__regex_added_begin_int	29075
29076	{ \flag_height:N \l__regex_begin_flag }	29076
29077	\int_set:Nn \l__regex_added_end_int	29077
29078	{ \flag_height:N \l__regex_end_flag }	29078
29079	\tex_afterassignment:D __regex_extract_check:w	29079
29080	__kernel_tl_gset:Nx \g__regex_internal_tl	29080
29081	{ \g__regex_internal_tl \if_false: { \fi: } }	29081
29082	\int_compare:nNnT	29082
29083	{ \l__regex_added_begin_int + \l__regex_added_end_int } > \c_zero_int	29083
29084	{	29084
29085	\msg_error:nneee { regex } { result-unbalanced }	29085
29086	{ splitting~or~extracting~submatches }	29086
29087	{ \int_use:N \l__regex_added_begin_int }	29087
29088	{ \int_use:N \l__regex_added_end_int }	29088
29089	}	29089
29090	\group_end:	29090
29091	__regex_extract_seq:N #1	29091
29092	}	29092
29093	\cs_gset_protected:Npn __regex_extract_seq:N #1	29093
29094	{	29094
29095	\seq_clear:N #1	29095
29096	\cs_set_eq:NN __regex_tmp:w __regex_extract_seq_loop:Nw	29096
29097	\exp_after:wN __regex_extract_seq:NNn	29097
29098	\exp_after:wN #1	29098
29099	\g__regex_internal_tl \use_none:nnn	29099
29100	}	29100
29101	\cs_new_protected:Npn __regex_extract_seq:NNn #1#2#3	29101

```

29102 { #3 #2 #1 \prg_do_nothing: } 29102
29103 \cs_new_protected:Npn \__regex_extract_seq_loop:Nw #1#2 \__regex_tmp:w #3 29103
29104 { 29104
29105     \seq_put_right:No #1 {#2} 29105
29106     #3 \__regex_extract_seq_loop:Nw #1 \prg_do_nothing: 29106
29107 } 29107
29108 \cs_new:Npn \__regex_extract_seq_aux:n #1 29108
29109 { 29109
29110     \__regex_tmp:w { } 29110
29111     \exp_after:wN \__regex_extract_seq_aux:ww 29111
29112     \int_value:w \__regex_submatch_balance:n {#1} \__regex_sep: #1 \__regex_sep: 29112
29113 } 29113
29114 \cs_new:Npn \__regex_extract_seq_aux:ww #1 \__regex_sep: #2 \__regex_sep: 29114
29115 { 29115
29116     \if_int_compare:w #1 < \c_zero_int 29116
29117     \prg_replicate:nn {-#1} 29117
29118     { 29118
29119         \flag_raise:N \l__regex_begin_flag 29119
29120         \exp_not:n { { \if_false: } \fi: } 29120
29121     } 29121
29122     \fi: 29122
29123     \__regex_query_submatch:n {#2} 29123
29124     \if_int_compare:w #1 > \c_zero_int 29124
29125     \prg_replicate:nn {#1} 29125
29126     { 29126
29127         \flag_raise:N \l__regex_end_flag 29127
29128         \exp_not:n { \if_false: { \fi: } } 29128
29129     } 29129
29130     \fi: 29130
29131 } 29131
29132 \cs_new_protected:Npn \__regex_extract_check:w 29132
29133 { 29133
29134     \exp_after:wN \__regex_extract_check:n 29134
29135     \exp_after:wN { \if_false: } \fi: 29135
29136 } 29136
29137 \cs_new_protected:Npn \__regex_extract_check:n #1 29137
29138 { 29138
29139     \tl_if_empty:nF {#1} 29139
29140     { 29140
29141         \int_incr:N \l__regex_added_begin_int 29141
29142         \int_incr:N \l__regex_added_end_int 29142
29143         \tex_afterassignment:D \__regex_extract_check:w 29143
29144         \__kernel_tl_gset:Nx \g__regex_internal_tl 29144
29145         { 29145
29146             \exp_after:wN \__regex_extract_check_loop:w 29146
29147             \g__regex_internal_tl 29147

```

```
29148         \__regex_tmp:w \__regex_extract_check_end:w 29148
29149         #1 29149
29150     } 29150
29151 } 29151
29152 } 29152
29153 \cs_new:Npn \__regex_extract_check_loop:w #1 \__regex_tmp:w #2 29153
29154 { 29154
29155     #2 29155
29156     \exp_not:o {#1} 29156
29157     \__regex_tmp:w { } 29157
29158     \__regex_extract_check_loop:w \prg_do_nothing: 29158
29159 } 29159
29160 \cs_new:Npn \__regex_extract_check_end:w 29160
29161     \exp_not:o #1#2 \__regex_extract_check_loop:w #3 \__regex_tmp:w 29161
29162 { 29162
29163     { \exp_not:o {#1} } 29163
29164     #3 29164
29165     \if_false: { \fi: } 29165
29166     \__regex_tmp:w 29166
29167 } 29167
29168 \cs_new_protected:Npn \__regex_extract: 29168
29169 { 29169
29170     \if_meaning:w \c_true_bool \g__regex_success_bool 29170
29171     \int_set_eq:NN \l__regex_zeroth_submatch_int \l__regex_submatch_int 29171
29172     \prg_replicate:nn \l__regex_capturing_group_int 29172
29173     { 29173
29174         \__kernel_intarray_gset:Nnn \g__regex_submatch_prev_intarray 29174
29175         \l__regex_submatch_int \c_zero_int 29175
29176         \__kernel_intarray_gset:Nnn \g__regex_submatch_case_intarray 29176
29177         \l__regex_submatch_int \c_zero_int 29177
29178         \int_incr:N \l__regex_submatch_int 29178
29179     } 29179
29180     \__kernel_intarray_gset:Nnn \g__regex_submatch_prev_intarray 29180
29181     \l__regex_zeroth_submatch_int \l__regex_start_pos_int 29181
29182     \__kernel_intarray_gset:Nnn \g__regex_submatch_case_intarray 29182
29183     \l__regex_zeroth_submatch_int \g__regex_case_int 29183
29184     \int_zero:N \l__regex_internal_a_int 29184
29185     \exp_after:wN \__regex_extract_aux:w \l__regex_success_submatches_tl 29185
29186     \prg_break_point: \__regex_use_none_delimit_by_q_recursion_stop:w , 29186
29187     \q__regex_recursion_stop 29187
29188     \fi: 29188
29189 } 29189
29190 \cs_new_protected:Npn \__regex_extract_aux:w #1 , 29190
29191 { 29191
29192     \prg_break: #1 \prg_break_point: 29192
29193     \if_int_compare:w \l__regex_internal_a_int < \l__regex_capturing_group_int 29193
```



```
29194 \__kernel_intarray_gset:Nnn \g__regex_submatch_begin_intarray 29194
29195 { \__regex_int_eval:w \l__regex_zeroth_submatch_int + \l__regex_internal_a_int } 29195 ✓
29196 {#1} 29196
29197 \else: 29197
29198 \__kernel_intarray_gset:Nnn \g__regex_submatch_end_intarray 29198
29199 { 29199
29200 \__regex_int_eval:w 29200
29201 \l__regex_zeroth_submatch_int + \l__regex_internal_a_int 29201
29202 - \l__regex_capturing_group_int 29202
29203 } 29203
29204 {#1} 29204
29205 \fi: 29205
29206 \int_incr:N \l__regex_internal_a_int 29206
29207 \__regex_extract_aux:w 29207
29208 } 29208
29209 \cs_new_protected:Npn \__regex_replace_once:nnN #1#2 29209
29210 { \__regex_replace_once_aux:nnN {#1} { \__regex_replacement:n {#2} } } 29210
29211 \cs_new_protected:Npn \__regex_replace_once_aux:nnN #1#2#3 29211
29212 { 29212
29213 \group_begin: 29213
29214 \__regex_single_match: 29214
29215 #1 29215
29216 \exp_args:No \__regex_match:n {#3} 29216
29217 \bool_if:NTF \g__regex_success_bool 29217
29218 { 29218
29219 \__regex_extract: 29219
29220 \exp_args:No \__regex_query_set:n {#3} 29220
29221 #2 29221
29222 \int_set:Nn \l__regex_balance_int 29222
29223 { \__regex_replacement_balance_one_match:n \l__regex_zeroth_submatch_int } 29223
29224 \__kernel_tl_set:Nx \l__regex_internal_a_tl 29224
29225 { 29225
29226 \__regex_replacement_do_one_match:n \l__regex_zeroth_submatch_int 29226
29227 \__regex_query_range:nn 29227
29228 { 29228
29229 \__kernel_intarray_item:Nn \g__regex_submatch_end_intarray 29229
29230 \l__regex_zeroth_submatch_int 29230
29231 } 29231
29232 \l__regex_max_pos_int 29232
29233 } 29233
29234 \__regex_group_end_replace:N #3 29234
29235 } 29235
29236 { \group_end: } 29236
29237 } 29237
29238 \cs_new_protected:Npn \__regex_replace_all:nnN #1#2 29238
29239 { \__regex_replace_all_aux:nnN {#1} { \__regex_replacement:n {#2} } } 29239
```



```
29239 \cs_new_protected:Npn \__regex_replace_all_aux:nnN #1#2#3
29240 {
29241   \group_begin:
29242     \__regex_multi_match:n { \__regex_extract: }
29243     #1
29244     \exp_args:No \__regex_match:n {#3}
29245     \exp_args:No \__regex_query_set:n {#3}
29246     #2
29247     \int_set:Nn \l__regex_balance_int
29248     {
29249       \c_zero_int
29250       \int_step_function:nnnN
29251         \l__regex_min_submatch_int
29252         \l__regex_capturing_group_int
29253         { \l__regex_submatch_int - \c_one_int }
29254         \__regex_replacement_balance_one_match:n
29255     }
29256     \__kernel_tl_set:Nx \l__regex_internal_a_tl
29257     {
29258       \int_step_function:nnnN
29259         \l__regex_min_submatch_int
29260         \l__regex_capturing_group_int
29261         { \l__regex_submatch_int - \c_one_int }
29262         \__regex_replacement_do_one_match:n
29263         \__regex_query_range:nn
29264         \l__regex_start_pos_int \l__regex_max_pos_int
29265     }
29266     \__regex_group_end_replace:N #3
29267   }
29268   \cs_new_protected:Npn \__regex_group_end_replace:N #1
29269   {
29270     \int_set:Nn \l__regex_added_begin_int
29271     { \int_max:nn { - \l__regex_balance_int } \c_zero_int }
29272     \int_set:Nn \l__regex_added_end_int
29273     { \int_max:nn \l__regex_balance_int \c_zero_int }
29274     \__regex_group_end_replace_try:
29275     \int_compare:nNnT { \l__regex_added_begin_int + \l__regex_added_end_int }
29276       > \c_zero_int
29277     {
29278       \msg_error:nneee { regex } { result-unbalanced }
29279       { replacing } { \int_use:N \l__regex_added_begin_int }
29280       { \int_use:N \l__regex_added_end_int }
29281     }
29282   \group_end:
29283   \tl_set_eq:NN #1 \g__regex_internal_tl
29284 }
```

```
29285 \cs_new_protected:Npn \__regex_group_end_replace_try: 29285
29286 { 29286
29287 \tex_afterassignment:D \__regex_group_end_replace_check:w 29287
29288 \__kernel_tl_gset:Nx \g__regex_internal_tl 29288
29289 { 29289
29290 \prg_replicate:nn \l__regex_added_begin_int { { \if_false: } \fi: } 29290
29291 \l__regex_internal_a_tl 29291
29292 \prg_replicate:nn \l__regex_added_end_int { \if_false: { \fi: } } 29292
29293 \if_false: { \fi: } 29293
29294 } 29294
29295 } 29295
29296 \cs_new_protected:Npn \__regex_group_end_replace_check:w 29296
29297 { 29297
29298 \exp_after:wN \__regex_group_end_replace_check:n 29298
29299 \exp_after:wN { \if_false: } \fi: 29299
29300 } 29300
29301 \cs_new_protected:Npn \__regex_group_end_replace_check:n #1 29301
29302 { 29302
29303 \tl_if_empty:nF {#1} 29303
29304 { 29304
29305 \int_incr:N \l__regex_added_begin_int 29305
29306 \int_incr:N \l__regex_added_end_int 29306
29307 \__regex_group_end_replace_try: 29307
29308 } 29308
29309 } 29309
29310 \tl_new:N \l__regex_peek_true_tl 29310
29311 \tl_new:N \l__regex_peek_false_tl 29311
29312 \tl_new:N \l__regex_replacement_tl 29312
29313 \tl_new:N \l__regex_input_tl 29313
29314 \cs_new_eq:NN \__regex_input_item:n ? 29314
29315 \cs_new_protected:Npn \peek_regex:nTF #1 29315
29316 { 29316
29317 \__regex_peek:nnTF 29317
29318 { \__regex_build_aux:Nn \c_false_bool {#1} } 29318
29319 { \__regex_peek_end: } 29319
29320 } 29320
29321 \cs_new_protected:Npn \peek_regex:nT #1#2 29321
29322 { \peek_regex:nTF {#1} {#2} { } } 29322
29323 \cs_new_protected:Npn \peek_regex:nF #1 { \peek_regex:nTF {#1} { } } 29323
29324 \cs_new_protected:Npn \peek_regex:NTF #1 29324
29325 { 29325
29326 \__regex_peek:nnTF 29326
29327 { \__regex_build_aux:NN \c_false_bool #1 } 29327
29328 { \__regex_peek_end: } 29328
29329 } 29329
29330 \cs_new_protected:Npn \peek_regex:NT #1#2 29330
```

29331	{ \peek_regex:NTF #1 {#2} { } }	29331
29332	\cs_new_protected:Npn \peek_regex:NF #1 { \peek_regex:NTF {#1} { } }	29332
29333	\cs_new_protected:Npn \peek_regex_remove_once:nTF #1	29333
29334	{	29334
29335	__regex_peek:nnTF	29335
29336	{ __regex_build_aux:Nn \c_false_bool {#1} }	29336
29337	{ __regex_peek_remove_end:n {##1} }	29337
29338	}	29338
29339	\cs_new_protected:Npn \peek_regex_remove_once:nT #1#2	29339
29340	{ \peek_regex_remove_once:nTF {#1} {#2} { } }	29340
29341	\cs_new_protected:Npn \peek_regex_remove_once:nF #1	29341
29342	{ \peek_regex_remove_once:nTF {#1} { } }	29342
29343	\cs_new_protected:Npn \peek_regex_remove_once:NTF #1	29343
29344	{	29344
29345	__regex_peek:nnTF	29345
29346	{ __regex_build_aux:NN \c_false_bool #1 }	29346
29347	{ __regex_peek_remove_end:n {##1} }	29347
29348	}	29348
29349	\cs_new_protected:Npn \peek_regex_remove_once:NT #1#2	29349
29350	{ \peek_regex_remove_once:NTF #1 {#2} { } }	29350
29351	\cs_new_protected:Npn \peek_regex_remove_once:NF #1	29351
29352	{ \peek_regex_remove_once:NTF #1 { } }	29352
29353	\cs_new_protected:Npn __regex_peek:nnTF #1	29353
29354	{	29354
29355	__regex_peek_aux:nnTF	29355
29356	{	29356
29357	__regex_disable_submatches:	29357
29358	#1	29358
29359	}	29359
29360	}	29360
29361	\cs_new_protected:Npn __regex_peek_aux:nnTF #1#2#3#4	29361
29362	{	29362
29363	\group_begin:	29363
29364	\tl_set:Nn \l__regex_peek_true_tl { \group_end: #3 }	29364
29365	\tl_set:Nn \l__regex_peek_false_tl { \group_end: #4 }	29365
29366	__regex_single_match:	29366
29367	#1	29367
29368	__regex_match_init:	29368
29369	\tl_build_begin:N \l__regex_input_tl	29369
29370	__regex_match_once_init:	29370
29371	\peek_analysis_map_inline:n	29371
29372	{	29372
29373	\tl_build_put_right:Nn \l__regex_input_tl	29373
29374	{ __regex_input_item:n {##1} }	29374
29375	__regex_match_one_token:nnN {##1} {##2} ##3	29375
29376	\use_none:nnn	29376

```
29377         \prg_break_point:Nn \__regex_maplike_break: 29377
29378         { \peek_analysis_map_break:n {#2} } 29378
29379     } 29379
29380 } 29380
29381 \cs_new_protected:Npn \__regex_peek_end: 29381
29382 { 29382
29383     \bool_if:NTF \g__regex_success_bool 29383
29384     { \__regex_peek_reinsert:N \l__regex_peek_true_tl } 29384
29385     { \__regex_peek_reinsert:N \l__regex_peek_false_tl } 29385
29386 } 29386
29387 \cs_new_protected:Npn \__regex_peek_remove_end:n #1 29387
29388 { 29388
29389     \bool_if:NTF \g__regex_success_bool 29389
29390     { \exp_args:NNo \use:nn \l__regex_peek_true_tl {#1} } 29390
29391     { \__regex_peek_reinsert:N \l__regex_peek_false_tl } 29391
29392 } 29392
29393 \cs_new_protected:Npn \__regex_peek_reinsert:N #1 29393
29394 { 29394
29395     \tl_build_end:N \l__regex_input_tl 29395
29396     \cs_set_eq:NN \__regex_input_item:n \__regex_reinsert_item:n 29396
29397     \exp_after:wN #1 \exp:w \l__regex_input_tl \exp_end: 29397
29398 } 29398
29399 \cs_new_protected:Npn \__regex_reinsert_item:n #1 29399
29400 { 29400
29401     \exp_after:wN \exp_after:wN 29401
29402     \exp_after:wN \exp_end: 29402
29403     \exp_after:wN \exp_after:wN 29403
29404     #1 29404
29405     \exp:w 29405
29406 } 29406
29407 \cs_new_protected:Npn \peek_regex_replace_once:nnTF #1 29407
29408 { \__regex_peek_replace:nnTF { \__regex_build_aux:Nn \c_false_bool {#1} } } 29408
29409 \cs_new_protected:Npn \peek_regex_replace_once:nnT #1#2#3 29409
29410 { \peek_regex_replace_once:nnTF {#1} {#2} {#3} { } } 29410
29411 \cs_new_protected:Npn \peek_regex_replace_once:nnF #1#2 29411
29412 { \peek_regex_replace_once:nnTF {#1} {#2} { } } 29412
29413 \cs_new_protected:Npn \peek_regex_replace_once:nn #1#2 29413
29414 { \peek_regex_replace_once:nnTF {#1} {#2} { } { } } 29414
29415 \cs_new_protected:Npn \peek_regex_replace_once:NnTF #1 29415
29416 { \__regex_peek_replace:nnTF { \__regex_build_aux:NN \c_false_bool #1 } } 29416
29417 \cs_new_protected:Npn \peek_regex_replace_once:NnT #1#2#3 29417
29418 { \peek_regex_replace_once:NnTF #1 {#2} {#3} { } } 29418
29419 \cs_new_protected:Npn \peek_regex_replace_once:NnF #1#2 29419
29420 { \peek_regex_replace_once:NnTF #1 {#2} { } } 29420
29421 \cs_new_protected:Npn \peek_regex_replace_once:Nn #1#2 29421
29422 { \peek_regex_replace_once:NnTF #1 {#2} { } { } } 29422
```

```
29423 \cs_new_protected:Npn \__regex_peek_replace:nnTF #1#2 29423
29424 { 29424
29425 \tl_set:Nn \l__regex_replacement_tl {#2} 29425
29426 \__regex_peek_aux:nnTF {#1} { \__regex_peek_replace_end: } 29426
29427 } 29427
29428 \cs_new_protected:Npn \__regex_peek_replace_end: 29428
29429 { 29429
29430 \bool_if:NTF \g__regex_success_bool 29430
29431 { 29431
29432 \__regex_extract: 29432
29433 \__regex_query_set_from_input_tl: 29433
29434 \cs_set_eq:NN \__regex_replacement_put:n \__regex_peek_replacement_put:n 29434
29435 \cs_set_eq:NN \__regex_replacement_put_submatch_aux:n 29435
29436 \__regex_peek_replacement_put_submatch_aux:n 29436
29437 \cs_set_eq:NN \__regex_input_item:n \__regex_reinsert_item:n 29437
29438 \cs_set_eq:NN \__regex_replacement_exp_not:N \__regex_peek_replacement_token:n 29438
29439 \cs_set_eq:NN \__regex_replacement_exp_not:V \__regex_peek_replacement_var:N 29439
29440 \exp_args:No \__regex_replacement:n { \l__regex_replacement_tl } 29440
29441 \use:e 29441
29442 { 29442
29443 \exp_not:n { \exp_after:wN \l__regex_peek_true_tl \exp:w } 29443
29444 \__regex_replacement_do_one_match:n \l__regex_zeroth_submatch_int 29444
29445 \__regex_query_range:nn 29445
29446 { 29446
29447 \__kernel_intarray_item:Nn \g__regex_submatch_end_intarray 29447
29448 \l__regex_zeroth_submatch_int 29448
29449 } 29449
29450 \l__regex_max_pos_int 29450
29451 \exp_end: 29451
29452 } 29452
29453 } 29453
29454 { \__regex_peek_reinsert:N \l__regex_peek_false_tl } 29454
29455 } 29455
29456 \cs_new_protected:Npn \__regex_query_set_from_input_tl: 29456
29457 { 29457
29458 \tl_build_end:N \l__regex_input_tl 29458
29459 \int_zero:N \l__regex_curr_pos_int 29459
29460 \cs_set_eq:NN \__regex_input_item:n \__regex_query_set_item:n 29460
29461 \__regex_query_set_item:n { } 29461
29462 \l__regex_input_tl 29462
29463 \__regex_query_set_item:n { } 29463
29464 \int_set_eq:NN \l__regex_max_pos_int \l__regex_curr_pos_int 29464
29465 } 29465
29466 \cs_new_protected:Npn \__regex_query_set_item:n #1 29466
29467 { 29467
29468 \int_incr:N \l__regex_curr_pos_int 29468
```

```
29469 \__regex_toks_set:Nn \l__regex_curr_pos_int { \__regex_input_item:n {#1} } 29469
29470 } 29470
29471 \cs_new_protected:Npn \__regex_peek_replacement_put:n #1 29471
29472 { 29472
29473 \if_case:w \l__regex_replacement_csnames_int 29473
29474 \tl_build_put_right:Nn \l__regex_build_tl 29474
29475 { \exp_not:N \__regex_reinsert_item:n {#1} } 29475
29476 \else: 29476
29477 \tl_build_put_right:Nn \l__regex_build_tl {#1} 29477
29478 \fi: 29478
29479 } 29479
29480 \cs_new_protected:Npn \__regex_peek_replacement_token:n #1 29480
29481 { \exp_after:wN \exp_end: \exp_after:wN #1 \exp:w } 29481
29482 \cs_new_protected:Npn \__regex_peek_replacement_put_submatch_aux:n #1 29482
29483 { 29483
29484 \if_case:w \l__regex_replacement_csnames_int 29484
29485 \tl_build_put_right:Nn \l__regex_build_tl 29485
29486 { \__regex_query_submatch:n { \__regex_int_eval:w #1 + ##1 \scan_stop: } } 29486
29487 \else: 29487
29488 \tl_build_put_right:Nn \l__regex_build_tl 29488
29489 { 29489
29490 \exp:w 29490
29491 \__regex_query_submatch:n { \__regex_int_eval:w #1 + ##1 \scan_stop: } 29491
29492 \exp_end: 29492
29493 } 29493
29494 \fi: 29494
29495 } 29495
29496 \cs_new_protected:Npn \__regex_peek_replacement_var:N #1 29496
29497 { 29497
29498 \exp_after:wN \exp_last_unbraced:NV 29498
29499 \exp_after:wN \exp_end: 29499
29500 \exp_after:wN #1 29500
29501 \exp:w 29501
29502 } 29502
29503 \use:e 29503
29504 { 29504
29505 \msg_new:nnn { regex } { trailing-backslash } 29505
29506 { Trailing~'\iow_char:N\\'\in~regex~or~replacement. } 29506
29507 \msg_new:nnn { regex } { x-missing-rbrace } 29507
29508 { 29508
29509 Missing~brace~'\iow_char:N\}'in~regex~ 29509
29510 '...\iow_char:N\\x\iow_char:N{...#1'. 29510
29511 } 29511
29512 \msg_new:nnn { regex } { x-overflow } 29512
29513 { 29513
29514 Character~code~##1~too~large~in~ 29514
```



```

29515 \iow_char:N\{x\iow_char:N\{##2\iow_char:N\}~regex. 29515
29516 } 29516
29517 } 29517
29518 \msg_new:nnnn { regex } { invalid-quantifier } 29518
29519 { Braced~quantifier~'#1'~may~not~be~followed~by~'#2'. } 29519
29520 { 29520
29521 The~character~'#2'~is~invalid~in~the~braced~quantifier~'#1'.~ 29521
29522 The~only~valid~quantifiers~are~'~*~','~?~','~+~','~{<int>}',~ 29522
29523 '{<min>,<max>}'~and~'{<min>,<max>}',~optionally~followed~by~'?'. 29523
29524 } 29524
29525 \msg_new:nnnn { regex } { missing-rbrack } 29525
29526 { Missing~right~bracket~inserted~in~regular~expression. } 29526
29527 { 29527
29528 LaTeX~was~given~a~regular~expression~where~a~character~class~ 29528
29529 was~started~with~'[',~but~the~matching~']'~is~missing. 29529
29530 } 29530
29531 \msg_new:nnnn { regex } { missing-rparen } 29531
29532 { 29532
29533 Missing~right~ 29533
29534 \int_compare:nTF { #1 = 1 } { parenthesis } { parentheses } ~ 29534
29535 inserted~in~regular~expression. 29535
29536 } 29536
29537 { 29537
29538 LaTeX~was~given~a~regular~expression~with~\int_eval:n {#1} ~ 29538
29539 more~left~parentheses~than~right~parentheses. 29539
29540 } 29540
29541 \msg_new:nnnn { regex } { extra-rparen } 29541
29542 { Extra~right~parenthesis~ignored~in~regular~expression. } 29542
29543 { 29543
29544 LaTeX~came~across~a~closing~parenthesis~when~no~submatch~group~ 29544
29545 was~open.~The~parenthesis~will~be~ignored. 29545
29546 } 29546
29547 \msg_new:nnnn { regex } { bad-escape } 29547
29548 { 29548
29549 Invalid~escape~'\iow_char:N\{##1'~ 29549
29550 \__regex_if_in_cs:TF { within~a~control~sequence. } 29550
29551 { 29551
29552 \__regex_if_in_class:TF 29552
29553 { in~a~character~class. } 29553
29554 { following~a~category~test. } 29554
29555 } 29555
29556 } 29556
29557 { 29557
29558 The~escape~sequence~'\iow_char:N\{##1'~may~not~appear~ 29558
29559 \__regex_if_in_cs:TF 29559
29560 { 29560

```



```
29561         within~a~control~sequence~test~introduced~by~ 29561
29562         '\iow_char:N\\c\iow_char:N\{' . 29562
29563     } 29563
29564     { 29564
29565         \__regex_if_in_class:TF 29565
29566         { within~a~character~class~ } 29566
29567         { following~a~category~test~such~as~'\iow_char:N\\cL'~ } 29567
29568         because~it~does~not~match~exactly~one~character. 29568
29569     } 29569
29570 } 29570
29571 \msg_new:nnnn { regex } { range-missing-end } 29571
29572 { Invalid~end~point~for~range~'#1-#2'~in~character~class. } 29572
29573 { 29573
29574     The~end~point~'#2'~of~the~range~'#1-#2'~may~not~serve~as~an~ 29574
29575     end~point~for~a~range:~alphanumeric~characters~should~not~be~ 29575
29576     escaped,~and~non-alphanumeric~characters~should~be~escaped. 29576
29577 } 29577
29578 \msg_new:nnnn { regex } { range-backwards } 29578
29579 { Range~'[#1-#2]'~out~of~order~in~character~class. } 29579
29580 { 29580
29581     In~ranges~of~characters~'[x-y]'~appearing~in~character~classes,~ 29581
29582     the~first~character~code~must~not~be~larger~than~the~second.~ 29582
29583     Here,~'#1'~has~character~code~\int_eval:n {`#1},~while~ 29583
29584     '#2'~has~character~code~\int_eval:n {`#2}. 29584
29585 } 29585
29586 \msg_new:nnnn { regex } { c-bad-mode } 29586
29587 { Invalid~nested~'\iow_char:N\\c'~escape~in~regular~expression. } 29587
29588 { 29588
29589     The~'\iow_char:N\\c'~escape~cannot~be~used~within~ 29589
29590     a~control~sequence~test~'\iow_char:N\\c{...}'~ 29590
29591     nor~another~category~test.~ 29591
29592     To~combine~several~category~tests,~use~'\iow_char:N\\c[...]'. 29592
29593 } 29593
29594 \msg_new:nnnn { regex } { c-C-invalid } 29594
29595 { '\iow_char:N\\cC'~should~be~followed~by~'.'~or~'(',~not~'#1'. } 29595
29596 { 29596
29597     The~'\iow_char:N\\cC'~construction~restricts~the~next~item~to~be~a~ 29597
29598     control~sequence~or~the~next~group~to~be~made~of~control~sequences.~ 29598
29599     It~only~makes~sense~to~follow~it~by~'.'~or~by~a~group. 29599
29600 } 29600
29601 \msg_new:nnnn { regex } { cu-lbrace } 29601
29602 { Left~braces~must~be~escaped~in~'\iow_char:N\\#1{...}' . } 29602
29603 { 29603
29604     Constructions~such~as~'\iow_char:N\\#1{... \iow_char:N\{...}'~are~ 29604
29605     not~allowed~and~should~be~replaced~by~ 29605
29606     '\iow_char:N\\#1{... \token_to_str:N\{...}' . 29606
```

```

29607 }
29608 \msg_new:nnnn { regex } { c-lparen-in-class }
29609 { Catcode~test~cannot~apply~to~group~in~character~class }
29610 {
29611   Construction~such~as~'\iow_char:N\\cL(abc)'\~are~not~allowed~inside~a~
29612   class~'[...]\~because~classes~do~not~match~multiple~characters~at~once.
29613 }
29614 \msg_new:nnnn { regex } { c-missing-rbrace }
29615 { Missing~right~brace~inserted~for~'\iow_char:N\\c'\~escape. }
29616 {
29617   LaTeX~was~given~a~regular~expression~where~a~
29618   '\iow_char:N\\c\iow_char:N{\...'\~construction~was~not~ended~
29619   with~a~closing~brace~'\iow_char:N\}'.
29620 }
29621 \msg_new:nnnn { regex } { c-missing-rbrack }
29622 { Missing~right~bracket~inserted~for~'\iow_char:N\\c'\~escape. }
29623 {
29624   A~construction~'\iow_char:N\\c[...'\~appears~in~a~
29625   regular~expression,~but~the~closing~']'\~is~not~present.
29626 }
29627 \msg_new:nnnn { regex } { c-missing-category }
29628 { Invalid~character~'#1'\~following~'\iow_char:N\\c'\~escape. }
29629 {
29630   In~regular~expressions,~the~'\iow_char:N\\c'\~escape~sequence~
29631   may~only~be~followed~by~a~left~brace,~a~left~bracket,~or~a~
29632   capital~letter~representing~a~character~category,~namely~
29633   one~of~'ABCDELMOPSTU'.
29634 }
29635 \msg_new:nnnn { regex } { c-trailing }
29636 { Trailing~category~code~escape~'\iow_char:N\\c'\... }
29637 {
29638   A~regular~expression~ends~with~'\iow_char:N\\c'\~followed~
29639   by~a~letter.~It~will~be~ignored.
29640 }
29641 \msg_new:nnnn { regex } { u-missing-lbrace }
29642 { Missing~left~brace~following~'\iow_char:N\\u'\~escape. }
29643 {
29644   The~'\iow_char:N\\u'\~escape~sequence~must~be~followed~by~
29645   a~brace~group~with~the~name~of~the~variable~to~use.
29646 }
29647 \msg_new:nnnn { regex } { u-missing-rbrace }
29648 { Missing~right~brace~inserted~for~'\iow_char:N\\u'\~escape. }
29649 {
29650   LaTeX~
29651   \str_if_eq:eeTF { } {#2}
29652   { reached~the~end~of~the~string~ }

```

```

29653 { encountered~an~escaped~alphanumeric~character '\iow_char:N\|#2'~ } 29653
29654 when~parsing~the~argument~of~an~ 29654
29655 '\iow_char:N\\u\iow_char:N\{...\}'~escape. 29655
29656 } 29656
29657 \msg_new:nnnn { regex } { posix-unsupported } 29657
29658 { POSIX~collating~element~'[#1 ~ #1]'~not~supported. } 29658
29659 { 29659
29660 The~'[.foo.]'~and~'[=bar=]'~syntaxes~have~a~special~meaning~ 29660
29661 in~POSIX~regular~expressions.~This~is~not~supported~by~LaTeX.~ 29661
29662 Maybe~you~forgot~to~escape~a~left~bracket~in~a~character~class? 29662
29663 } 29663
29664 \msg_new:nnnn { regex } { posix-unknown } 29664
29665 { POSIX~class~'[:#1:]'~unknown. } 29665
29666 { 29666
29667 '[:#1:]'~is~not~among~the~known~POSIX~classes~ 29667
29668 '[:alnum:]',~'[:alpha:]',~'[:ascii:]',~'[:blank:]',~ 29668
29669 '[:cntrl:]',~'[:digit:]',~'[:graph:]',~'[:lower:]',~ 29669
29670 '[:print:]',~'[:punct:]',~'[:space:]',~'[:upper:]',~ 29670
29671 '[:word:]',~and~'[:xdigit:]'. 29671
29672 } 29672
29673 \msg_new:nnnn { regex } { posix-missing-close } 29673
29674 { Missing~closing~':'~for~POSIX~class. } 29674
29675 { The~POSIX~syntax~'#1'~must~be~followed~by~':',~not~'#2'. } 29675
29676 \msg_new:nnnn { regex } { result-unbalanced } 29676
29677 { Missing~brace~inserted~when~#1. } 29677
29678 { 29678
29679 LaTeX~was~asked~to~do~some~regular~expression~operation,~ 29679
29680 and~the~resulting~token~list~would~not~have~the~same~number~ 29680
29681 of~begin~group~and~end~group~tokens.~Braces~were~inserted:~ 29681
29682 #2~left,~#3~right. 29682
29683 } 29683
29684 \msg_new:nnnn { regex } { unknown-option } 29684
29685 { Unknown~option~'#1'~for~regular~expressions. } 29685
29686 { 29686
29687 The~only~available~option~is~'case-insensitive',~toggled~by~ 29687
29688 '(?i)'~and~'(?-i)'. 29688
29689 } 29689
29690 \msg_new:nnnn { regex } { special-group-unknown } 29690
29691 { Unknown~special~group~'#1~...~'in~a~regular~expression. } 29691
29692 { 29692
29693 The~only~valid~constructions~starting~with~'(?~are~ 29693
29694 '(:~...~)',~'(|~...~)',~'(?i)',~and~'(?-i)'. 29694
29695 } 29695
29696 \msg_new:nnnn { regex } { replacement-c } 29696
29697 { Misused~'\iow_char:N\c'~command~in~a~replacement~text. } 29697
29698 { 29698

```

29699	In~a~replacement~text,~the~'\iow_char:N\\c'~escape~sequence~	29699
29700	can~be~followed~by~one~of~the~letters~'ABCDELMOPSTU'~	29700
29701	or~a~brace~group,~not~by~'#1'.	29701
29702	}	29702
29703	\msg_new:nnnn { regex } { replacement-u }	29703
29704	{ Misused~'\iow_char:N\\u'~command~in~a~replacement~text. }	29704
29705	{	29705
29706	In~a~replacement~text,~the~'\iow_char:N\\u'~escape~sequence~	29706
29707	must~be~followed~by~a~brace~group~holding~the~name~of~the~	29707
29708	variable~to~use.	29708
29709	}	29709
29710	\msg_new:nnnn { regex } { replacement-g }	29710
29711	{	29711
29712	Missing~brace~for~the~'\iow_char:N\\g'~construction~	29712
29713	in~a~replacement~text.	29713
29714	}	29714
29715	{	29715
29716	In~the~replacement~text~for~a~regular~expression~search,~	29716
29717	submatches~are~represented~either~as~'\iow_char:N \\g{dd..d}',~	29717
29718	or~'\\d',~where~'d'~are~single~digits.~Here,~a~brace~is~missing.	29718
29719	}	29719
29720	\msg_new:nnnn { regex } { replacement-catcode-end }	29720
29721	{	29721
29722	Missing~character~for~the~'\iow_char:N\\c<category><character>'~	29722
29723	construction~in~a~replacement~text.	29723
29724	}	29724
29725	{	29725
29726	In~a~replacement~text,~the~'\iow_char:N\\c'~escape~sequence~	29726
29727	can~be~followed~by~one~of~the~letters~'ABCDELMOPSTU'~representing~	29727
29728	the~character~category.~Then,~a~character~must~follow.~LaTeX~	29728
29729	reached~the~end~of~the~replacement~when~looking~for~that.	29729
29730	}	29730
29731	\msg_new:nnnn { regex } { replacement-catcode-escaped }	29731
29732	{	29732
29733	Escaped~letter~or~digit~after~category~code~in~replacement~text.	29733
29734	}	29734
29735	{	29735
29736	In~a~replacement~text,~the~'\iow_char:N\\c'~escape~sequence~	29736
29737	can~be~followed~by~one~of~the~letters~'ABCDELMOPSTU'~representing~	29737
29738	the~character~category.~Then,~a~character~must~follow,~not~	29738
29739	'\iow_char:N\\#2'.	29739
29740	}	29740
29741	\msg_new:nnnn { regex } { replacement-catcode-in-cs }	29741
29742	{	29742
29743	Category~code~'\iow_char:N\\c#1#3'~ignored~inside~	29743
29744	'\iow_char:N\\c{\...}'~in~a~replacement~text.	29744

```
29745 }
29746 {
29747     In~a~replacement~text,~the~category~codes~of~the~argument~of~
29748     '\iow_char:N\\c\{...\}'~are~ignored~when~building~the~control~
29749     sequence~name.
29750 }
29751 \msg_new:nnnn { regex } { replacement-null-space }
29752 { TeX~cannot~build~a~space~token~with~character~code~0. }
29753 {
29754     You~asked~for~a~character~token~with~category~space,~
29755     and~character~code~0,~for~instance~through~
29756     '\iow_char:N\\cS\iow_char:N\\x00'.~
29757     This~specific~case~is~impossible~and~will~be~replaced~
29758     by~a~normal~space.
29759 }
29760 \msg_new:nnnn { regex } { replacement-missing-rbrace }
29761 { Missing~right~brace~inserted~in~replacement~text. }
29762 {
29763     There~ \int_compare:nTF { #1 = 1 } { was } { were } ~ #1~
29764     missing~right~\int_compare:nTF { #1 = 1 } { brace } { braces } .
29765 }
29766 \msg_new:nnnn { regex } { replacement-missing-rparen }
29767 { Missing~right~parenthesis~inserted~in~replacement~text. }
29768 {
29769     There~ \int_compare:nTF { #1 = 1 } { was } { were } ~ #1~
29770     missing~right~
29771     \int_compare:nTF { #1 = 1 } { parenthesis } { parentheses } .
29772 }
29773 \msg_new:nnn { regex } { submatch-too-big }
29774 { Submatch~#1~used~but~regex~only~has~#2~group(s) }
29775 \msg_new:nnnn { regex } { backwards-quantifier }
29776 { Quantifer~"{#1,#2}"~is~backwards. }
29777 { The~values~given~in~a~quantifier~must~be~in~order. }
29778 \msg_new:nnnn { regex } { case-odd }
29779 { #1~with~odd~number~of~items }
29780 {
29781     There~must~be~a~#2~part~for~each~regex:~
29782     found~odd~number~of~items~(#3)~in\\
29783     \iow_indent:n {#4}
29784 }
29785 \msg_new:nnn { regex } { show }
29786 {
29787     >~Compiled~regex~
29788     \tl_if_empty:nTF {#1} { variable~ #2 } { {#1} } :
29789     #3
29790 }
```

```

29791 \prop_gput:Nnn \g_msg_module_name_prop { regex } { LaTeX }
29792 \prop_gput:Nnn \g_msg_module_type_prop { regex } { }
29793 \cs_new:Npn \__regex_msg_repeated:nnN #1#2#3
29794 {
29795   \str_if_eq:eeF { #1 #2 } { 1 0 }
29796   {
29797     , ~ repeated ~
29798     \int_case:nnF {#2}
29799     {
29800       { -1 } { #1~or~more~times,~\bool_if:NTF #3 { lazy } { greedy } }
29801       { 0 } { #1~times }
29802     }
29803     {
29804       between~#1~and~\int_eval:n {#1+#2}~times,~
29805       \bool_if:NTF #3 { lazy } { greedy }
29806     }
29807   }
29808 }
29809 \cs_new_protected:Npn \__regex_trace_push:nnN #1#2#3
29810 { \__regex_trace:nne {#1} {#2} { entering~ \token_to_str:N #3 } }
29811 \cs_new_protected:Npn \__regex_trace_pop:nnN #1#2#3
29812 { \__regex_trace:nne {#1} {#2} { leaving~ \token_to_str:N #3 } }
29813 \cs_new_protected:Npn \__regex_trace:nne #1#2#3
29814 {
29815   \int_compare:nNnF
29816   { \int_use:c { g__regex_trace_#1_int } } < {#2}
29817   { \iow_term:e { Trace:~#3 } }
29818 }
29819 \int_new:N \g__regex_trace_regex_int
29820 \cs_new_protected:Npn \__regex_trace_states:n #1
29821 {
29822   \int_step_inline:nnn
29823   \l__regex_min_state_int
29824   { \l__regex_max_state_int - \c_one_int }
29825   {
29826     \__regex_trace:nne { regex } {#1}
29827     { \iow_char:N \\toks ##1 = { \__regex_toks_use:w ##1 } }
29828   }
29829 }
29830 %% File: l3box.dtx
29831 \cs_new_eq:NN \__box_dim_eval:w \tex_dimexpr:D
29832 \cs_new:Npn \__box_dim_eval:n #1
29833 { \__box_dim_eval:w #1 \scan_stop: }
29834 \cs_new_protected:Npn \__kernel_kern:n #1
29835 { \tex_kern:D \__box_dim_eval:n {#1} }
29836 \cs_new_protected:Npn \box_new:N #1

```


29837	{	29837
29838	_kernel_chk_if_free_cs:N #1	29838
29839	\cs:w newbox \cs_end: #1	29839
29840	}	29840
29841	\cs_generate_variant:Nn \box_new:N { c }	29841
29842	\cs_new_protected:Npn \box_clear:N #1	29842
29843	{ \box_set_eq:NN #1 \c_empty_box }	29843
29844	\cs_new_protected:Npn \box_gclear:N #1	29844
29845	{ \box_gset_eq:NN #1 \c_empty_box }	29845
29846	\cs_generate_variant:Nn \box_clear:N { c }	29846
29847	\cs_generate_variant:Nn \box_gclear:N { c }	29847
29848	\cs_new_protected:Npn \box_clear_new:N #1	29848
29849	{ \box_if_exist:NTF #1 { \box_clear:N #1 } { \box_new:N #1 } }	29849
29850	\cs_new_protected:Npn \box_gclear_new:N #1	29850
29851	{ \box_if_exist:NTF #1 { \box_gclear:N #1 } { \box_new:N #1 } }	29851
29852	\cs_generate_variant:Nn \box_clear_new:N { c }	29852
29853	\cs_generate_variant:Nn \box_gclear_new:N { c }	29853
29854	\cs_new_protected:Npn \box_set_eq:NN #1#2	29854
29855	{ \tex_setbox:D #1 \tex_copy:D #2 }	29855
29856	\cs_new_protected:Npn \box_gset_eq:NN #1#2	29856
29857	{ \tex_global:D \tex_setbox:D #1 \tex_copy:D #2 }	29857
29858	\cs_generate_variant:Nn \box_set_eq:NN { c , Nc , cc }	29858
29859	\cs_generate_variant:Nn \box_gset_eq:NN { c , Nc , cc }	29859
29860	\cs_new_protected:Npn \box_set_eq_drop:NN #1#2	29860
29861	{ \tex_setbox:D #1 \tex_box:D #2 }	29861
29862	\cs_new_protected:Npn \box_gset_eq_drop:NN #1#2	29862
29863	{ \tex_global:D \tex_setbox:D #1 \tex_box:D #2 }	29863
29864	\cs_generate_variant:Nn \box_set_eq_drop:NN { c , Nc , cc }	29864
29865	\cs_generate_variant:Nn \box_gset_eq_drop:NN { c , Nc , cc }	29865
29866	\prg_new_eq_conditional:NNn \box_if_exist:N \cs_if_exist:N	29866
29867	{ TF , T , F , p }	29867
29868	\prg_new_eq_conditional:NNn \box_if_exist:c \cs_if_exist:c	29868
29869	{ TF , T , F , p }	29869
29870	\cs_new_eq:NN \box_ht:N \tex_ht:D	29870
29871	\cs_new_eq:NN \box_dp:N \tex_dp:D	29871
29872	\cs_new_eq:NN \box_wd:N \tex_wd:D	29872
29873	\cs_generate_variant:Nn \box_ht:N { c }	29873
29874	\cs_generate_variant:Nn \box_dp:N { c }	29874
29875	\cs_generate_variant:Nn \box_wd:N { c }	29875
29876	\cs_new_protected:Npn \box_ht_plus_dp:N #1	29876
29877	{ _box_dim_eval:n { \box_ht:N #1 + \box_dp:N #1 } }	29877
29878	\cs_generate_variant:Nn \box_ht_plus_dp:N { c }	29878
29879	\cs_new_protected:Npn \box_set_dp:Nn #1#2	29879
29880	{	29880
29881	\tex_setbox:D #1 = \tex_copy:D #1	29881
29882	\box_dp:N #1 _box_dim_eval:n {#2}	29882

29883	}	29883
29884	\cs_generate_variant:Nn \box_set_dp:Nn { c }	29884
29885	\cs_new_protected:Npn \box_gset_dp:Nn #1#2	29885
29886	{ \box_dp:N #1 __box_dim_eval:n {#2} }	29886
29887	\cs_generate_variant:Nn \box_gset_dp:Nn { c }	29887
29888	\cs_new_protected:Npn \box_set_ht:Nn #1#2	29888
29889	{	29889
29890	\tex_setbox:D #1 = \tex_copy:D #1	29890
29891	\box_ht:N #1 __box_dim_eval:n {#2}	29891
29892	}	29892
29893	\cs_generate_variant:Nn \box_set_ht:Nn { c }	29893
29894	\cs_new_protected:Npn \box_gset_ht:Nn #1#2	29894
29895	{ \box_ht:N #1 __box_dim_eval:n {#2} }	29895
29896	\cs_generate_variant:Nn \box_gset_ht:Nn { c }	29896
29897	\cs_new_protected:Npn \box_set_wd:Nn #1#2	29897
29898	{	29898
29899	\tex_setbox:D #1 = \tex_copy:D #1	29899
29900	\box_wd:N #1 __box_dim_eval:n {#2}	29900
29901	}	29901
29902	\cs_generate_variant:Nn \box_set_wd:Nn { c }	29902
29903	\cs_new_protected:Npn \box_gset_wd:Nn #1#2	29903
29904	{ \box_wd:N #1 __box_dim_eval:n {#2} }	29904
29905	\cs_generate_variant:Nn \box_gset_wd:Nn { c }	29905
29906	\cs_new_eq:NN \box_use_drop:N \tex_box:D	29906
29907	\cs_new_eq:NN \box_use:N \tex_copy:D	29907
29908	\cs_generate_variant:Nn \box_use_drop:N { c }	29908
29909	\cs_generate_variant:Nn \box_use:N { c }	29909
29910	\cs_new_protected:Npn \box_move_left:nn #1#2	29910
29911	{ \tex_moveleft:D __box_dim_eval:n {#1} #2 }	29911
29912	\cs_new_protected:Npn \box_move_right:nn #1#2	29912
29913	{ \tex_moveright:D __box_dim_eval:n {#1} #2 }	29913
29914	\cs_new_protected:Npn \box_move_up:nn #1#2	29914
29915	{ \tex_raise:D __box_dim_eval:n {#1} #2 }	29915
29916	\cs_new_protected:Npn \box_move_down:nn #1#2	29916
29917	{ \tex_lower:D __box_dim_eval:n {#1} #2 }	29917
29918	\cs_new_eq:NN \if_hbox:N \tex_ifhbox:D	29918
29919	\cs_new_eq:NN \if_vbox:N \tex_ifvbox:D	29919
29920	\cs_new_eq:NN \if_box_empty:N \tex_ifvoid:D	29920
29921	\prg_new_conditional:Npnn \box_if_horizontal:N #1 { p , T , F , TF }	29921
29922	{ \if_hbox:N #1 \prg_return_true: \else: \prg_return_false: \fi: }	29922
29923	\prg_new_conditional:Npnn \box_if_vertical:N #1 { p , T , F , TF }	29923
29924	{ \if_vbox:N #1 \prg_return_true: \else: \prg_return_false: \fi: }	29924
29925	\prg_generate_conditional_variant:Nnn \box_if_horizontal:N	29925
29926	{ c } { p , T , F , TF }	29926
29927	\prg_generate_conditional_variant:Nnn \box_if_vertical:N	29927
29928	{ c } { p , T , F , TF }	29928

```

29929 \prg_new_conditional:Npnn \box_if_empty:N #1{ p , T , F , TF }
29930 { \if_box_empty:N #1 \prg_return_true: \else: \prg_return_false: \fi: }
29931 \prg_generate_conditional_variant:Nnn \box_if_empty:N
29932 { c } { p , T , F , TF }
29933 \cs_new_protected:Npn \box_set_to_last:N #1
29934 { \tex_setbox:D #1 \tex_lastbox:D }
29935 \cs_new_protected:Npn \box_gset_to_last:N #1
29936 { \tex_global:D \tex_setbox:D #1 \tex_lastbox:D }
29937 \cs_generate_variant:Nn \box_set_to_last:N { c }
29938 \cs_generate_variant:Nn \box_gset_to_last:N { c }
29939 \box_new:N \c_empty_box
29940 \box_new:N \l_tmpa_box
29941 \box_new:N \l_tmpb_box
29942 \box_new:N \g_tmpa_box
29943 \box_new:N \g_tmpb_box
29944 \cs_new_protected:Npn \box_show:N #1
29945 { \box_show:Nnn #1 \c_max_int \c_max_int }
29946 \cs_generate_variant:Nn \box_show:N { c }
29947 \cs_new_protected:Npn \box_show:Nnn #1#2#3
29948 { \__box_show:NNff 1 #1 { \int_eval:n {#2} } { \int_eval:n {#3} } }
29949 \cs_generate_variant:Nn \box_show:Nnn { c }
29950 \cs_new_protected:Npn \box_log:N #1
29951 { \box_log:Nnn #1 \c_max_int \c_max_int }
29952 \cs_generate_variant:Nn \box_log:N { c }
29953 \cs_new_protected:Npn \box_log:Nnn
29954 { \exp_args:No \__box_log:nNnn { \tex_the:D \tex_interactionmode:D } }
29955 \cs_new_protected:Npn \__box_log:nNnn #1#2#3#4
29956 {
29957 \int_gset:Nn \tex_interactionmode:D { 0 }
29958 \__box_show:NNff 0 #2 { \int_eval:n {#3} } { \int_eval:n {#4} }
29959 \int_gset:Nn \tex_interactionmode:D {#1}
29960 }
29961 \cs_generate_variant:Nn \box_log:Nnn { c }
29962 \cs_new_protected:Npn \__box_show:NNnn #1#2#3#4
29963 {
29964 \box_if_exist:NTF #2
29965 {
29966 \group_begin:
29967 \int_set:Nn \tex_showboxbreadth:D {#3}
29968 \int_set:Nn \tex_showboxdepth:D {#4}
29969 \int_set:Nn \tex_tracingonline:D {#1}
29970 \int_set:Nn \tex_errorcontextlines:D { -1 }
29971 \tex_showbox:D \use:n {#2}
29972 \group_end:
29973 }
29974 {

```

```
29975         \msg_error:nne { kernel } { variable-not-defined } 29975
29976         { \token_to_str:N #2 } 29976
29977     } 29977
29978 } 29978
29979 \cs_generate_variant:Nn \__box_show:NNnn { NNff } 29979
29980 \cs_new_protected:Npn \hbox:n #1 29980
29981     { \tex_hbox:D \scan_stop: { \color_group_begin: #1 \color_group_end: } } 29981
29982 \cs_new_protected:Npn \hbox_set:Nn #1#2 29982
29983     { 29983
29984         \tex_setbox:D #1 \tex_hbox:D 29984
29985         { \color_group_begin: #2 \color_group_end: } 29985
29986     } 29986
29987 \cs_new_protected:Npn \hbox_gset:Nn #1#2 29987
29988     { 29988
29989         \tex_global:D \tex_setbox:D #1 \tex_hbox:D 29989
29990         { \color_group_begin: #2 \color_group_end: } 29990
29991     } 29991
29992 \cs_generate_variant:Nn \hbox_set:Nn { c } 29992
29993 \cs_generate_variant:Nn \hbox_gset:Nn { c } 29993
29994 \cs_new_protected:Npn \hbox_set_to_wd:Nnn #1#2#3 29994
29995     { 29995
29996         \tex_setbox:D #1 \tex_hbox:D to \__box_dim_eval:n {#2} 29996
29997         { \color_group_begin: #3 \color_group_end: } 29997
29998     } 29998
29999 \cs_new_protected:Npn \hbox_gset_to_wd:Nnn #1#2#3 29999
30000     { 30000
30001         \tex_global:D \tex_setbox:D #1 \tex_hbox:D to \__box_dim_eval:n {#2} 30001
30002         { \color_group_begin: #3 \color_group_end: } 30002
30003     } 30003
30004 \cs_generate_variant:Nn \hbox_set_to_wd:Nnn { c } 30004
30005 \cs_generate_variant:Nn \hbox_gset_to_wd:Nnn { c } 30005
30006 \cs_new_protected:Npn \hbox_set:Nw #1 30006
30007     { 30007
30008         \tex_setbox:D #1 \tex_hbox:D 30008
30009         \c_group_begin_token 30009
30010         \color_group_begin: 30010
30011     } 30011
30012 \cs_new_protected:Npn \hbox_gset:Nw #1 30012
30013     { 30013
30014         \tex_global:D \tex_setbox:D #1 \tex_hbox:D 30014
30015         \c_group_begin_token 30015
30016         \color_group_begin: 30016
30017     } 30017
30018 \cs_generate_variant:Nn \hbox_set:Nw { c } 30018
30019 \cs_generate_variant:Nn \hbox_gset:Nw { c } 30019
30020 \cs_new_protected:Npn \hbox_set_end: 30020
```

```

30021 {
30022     \color_group_end:
30023     \c_group_end_token
30024 }
30025 \cs_new_eq:NN \hbox_gset_end: \hbox_set_end:
30026 \cs_new_protected:Npn \hbox_set_to_wd:Nnw #1#2
30027 {
30028     \tex_setbox:D #1 \tex_hbox:D to \__box_dim_eval:n {#2}
30029     \c_group_begin_token
30030     \color_group_begin:
30031 }
30032 \cs_new_protected:Npn \hbox_gset_to_wd:Nnw #1#2
30033 {
30034     \tex_global:D \tex_setbox:D #1 \tex_hbox:D to \__box_dim_eval:n {#2}
30035     \c_group_begin_token
30036     \color_group_begin:
30037 }
30038 \cs_generate_variant:Nn \hbox_set_to_wd:Nnw { c }
30039 \cs_generate_variant:Nn \hbox_gset_to_wd:Nnw { c }
30040 \cs_new_protected:Npn \hbox_to_wd:nn #1#2
30041 {
30042     \tex_hbox:D to \__box_dim_eval:n {#1}
30043     { \color_group_begin: #2 \color_group_end: }
30044 }
30045 \cs_new_protected:Npn \hbox_to_zero:n #1
30046 {
30047     \tex_hbox:D to \c_zero_dim
30048     { \color_group_begin: #1 \color_group_end: }
30049 }
30050 \cs_new_protected:Npn \hbox_overlap_center:n #1
30051 { \hbox_to_zero:n { \tex_hss:D #1 \tex_hss:D } }
30052 \cs_new_protected:Npn \hbox_overlap_left:n #1
30053 { \hbox_to_zero:n { \tex_hss:D #1 } }
30054 \cs_new_protected:Npn \hbox_overlap_right:n #1
30055 { \hbox_to_zero:n { #1 \tex_hss:D } }
30056 \cs_new_eq:NN \hbox_unpack:N \tex_unhcopy:D
30057 \cs_new_eq:NN \hbox_unpack_drop:N \tex_unhbox:D
30058 \cs_generate_variant:Nn \hbox_unpack:N { c }
30059 \cs_generate_variant:Nn \hbox_unpack_drop:N { c }
30060 \cs_new_protected:Npn \vbox:n #1
30061 { \tex_vbox:D { \color_group_begin: #1 \par \color_group_end: } }
30062 \cs_new_protected:Npn \vbox_top:n #1
30063 { \tex_vtop:D { \color_group_begin: #1 \par \color_group_end: } }
30064 \cs_new_protected:Npn \vbox_to_ht:nn #1#2
30065 {
30066     \tex_vbox:D to \__box_dim_eval:n {#1}

```

```
30067 { \color_group_begin: #2 \par \color_group_end: } 30067
30068 } 30068
30069 \cs_new_protected:Npn \vbox_to_zero:n #1 30069
30070 { 30070
30071 \tex_vbox:D to \c_zero_dim 30071
30072 { \color_group_begin: #1 \par \color_group_end: } 30072
30073 } 30073
30074 \cs_new_protected:Npn \vbox_set:Nn #1#2 30074
30075 { 30075
30076 \tex_setbox:D #1 \tex_vbox:D 30076
30077 { \color_group_begin: #2 \par \color_group_end: } 30077
30078 } 30078
30079 \cs_new_protected:Npn \vbox_gset:Nn #1#2 30079
30080 { 30080
30081 \tex_global:D \tex_setbox:D #1 \tex_vbox:D 30081
30082 { \color_group_begin: #2 \par \color_group_end: } 30082
30083 } 30083
30084 \cs_generate_variant:Nn \vbox_set:Nn { c } 30084
30085 \cs_generate_variant:Nn \vbox_gset:Nn { c } 30085
30086 \cs_new_protected:Npn \vbox_set_top:Nn #1#2 30086
30087 { 30087
30088 \tex_setbox:D #1 \tex_vtop:D 30088
30089 { \color_group_begin: #2 \par \color_group_end: } 30089
30090 } 30090
30091 \cs_new_protected:Npn \vbox_gset_top:Nn #1#2 30091
30092 { 30092
30093 \tex_global:D \tex_setbox:D #1 \tex_vtop:D 30093
30094 { \color_group_begin: #2 \par \color_group_end: } 30094
30095 } 30095
30096 \cs_generate_variant:Nn \vbox_set_top:Nn { c } 30096
30097 \cs_generate_variant:Nn \vbox_gset_top:Nn { c } 30097
30098 \cs_new_protected:Npn \vbox_set_to_ht:Nnn #1#2#3 30098
30099 { 30099
30100 \tex_setbox:D #1 \tex_vbox:D to \__box_dim_eval:n {#2} 30100
30101 { \color_group_begin: #3 \par \color_group_end: } 30101
30102 } 30102
30103 \cs_new_protected:Npn \vbox_gset_to_ht:Nnn #1#2#3 30103
30104 { 30104
30105 \tex_global:D \tex_setbox:D #1 \tex_vbox:D to \__box_dim_eval:n {#2} 30105
30106 { \color_group_begin: #3 \par \color_group_end: } 30106
30107 } 30107
30108 \cs_generate_variant:Nn \vbox_set_to_ht:Nnn { c } 30108
30109 \cs_generate_variant:Nn \vbox_gset_to_ht:Nnn { c } 30109
30110 \cs_new_protected:Npn \vbox_set:Nw #1 30110
30111 { 30111
30112 \tex_setbox:D #1 \tex_vbox:D 30112
```

```

30113         \c_group_begin_token
30114         \color_group_begin:
30115     }
30116 \cs_new_protected:Npn \vbox_gset:Nw #1
30117 {
30118     \tex_global:D \tex_setbox:D #1 \tex_vbox:D
30119     \c_group_begin_token
30120     \color_group_begin:
30121 }
30122 \cs_generate_variant:Nn \vbox_set:Nw { c }
30123 \cs_generate_variant:Nn \vbox_gset:Nw { c }
30124 \cs_new_protected:Npn \vbox_set_end:
30125 {
30126     \par
30127     \color_group_end:
30128     \c_group_end_token
30129 }
30130 \cs_new_eq:NN \vbox_gset_end: \vbox_set_end:
30131 \cs_new_protected:Npn \vbox_set_to_ht:Nnw #1#2
30132 {
30133     \tex_setbox:D #1 \tex_vbox:D to \__box_dim_eval:n {#2}
30134     \c_group_begin_token
30135     \color_group_begin:
30136 }
30137 \cs_new_protected:Npn \vbox_gset_to_ht:Nnw #1#2
30138 {
30139     \tex_global:D \tex_setbox:D #1 \tex_vbox:D to \__box_dim_eval:n {#2}
30140     \c_group_begin_token
30141     \color_group_begin:
30142 }
30143 \cs_generate_variant:Nn \vbox_set_to_ht:Nnw { c }
30144 \cs_generate_variant:Nn \vbox_gset_to_ht:Nnw { c }
30145 \cs_new_eq:NN \vbox_unpack:N \tex_unvcopy:D
30146 \cs_new_eq:NN \vbox_unpack_drop:N \tex_unvbox:D
30147 \cs_generate_variant:Nn \vbox_unpack:N { c }
30148 \cs_generate_variant:Nn \vbox_unpack_drop:N { c }
30149 \cs_new_protected:Npn \vbox_set_split_to_ht:NNn #1#2#3
30150 { \tex_setbox:D #1 \tex_vsplit:D #2 to \__box_dim_eval:n {#3} }
30151 \cs_generate_variant:Nn \vbox_set_split_to_ht:NNn { c , Nc , cc }
30152 \cs_new_protected:Npn \vbox_gset_split_to_ht:NNn #1#2#3
30153 {
30154     \tex_global:D \tex_setbox:D #1
30155     \tex_vsplit:D #2 to \__box_dim_eval:n {#3}
30156 }
30157 \cs_generate_variant:Nn \vbox_gset_split_to_ht:NNn { c , Nc , cc }
30158 \fp_new:N \l__box_angle_fp

```



```

30159 \fp_new:N \l__box_cos_fp
30160 \fp_new:N \l__box_sin_fp
30161 \dim_new:N \l__box_top_dim
30162 \dim_new:N \l__box_bottom_dim
30163 \dim_new:N \l__box_left_dim
30164 \dim_new:N \l__box_right_dim
30165 \dim_new:N \l__box_top_new_dim
30166 \dim_new:N \l__box_bottom_new_dim
30167 \dim_new:N \l__box_left_new_dim
30168 \dim_new:N \l__box_right_new_dim
30169 \box_new:N \l__box_internal_box
30170 \cs_new_protected:Npn \box_rotate:Nn #1#2
30171 { \__box_rotate:NnN #1 {#2} \hbox_set:Nn }
30172 \cs_generate_variant:Nn \box_rotate:Nn { c }
30173 \cs_new_protected:Npn \box_grotate:Nn #1#2
30174 { \__box_rotate:NnN #1 {#2} \hbox_gset:Nn }
30175 \cs_generate_variant:Nn \box_grotate:Nn { c }
30176 \cs_new_protected:Npn \__box_rotate:NnN #1#2#3
30177 {
30178     #3 #1
30179     {
30180         \fp_set:Nn \l__box_angle_fp {#2}
30181         \fp_set:Nn \l__box_sin_fp { sind ( \l__box_angle_fp ) }
30182         \fp_set:Nn \l__box_cos_fp { cosd ( \l__box_angle_fp ) }
30183         \__box_rotate:N #1
30184     }
30185 }
30186 \cs_new_protected:Npn \__box_rotate:N #1
30187 {
30188     \dim_set:Nn \l__box_top_dim { \box_ht:N #1 }
30189     \dim_set:Nn \l__box_bottom_dim { -\box_dp:N #1 }
30190     \dim_set:Nn \l__box_right_dim { \box_wd:N #1 }
30191     \dim_zero:N \l__box_left_dim
30192     \fp_compare:nNnTF \l__box_sin_fp > \c_zero_fp
30193     {
30194         \fp_compare:nNnTF \l__box_cos_fp > \c_zero_fp
30195         { \__box_rotate_quadrant_one: }
30196         { \__box_rotate_quadrant_two: }
30197     }
30198     {
30199         \fp_compare:nNnTF \l__box_cos_fp < \c_zero_fp
30200         { \__box_rotate_quadrant_three: }
30201         { \__box_rotate_quadrant_four: }
30202     }
30203     \hbox_set:Nn \l__box_internal_box { \box_use:N #1 }
30204     \hbox_set:Nn \l__box_internal_box

```



```

30205 {
30206     \__kernel_kern:n { -\l__box_left_new_dim }
30207     \hbox:n
30208     {
30209         \__box_backend_rotate:Nn
30210         \l__box_internal_box
30211         \l__box_angle_fp
30212     }
30213 }
30214 \box_set_ht:Nn \l__box_internal_box { \l__box_top_new_dim }
30215 \box_set_dp:Nn \l__box_internal_box { -\l__box_bottom_new_dim }
30216 \box_set_wd:Nn \l__box_internal_box
30217 { \l__box_right_new_dim - \l__box_left_new_dim }
30218 \box_use_drop:N \l__box_internal_box
30219 }
30220 \cs_new_protected:Npn \__box_rotate_xdir:nnN #1#2#3
30221 {
30222     \dim_set:Nn #3
30223     {
30224         \fp_to_dim:n
30225         {
30226             \l__box_cos_fp * \dim_to_fp:n {#1}
30227             - \l__box_sin_fp * \dim_to_fp:n {#2}
30228         }
30229     }
30230 }
30231 \cs_new_protected:Npn \__box_rotate_ydir:nnN #1#2#3
30232 {
30233     \dim_set:Nn #3
30234     {
30235         \fp_to_dim:n
30236         {
30237             \l__box_sin_fp * \dim_to_fp:n {#1}
30238             + \l__box_cos_fp * \dim_to_fp:n {#2}
30239         }
30240     }
30241 }
30242 \cs_new_protected:Npn \__box_rotate_quadrant_one:
30243 {
30244     \__box_rotate_ydir:nnN \l__box_right_dim \l__box_top_dim
30245     \l__box_top_new_dim
30246     \__box_rotate_ydir:nnN \l__box_left_dim \l__box_bottom_dim
30247     \l__box_bottom_new_dim
30248     \__box_rotate_xdir:nnN \l__box_left_dim \l__box_top_dim
30249     \l__box_left_new_dim
30250     \__box_rotate_xdir:nnN \l__box_right_dim \l__box_bottom_dim

```

```
30251         \l__box_right_new_dim
30252     }
30253 \cs_new_protected:Npn \__box_rotate_quadrant_two:
30254 {
30255     \__box_rotate_ydir:nnN \l__box_right_dim \l__box_bottom_dim
30256     \l__box_top_new_dim
30257     \__box_rotate_ydir:nnN \l__box_left_dim \l__box_top_dim
30258     \l__box_bottom_new_dim
30259     \__box_rotate_xdir:nnN \l__box_right_dim \l__box_top_dim
30260     \l__box_left_new_dim
30261     \__box_rotate_xdir:nnN \l__box_left_dim \l__box_bottom_dim
30262     \l__box_right_new_dim
30263 }
30264 \cs_new_protected:Npn \__box_rotate_quadrant_three:
30265 {
30266     \__box_rotate_ydir:nnN \l__box_left_dim \l__box_bottom_dim
30267     \l__box_top_new_dim
30268     \__box_rotate_ydir:nnN \l__box_right_dim \l__box_top_dim
30269     \l__box_bottom_new_dim
30270     \__box_rotate_xdir:nnN \l__box_right_dim \l__box_bottom_dim
30271     \l__box_left_new_dim
30272     \__box_rotate_xdir:nnN \l__box_left_dim \l__box_top_dim
30273     \l__box_right_new_dim
30274 }
30275 \cs_new_protected:Npn \__box_rotate_quadrant_four:
30276 {
30277     \__box_rotate_ydir:nnN \l__box_left_dim \l__box_top_dim
30278     \l__box_top_new_dim
30279     \__box_rotate_ydir:nnN \l__box_right_dim \l__box_bottom_dim
30280     \l__box_bottom_new_dim
30281     \__box_rotate_xdir:nnN \l__box_left_dim \l__box_bottom_dim
30282     \l__box_left_new_dim
30283     \__box_rotate_xdir:nnN \l__box_right_dim \l__box_top_dim
30284     \l__box_right_new_dim
30285 }
30286 \fp_new:N \l__box_scale_x_fp
30287 \fp_new:N \l__box_scale_y_fp
30288 \cs_new_protected:Npn \box_resize_to_wd_and_ht_plus_dp:Nnn #1#2#3
30289 {
30290     \__box_resize_to_wd_and_ht_plus_dp:NnnN #1 {#2} {#3}
30291     \hbox_set:Nn
30292 }
30293 \cs_generate_variant:Nn \box_resize_to_wd_and_ht_plus_dp:Nnn { c }
30294 \cs_new_protected:Npn \box_gresize_to_wd_and_ht_plus_dp:Nnn #1#2#3
30295 {
30296     \__box_resize_to_wd_and_ht_plus_dp:NnnN #1 {#2} {#3}
```

```

30297 \hbox_gset:Nn
30298 }
30299 \cs_generate_variant:Nn \box_gresize_to_wd_and_ht_plus_dp:Nnn { c }
30300 \cs_new_protected:Npn \__box_resize_to_wd_and_ht_plus_dp:NnnN #1#2#3#4
30301 {
30302   #4 #1
30303   {
30304     \__box_resize_set_corners:N #1
30305     \fp_set:Nn \l__box_scale_x_fp
30306       { \dim_to_fp:n {#2} / \dim_to_fp:n { \l__box_right_dim } }
30307     \fp_set:Nn \l__box_scale_y_fp
30308     {
30309       \dim_to_fp:n {#3}
30310       / \dim_to_fp:n { \l__box_top_dim - \l__box_bottom_dim }
30311     }
30312     \__box_resize:N #1
30313   }
30314 }
30315 \cs_new_protected:Npn \__box_resize_set_corners:N #1
30316 {
30317   \dim_set:Nn \l__box_top_dim { \box_ht:N #1 }
30318   \dim_set:Nn \l__box_bottom_dim { -\box_dp:N #1 }
30319   \dim_set:Nn \l__box_right_dim { \box_wd:N #1 }
30320   \dim_zero:N \l__box_left_dim
30321 }
30322 \cs_new_protected:Npn \__box_resize:N #1
30323 {
30324   \__box_resize:NNN \l__box_right_new_dim
30325     \l__box_scale_x_fp \l__box_right_dim
30326   \__box_resize:NNN \l__box_bottom_new_dim
30327     \l__box_scale_y_fp \l__box_bottom_dim
30328   \__box_resize:NNN \l__box_top_new_dim
30329     \l__box_scale_y_fp \l__box_top_dim
30330   \__box_resize_common:N #1
30331 }
30332 \cs_new_protected:Npn \__box_resize:NNN #1#2#3
30333 {
30334   \dim_set:Nn #1
30335     { \fp_to_dim:n { \fp_abs:n { #2 } * \dim_to_fp:n { #3 } } }
30336 }
30337 \cs_new_protected:Npn \box_resize_to_ht:Nn #1#2
30338 { \__box_resize_to_ht:NnN #1 {#2} \hbox_set:Nn }
30339 \cs_generate_variant:Nn \box_resize_to_ht:Nn { c }
30340 \cs_new_protected:Npn \box_gresize_to_ht:Nn #1#2
30341 { \__box_resize_to_ht:NnN #1 {#2} \hbox_gset:Nn }
30342 \cs_generate_variant:Nn \box_gresize_to_ht:Nn { c }

```

```
30343 \cs_new_protected:Npn \__box_resize_to_ht:NnN #1#2#3 30343
30344 { 30344
30345 #3 #1 30345
30346 { 30346
30347 \__box_resize_set_corners:N #1 30347
30348 \fp_set:Nn \l__box_scale_y_fp 30348
30349 { 30349
30350 \dim_to_fp:n {#2} 30350
30351 / \dim_to_fp:n { \l__box_top_dim } 30351
30352 } 30352
30353 \fp_set_eq:NN \l__box_scale_x_fp \l__box_scale_y_fp 30353
30354 \__box_resize:N #1 30354
30355 } 30355
30356 } 30356
30357 \cs_new_protected:Npn \box_resize_to_ht_plus_dp:Nn #1#2 30357
30358 { \__box_resize_to_ht_plus_dp:NnN #1 {#2} \hbox_set:Nn } 30358
30359 \cs_generate_variant:Nn \box_resize_to_ht_plus_dp:Nn { c } 30359
30360 \cs_new_protected:Npn \box_gresize_to_ht_plus_dp:Nn #1#2 30360
30361 { \__box_resize_to_ht_plus_dp:NnN #1 {#2} \hbox_gset:Nn } 30361
30362 \cs_generate_variant:Nn \box_gresize_to_ht_plus_dp:Nn { c } 30362
30363 \cs_new_protected:Npn \__box_resize_to_ht_plus_dp:NnN #1#2#3 30363
30364 { 30364
30365 #3 #1 30365
30366 { 30366
30367 \__box_resize_set_corners:N #1 30367
30368 \fp_set:Nn \l__box_scale_y_fp 30368
30369 { 30369
30370 \dim_to_fp:n {#2} 30370
30371 / \dim_to_fp:n { \l__box_top_dim - \l__box_bottom_dim } 30371
30372 } 30372
30373 \fp_set_eq:NN \l__box_scale_x_fp \l__box_scale_y_fp 30373
30374 \__box_resize:N #1 30374
30375 } 30375
30376 } 30376
30377 \cs_new_protected:Npn \box_resize_to_wd:Nn #1#2 30377
30378 { \__box_resize_to_wd:NnN #1 {#2} \hbox_set:Nn } 30378
30379 \cs_generate_variant:Nn \box_resize_to_wd:Nn { c } 30379
30380 \cs_new_protected:Npn \box_gresize_to_wd:Nn #1#2 30380
30381 { \__box_resize_to_wd:NnN #1 {#2} \hbox_gset:Nn } 30381
30382 \cs_generate_variant:Nn \box_gresize_to_wd:Nn { c } 30382
30383 \cs_new_protected:Npn \__box_resize_to_wd:NnN #1#2#3 30383
30384 { 30384
30385 #3 #1 30385
30386 { 30386
30387 \__box_resize_set_corners:N #1 30387
30388 \fp_set:Nn \l__box_scale_x_fp 30388
```

```
30389 { \dim_to_fp:n {#2} / \dim_to_fp:n { \l__box_right_dim } } 30389
30390 \fp_set_eq:NN \l__box_scale_y_fp \l__box_scale_x_fp 30390
30391 \__box_resize:N #1 30391
30392 } 30392
30393 } 30393
30394 \cs_new_protected:Npn \box_resize_to_wd_and_ht:Nnn #1#2#3 30394
30395 { \__box_resize_to_wd_and_ht:NnnN #1 {#2} {#3} \hbox_set:Nn } 30395
30396 \cs_generate_variant:Nn \box_resize_to_wd_and_ht:Nnn { c } 30396
30397 \cs_new_protected:Npn \box_gresize_to_wd_and_ht:Nnn #1#2#3 30397
30398 { \__box_resize_to_wd_and_ht:NnnN #1 {#2} {#3} \hbox_gset:Nn } 30398
30399 \cs_generate_variant:Nn \box_gresize_to_wd_and_ht:Nnn { c } 30399
30400 \cs_new_protected:Npn \__box_resize_to_wd_and_ht:NnnN #1#2#3#4 30400
30401 { 30401
30402 #4 #1 30402
30403 { 30403
30404 \__box_resize_set_corners:N #1 30404
30405 \fp_set:Nn \l__box_scale_x_fp 30405
30406 { \dim_to_fp:n {#2} / \dim_to_fp:n { \l__box_right_dim } } 30406
30407 \fp_set:Nn \l__box_scale_y_fp 30407
30408 { 30408
30409 \dim_to_fp:n {#3} 30409
30410 / \dim_to_fp:n { \l__box_top_dim } 30410
30411 } 30411
30412 \__box_resize:N #1 30412
30413 } 30413
30414 } 30414
30415 \cs_new_protected:Npn \box_scale:Nnn #1#2#3 30415
30416 { \__box_scale:NnnN #1 {#2} {#3} \hbox_set:Nn } 30416
30417 \cs_generate_variant:Nn \box_scale:Nnn { c } 30417
30418 \cs_new_protected:Npn \box_gscale:Nnn #1#2#3 30418
30419 { \__box_scale:NnnN #1 {#2} {#3} \hbox_gset:Nn } 30419
30420 \cs_generate_variant:Nn \box_gscale:Nnn { c } 30420
30421 \cs_new_protected:Npn \__box_scale:NnnN #1#2#3#4 30421
30422 { 30422
30423 #4 #1 30423
30424 { 30424
30425 \fp_set:Nn \l__box_scale_x_fp {#2} 30425
30426 \fp_set:Nn \l__box_scale_y_fp {#3} 30426
30427 \__box_scale:N #1 30427
30428 } 30428
30429 } 30429
30430 \cs_new_protected:Npn \__box_scale:N #1 30430
30431 { 30431
30432 \dim_set:Nn \l__box_top_dim { \box_ht:N #1 } 30432
30433 \dim_set:Nn \l__box_bottom_dim { -\box_dp:N #1 } 30433
30434 \dim_set:Nn \l__box_right_dim { \box_wd:N #1 } 30434
```

```
30435 \dim_zero:N \l__box_left_dim 30435
30436 \dim_set:Nn \l__box_top_new_dim 30436
30437 { \fp_abs:n { \l__box_scale_y_fp } \l__box_top_dim } 30437
30438 \dim_set:Nn \l__box_bottom_new_dim 30438
30439 { \fp_abs:n { \l__box_scale_y_fp } \l__box_bottom_dim } 30439
30440 \dim_set:Nn \l__box_right_new_dim 30440
30441 { \fp_abs:n { \l__box_scale_x_fp } \l__box_right_dim } 30441
30442 \__box_resize_common:N #1 30442
30443 } 30443
30444 \cs_new_protected:Npn \box_autosize_to_wd_and_ht:Nnn #1#2#3 30444
30445 { \__box_autosize:NnnnN #1 {#2} {#3} { \box_ht:N #1 } \hbox_set:Nn } 30445
30446 \cs_generate_variant:Nn \box_autosize_to_wd_and_ht:Nnn { c } 30446
30447 \cs_new_protected:Npn \box_gautosize_to_wd_and_ht:Nnn #1#2#3 30447
30448 { \__box_autosize:NnnnN #1 {#2} {#3} { \box_ht:N #1 } \hbox_gset:Nn } 30448
30449 \cs_generate_variant:Nn \box_gautosize_to_wd_and_ht:Nnn { c } 30449
30450 \cs_new_protected:Npn \box_autosize_to_wd_and_ht_plus_dp:Nnn #1#2#3 30450
30451 { 30451
30452 \__box_autosize:NnnnN #1 {#2} {#3} { \box_ht:N #1 + \box_dp:N #1 } 30452
30453 \hbox_set:Nn 30453
30454 } 30454
30455 \cs_generate_variant:Nn \box_autosize_to_wd_and_ht_plus_dp:Nnn { c } 30455
30456 \cs_new_protected:Npn \box_gautosize_to_wd_and_ht_plus_dp:Nnn #1#2#3 30456
30457 { 30457
30458 \__box_autosize:NnnnN #1 {#2} {#3} { \box_ht:N #1 + \box_dp:N #1 } 30458
30459 \hbox_gset:Nn 30459
30460 } 30460
30461 \cs_generate_variant:Nn \box_gautosize_to_wd_and_ht_plus_dp:Nnn { c } 30461
30462 \cs_new_protected:Npn \__box_autosize:NnnnN #1#2#3#4#5 30462
30463 { 30463
30464 #5 #1 30464
30465 { 30465
30466 \fp_set:Nn \l__box_scale_x_fp { ( \dim_to_fp:n {#2} ) / \box_wd:N #1 } 30466
30467 \fp_set:Nn \l__box_scale_y_fp 30467
30468 { ( \dim_to_fp:n {#3} ) / ( \dim_to_fp:n {#4} ) } 30468
30469 \fp_compare:nNnTF \l__box_scale_x_fp > \l__box_scale_y_fp 30469
30470 { \fp_set_eq:NN \l__box_scale_x_fp \l__box_scale_y_fp } 30470
30471 { \fp_set_eq:NN \l__box_scale_y_fp \l__box_scale_x_fp } 30471
30472 \__box_scale:N #1 30472
30473 } 30473
30474 } 30474
30475 \cs_new_protected:Npn \__box_resize_common:N #1 30475
30476 { 30476
30477 \hbox_set:Nn \l__box_internal_box 30477
30478 { 30478
30479 \__box_backend_scale:Nnn 30479
30480 #1 30480
```



```

30481         \l__box_scale_x_fp                                30481
30482         \l__box_scale_y_fp                                30482
30483     }                                                        30483
30484 \fp_compare:nNnTF \l__box_scale_y_fp > \c_zero_fp          30484
30485 {                                                            30485
30486     \box_set_ht:Nn \l__box_internal_box { \l__box_top_new_dim } 30486
30487     \box_set_dp:Nn \l__box_internal_box { -\l__box_bottom_new_dim } 30487
30488 }                                                            30488
30489 {                                                            30489
30490     \box_set_dp:Nn \l__box_internal_box { \l__box_top_new_dim } 30490
30491     \box_set_ht:Nn \l__box_internal_box { -\l__box_bottom_new_dim } 30491
30492 }                                                            30492
30493 \fp_compare:nNnTF \l__box_scale_x_fp < \c_zero_fp          30493
30494 {                                                            30494
30495     \hbox_to_wd:nn { \l__box_right_new_dim }                 30495
30496     {                                                         30496
30497         \__kernel_kern:n { \l__box_right_new_dim }           30497
30498         \box_use_drop:N \l__box_internal_box                 30498
30499         \tex_hss:D                                           30499
30500     }                                                         30500
30501 }                                                            30501
30502 {                                                            30502
30503     \box_set_wd:Nn \l__box_internal_box { \l__box_right_new_dim } 30503
30504     \hbox:n                                                  30504
30505     {                                                         30505
30506         \__kernel_kern:n { Opt }                             30506
30507         \box_use_drop:N \l__box_internal_box                 30507
30508         \tex_hss:D                                           30508
30509     }                                                         30509
30510 }                                                            30510
30511 }                                                            30511
30512 \cs_new_protected:Npn \box_set_clipped:N #1                 30512
30513 { \hbox_set:Nn #1 { \__box_backend_clip:N #1 } }           30513
30514 \cs_generate_variant:Nn \box_set_clipped:N { c }           30514
30515 \cs_new_protected:Npn \box_gset_clipped:N #1               30515
30516 { \hbox_gset:Nn #1 { \__box_backend_clip:N #1 } }         30516
30517 \cs_generate_variant:Nn \box_gset_clipped:N { c }           30517
30518 \cs_new_protected:Npn \box_set_trim:Nnnnn #1#2#3#4#5       30518
30519 { \__box_set_trim:NnnnnN #1 {#2} {#3} {#4} {#5} \box_set_eq:NN } 30519
30520 \cs_generate_variant:Nn \box_set_trim:Nnnnn { c }           30520
30521 \cs_new_protected:Npn \box_gset_trim:Nnnnn #1#2#3#4#5      30521
30522 { \__box_set_trim:NnnnnN #1 {#2} {#3} {#4} {#5} \box_gset_eq:NN } 30522
30523 \cs_generate_variant:Nn \box_gset_trim:Nnnnn { c }           30523
30524 \cs_new_protected:Npn \__box_set_trim:NnnnnN #1#2#3#4#5#6  30524
30525 {                                                            30525
30526     \hbox_set:Nn \l__box_internal_box                        30526

```



```
30527 { 30527
30528 \__kernel_kern:n { -#2 } 30528
30529 \box_use:N #1 30529
30530 \__kernel_kern:n { -#4 } 30530
30531 } 30531
30532 \dim_compare:nNnTF { \box_dp:N #1 } > {#3} 30532
30533 { 30533
30534 \hbox_set:Nn \l__box_internal_box 30534
30535 { 30535
30536 \box_move_down:nn \c_zero_dim 30536
30537 { \box_use_drop:N \l__box_internal_box } 30537
30538 } 30538
30539 \box_set_dp:Nn \l__box_internal_box { \box_dp:N #1 - (#3) } 30539
30540 } 30540
30541 { 30541
30542 \hbox_set:Nn \l__box_internal_box 30542
30543 { 30543
30544 \box_move_down:nn { (#3) - \box_dp:N #1 } 30544
30545 { \box_use_drop:N \l__box_internal_box } 30545
30546 } 30546
30547 \box_set_dp:Nn \l__box_internal_box \c_zero_dim 30547
30548 } 30548
30549 \dim_compare:nNnTF { \box_ht:N \l__box_internal_box } > {#5} 30549
30550 { 30550
30551 \hbox_set:Nn \l__box_internal_box 30551
30552 { 30552
30553 \box_move_up:nn \c_zero_dim 30553
30554 { \box_use_drop:N \l__box_internal_box } 30554
30555 } 30555
30556 \box_set_ht:Nn \l__box_internal_box 30556
30557 { \box_ht:N \l__box_internal_box - (#5) } 30557
30558 } 30558
30559 { 30559
30560 \hbox_set:Nn \l__box_internal_box 30560
30561 { 30561
30562 \box_move_up:nn { (#5) - \box_ht:N \l__box_internal_box } 30562
30563 { \box_use_drop:N \l__box_internal_box } 30563
30564 } 30564
30565 \box_set_ht:Nn \l__box_internal_box \c_zero_dim 30565
30566 } 30566
30567 #6 #1 \l__box_internal_box 30567
30568 } 30568
30569 \cs_new_protected:Npn \box_set_viewport:Nnnnn #1#2#3#4#5 30569
30570 { \__box_set_viewport:NnnnnN #1 {#2} {#3} {#4} {#5} \box_set_eq:NN } 30570
30571 \cs_generate_variant:Nn \box_set_viewport:Nnnnn { c } 30571
30572 \cs_new_protected:Npn \box_gset_viewport:Nnnnn #1#2#3#4#5 30572
```

```
30573 { \__box_set_viewport:NnnnnN #1 {#2} {#3} {#4} {#5} \box_gset_eq:NN } 30573
30574 \cs_generate_variant:Nn \box_gset_viewport:Nnnnn { c } 30574
30575 \cs_new_protected:Npn \__box_set_viewport:NnnnnN #1#2#3#4#5#6 30575
30576 { 30576
30577 \hbox_set:Nn \l__box_internal_box 30577
30578 { 30578
30579 \__kernel_kern:n { -#2 } 30579
30580 \box_use:N #1 30580
30581 \__kernel_kern:n { #4 - \box_wd:N #1 } 30581
30582 } 30582
30583 \dim_compare:nNnTF {#3} < \c_zero_dim 30583
30584 { 30584
30585 \hbox_set:Nn \l__box_internal_box 30585
30586 { 30586
30587 \box_move_down:nn \c_zero_dim 30587
30588 { \box_use_drop:N \l__box_internal_box } 30588
30589 } 30589
30590 \box_set_dp:Nn \l__box_internal_box { - \__box_dim_eval:n {#3} } 30590
30591 } 30591
30592 { 30592
30593 \hbox_set:Nn \l__box_internal_box 30593
30594 { \box_move_down:nn {#3} { \box_use_drop:N \l__box_internal_box } } 30594
30595 \box_set_dp:Nn \l__box_internal_box \c_zero_dim 30595
30596 } 30596
30597 \dim_compare:nNnTF {#5} > \c_zero_dim 30597
30598 { 30598
30599 \hbox_set:Nn \l__box_internal_box 30599
30600 { 30600
30601 \box_move_up:nn \c_zero_dim 30601
30602 { \box_use_drop:N \l__box_internal_box } 30602
30603 } 30603
30604 \box_set_ht:Nn \l__box_internal_box 30604
30605 { 30605
30606 (#5) 30606
30607 \dim_compare:nNnT {#3} > \c_zero_dim 30607
30608 { - (#3) } 30608
30609 } 30609
30610 } 30610
30611 { 30611
30612 \hbox_set:Nn \l__box_internal_box 30612
30613 { 30613
30614 \box_move_up:nn { - \__box_dim_eval:n {#5} } 30614
30615 { \box_use_drop:N \l__box_internal_box } 30615
30616 } 30616
30617 \box_set_ht:Nn \l__box_internal_box \c_zero_dim 30617
30618 } 30618
```

```
30619 #6 #1 \l__box_internal_box 30619
30620 } 30620
30621 %% File: l3color.dtx 30621
30622 \cs_new_eq:NN \color_group_begin: \group_begin: 30622
30623 \cs_new_eq:NN \color_group_end: \group_end: 30623
30624 \cs_new_protected:Npn \color_ensure_current: 30624
30625 { \__color_select:N \l__color_current_tl } 30625
30626 \scan_new:N \s__color_stop 30626
30627 \cs_new_protected:Npn \__color_select:N #1 30627
30628 { 30628
30629 \exp_after:wN \__color_select:nn #1 30629
30630 \group_insert_after:N \__color_backend_reset: 30630
30631 } 30631
30632 \cs_new_protected:Npn \__color_select_math:N #1 30632
30633 { \exp_after:wN \__color_select:nn #1 } 30633
30634 \cs_new_protected:Npn \__color_select:nn #1#2 30634
30635 { \use:c { __color_backend_select_ #1 :n } {#2} } 30635
30636 \tl_new:N \l__color_current_tl 30636
30637 \tl_set:Nn \l__color_current_tl { { gray } { 0 } } 30637
30638 \int_new:N \l__color_internal_int 30638
30639 \tl_new:N \l__color_internal_tl 30639
30640 \scan_new:N \s__color_mark 30640
30641 \bool_new:N \l__color_ignore_error_bool 30641
30642 \prg_new_conditional:Npnn \color_if_exist:n #1 { p , T, F, TF } 30642
30643 { 30643
30644 \prop_if_exist:cTF { l__color_named_ #1 _prop } 30644
30645 { 30645
30646 \prop_if_empty:cTF { l__color_named_ #1 _prop } 30646
30647 \prg_return_false: 30647
30648 \prg_return_true: 30648
30649 } 30649
30650 \prg_return_false: 30650
30651 } 30651
30652 \cs_new:Npn \__color_model:N #1 { \exp_after:wN \use_i:nn #1 } 30652
30653 \cs_new:Npn \__color_values:N #1 { \exp_after:wN \use_ii:nn #1 } 30653
30654 \cs_new_protected:Npn \__color_extract:nNN #1#2#3 30654
30655 { 30655
30656 \tl_set_eq:Nc #2 { l__color_named_ #1 _tl } 30656
30657 \prop_get:cVN { l__color_named_ #1 _prop } #2 #3 30657
30658 } 30658
30659 \cs_generate_variant:Nn \__color_extract:nNN { V } 30659
30660 \cs_new_protected:Npn \__color_convert:nnN #1#2#3 30660
30661 { \__color_convert:nnVN {#1} {#2} #3 #3 } 30661
30662 \cs_generate_variant:Nn \__color_convert:nnN { VV } 30662
30663 \cs_generate_variant:Nn \exp_last_unbraced:Nf { c } 30663
30664 \cs_new_protected:Npn \__color_convert:nnnN #1#2#3#4 30664
```

```

30665 {
30666 \tl_set:Nc #4
30667 {
30668 \cs_if_exist_use:cTF { __color_convert_ #1 _ #2 :w }
30669 { #3 \s__color_stop }
30670 {
30671 \cs_if_exist:cTF { __color_convert_ \use:c { c__color_fallback_ #1 _tl } _
30672 #2 :w }
30673 {
30674 \exp_last_unbraced:cf
30675 { __color_convert_ \use:c { c__color_fallback_ #1 _tl } _ #2 :w }
30676 { \use:c { __color_convert_ #1 _ \use:c { c__color_fallback_ #1 _tl }
30677 :w } #3 \s__color_stop }
30678 \s__color_stop
30679 }
30680 {
30681 \exp_last_unbraced:cf
30682 { __color_convert_ \use:c { c__color_fallback_ #2 _tl } _ #2 :w }
30683 {
30684 \cs_if_exist_use:cTF { __color_convert_ #1 _ \use:c {
30685 c__color_fallback_ #2 _tl } :w }
30686 { #3 \s__color_stop }
30687 {
30688 \exp_last_unbraced:cf
30689 { __color_convert_ \use:c { c__color_fallback_ #1 _tl } _
30690 \use:c { c__color_fallback_ #2 _tl } :w }
30691 { \use:c { __color_convert_ #1 _ \use:c { c__color_fallback_
30692 #1 _tl } :w } #3 \s__color_stop }
30693 \s__color_stop
30694 }
30695 }
30696 }
30697 \cs_generate_variant:Nn \__color_convert:nnnN { nV , nnV }
30698 \cs_new:Npn \__color_convert_gray_gray:w #1 \s__color_stop
30699 { #1 }
30700 \cs_new:Npn \__color_convert_gray_rgb:w #1 \s__color_stop
30701 { #1 ~ #1 ~ #1 }
30702 \cs_new:Npn \__color_convert_gray_cmyk:w #1 \s__color_stop
30703 { 0 ~ 0 ~ 0 ~ \fp_eval:n { 1 - #1 } }
30704 \cs_new:Npn \__color_convert_rgb_gray:w #1 ~ #2 ~ #3 \s__color_stop
30705 { \fp_eval:n { 0.3 * #1 + 0.59 * #2 + 0.11 * #3 } }
30706 \cs_new:Npn \__color_convert_rgb_rgb:w #1 \s__color_stop

```

```

30706 { #1 }
30707 \cs_new:Npn \__color_convert_rgb_cmyk:w #1 ~ #2 ~ #3 \s__color_stop
30708 {
30709     \exp_args:Neee \__color_convert_rgb_cmyk:nnn
30710     { \fp_eval:n { 1 - #1 } }
30711     { \fp_eval:n { 1 - #2 } }
30712     { \fp_eval:n { 1 - #3 } }
30713 }
30714 \cs_new:Npn \__color_convert_rgb_cmyk:nnn #1#2#3
30715 {
30716     \exp_args:Ne \__color_convert_rgb_cmyk:nnnn
30717     { \fp_eval:n { min( #1, #2 , #3 ) } } {#1} {#2} {#3}
30718 }
30719 \cs_new:Npn \__color_convert_rgb_cmyk:nnnn #1#2#3#4
30720 {
30721     \fp_eval:n { min ( 1 , max ( 0 , #2 - #1 ) ) } \c_space_tl
30722     \fp_eval:n { min ( 1 , max ( 0 , #3 - #1 ) ) } \c_space_tl
30723     \fp_eval:n { min ( 1 , max ( 0 , #4 - #1 ) ) } \c_space_tl
30724     #1
30725 }
30726 \cs_new:Npn \__color_convert_cmyk_gray:w #1 ~ #2 ~ #3 ~ #4 \s__color_stop
30727 { \fp_eval:n { 1 - min ( 1 , 0.3 * #1 + 0.59 * #2 + 0.11 * #3 + #4 ) } }
30728 \cs_new:Npn \__color_convert_cmyk_rgb:w #1 ~ #2 ~ #3 ~ #4 \s__color_stop
30729 {
30730     \fp_eval:n { 1 - min ( 1 , #1 + #4 ) } \c_space_tl
30731     \fp_eval:n { 1 - min ( 1 , #2 + #4 ) } \c_space_tl
30732     \fp_eval:n { 1 - min ( 1 , #3 + #4 ) }
30733 }
30734 \cs_new:Npn \__color_convert_cmyk_cmyk:w #1 \s__color_stop
30735 { #1 }
30736 \tl_new:N \l__color_model_tl
30737 \tl_new:N \l__color_value_tl
30738 \tl_new:N \l__color_next_model_tl
30739 \tl_new:N \l__color_next_value_tl
30740 \cs_new_protected:Npe \__color_parse:nN #1#2
30741 {
30742     \tl_set:Ne \exp_not:c { l__color_named_ . _tl }
30743     { \exp_not:N \__color_model:N \exp_not:N \l__color_current_tl }
30744     \prop_put:NVe \exp_not:c { l__color_named_ . _prop }
30745     \exp_not:c { l__color_named_ . _tl }
30746     { \exp_not:N \__color_values:N \exp_not:N \l__color_current_tl }
30747     \exp_not:N \exp_args:Ne \exp_not:N \__color_parse_aux:nN
30748     { \exp_not:N \tl_to_str:n {#1} } #2
30749 }
30750 \cs_new_protected:Npn \__color_parse_aux:nN #1#2
30751 {

```

```
30752 \color_if_exist:nTF {#1}
30753 { \__color_parse_set_eq:Nn #2 {#1} }
30754 { \__color_parse:Nw #2#1 ! \s__color_stop }
30755 \__color_check_model:N #2
30756 }
30757 \cs_new_protected:Npn \__color_parse_set_eq:Nn #1#2
30758 {
30759 \tl_if_empty:NTF \l_color_fixed_model_tl
30760 { \exp_args:Nv \__color_parse_set_eq:nNn { l__color_named_ #2 _tl } }
30761 { \exp_args:NV \__color_parse_set_eq:nNn \l_color_fixed_model_tl }
30762 #1 {#2}
30763 }
30764 \cs_new_protected:Npn \__color_parse_set_eq:nNn #1#2#3
30765 {
30766 \prop_get:cnNTF
30767 { l__color_named_ #3 _prop } {#1}
30768 \l__color_value_tl
30769 { \tl_set:Ne #2 { {#1} { \l__color_value_tl } } }
30770 {
30771 \tl_set_eq:Nc \l__color_model_tl { l__color_named_ #3 _tl }
30772 \prop_get:cVN { l__color_named_ #3 _prop } \l__color_model_tl
30773 \l__color_value_tl
30774 \__color_convert:nnN
30775 \l__color_model_tl {#1} \l__color_value_tl
30776 \tl_set:Ne #2
30777 {
30778 {#1}
30779 { \l__color_value_tl }
30780 }
30781 }
30782 }
30783 \cs_new_protected:Npn \__color_parse:Nw #1#2 ! #3 \s__color_stop
30784 {
30785 \color_if_exist:nTF {#2}
30786 {
30787 \tl_if_blank:nTF {#3}
30788 { \__color_parse_set_eq:Nn #1 {#2} }
30789 { \__color_parse_loop_init:Nnn #1 {#2} {#3} }
30790 }
30791 {
30792 \msg_error:nnn { color } { unknown-color } {#2}
30793 \tl_set:Nn \l__color_current_tl { { gray } { 0 } }
30794 }
30795 }
30796 \cs_new_protected:Npn \__color_parse_loop_init:Nnn #1#2#3
30797 {
```

```
30798 \group_begin: 30798
30799 \__color_extract:nNN {#2} \l__color_model_tl \l__color_value_tl 30799
30800 \__color_parse_loop:w #3 ! ! ! ! \s__color_stop 30800
30801 \tl_set:Nx \l__color_internal_tl 30801
30802 { { \l__color_model_tl } { \l__color_value_tl } } 30802
30803 \exp_args:NNNV \group_end: 30803
30804 \tl_set:Nn #1 \l__color_internal_tl 30804
30805 } 30805
30806 \cs_new_protected:Npn \__color_parse_loop:w #1 ! #2 ! #3 ! #4 ! #5 \s__color_stop 30806
30807 { 30807
30808 \tl_if_blank:nF {#1} 30808
30809 { 30809
30810 \bool_lazy_and:nnTF 30810
30811 { \fp_compare_p:nNn {#1} > { 0 } } 30811
30812 { \fp_compare_p:nNn {#1} < { 100 } } 30812
30813 { 30813
30814 \use:e 30814
30815 { 30815
30816 \__color_parse_loop:nn {#1} 30816
30817 { \tl_if_blank:nTF {#2} { white } {#2} } 30817
30818 } 30818
30819 } 30819
30820 { \__color_parse_loop_check:nn {#1} {#2} } 30820
30821 } 30821
30822 \tl_if_blank:nF {#3} 30822
30823 { \__color_parse_loop:w #3 ! #4 ! #5 \s__color_stop } 30823
30824 \__color_parse_end: 30824
30825 } 30825
30826 \cs_new_protected:Npn \__color_parse_loop_check:nn #1#2 30826
30827 { 30827
30828 \bool_if:NF \l__color_ignore_error_bool 30828
30829 { 30829
30830 \bool_lazy_or:nnT 30830
30831 { \fp_compare_p:nNn {#1} < { 0 } } 30831
30832 { \fp_compare_p:nNn {#1} > { 100 } } 30832
30833 { \msg_error:nnnnn { color } { out-of-range } {#1} { 0 } { 100 } } 30833
30834 } 30834
30835 \fp_compare:nNnF {#1} > \c_zero_fp 30835
30836 { 30836
30837 \tl_if_blank:nTF {#2} 30837
30838 { \__color_extract:nNN { white } } 30838
30839 { \__color_extract:nNN {#2} } 30839
30840 \l__color_model_tl \l__color_value_tl 30840
30841 } 30841
30842 } 30842
30843 \cs_new_protected:Npn \__color_parse_loop:nn #1#2 30843
```



```
30844 { 30844
30845 \color_if_exist:nTF {#2} 30845
30846 { 30846
30847 \__color_extract:nNN {#2} \l__color_next_model_tl \l__color_next_value_tl 30847
30848 \tl_if_eq:NNF \l__color_model_tl \l__color_next_model_tl 30848
30849 { 30849
30850 \str_if_eq:VnTF \l__color_model_tl { gray } 30850
30851 { \__color_parse_gray:n {#2} } 30851
30852 { \__color_parse_std:n {#2} } 30852
30853 } 30853
30854 \tl_set:Ne \l__color_value_tl 30854
30855 { 30855
30856 \__color_parse_mix:NVVn 30856
30857 \l__color_model_tl \l__color_value_tl \l__color_next_value_tl {#1} 30857
30858 } 30858
30859 } 30859
30860 { 30860
30861 \msg_error:nnn { color } { unknown-color } {#2} 30861
30862 \__color_extract:nNN { black } \l__color_model_tl \l__color_value_tl 30862
30863 \__color_parse_break:w 30863
30864 } 30864
30865 } 30865
30866 \cs_new_protected:Npn \__color_parse_gray:n #1 30866
30867 { 30867
30868 \tl_set_eq:NN \l__color_model_tl \l__color_next_model_tl 30868
30869 \tl_set:Nn \l__color_next_model_tl { gray } 30869
30870 \exp_args:NnV \__color_convert:nnN { gray } \l__color_model_tl 30870
30871 \l__color_value_tl 30871
30872 \prop_get:cVN { l__color_named_ #1 _prop } \l__color_model_tl 30872
30873 \l__color_next_value_tl 30873
30874 } 30874
30875 \cs_new_protected:Npn \__color_parse_std:n #1 30875
30876 { 30876
30877 \prop_get:cVNF { l__color_named_ #1 _prop } 30877
30878 \l__color_model_tl 30878
30879 \l__color_next_value_tl 30879
30880 { 30880
30881 \__color_convert:VVN 30881
30882 \l__color_next_model_tl 30882
30883 \l__color_model_tl 30883
30884 \l__color_next_value_tl 30884
30885 } 30885
30886 } 30886
30887 \cs_new_protected:Npn \__color_parse_break:w #1 \__color_parse_end: { } 30887
30888 \cs_new_protected:Npn \__color_parse_end: { } 30888
30889 \cs_new:Npn \__color_parse_mix:Nnnn #1#2#3#4 30889
```

```
30890 {
30891     \exp_args:Nf \__color_parse_mix:nNnn
30892     { \fp_eval:n { #4 / 100 } }
30893     #1 {#2} {#3}
30894 }
30895 \cs_generate_variant:Nn \__color_parse_mix:Nnnn { NVV }
30896 \cs_new:Npn \__color_parse_mix:nNnn #1#2#3#4
30897 {
30898     \use:c { __color_parse_mix_ #2 :nw } {#1}
30899     #3 \s__color_mark #4 \s__color_stop
30900 }
30901 \cs_new:Npn \__color_parse_mix_gray:nw #1#2 \s__color_mark #3 \s__color_stop
30902 { \fp_eval:n { #2 * #1 + #3 * ( 1 - #1 ) } }
30903 \cs_new:Npn \__color_parse_mix_rgb:nw
30904 #1#2 ~ #3 ~ #4 \s__color_mark #5 ~ #6 ~ #7 \s__color_stop
30905 {
30906     \fp_eval:n { #2 * #1 + #5 * ( 1 - #1 ) } \c_space_tl
30907     \fp_eval:n { #3 * #1 + #6 * ( 1 - #1 ) } \c_space_tl
30908     \fp_eval:n { #4 * #1 + #7 * ( 1 - #1 ) }
30909 }
30910 \cs_new:Npn \__color_parse_mix_cmyk:nw
30911 #1#2 ~ #3 ~ #4 ~ #5 \s__color_mark #6 ~ #7 ~ #8 ~ #9 \s__color_stop
30912 {
30913     \fp_eval:n { #2 * #1 + #6 * ( 1 - #1 ) } \c_space_tl
30914     \fp_eval:n { #3 * #1 + #7 * ( 1 - #1 ) } \c_space_tl
30915     \fp_eval:n { #4 * #1 + #8 * ( 1 - #1 ) } \c_space_tl
30916     \fp_eval:n { #5 * #1 + #9 * ( 1 - #1 ) }
30917 }
30918 \cs_new:Npn \__color_parse_model_gray:w #1 , #2 \s__color_stop
30919 { { gray } { \__color_parse_number:n {#1} } }
30920 \cs_new:Npn \__color_parse_model_rgb:w #1 , #2 , #3 , #4 \s__color_stop
30921 {
30922     { rgb }
30923     {
30924         \__color_parse_number:n {#1} ~
30925         \__color_parse_number:n {#2} ~
30926         \__color_parse_number:n {#3}
30927     }
30928 }
30929 \cs_new:Npn \__color_parse_model_cmyk:w #1 , #2 , #3 , #4 , #5 \s__color_stop
30930 {
30931     { cmyk }
30932     {
30933         \__color_parse_number:n {#1} ~
30934         \__color_parse_number:n {#2} ~
30935         \__color_parse_number:n {#3} ~
```

```
30936 \__color_parse_number:n {#4} 30936
30937 } 30937
30938 } 30938
30939 \cs_new:Npn \__color_parse_number:n #1 30939
30940 { \__color_parse_number:w #1 . 0 . \s__color_stop } 30940
30941 \cs_new:Npn \__color_parse_number:w #1 . #2 . #3 \s__color_stop 30941
30942 { \tl_if_blank:nTF {#1} { 0 } {#1} . #2 } 30942
30943 \cs_new:Npn \__color_parse_model_Gray:w #1 , #2 \s__color_stop 30943
30944 { { gray } { \fp_eval:n { #1 / 15 } } } 30944
30945 \cs_new:Npn \__color_parse_model_hsb:w #1 , #2 , #3 , #4 \s__color_stop 30945
30946 { \__color_parse_model_hsb:nnn {#1} {#2} {#3} } 30946
30947 \cs_new:Npn \__color_parse_model_Hsb:w #1 , #2 , #3 , #4 \s__color_stop 30947
30948 { 30948
30949 \exp_args:Ne \__color_parse_model_hsb:nnn { \fp_eval:n { #1 / 360 } } 30949
30950 {#2} {#3} 30950
30951 } 30951
30952 \cs_new:Npn \__color_parse_model_hsb:nnn #1#2#3 30952
30953 { 30953
30954 { rgb } 30954
30955 { 30955
30956 \exp_args:Ne \__color_parse_model_hsb_aux:nnn 30956
30957 { \fp_eval:n { 6 * (#1) } } {#2} {#3} 30957
30958 } 30958
30959 } 30959
30960 \cs_new:Npn \__color_parse_model_hsb_aux:nnn #1#2#3 30960
30961 { 30961
30962 \exp_args:Nee \__color_parse_model_hsb_aux:nnnn 30962
30963 { \fp_eval:n { floor(#1) } } { \fp_eval:n { #1 - floor(#1) } } 30963
30964 {#2} {#3} 30964
30965 } 30965
30966 \cs_new:Npn \__color_parse_model_hsb_aux:nnnn #1#2#3#4 30966
30967 { 30967
30968 \use:e 30968
30969 { 30969
30970 \exp_not:N \__color_parse_model_hsb_aux:nnnnn 30970
30971 { \__color_parse_number:n {#4} } 30971
30972 { \fp_eval:n { round(#4 * (1 - #3) ,5) } } 30972
30973 { \fp_eval:n { round(#4 * (1 - #3 * #2) ,5) } } 30973
30974 { \fp_eval:n { round(#4 * (1 - #3 * (1 - #2) ) ,5) } } 30974
30975 {#1} 30975
30976 } 30976
30977 } 30977
30978 \cs_new:Npn \__color_parse_model_hsb_aux:nnnnn #1#2#3#4#5 30978
30979 { \use:c { __color_parse_model_hsb_ #5 :nnnn } {#1} {#2} {#3} {#4} } 30979
30980 \cs_new:cpn { __color_parse_model_hsb_0:nnnn } #1#2#3#4 { #1 ~ #4 ~ #2 } 30980
30981 \cs_new:cpn { __color_parse_model_hsb_1:nnnn } #1#2#3#4 { #3 ~ #1 ~ #2 } 30981
```

```
30982 \cs_new:cpn { __color_parse_model_hsb_2:nnnn } #1#2#3#4 { #2 ~ #1 ~ #4 } 30982
30983 \cs_new:cpn { __color_parse_model_hsb_3:nnnn } #1#2#3#4 { #2 ~ #3 ~ #1 } 30983
30984 \cs_new:cpn { __color_parse_model_hsb_4:nnnn } #1#2#3#4 { #4 ~ #2 ~ #1 } 30984
30985 \cs_new:cpn { __color_parse_model_hsb_5:nnnn } #1#2#3#4 { #1 ~ #2 ~ #3 } 30985
30986 \cs_new:cpn { __color_parse_model_hsb_6:nnnn } #1#2#3#4 { #1 ~ #2 ~ #2 } 30986
30987 \cs_new:Npn \__color_parse_model_HSB:w #1 , #2 , #3 , #4 \s__color_stop 30987
30988 { 30988
30989   \exp_args:Neee \__color_parse_model_hsb:nnn 30989
30990   { \fp_eval:n { round((#1) / 240,5) } } 30990
30991   { \fp_eval:n { round((#2) / 240,5) } } 30991
30992   { \fp_eval:n { round((#3) / 240,5) } } 30992
30993 } 30993
30994 \cs_new:Npn \__color_parse_model_HTML:w #1 , #2 \s__color_stop 30994
30995 { \__color_parse_model_HTML_aux:w #1 0 0 0 0 0 0 \s__color_stop } 30995
30996 \cs_new:Npn \__color_parse_model_HTML_aux:w #1#2#3#4#5#6#7 \s__color_stop 30996
30997 { 30997
30998   { rgb } 30998
30999   { 30999
31000     \fp_eval:n { round(\int_from_hex:n {#1#2} / 255,5) } ~ 31000
31001     \fp_eval:n { round(\int_from_hex:n {#3#4} / 255,5) } ~ 31001
31002     \fp_eval:n { round(\int_from_hex:n {#5#6} / 255,5) } 31002
31003   } 31003
31004 } 31004
31005 \cs_new:Npn \__color_parse_model_RGB:w #1 , #2 , #3 , #4 \s__color_stop 31005
31006 { 31006
31007   { rgb } 31007
31008   { 31008
31009     \fp_eval:n { round((#1) / 255,5) } ~ 31009
31010     \fp_eval:n { round((#2) / 255,5) } ~ 31010
31011     \fp_eval:n { round((#3) / 255,5) } 31011
31012   } 31012
31013 } 31013
31014 \cs_new:Npn \__color_parse_model_wave:w #1 , #2 \s__color_stop 31014
31015 { 31015
31016   { rgb } 31016
31017   { 31017
31018     \fp_compare:nNnTF {#1} < { 420 } 31018
31019     { \__color_parse_model_wave_auxi:nn {#1} { 0.3 + 0.7 * (#1 - 380) / 40 } 31019
31020     } 31020
31021     { 31021
31022       \fp_compare:nNnTF {#1} > { 700 } 31022
31023       { \__color_parse_model_wave_auxi:nn {#1} { 0.3 + 0.7 * (#1 - 780) / -80 } } 31023
31024       { \__color_parse_model_wave_auxi:nn {#1} { 1 } } 31024
31025     } 31025
31026   } 31026
31027 } 31027
```

```
31028 \cs_new:Npn \__color_parse_model_wave_auxi:nn #1#2 31028
31029 { 31029
31030 \fp_compare:nNnTF {#1} < { 440 } 31030
31031 { 31031
31032 \__color_parse_model_wave_auxii:nn 31032
31033 { 4 + \__color_parse_model_wave_rho:n { (#1 - 440) / -60 } } 31033
31034 {#2} 31034
31035 } 31035
31036 { 31036
31037 \fp_compare:nNnTF {#1} < { 490 } 31037
31038 { 31038
31039 \__color_parse_model_wave_auxii:nn 31039
31040 { 4 - \__color_parse_model_wave_rho:n { (#1 - 440) / 50 } } 31040
31041 {#2} 31041
31042 } 31042
31043 { 31043
31044 \fp_compare:nNnTF {#1} < { 510 } 31044
31045 { 31045
31046 \__color_parse_model_wave_auxii:nn 31046
31047 { 2 + \__color_parse_model_wave_rho:n { (#1 - 510) / -20 } } 31047
31048 {#2} 31048
31049 } 31049
31050 { 31050
31051 \fp_compare:nNnTF {#1} < { 580 } 31051
31052 { 31052
31053 \__color_parse_model_wave_auxii:nn 31053
31054 { 2 - \__color_parse_model_wave_rho:n { (#1 - 510) / 70 } } 31054
31055 {#2} 31055
31056 } 31056
31057 { 31057
31058 \fp_compare:nNnTF {#1} < { 645 } 31058
31059 { 31059
31060 \__color_parse_model_wave_auxii:nn 31060
31061 { \__color_parse_model_wave_rho:n { (#1 - 645) / -65 } } 31061
31062 {#2} 31062
31063 } 31063
31064 { \__color_parse_model_wave_auxii:nn { 0 } {#2} } 31064
31065 } 31065
31066 } 31066
31067 } 31067
31068 } 31068
31069 } 31069
31070 \cs_new:Npn \__color_parse_model_wave_auxii:nn #1#2 31070
31071 { 31071
31072 \exp_args:Neee \__color_parse_model_hsb_aux:nnn 31072
31073 { \fp_eval:n {#1} } 31073
```

```

31074 { 1 }
31075 { \_color_parse_model_wave_rho:n {#2} }
31076 }
31077 \cs_new:Npn \_color_parse_model_wave_rho:n #1
31078 { \fp_eval:n { min(1, max(0,#1) ) } }
31079 \cs_new:Npn \_color_parse_model_cmy:w #1 , #2 , #3 , #4 \s_color_stop
31080 {
31081 { cmyk }
31082 { \_color_convert_rgb_cmyk:nnn {#1} {#2} {#3} }
31083 }
31084 \cs_new:Npn \_color_parse_model_tHsb:w #1 , #2 , #3 , #4 \s_color_stop
31085 {
31086 \exp_args:Ne \_color_parse_model_hsb:nnn
31087 { \_color_parse_model_tHsb:n {#1} } {#2} {#3}
31088 }
31089 \cs_new:Npn \_color_parse_model_tHsb:n #1
31090 {
31091 \_color_parse_model_tHsb:nw {#1}
31092 0 , 0 ;
31093 60 , 30 ;
31094 120 , 60 ;
31095 180 , 120 ;
31096 210 , 180 ;
31097 240 , 240 ;
31098 360 , 360 ;
31099 \q_recursion_tail , ;
31100 \q_recursion_stop
31101 }
31102 \cs_new:Npn \_color_parse_model_tHsb:nw #1 #2 , #3 ; #4 , #5 ;
31103 {
31104 \quark_if_recursion_tail_stop_do:nn {#4} { 0 }
31105 \fp_compare:nNnTF {#1} > {#4}
31106 { \_color_parse_model_tHsb:nw {#1} #4 , #5 ; }
31107 {
31108 \use_i_delimit_by_q_recursion_stop:nw
31109 { \fp_eval:n { ((#1 - #2) / (#4 - #2) * (#5 - #3) + #3) / 360 } }
31110 }
31111 }
31112 \cs_new:cpn { \_color_parse_model_&spot:w } #1 , #2 \s_color_stop
31113 { { gray } { #1 } }
31114 \tl_new:N \l_color_fixed_model_tl
31115 \cs_new_protected:Npn \_color_check_model:N #1
31116 {
31117 \tl_if_empty:NF \l_color_fixed_model_tl
31118 {
31119 \exp_after:wN \_color_check_model:nn #1

```

```
31120     \tl_if_eq:NNF \l__color_model_tl \l_color_fixed_model_tl 31120
31121     { 31121
31122         \__color_convert:VVN \l__color_model_tl \l_color_fixed_model_tl 31122
31123         \l__color_value_tl 31123
31124     } 31124
31125     \tl_set:Ne #1 31125
31126     { { \l_color_fixed_model_tl } { \l__color_value_tl } } 31126
31127 } 31127
31128 } 31128
31129 \cs_new_protected:Npn \__color_check_model:nn #1#2 31129
31130 { 31130
31131     \tl_set:Nn \l__color_model_tl {#1} 31131
31132     \tl_set:Nn \l__color_value_tl {#2} 31132
31133 } 31133
31134 \cs_new_protected:Npe \__color_finalise_current: 31134
31135 { 31135
31136     \tl_set:Ne \exp_not:c { l__color_named_ . _tl } 31136
31137     { \exp_not:N \__color_model:N \exp_not:N \l__color_current_tl } 31137
31138     \prop_clear:N \exp_not:c { l__color_named_ . _prop } 31138
31139     \prop_put:NVe \exp_not:c { l__color_named_ . _prop } 31139
31140     \exp_not:c { l__color_named_ . _tl } 31140
31141     { \exp_not:N \__color_values:N \exp_not:N \l__color_current_tl } 31141
31142 } 31142
31143 \cs_new_protected:Npn \color_select:n #1 31143
31144 { 31144
31145     \__color_parse:nN {#1} \l__color_current_tl 31145
31146     \__color_finalise_current: 31146
31147     \__color_select:N \l__color_current_tl 31147
31148 } 31148
31149 \cs_generate_variant:Nn \color_select:n { V } 31149
31150 \cs_new_protected:Npn \color_select:nn #1#2 31150
31151 { 31151
31152     \__color_select_main:Nnn \l__color_current_tl {#1} {#2} 31152
31153     \__color_finalise_current: 31153
31154     \__color_select:N \l__color_current_tl 31154
31155 } 31155
31156 \cs_generate_variant:Nn \color_select:nn { nV , V , VV } 31156
31157 \cs_new_protected:Npn \__color_select_main:Nnn #1#2#3 31157
31158 { 31158
31159     \use:e 31159
31160     { 31160
31161         \exp_not:N \__color_select_main:Nw \exp_not:N #1 31161
31162         \exp_not:n {#2} / / \exp_not:N \s__color_mark 31162
31163         #3 / / \exp_not:N \s__color_stop 31163
31164     } 31164
31165 } 31165
```



```
31166 \cs_new_protected:Npn \__color_select_main:Nw 31166
31167 #1 #2 / #3 / #4 \s__color_mark #5 / #6 / #7 \s__color_stop 31167
31168 { 31168
31169 \__color_select:nnN {#2} {#5} #1 31169
31170 \bool_lazy_or:nnF 31170
31171 { \tl_if_empty_p:N \l_color_fixed_model_tl } 31171
31172 { \str_if_eq_p:nV {#2} \l_color_fixed_model_tl } 31172
31173 { \__color_select_loop:Nw #1 #3 / #4 \s__color_mark #6 / #7 \s__color_stop } 31173
31174 } 31174
31175 \cs_new_protected:Npn \__color_select_loop:Nw 31175
31176 #1 #2 / #3 \s__color_mark #4 / #5 \s__color_stop 31176
31177 { 31177
31178 \str_if_eq:nVTF {#2} \l_color_fixed_model_tl 31178
31179 { \__color_select:nnN {#2} {#4} #1 } 31179
31180 { 31180
31181 \tl_if_blank:nTF {#2} 31181
31182 { \exp_after:wN \__color_select_swap:Nnn \exp_after:wN #1 #1 } 31182
31183 { \__color_select_loop:Nw #1 #3 \s__color_mark #5 \s__color_stop } 31183
31184 } 31184
31185 } 31185
31186 \cs_new_protected:Npn \__color_select:nnN #1#2#3 31186
31187 { 31187
31188 \cs_if_exist:cTF { __color_parse_model_ #1 :w } 31188
31189 { 31189
31190 \tl_set:Ne #3 31190
31191 { \use:c { __color_parse_model_ #1 :w } #2 , 0 , 0 , 0 , 0 \s__color_stop } 31191
31192 } 31192
31193 { \msg_error:nnn { color } { unknown-model } {#1} } 31193
31194 } 31194
31195 \cs_new_protected:Npn \__color_select_swap:Nnn #1#2#3 31195
31196 { 31196
31197 \__color_convert:nVnN {#2} \l_color_fixed_model_tl {#3} \l__color_value_tl 31197
31198 \tl_set:Ne #1 31198
31199 { { \l_color_fixed_model_tl } { \l__color_value_tl } } 31199
31200 } 31200
31201 \tl_new:N \l_color_math_active_tl 31201
31202 \tl_set:Nn \l_color_math_active_tl { ' } 31202
31203 \seq_new:N \g__color_math_seq 31203
31204 \cs_new_protected:Npn \color_math:nn #1#2 31204
31205 { 31205
31206 \__color_math:nn {#2} 31206
31207 { \__color_parse:nN {#1} \l__color_current_tl } 31207
31208 } 31208
31209 \cs_new_protected:Npn \color_math:nnn #1#2#3 31209
31210 { 31210
31211 \__color_math:nn {#3} 31211
```

```
31212 { \_color_select_main:Nnn \l\_color_current_tl {#1} {#2} } 31212
31213 } 31213
31214 \cs_new_protected:Npn \_color_math:nn #1#2 31214
31215 { 31215
31216 \seq_gpush:NV \g\_color_math_seq \l\_color_current_tl 31216
31217 #2 31217
31218 \_color_select_math:N \l\_color_current_tl 31218
31219 #1 31219
31220 \_color_math_scan:w 31220
31221 } 31221
31222 \cs_new_protected:Npn \_color_math_scan:w 31222
31223 { 31223
31224 \peek_remove_filler:n 31224
31225 { 31225
31226 \group_align_safe_begin: 31226
31227 \peek_catcode:NTF \c_alignment_token 31227
31228 { 31228
31229 \group_align_safe_end: 31229
31230 \_color_math_scan_end: 31230
31231 } 31231
31232 { 31232
31233 \group_align_safe_end: 31233
31234 \_color_math_scan_auxi: 31234
31235 } 31235
31236 } 31236
31237 } 31237
31238 \cs_new_protected:Npn \_color_math_scan_auxi: 31238
31239 { 31239
31240 \token_case_catcode:NnTF \l_peek_token 31240
31241 { 31241
31242 \c_math_subscript_token { } 31242
31243 \c_math_superscript_token { } 31243
31244 } 31244
31245 { \_color_math_scripts:Nw } 31245
31246 { 31246
31247 \token_case_meaning:NnTF \l_peek_token 31247
31248 { 31248
31249 \tex_limits:D { \tex_limits:D } 31249
31250 \tex_nolimits:D { \tex_nolimits:D } 31250
31251 \tex_displaylimits:D { \tex_displaylimits:D } 31251
31252 } 31252
31253 { \_color_math_scan:w \use_none:n } 31253
31254 { \_color_math_scan_auxii: } 31254
31255 } 31255
31256 } 31256
31257 \cs_new_protected:Npn \_color_math_scan_auxii: 31257
```

```

31258 {
31259     \tl_map_inline:Nn \l_color_math_active_tl
31260     {
31261         \token_if_eq_meaning:NNT \l_peek_token ##1
31262         {
31263             \tl_map_break:n
31264             {
31265                 \use_i:nn
31266                 { \_color_math_scan_auxiii:N ##1 }
31267             }
31268         }
31269         \_color_math_scan_end:
31270     }
31271 }
31272 \cs_new_protected:Npn \_color_math_scan_auxiii:N #1
31273 {
31274     \exp_after:wN \exp_after:wN \exp_after:wN \_color_math_scan:w
31275     \char_generate:nn { `#1 } { 13 }
31276 }
31277 \cs_new_protected:Npn \_color_math_scan_end:
31278 {
31279     \_color_backend_reset:
31280     \seq_gpop:NN \g_color_math_seq \l_color_current_tl
31281 }
31282 \cs_new_protected:Npn \_color_math_scripts:Nw #1
31283 {
31284     #1
31285     \c_group_begin_token
31286     \c_group_begin_token
31287     \seq_get:NN \g_color_math_seq \l_color_current_tl
31288     \_color_select:N \l_color_current_tl
31289     \group_insert_after:N \c_group_end_token
31290     \group_insert_after:N \_color_math_scan:w
31291     \peek_remove_filler:n
31292     {
31293         \peek_catcode_remove:NF \c_group_begin_token
31294         { \_color_math_script_aux:N }
31295     }
31296 }
31297 \cs_new_protected:Npn \_color_math_script_aux:N #1 { #1 \c_group_end_token }
31298 \cs_new_protected:Npn \color_fill:n #1
31299 {
31300     \_color_parse:nN {#1} \l_color_current_tl
31301     \exp_after:wN \_color_draw:nnn \l_color_current_tl { fill }
31302 }
31303 \cs_new_protected:Npn \color_stroke:n #1

```

31304	{	31304
31305	__color_parse:nN {#1} \l__color_current_tl	31305
31306	exp_after:wN __color_draw:nnn \l__color_current_tl { stroke }	31306
31307	}	31307
31308	\cs_new_protected:Npn \color_fill:nn #1#2	31308
31309	{	31309
31310	__color_select_main:Nnn \l__color_current_tl {#1} {#2}	31310
31311	exp_after:wN __color_draw:nnn \l__color_current_tl { fill }	31311
31312	}	31312
31313	\cs_new_protected:Npn \color_stroke:nn #1#2	31313
31314	{	31314
31315	__color_select_main:Nnn \l__color_current_tl {#1} {#2}	31315
31316	exp_after:wN __color_draw:nnn \l__color_current_tl { stroke }	31316
31317	}	31317
31318	\cs_new_protected:Npn __color_draw:nnn #1#2#3	31318
31319	{	31319
31320	use:c { __color_backend_ #3 _ #1 :n } {#2}	31320
31321	exp_args:Nc \group_insert_after:N { __color_backend_ #3 _ reset: }	31321
31322	}	31322
31323	\tl_new:N \l__color_named_tl	31323
31324	\cs_new_protected:Npn \color_set:nn #1#2	31324
31325	{	31325
31326	exp_args:NV __color_set:nnn	31326
31327	\l_color_fixed_model_tl {#1} {#2}	31327
31328	}	31328
31329	\cs_new_protected:Npn __color_set:nnn #1#2#3	31329
31330	{	31330
31331	\tl_clear:N \l_color_fixed_model_tl	31331
31332	__color_set:nn {#2} {#3}	31332
31333	\tl_set:Nn \l_color_fixed_model_tl {#1}	31333
31334	}	31334
31335	\cs_new_protected:Npn __color_set:nn #1#2	31335
31336	{	31336
31337	str_if_eq:nnF {#1} { . }	31337
31338	{	31338
31339	__color_parse:nN {#2} \l__color_named_tl	31339
31340	\tl_clear_new:c { l__color_named_ #1 _tl }	31340
31341	\tl_set:ce { l__color_named_ #1 _tl }	31341
31342	{ __color_model:N \l__color_named_tl }	31342
31343	\prop_clear_new:c { l__color_named_ #1 _prop }	31343
31344	\prop_put:cve { l__color_named_ #1 _prop } { l__color_named_ #1 _tl }	31344
31345	{ __color_values:N \l__color_named_tl }	31345
31346	__color_set:nnw {#1} {#2} #2 ! \s__color_stop	31346
31347	}	31347
31348	}	31348
31349	\cs_new_protected:Npn __color_set:nnw #1#2#3 ! #4 \s__color_stop	31349

```

31350 {
31351     \clist_map_inline:nn { cmyk , gray , rgb }
31352     {
31353         \prop_get:cnNT { l__color_named_ #3 _prop } {##1} \l__color_internal_tl
31354         {
31355             \prop_if_in:cnF { l__color_named_ #1 _prop } {##1}
31356             {
31357                 \group_begin:
31358                     \bool_set_true:N \l__color_ignore_error_bool
31359                     \tl_set:cn { l__color_named_ #3 _tl } {##1}
31360                     \__color_parse:nN {#2} \l__color_internal_tl
31361                 \exp_args:NNNV \group_end:
31362                 \tl_set:Nn \l__color_internal_tl \l__color_internal_tl
31363                 \prop_put:cee { l__color_named_ #1 _prop }
31364                     { \__color_model:N \l__color_internal_tl }
31365                     { \__color_values:N \l__color_internal_tl }
31366             }
31367         }
31368     }
31369 }
31370 \cs_new_protected:Npn \color_set:nnn #1#2#3
31371 {
31372     \str_if_eq:nnF {#1} { . }
31373     {
31374         \tl_clear_new:c { l__color_named_ #1 _tl }
31375         \prop_clear_new:c { l__color_named_ #1 _prop }
31376         \exp_args:Ne \__color_set_aux:nnn { \tl_to_str:n {#2} }
31377             {#1} {#3}
31378     }
31379 }
31380 \cs_new_protected:Npe \__color_set_aux:nnn #1#2#3
31381 {
31382     \exp_not:N \__color_set_colon:nnw {#2} {#3}
31383     #1 \c_colon_str \c_colon_str \exp_not:N \s__color_stop
31384 }
31385 \use:e
31386 {
31387     \cs_new_protected:Npn \exp_not:N \__color_set_colon:nnw
31388         #1#2 #3 \c_colon_str #4 \c_colon_str
31389         #5 \exp_not:N \s__color_stop
31390 }
31391 {
31392     \tl_if_blank:nTF {#4}
31393     { \__color_set_loop:nw {#1} #3 }
31394     { \__color_set_loop:nw {#1} #4 }
31395     // \s__color_mark #2 // \s__color_stop

```

```

31396 } 31396
31397 \cs_new_protected:Npn \__color_set_loop:nw 31397
31398 #1#2 / #3 \s__color_mark #4 / #5 \s__color_stop 31398
31399 { 31399
31400 \tl_if_blank:nF {#2} 31400
31401 { 31401
31402 \__color_select:nnN {#2} {#4} \l__color_named_tl 31402
31403 \tl_set:Ne \l__color_internal_tl { \__color_model:N \l__color_named_tl } 31403
31404 \tl_if_empty:cT { l__color_named_ #1 _tl } 31404
31405 { \tl_set_eq:cN { l__color_named_ #1 _tl } \l__color_internal_tl } 31405
31406 \prop_put:cVe { l__color_named_ #1 _prop } \l__color_internal_tl 31406
31407 { \__color_values:N \l__color_named_tl } 31407
31408 \__color_set_loop:nw {#1} #3 \s__color_mark #5 \s__color_stop 31408
31409 } 31409
31410 } 31410
31411 \cs_new_protected:Npn \color_set_eq:nn #1#2 31411
31412 { 31412
31413 \color_if_exist:nTF {#2} 31413
31414 { 31414
31415 \tl_clear_new:c { l__color_named_ #1 _tl } 31415
31416 \prop_clear_new:c { l__color_named_ #1 _prop } 31416
31417 \str_if_eq:nnTF {#2} { . } 31417
31418 { 31418
31419 \tl_set:ce { l__color_named_ #1 _tl } 31419
31420 { \__color_model:N \l__color_current_tl } 31420
31421 \prop_put:cve { l__color_named_ #1 _prop } { l__color_named_ #1 _tl } 31421
31422 { \__color_values:N \l__color_current_tl } 31422
31423 } 31423
31424 { 31424
31425 \tl_set_eq:cc { l__color_named_ #1 _tl } { l__color_named_ #2 _tl } 31425
31426 \prop_set_eq:cc { l__color_named_ #1 _prop } { l__color_named_ #2 _prop } 31426
31427 } 31427
31428 } 31428
31429 { 31429
31430 \msg_error:nnn { color } { unknown-color } {#2} 31430
31431 } 31431
31432 } 31432
31433 \color_set:nnn { black } { gray } { 0 } 31433
31434 \color_set:nnn { white } { gray } { 1 } 31434
31435 \color_set:nnn { cyan } { cmyk } { 1 , 0 , 0 , 0 } 31435
31436 \color_set:nnn { magenta } { cmyk } { 0 , 1 , 0 , 0 } 31436
31437 \color_set:nnn { yellow } { cmyk } { 0 , 0 , 1 , 0 } 31437
31438 \color_set:nnn { red } { rgb } { 1 , 0 , 0 } 31438
31439 \color_set:nnn { green } { rgb } { 0 , 1 , 0 } 31439
31440 \color_set:nnn { blue } { rgb } { 0 , 0 , 1 } 31440
31441 \prop_new:c { l__color_named_._prop } 31441

```

```
31442 \tl_new:c { l__color_named_.tl } 31442
31443 \tl_set:ce { l__color_named_.tl } { \__color_model:N \l__color_current_tl } 31443
31444 \prop_put:cve { l__color_named_.prop } { l__color_named_.tl } 31444
31445 { \__color_values:N \l__color_current_tl } 31445
31446 \cs_new_protected:Npn \color_export:nnN #1#2#3 31446
31447 { 31447
31448 \group_begin: 31448
31449 \tl_if_exist:cT { c__color_export_ #2_tl } 31449
31450 { \tl_set_eq:Nc \l_color_fixed_model_tl { c__color_export_ #2_tl } } 31450
31451 \__color_parse:nN {#1} #3 31451
31452 \__color_export:nN {#2} #3 31452
31453 \exp_args:NNNV \group_end: 31453
31454 \tl_set:Nn #3 #3 31454
31455 } 31455
31456 \cs_new_protected:Npn \color_export:nnnN #1#2#3#4 31456
31457 { 31457
31458 \__color_select_main:Nnn #4 {#1} {#2} 31458
31459 \__color_export:nN {#3} #4 31459
31460 } 31460
31461 \cs_new_protected:Npn \__color_export:nN #1#2 31461
31462 { \exp_after:wN \__color_export:nnnN #2 {#1} #2 } 31462
31463 \cs_new:Npn \__color_export:nnnN #1#2#3#4 31463
31464 { 31464
31465 \cs_if_exist_use:cF { __color_export_format_ #3 :nnN } 31465
31466 { 31466
31467 \msg_error:nnn { color } { unknown-export-format } {#3} 31467
31468 \use_none:nnn 31468
31469 } 31469
31470 {#1} {#2} #4 31470
31471 } 31471
31472 \cs_new_protected:Npn \__color_export_format_backend:nnN #1#2#3 31472
31473 { \tl_set:Nn #3 { {#1} {#2} } } 31473
31474 \cs_new_protected:Npn \__color_export:nnnNN #1#2#3#4#5 31474
31475 { 31475
31476 \str_if_eq:nnTF {#2} {#1} 31476
31477 { #5 #4 #3 \s__color_stop } 31477
31478 { 31478
31479 \__color_convert:nnnN {#2} {#1} {#3} #4 31479
31480 \exp_after:wN #5 \exp_after:wN #4 31480
31481 #4 \s__color_stop 31481
31482 } 31482
31483 } 31483
31484 \tl_const:cn { c__color_export_comma-sep-cmyk_tl } { cmyk } 31484
31485 \tl_const:cn { c__color_export_comma-sep-rgb_tl } { rgb } 31485
31486 \tl_const:Nn \c__color_export_HTML_tl { rgb } 31486
31487 \tl_const:cn { c__color_export_space-sep-cmyk_tl } { cmyk } 31487
```



```
31488 \tl_const:cn { c__color_export_space-sep-rgb_tl } { rgb } 31488
31489 \group_begin: 31489
31490 \cs_set_protected:Npn \__color_tmp:w #1#2 31490
31491 { 31491
31492 \cs_new_protected:cpe { __color_export_format_ #1 :nnN } ##1##2##3 31492
31493 { 31493
31494 \exp_not:N \__color_export:nnnNN {#2} {##1} {##2} ##3 31494
31495 \exp_not:c { __color_export_ #1 :Nw } 31495
31496 } 31496
31497 } 31497
31498 \__color_tmp:w { comma-sep-cmyk } { cmyk } 31498
31499 \__color_tmp:w { comma-sep-rgb } { rgb } 31499
31500 \__color_tmp:w { HTML } { rgb } 31500
31501 \__color_tmp:w { space-sep-cmyk } { cmyk } 31501
31502 \__color_tmp:w { space-sep-rgb } { rgb } 31502
31503 \group_end: 31503
31504 \cs_new_protected:cpn { __color_export_comma-sep-cmyk:Nw } 31504
31505 #1#2 ~ #3 ~ #4 ~ #5 \s__color_stop 31505
31506 { \tl_set:Nn #1 { #2 , #3 , #4 , #5 } } 31506
31507 \cs_new_protected:cpn { __color_export_space-sep-cmyk:Nw } #1#2 \s__color_stop 31507
31508 { \tl_set:Nn #1 {#2} } 31508
31509 \cs_new_protected:cpn { __color_export_comma-sep-rgb:Nw } #1#2 ~ #3 ~ #4 \s__color_stop 31509
31510 { \tl_set:Nn #1 { #2 , #3 , #4 } } 31510
31511 \cs_new_protected:Npn \__color_export_HTML:Nw #1#2 ~ #3 ~ #4 \s__color_stop 31511
31512 { 31512
31513 \tl_set:Nn #1 31513
31514 { 31514
31515 \__color_export_HTML:n {#2} 31515
31516 \__color_export_HTML:n {#3} 31516
31517 \__color_export_HTML:n {#4} 31517
31518 } 31518
31519 } 31519
31520 \cs_new:Npn \__color_export_HTML:n #1 31520
31521 { 31521
31522 \fp_compare:nNnTF {#1} = { 0 } 31522
31523 { 00 } 31523
31524 { 31524
31525 \fp_compare:nNnT { #1 * 255 } < { 16 } { 0 } 31525
31526 \int_to_Hex:n { \fp_to_int:n { #1 * 255 } } 31526
31527 } 31527
31528 } 31528
31529 \cs_new_protected:cpn { __color_export_space-sep-rgb:Nw } #1#2 \s__color_stop 31529
31530 { \tl_set:Nn #1 {#2} } 31530
31531 \prop_new:N \l__color_internal_prop 31531
31532 \int_new:N \g__color_model_int 31532
31533 \tl_const:Nn \c__color_fallback_cmyk_tl { cmyk } 31533
```

```
31534 \tl_const:Nn \c__color_fallback_gray_tl { gray } 31534
31535 \tl_const:Nn \c__color_fallback_rgb_tl { rgb } 31535
31536 \prop_new:N \g__color_colorants_prop 31536
31537 \prop_gput:Nnn \g__color_colorants_prop { black } { Black } 31537
31538 \prop_gput:Nnn \g__color_colorants_prop { blue } { Blue } 31538
31539 \prop_gput:Nnn \g__color_colorants_prop { cyan } { Cyan } 31539
31540 \prop_gput:Nnn \g__color_colorants_prop { green } { Green } 31540
31541 \prop_gput:Nnn \g__color_colorants_prop { magenta } { Magenta } 31541
31542 \prop_gput:Nnn \g__color_colorants_prop { none } { None } 31542
31543 \prop_gput:Nnn \g__color_colorants_prop { red } { Red } 31543
31544 \prop_gput:Nnn \g__color_colorants_prop { yellow } { Yellow } 31544
31545 \tl_const:Nn \c__color_model_whitepoint_CIELAB_a_tl { 1.0985 ~ 1 ~ 0.3558 } 31545
31546 \tl_const:Nn \c__color_model_whitepoint_CIELAB_b_tl { 0.9807 ~ 1 ~ 1.1822 } 31546
31547 \tl_const:Nn \c__color_model_whitepoint_CIELAB_e_tl { 1 ~ 1 ~ 1 } 31547
31548 \tl_const:cn { c__color_model_whitepoint_CIELAB_d50_tl } { 0.9642 ~ 1 ~ 0.8251 } 31548
31549 \tl_const:cn { c__color_model_whitepoint_CIELAB_d55_tl } { 0.9568 ~ 1 ~ 0.9214 } 31549
31550 \tl_const:cn { c__color_model_whitepoint_CIELAB_d65_tl } { 0.9504 ~ 1 ~ 1.0888 } 31550
31551 \tl_const:cn { c__color_model_whitepoint_CIELAB_d75_tl } { 0.9497 ~ 1 ~ 1.2261 } 31551
31552 \tl_const:Nn \c__color_model_range_CIELAB_tl { 0 ~ 100 ~ -128 ~ 127 ~ -128 ~ 127 } 31552
31553 \prop_new:N \g__color_alternative_model_prop 31553
31554 \clist_map_inline:nn { cyan , magenta , yellow , black } 31554
31555 { \prop_gput:Nnn \g__color_alternative_model_prop {#1} { cmyk } } 31555
31556 \clist_map_inline:nn { red , green , blue } 31556
31557 { \prop_gput:Nnn \g__color_alternative_model_prop {#1} { rgb } } 31557
31558 \prop_new:N \g__color_alternative_values_prop 31558
31559 \prop_gput:Nnn \g__color_alternative_values_prop { cyan } { 1 , 0 , 0 , 0 } 31559
31560 \prop_gput:Nnn \g__color_alternative_values_prop { magenta } { 0 , 1 , 0 , 0 } 31560
31561 \prop_gput:Nnn \g__color_alternative_values_prop { yellow } { 0 , 0 , 1 , 0 } 31561
31562 \prop_gput:Nnn \g__color_alternative_values_prop { black } { 0 , 0 , 0 , 1 } 31562
31563 \prop_gput:Nnn \g__color_alternative_values_prop { red } { 1 , 0 , 0 } 31563
31564 \prop_gput:Nnn \g__color_alternative_values_prop { green } { 0 , 1 , 0 } 31564
31565 \prop_gput:Nnn \g__color_alternative_values_prop { blue } { 0 , 0 , 1 } 31565
31566 \cs_new_protected:Npn \color_model_new:nnn #1#2#3 31566
31567 { 31567
31568 \exp_args:Nee \__color_model_new:nnn 31568
31569 { \tl_to_str:n {#1} } 31569
31570 { \str_casefold:n {#2} } {#3} 31570
31571 } 31571
31572 \cs_new_protected:Npn \__color_model_new:nnn #1#2#3 31572
31573 { 31573
31574 \cs_if_exist:cTF { __color_parse_model_ #1 :w } 31574
31575 { 31575
31576 \msg_error:nnn { color } { model-already-defined } {#1} 31576
31577 } 31577
31578 { 31578
31579 \cs_if_exist:cTF { __color_model_ #2 :n } 31579
```

```

31580         {
31581             \prop_set_from_keyval:Nn \l__color_internal_prop {#3}
31582             \use:c { __color_model_ #2 :n } {#1}
31583         }
31584     {
31585         \msg_error:nnn { color } { unknown-model-type } {#2}
31586     }
31587 }
31588 }
31589 \cs_new_protected:Npn \__color_model_init:nnn #1#2#3
31590 {
31591     \int_gincr:N \g__color_model_int
31592     \clist_map_inline:nn { fill , stroke , select }
31593     {
31594         \cs_new_protected:cpe { __color_backend_ ##1 _ #1 :n } ####1
31595         {
31596             \exp_not:c { __color_backend_ ##1 _ #2 :nn }
31597             { color \int_use:N \g__color_model_int } {####1}
31598         }
31599     }
31600     \cs_new_protected:cpe { __color_model_ #1 _white: }
31601     {
31602         \prop_put:Nnn \exp_not:N \l__color_named_white_prop {#1}
31603         { \exp_not:n {#3} }
31604         \exp_not:N \int_compare:nNnF { \tex_currentgrouplevel:D } = 0
31605         { \group_insert_after:N \exp_not:c { __color_model_ #1 _ white: } }
31606     }
31607     \use:c { __color_model_ #1 _white: }
31608 }
31609 \cs_generate_variant:Nn \__color_model_init:nnn { nne }
31610 \cs_new_protected:Npn \__color_model_separation:n #1
31611 {
31612     \prop_get:NnNTF \l__color_internal_prop { name }
31613     \l__color_internal_tl
31614     {
31615         \exp_args:NV \__color_model_separation:nn
31616         \l__color_internal_tl {#1}
31617     }
31618     {
31619         \msg_error:nnn { color }
31620         { separation-requires-name } {#1}
31621     }
31622 }
31623 \cs_new_protected:Npn \__color_model_separation:nn #1#2
31624 {
31625     \prop_get:NnNTF \l__color_internal_prop { alternative-model }

```

```
31626 \l__color_internal_tl 31626
31627 { 31627
31628 \exp_args:NV \__color_model_separation:nnn 31628
31629 \l__color_internal_tl {#2} {#1} 31629
31630 } 31630
31631 { 31631
31632 \msg_error:nnn { color } 31632
31633 { separation-alternative-model } {#2} 31633
31634 } 31634
31635 } 31635
31636 \cs_new_protected:Npn \__color_model_separation:nnn #1#2#3 31636
31637 { 31637
31638 \cs_if_exist:cTF { __color_model_separation_ #1 :nnnnnn } 31638
31639 { 31639
31640 \prop_get:NnNTF \l__color_internal_prop { alternative-values } 31640
31641 \l__color_internal_tl 31641
31642 { 31642
31643 \exp_after:wN \__color_model_separation:w \l__color_internal_tl 31643
31644 , 0 , 0 , 0 , 0 \s__color_stop {#2} {#3} {#1} 31644
31645 } 31645
31646 { 31646
31647 \msg_error:nnn { color } 31647
31648 { separation-alternative-values } {#2} 31648
31649 } 31649
31650 } 31650
31651 { 31651
31652 \msg_error:nnn { color } 31652
31653 { unknown-alternative-model } {#1} 31653
31654 } 31654
31655 } 31655
31656 \cs_new_protected:Npn \__color_model_separation:w 31656
31657 #1 , #2 , #3 , #4 , #5 \s__color_stop #6#7#8 31657
31658 { 31658
31659 \__color_model_init:nnn {#6} { separation } { 0 } 31659
31660 \cs_new_eq:cN { __color_parse_mix_ #6 :nw } \__color_parse_mix_gray:nw 31660
31661 \cs_new:cpn { __color_parse_model_ #6 :w } ##1 , ##2 \s__color_stop 31661
31662 { {#6} { \__color_parse_number:n {##1} } } 31662
31663 \use:c { __color_model_separation_ #8 :nnnnnn } 31663
31664 {#6} {#7} {#1} {#2} {#3} {#4} 31664
31665 \prop_gput:Nnn \g__color_alternative_model_prop {#6} {#8} 31665
31666 \prop_gput:Nne \g__color_colorants_prop {#6} 31666
31667 { \str_convert_pdfname:n {#7} } 31667
31668 } 31668
31669 \cs_new_protected:Npn \__color_model_separation_cmyk:nnnnnn #1#2#3#4#5#6 31669
31670 { 31670
31671 \tl_const:cn { c__color_fallback_ #1 _tl } { cmyk } 31671
```

```
31672 \cs_new:cpn { __color_convert_ #1 _cmyk:w } ##1 \s__color_stop 31672
31673 { 31673
31674 \fp_eval:n {##1 * #3} ~ 31674
31675 \fp_eval:n {##1 * #4} ~ 31675
31676 \fp_eval:n {##1 * #5} ~ 31676
31677 \fp_eval:n {##1 * #6} 31677
31678 } 31678
31679 \cs_new:cpn { __color_convert_cmyk_ #1 :w } ##1 \s__color_stop { 1 } 31679
31680 \prop_gput:Nnn \g__color_alternative_values_prop {#1} { #3 , #4 , #5 , #6 } 31680
31681 \__color_backend_separation_init:nnnnn {#2} { /DeviceCMYK } { } 31681
31682 { 0 ~ 0 ~ 0 ~ 0 } { #3 ~ #4 ~ #5 ~ #6 } 31682
31683 } 31683
31684 \cs_new_protected:Npn \__color_model_separation_rgb:nnnnnn #1#2#3#4#5#6 31684
31685 { 31685
31686 \tl_const:cn { c__color_fallback_ #1 _tl } { rgb } 31686
31687 \cs_new:cpn { __color_convert_ #1 _rgb:w } ##1 \s__color_stop 31687
31688 { 31688
31689 \fp_eval:n {##1 * #3} ~ 31689
31690 \fp_eval:n {##1 * #4} ~ 31690
31691 \fp_eval:n {##1 * #5} 31691
31692 } 31692
31693 \cs_new:cpn { __color_convert_rgb_ #1 :w } ##1 \s__color_stop { 1 } 31693
31694 \prop_gput:Nnn \g__color_alternative_values_prop {#1} { #3 , #4 , #5 } 31694
31695 \__color_backend_separation_init:nnnnn {#2} { /DeviceRGB } { } 31695
31696 { 0 ~ 0 ~ 0 } { #3 ~ #4 ~ #5 } 31696
31697 } 31697
31698 \cs_new_protected:Npn \__color_model_separation_gray:nnnnnn #1#2#3#4#5#6 31698
31699 { 31699
31700 \tl_const:cn { c__color_fallback_ #1 _tl } { gray } 31700
31701 \cs_new:cpn { __color_convert_ #1 _gray:w } ##1 \s__color_stop 31701
31702 { \fp_eval:n {##1 * #3} } 31702
31703 \cs_new:cpn { __color_convert_gray_ #1 :w } ##1 \s__color_stop { 1 } 31703
31704 \prop_gput:Nnn \g__color_alternative_values_prop {#1} {#3} 31704
31705 \__color_backend_separation_init:nnnnn {#2} { /DeviceGray } { } { 0 } {#3} 31705
31706 } 31706
31707 \cs_new_protected:Npn \__color_model_convert:nnn #1#2#3 31707
31708 { 31708
31709 \cs_new:cpe { __color_convert_ #1 _ #3 :w } ##1 \s__color_stop 31709
31710 { 31710
31711 \exp_not:N \exp_args:NNe \exp_not:N \use:nn 31711
31712 \exp_not:c { __color_convert_ #2 _ #3 :w } 31712
31713 { \exp_not:c { __color_convert_ #1 _ #2 :w } ##1 \s__color_stop } 31713
31714 \c_space_tl \exp_not:N \s__color_stop 31714
31715 } 31715
31716 } 31716
31717 \cs_new_protected:Npn \__color_model_separation_CIELAB:nnnnnn #1#2#3#4#5#6 31717
```

```

31718 {
31719     \prop_get:NnNF \l__color_internal_prop { illuminant }
31720     \l__color_internal_tl
31721     {
31722         \msg_error:nnn { color }
31723         { CIELAB-requires-illuminant } {#1}
31724         \tl_set:Nn \l__color_internal_tl { d50 }
31725     }
31726     \exp_args:NV \__color_model_separation_CIELAB:nnnnnnn
31727     \l__color_internal_tl {#1} {#2} {#3} {#4} {#5} {#6}
31728 }
31729 \cs_new_protected:Npn \__color_model_separation_CIELAB:nnnnnnn #1#2#3#4#5#6#7
31730 {
31731     \tl_if_exist:cTF { c__color_model_whitepoint_CIELAB_ #1 _tl }
31732     {
31733         \__color_backend_separation_init_CIELAB:nnn {#1} {#3} { #4 ~ #5 ~ #6 }
31734         \tl_const:cn { c__color_fallback_ #2 _tl } { gray }
31735         \cs_new:cpn { __color_convert_ #2 _gray:w } ##1 \s__color_stop
31736         { 0 }
31737         \cs_new:cpn { __color_convert_gray_ #2 :w } ##1 \s__color_stop
31738         { 1 }
31739     }
31740     {
31741         \msg_error:nnn { color }
31742         { unknown-CIELAB-illuminant } {#1}
31743     }
31744 }
31745 \cs_new_protected:Npn \__color_model_devicen:n #1
31746 {
31747     \prop_get:NnNTF \l__color_internal_prop { names }
31748     \l__color_internal_tl
31749     {
31750         \exp_args:NV \__color_model_devicen:nn
31751         \l__color_internal_tl {#1}
31752     }
31753     {
31754         \msg_error:nnn { color }
31755         { DeviceN-requires-names } {#1}
31756     }
31757 }
31758 \cs_new_protected:Npn \__color_model_devicen:nn #1#2
31759 {
31760     \tl_clear:N \l__color_model_tl
31761     \clist_map_inline:nn {#1}
31762     {
31763         \prop_get:NnNTF \g__color_alternative_model_prop {##1}

```

```

31764         \l__color_internal_tl
31765     {
31766         \tl_if_empty:NTF \l__color_model_tl
31767         { \tl_set_eq:NN \l__color_model_tl \l__color_internal_tl }
31768         {
31769             \str_if_eq:VVF \l__color_model_tl \l__color_internal_tl
31770             {
31771                 \msg_error:nnn { color }
31772                 { DeviceN-inconsistent-alternative }
31773                 {#2}
31774                 \clist_map_break:n { \use_none:nnnn }
31775             }
31776         }
31777     }
31778     {
31779         \str_if_eq:nnF {##1} { none }
31780         {
31781             \msg_error:nnn { color }
31782             { DeviceN-no-alternative }
31783             {#2}
31784         }
31785     }
31786 }
31787 \tl_if_empty:NTF \l__color_model_tl
31788 {
31789     \msg_error:nnn { color }
31790     { DeviceN-no-alternative } {#2}
31791 }
31792 { \exp_args:NV \__color_model_devicen:nnn \l__color_model_tl {#1} {#2} }
31793 }
31794 \cs_new_protected:Npn \__color_model_devicen:nnn #1#2#3
31795 {
31796     \exp_args:Ne \__color_model_devicen:nnnn
31797     { \clist_count:n {#2} } {#1} {#2} {#3}
31798 }
31799 \cs_new_protected:Npn \__color_model_devicen:nnnn #1#2#3#4
31800 {
31801     \__color_model_init:nne {#4} { devicen }
31802     {
31803         0 \prg_replicate:nn { #1 - 1 } { ~ 0 }
31804     }
31805     \cs_if_exist_use:cF { __color_model_devicen_parse_ #1 :nn }
31806     { \__color_model_devicen_parse_generic:nn }
31807     {#4} {#1}
31808     \__color_model_devicen_init:nnn {#1} {#2} {#3}
31809     \__color_model_devicen_convert:nnne {#4} {#2} {#3}

```



```

31810 {
31811     1 \prg_replicate:nn { #1 - 1 } { ~ 1 }
31812 }
31813 }
31814 \cs_new_protected:cpn { __color_model_devicen_parse_1:nn } #1#2
31815 {
31816     \cs_new:cpn { __color_parse_model_ #1 :w } ##1 , ##2 \s__color_stop
31817     { {#1} { \__color_parse_number:n {##1} } }
31818     \cs_new_eq:cN { __color_parse_mix_ #1 :nw } \__color_parse_mix_gray:nw
31819 }
31820 \cs_new_protected:cpn { __color_model_devicen_parse_2:nn } #1#2
31821 {
31822     \cs_new:cpn { __color_parse_model_ #1 :w } ##1 , ##2 , ##3 \s__color_stop
31823     { {#1} { \__color_parse_number:n {##1} ~ \__color_parse_number:n {##2} } }
31824     \cs_new:cpn { __color_parse_mix_ #1 :nw }
31825     ##1##2 ~ ##3 \s__color_mark ##4 ~ ##5 \s__color_stop
31826     {
31827         \fp_eval:n { ##2 * ##1 + ##4 * ( 1 - ##1 ) } \c_space_tl
31828         \fp_eval:n { ##3 * ##1 + ##5 * ( 1 - ##1 ) }
31829     }
31830 }
31831 \cs_new_protected:cpn { __color_model_devicen_parse_3:nn } #1#2
31832 {
31833     \cs_new:cpn { __color_parse_model_ #1 :w } ##1 , ##2 , ##3 , ##4 \s__color_stop
31834     {
31835         {#1}
31836         {
31837             \__color_parse_number:n {##1} ~
31838             \__color_parse_number:n {##2} ~
31839             \__color_parse_number:n {##3}
31840         }
31841     }
31842     \cs_new_eq:cN { __color_parse_mix_ #1 :nw } \__color_parse_mix_rgb:nw
31843 }
31844 \cs_new_protected:cpn { __color_model_devicen_parse_4:nn } #1#2
31845 {
31846     \cs_new:cpn { __color_parse_model_ #1 :w }
31847     ##1 , ##2 , ##3 , ##4 , ##5 \s__color_stop
31848     {
31849         {#1}
31850         {
31851             \__color_parse_number:n {##1} ~
31852             \__color_parse_number:n {##2} ~
31853             \__color_parse_number:n {##3} ~
31854             \__color_parse_number:n {##4}
31855         }

```

```
31856 } 31856
31857 \cs_new_eq:cN { __color_parse_mix_ #1 :nw } \__color_parse_mix_cmyk:nw 31857
31858 } 31858
31859 \cs_new_protected:Npn \__color_model_devicen_parse_generic:nn #1#2 31859
31860 { 31860
31861 \cs_new:cpn { __color_parse_model_ #1 :w } ##1 , ##2 \s__color_stop 31861
31862 { 31862
31863 {#1} 31863
31864 { \__color_model_devicen_parse:nw {#2} ##1 , ##2 , \q_nil , \s__color_stop } 31864
31865 } 31865
31866 \cs_new:cpe { __color_parse_mix_ #1 :nw } 31866
31867 ##1 ##2 \s__color_mark ##3 \s__color_stop 31867
31868 { 31868
31869 \exp_not:N \__color_model_devicen_mix:nw {##1} 31869
31870 ##2 \c_space_tl \exp_not:N \q_nil \c_space_tl \exp_not:N \s__color_mark 31870
31871 ##3 \c_space_tl \exp_not:N \q_nil \c_space_tl \exp_not:N \s__color_stop 31871
31872 } 31872
31873 } 31873
31874 \cs_new:Npn \__color_model_devicen_parse:nw #1#2 , #3 \s__color_stop 31874
31875 { 31875
31876 \int_compare:nNnT {#1} > 0 31876
31877 { 31877
31878 \quark_if_nil:nTF {#2} 31878
31879 { \prg_replicate:nn {#1} { 0 ~ } } 31879
31880 { 31880
31881 \__color_parse_number:n {#2} 31881
31882 \int_compare:nNnT {#1} > 1 { ~ } 31882
31883 \exp_args:Nf \__color_model_devicen_parse:nw 31883
31884 { \int_eval:n { #1 - 1 } } #3 \s__color_stop 31884
31885 } 31885
31886 } 31886
31887 } 31887
31888 \cs_new:Npn \__color_model_devicen_mix:nw #1#2 ~ #3 \s__color_mark #4 ~ #5 \s__color_stop 31888
31889 { 31889
31890 \fp_eval:n { #2 * #1 + #4 * ( 1 - #1 ) } 31890
31891 \quark_if_nil:oF { \tl_head:w #3 \q_stop } 31891
31892 { 31892
31893 \c_space_tl 31893
31894 \__color_model_devicen_mix:nw {#1} #3 \s__color_mark #5 \s__color_stop 31894
31895 } 31895
31896 } 31896
31897 \cs_new_protected:Npn \__color_model_devicen_init:nnn #1#2#3 31897
31898 { 31898
31899 \exp_args:Ne \__color_model_devicen_init:nnnn 31899
31900 { 31900
31901 \str_case:nn {#2} 31901
```

```

31902         {
31903             { cmyk } { 4 }
31904             { gray } { 1 }
31905             { rgb } { 3 }
31906         }
31907     }
31908     {#1} {#2} {#3}
31909 }
31910 \cs_new_protected:Npn \__color_model_devicen_init:nnnn #1#2#3#4
31911 {
31912     \tl_set:Nc \l__color_internal_tl
31913     { \prg_replicate:nn {#1} { 1.0 ~ } }
31914     \int_zero:N \l__color_internal_int
31915     \clist_map_inline:nn {#4}
31916     {
31917         \int_incr:N \l__color_internal_int
31918         \prop_get:NnN \g__color_alternative_values_prop {##1}
31919         \l__color_value_tl
31920         \exp_after:wN \__color_model_devicen_transform:w
31921         \l__color_value_tl , 0 , 0 , 0 , \s__color_stop {#1} {#2}
31922     }
31923     \tl_put_right:Nc \l__color_internal_tl
31924     {
31925         \prg_replicate:nn {#1}
31926         { neg ~ 1.0 ~ add ~ #1 ~ -1 ~ roll ~ }
31927         \int_eval:n { #2 + #1 } ~ #1 ~ roll
31928         \prg_replicate:nn {#2} { ~ pop } ~
31929         #1 ~ 1 ~ roll
31930     }
31931     \use:e
31932     {
31933         \__color_backend_devicen_init:nnn
31934         {
31935             \clist_map_function:nN {#4}
31936             \__color_model_devicen_colorant:n
31937         }
31938         {
31939             \str_case:nn {#3}
31940             {
31941                 { cmyk } { /DeviceCMYK }
31942                 { gray } { /DeviceGray }
31943                 { rgb } { /DeviceRGB }
31944             }
31945         }
31946         { \exp_not:V \l__color_internal_tl }
31947     }

```

```

31948 }
31949 \cs_new_protected:Npn \__color_model_devicen_transform:w
31950 #1 , #2 , #3 , #4 , #5 \s__color_stop #6#7
31951 {
31952   \use:c { __color_model_devicen_transform_ #6 :nnnnn }
31953   {#1} {#2} {#3} {#4} {#7}
31954 }
31955 \cs_new_protected:cpn { __color_model_devicen_transform_1:nnnnn } #1#2#3#4#5
31956 { \__color_model_devicen_transform:nnn {#5} { 1 } {#1} }
31957 \cs_new_protected:cpn { __color_model_devicen_transform_3:nnnnn } #1#2#3#4#5
31958 {
31959   \clist_map_inline:nn { #1 , #2 , #3 }
31960   { \__color_model_devicen_transform:nnn {#5} { 3 } {##1} }
31961 }
31962 \cs_new_protected:cpn { __color_model_devicen_transform_4:nnnnn } #1#2#3#4#5
31963 {
31964   \clist_map_inline:nn { #1 , #2 , #3 , #4 }
31965   { \__color_model_devicen_transform:nnn {#5} { 4 } {##1} }
31966 }
31967 \cs_new_protected:Npn \__color_model_devicen_transform:nnn #1#2#3
31968 {
31969   \tl_put_right:Ne \l__color_internal_tl
31970   {
31971     \fp_compare:nNnF {#3} = \c_zero_fp
31972     {
31973       \int_eval:n { #1 - \l__color_internal_int + #2 } ~ index ~
31974       -#3 ~ mul ~ 1.0 ~ add ~ mul ~
31975     }
31976     #2 ~ -1 ~ roll ~
31977   }
31978 }
31979 \cs_new:Npn \__color_model_devicen_colorant:n #1
31980 {
31981   / \prop_item:Nn \g__color_colorants_prop {#1} ~
31982 }
31983 \cs_new_protected:Npn \__color_model_devicen_convert:nnnn #1#2#3
31984 {
31985   \use:c { __color_model_devicen_convert_ #2 :nnn } {#1} {#3}
31986 }
31987 \cs_generate_variant:Nn \__color_model_devicen_convert:nnnn { nnne }
31988 \cs_new_protected:Npn \__color_model_devicen_convert_cmyk:nnn #1#2
31989 {
31990   \tl_const:cn { c__color_fallback_ #1 _tl } { cmyk }
31991   \__color_model_devicen_convert:nnnnn {#1} { cmyk } { 4 } {#2}
31992 }
31993 \cs_new_protected:Npn \__color_model_devicen_convert_gray:nnn #1#2

```

```

31994 { 31994
31995 \tl_const:cn { c__color_fallback_ #1 _tl } { gray } 31995
31996 \__color_model_devicen_convert:nnnnn {#1} { gray } { 1 } {#2} 31996
31997 } 31997
31998 \cs_new_protected:Npn \__color_model_devicen_convert_rgb:nnn #1#2 31998
31999 { 31999
32000 \tl_const:cn { c__color_fallback_ #1 _tl } { rgb } 32000
32001 \__color_model_devicen_convert:nnnnn {#1} { rgb } { 3 } {#2} 32001
32002 } 32002
32003 \cs_new_protected:Npn \__color_model_devicen_convert:nnnnn #1#2#3#4#5 32003
32004 { 32004
32005 \cs_new:cpn { __color_convert_ #2 _ #1 :w } ##1 \s__color_stop {#5} 32005
32006 \cs_new:cpe { __color_convert_ #1 _ #2 :w } ##1 \s__color_stop 32006
32007 { 32007
32008 \exp_not:c { __color_convert_devicen_ #2 : \prg_replicate:nn {#3} { n } w } 32008
32009 \prg_replicate:nn {#3} { { 1 } } 32009
32010 ##1 ~ \exp_not:N \s__color_mark 32010
32011 \clist_map_function:nN {#4} \__color_model_devicen_convert:n 32011
32012 {} 32012
32013 \exp_not:N \s__color_stop 32013
32014 } 32014
32015 } 32015
32016 \cs_new:Npn \__color_model_devicen_convert:n #1 32016
32017 { 32017
32018 { 32018
32019 \exp_args:Ne \__color_model_devicen_convert_aux:n 32019
32020 { \prop_item:Nn \g__color_alternative_values_prop {#1} } 32020
32021 } 32021
32022 } 32022
32023 \cs_new:Npn \__color_model_devicen_convert_aux:n #1 32023
32024 { \__color_model_devicen_convert_aux:w #1 , , , , \s__color_stop } 32024
32025 \cs_new:Npn \__color_model_devicen_convert_aux:w #1 , #2 , #3 , #4 , #5 \s__color_stop 32025
32026 { 32026
32027 {#1} 32027
32028 \tl_if_blank:nF {#2} 32028
32029 { 32029
32030 {#2} 32030
32031 \tl_if_blank:nF {#3} 32031
32032 { 32032
32033 {#3} 32033
32034 \tl_if_blank:nF {#4} { {#4} } 32034
32035 } 32035
32036 } 32036
32037 } 32037
32038 \cs_new:Npn \__color_convert_devicen_cmyk:nnnnw 32038
32039 #1#2#3#4#5 ~ #6 \s__color_mark #7#8 \s__color_stop 32039

```

```
32040 { 32040
32041 \__color_convert_devicen_cmyk:nnnnnnnnn {#5} {#1} {#2} {#3} {#4} #7 32041
32042 #6 \s__color_mark #8 \s__color_stop 32042
32043 } 32043
32044 \cs_new:Npn \__color_convert_devicen_cmyk:nnnnnnnnn #1#2#3#4#5#6#7#8#9 32044
32045 { 32045
32046 \use:e 32046
32047 { 32047
32048 \exp_not:N \__color_convert_devicen_cmyk_aux:nnnnw 32048
32049 { \fp_eval:n { #2 * (1 - (#1 * #6)) } } 32049
32050 { \fp_eval:n { #3 * (1 - (#1 * #7)) } } 32050
32051 { \fp_eval:n { #4 * (1 - (#1 * #8)) } } 32051
32052 { \fp_eval:n { #5 * (1 - (#1 * #9)) } } 32052
32053 } 32053
32054 } 32054
32055 \cs_new:Npn \__color_convert_devicen_cmyk_aux:nnnnw 32055
32056 #1#2#3#4 #5 \s__color_mark #6 \s__color_stop 32056
32057 { 32057
32058 \tl_if_blank:nTF {#5} 32058
32059 { 32059
32060 \fp_eval:n { 1 - #1 } ~ 32060
32061 \fp_eval:n { 1 - #2 } ~ 32061
32062 \fp_eval:n { 1 - #3 } ~ 32062
32063 \fp_eval:n { 1 - #4 } 32063
32064 } 32064
32065 { 32065
32066 \__color_convert_devicen_cmyk:nnnnw {#1} {#2} {#3} {#4} 32066
32067 #5 \s__color_mark #6 \s__color_stop 32067
32068 } 32068
32069 } 32069
32070 \cs_new:Npn \__color_convert_devicen_gray:nw 32070
32071 #1#2 ~ #3 \s__color_mark #4#5 \s__color_stop 32071
32072 { 32072
32073 \__color_convert_devicen_gray:nnn {#2} {#1} #4 32073
32074 #3 \s__color_mark #5 \s__color_stop 32074
32075 } 32075
32076 \cs_new:Npn \__color_convert_devicen_gray:nnn #1#2#3 32076
32077 { 32077
32078 \exp_args:Ne \__color_convert_devicen_gray_aux:nw 32078
32079 { \fp_eval:n { #2 * (1 - (#1 * #3)) } } 32079
32080 } 32080
32081 \cs_new:Npn \__color_convert_devicen_gray_aux:nw 32081
32082 #1 #2 \s__color_mark #3 \s__color_stop 32082
32083 { 32083
32084 \tl_if_blank:nTF {#2} 32084
32085 { \fp_eval:n { 1 - #1 } } 32085
```

```
32086 {
32087     \__color_convert_devicen_gray:nw {#1}
32088     #2 \s__color_mark #3 \s__color_stop
32089 }
32090 }
32091 \cs_new:Npn \__color_convert_devicen_rgb:nnnw
32092 #1#2#3#4 ~ #5 \s__color_mark #6#7 \s__color_stop
32093 {
32094     \__color_convert_devicen_rgb:nnnnnnn {#4} {#1} {#2} {#3} #6
32095     #5 \s__color_mark #7 \s__color_stop
32096 }
32097 \cs_new:Npn \__color_convert_devicen_rgb:nnnnnnn #1#2#3#4#5#6#7
32098 {
32099     \use:e
32100     {
32101         \exp_not:N \__color_convert_devicen_rgb_aux:nnnw
32102         { \fp_eval:n { #2 * (1 - (#1 * #5)) } }
32103         { \fp_eval:n { #3 * (1 - (#1 * #6)) } }
32104         { \fp_eval:n { #4 * (1 - (#1 * #7)) } }
32105     }
32106 }
32107 \cs_new:Npn \__color_convert_devicen_rgb_aux:nnnw
32108 #1#2#3 #4 \s__color_mark #5 \s__color_stop
32109 {
32110     \tl_if_blank:nTF {#4}
32111     {
32112         \fp_eval:n { 1 - #1 } ~
32113         \fp_eval:n { 1 - #2 } ~
32114         \fp_eval:n { 1 - #3 }
32115     }
32116     {
32117         \__color_convert_devicen_rgb:nnnw {#1} {#2} {#3}
32118         #4 \s__color_mark #5 \s__color_stop
32119     }
32120 }
32121 \prop_const_from_keyval:Nn \c__color_icc_colorspace_signatures_prop
32122 {
32123     47524159 = {1} {1} {0} {},
32124     52474220 = {3} {0~0~0} {1~1~1} {},
32125     434D594B = {4} {0~0~0~1} {0~0~0~0} {},
32126     4C616220 = {3} {0~0~0} {100~0~0} {0~100~-128~127~-128~127}
32127 }
32128 \cs_new_protected:Npn \__color_model_iccbased:n #1
32129 {
32130     \prop_get:NnNTF \l__color_internal_prop { file }
32131     \l__color_internal_tl
```



```
32132 {
32133     \exp_args:NV \__color_model_iccbased:nn
32134     \l__color_internal_tl {#1}
32135 }
32136 {
32137     \msg_error:nnn { color }
32138     { ICCBased-requires-file } {#1}
32139 }
32140 }
32141 \cs_new_protected:Npn \__color_model_iccbased:nn #1#2
32142 {
32143     \prop_get:NeNTF \c__color_icc_colorspace_signatures_prop
32144     { \file_hex_dump:nnn { #1 } { 17 } { 20 } } \l__color_internal_tl
32145     {
32146         \exp_last_unbraced:NV \__color_model_iccbased_aux:nnnnnn
32147         \l__color_internal_tl { #2 } { #1 }
32148     }
32149     {
32150         \msg_error:nnn { color }
32151         { ICCBased-unsupported-colorspace } {#2}
32152     }
32153 }
32154 \cs_new_protected:Npn \__color_model_iccbased_aux:nnnnnn #1#2#3#4#5#6
32155 {
32156     \__color_model_init:nnn {#5} { iccbased } {#3}
32157     \tl_const:cn { c__color_fallback_ #5 _tl } { gray }
32158     \cs_new:cpn { __color_convert_ #5 _gray:w } ##1 \s__color_stop { 0 }
32159     \cs_new:cpn { __color_convert_gray_ #5 :w } ##1 \s__color_stop { #2 }
32160     \use:c { __color_model_devicen_parse_ #1 :nn } {#5} {#1}
32161     \exp_args:Ne \__color_backend_iccbased_init:nnn
32162     { \file_full_name:n {#6} } {#1} {#4}
32163 }
32164 \cs_new_protected:Npn \color_profile_apply:nn #1#2
32165 {
32166     \exp_args:Ne \__color_profile_apply:nn
32167     { \file_full_name:n {#1} } {#2}
32168 }
32169 \cs_new_protected:Npn \__color_profile_apply:nn #1#2
32170 {
32171     \cs_if_exist_use:cF { __color_profile_apply_ \tl_to_str:n {#2} :n }
32172     {
32173         \msg_error:nnn { color } { ICC-Device-unknown } {#2}
32174         \use_none:n
32175     }
32176     {#1}
32177 }
```

```
32178 \cs_new_protected:Npn \__color_profile_apply_gray:n #1 32178
32179 { 32179
32180 \int_gincr:N \g__color_model_int 32180
32181 \__color_backend_iccbased_device:nnn {#1} { Gray } { 1 } 32181
32182 } 32182
32183 \cs_new_protected:Npn \__color_profile_apply_rgb:n #1 32183
32184 { 32184
32185 \int_gincr:N \g__color_model_int 32185
32186 \__color_backend_iccbased_device:nnn {#1} { RGB } { 3 } 32186
32187 } 32187
32188 \cs_new_protected:Npn \__color_profile_apply_cmyk:n #1 32188
32189 { 32189
32190 \int_gincr:N \g__color_model_int 32190
32191 \__color_backend_iccbased_device:nnn {#1} { CMYK } { 4 } 32191
32192 } 32192
32193 \cs_new_protected:Npn \color_show:n 32193
32194 { \__color_show:Nn \msg_show:nneeee } 32194
32195 \cs_new_protected:Npn \color_log:n 32195
32196 { \__color_show:Nn \msg_log:nneeee } 32196
32197 \cs_new_protected:Npn \__color_show:Nn #1#2 32197
32198 { 32198
32199 #1 { color } { show } 32199
32200 {#2} 32200
32201 { 32201
32202 \color_if_exist:nT {#2} 32202
32203 { 32203
32204 \exp_args:Nv \__color_show:n { l__color_named_ #2 _tl } 32204
32205 \prop_map_function:cN 32205
32206 { l__color_named_ #2 _prop } 32206
32207 \msg_show_item_unbraced:nn 32207
32208 } 32208
32209 } 32209
32210 { } 32210
32211 { } 32211
32212 } 32212
32213 \cs_new:Npn \__color_show:n #1 32213
32214 { 32214
32215 \msg_show_item_unbraced:nn { model } {#1} 32215
32216 } 32216
32217 \msg_new:nnnn { color } { CIELAB-requires-illuminant } 32217
32218 { CIELAB~color~space~'#1'~require~an~illuminant. } 32218
32219 { 32219
32220 LaTeX~has~been~asked~to~create~a~separation~color~space~using~ 32220
32221 CIELAB~specifications,~but~no~\\ \\ 32221
32222 \iow_indent:n { illuminant~==<basis> } 32222
32223 \\ \\ 32223
```

```
32224 key~was~given~with~the~correct~information.~LaTeX~will~use~illuminant~ 32224
32225 'd50'~for~recovery. 32225
32226 } 32226
32227 \msg_new:nnnn { color } { conversion-not-available } 32227
32228 { No~model~conversion~available~from~'#1'~to~'#2'. } 32228
32229 { 32229
32230 LaTeX~has~been~asked~to~convert~a~color~from~model~'#1'~ 32230
32231 to~model~'#2',~but~there~is~no~method~available~to~do~that. 32231
32232 } 32232
32233 \msg_new:nnnn { color } { DeviceN-inconsistent-alternative } 32233
32234 { DeviceN~color~spaces~require~a~single~alternative~space. } 32234
32235 { 32235
32236 LaTeX~has~been~asked~to~create~a~DeviceN~color~space~'#1',~ 32236
32237 but~the~constituent~colors~do~not~have~a~common~alternative~ 32237
32238 color. 32238
32239 } 32239
32240 \msg_new:nnnn { color } { DeviceN-no-alternative } 32240
32241 { DeviceN~color~spaces~require~an~alternative~space. } 32241
32242 { 32242
32243 LaTeX~has~been~asked~to~create~a~DeviceN~color~space~'#1',~ 32243
32244 but~the~constituent~colors~do~not~all~have~a~device-based~alternative. 32244
32245 } 32245
32246 \msg_new:nnnn { color } { DeviceN-requires-names } 32246
32247 { DeviceN~color~space~'#1'~require~a~list~of~names. } 32247
32248 { 32248
32249 LaTeX~has~been~asked~to~create~a~DeviceN~color~space,~ 32249
32250 but~no~\\ \\ 32250
32251 \iow_indent:n { names~==<names> } 32251
32252 \\ \\ 32252
32253 key~was~given~with~the~correct~information. 32253
32254 } 32254
32255 \msg_new:nnnn { color } { ICC-Device-unknown } 32255
32256 { Unknown~device~color~space~'#1'. } 32256
32257 { 32257
32258 LaTeX~has~been~asked~to~apply~an~ICC~profile~but~the~device~color~space~ 32258
32259 '#1'~is~unknown. 32259
32260 } 32260
32261 \msg_new:nnnn { color } { ICCBased-unsupported-colorspace } 32261
32262 { ICCBased~color~space~'#1'~uses~an~unsupported~data~color~space. } 32262
32263 { 32263
32264 LaTeX~has~been~asked~to~create~a~ICCBased~colorspace,~but~the~ 32264
32265 used~data~colorspace~is~not~supported.~ICC~profiles~used~for~ 32265
32266 defining~a~ICCBased~colorspace~should~use~a~Lab,~RGB,~or~ 32266
32267 CMYK~data~colorspace.~LaTeX~will~ignore~this~request. 32267
32268 } 32268
32269 \msg_new:nnnn { color } { ICCBased-requires-file } 32269
```

```
32270 { ICCBased~color~space~'#1'~require~an~file. } 32270
32271 { 32271
32272 LaTeX~has~been~asked~to~create~an~ICCBased~color~space,~but~no~\\ \\ 32272
32273 \iow_indent:n { file~==<name> } 32273
32274 \\ \\ 32274
32275 key~was~given~with~the~correct~information.~LaTeX~will~ignore~this~ 32275
32276 request. 32276
32277 } 32277
32278 \msg_new:nnnn { color } { model-already-defined } 32278
32279 { Color~model~'#1'~already~defined. } 32279
32280 { 32280
32281 LaTeX~was~asked~to~define~a~new~color~model~called~'#1',~but~ 32281
32282 this~color~model~already~exists. 32282
32283 } 32283
32284 \msg_new:nnnn { color } { out-of-range } 32284
32285 { Input~value~#1~out~of~range~[#2,~#3]. } 32285
32286 { 32286
32287 LaTeX~was~expecting~a~value~in~the~range~[#2,~#3]~as~part~of~a~color,~ 32287
32288 but~you~gave~#1.~LaTeX~will~assume~you~meant~the~limit~of~the~range~ 32288
32289 and~continue. 32289
32290 } 32290
32291 \msg_new:nnnn { color } { separation-alternative-model } 32291
32292 { Separation~color~space~'#1'~require~an~alternative~model. } 32292
32293 { 32293
32294 LaTeX~has~been~asked~to~create~a~separation~color~space,~ 32294
32295 but~no~\\ \\ 32295
32296 \iow_indent:n { alternative-model~==<model> } 32296
32297 \\ \\ 32297
32298 key~was~given~with~the~correct~information. 32298
32299 } 32299
32300 \msg_new:nnnn { color } { separation-alternative-values } 32300
32301 { Separation~color~space~'#1'~require~values~for~the~alternative~space. } 32301
32302 { 32302
32303 LaTeX~has~been~asked~to~create~a~separation~color~space,~ 32303
32304 but~no~\\ \\ 32304
32305 \iow_indent:n { alternative-values~==<model> } 32305
32306 \\ \\ 32306
32307 key~was~given~with~the~correct~information. 32307
32308 } 32308
32309 \msg_new:nnnn { color } { separation-requires-name } 32309
32310 { Separation~color~space~'#1'~require~a~formal~name. } 32310
32311 { 32311
32312 LaTeX~has~been~asked~to~create~a~separation~color~space,~ 32312
32313 but~no~\\ \\ 32313
32314 \iow_indent:n { name~==<formal~name> } 32314
32315 \\ \\ 32315
```

32316	key~was~given~with~the~correct~information.	32316
32317	}	32317
32318	\msg_new:nnn { color } { unhandled-model }	32318
32319	{	32319
32320	Unhandled~color~model~in~LaTeX2e~value~"#1":	32320
32321	\\ \\	32321
32322	falling~back~on~grayscale.	32322
32323	}	32323
32324	\msg_new:nnnn { color } { unknown-color }	32324
32325	{ Unknown~color~'#1'. }	32325
32326	{	32326
32327	LaTeX~has~been~asked~to~use~a~color~named~'#1',~	32327
32328	but~this~has~never~been~defined.	32328
32329	}	32329
32330	\msg_new:nnnn { color } { unknown-alternative-model }	32330
32331	{ Separation~color~space~'#1'~require~an~valid~alternative~space. }	32331
32332	{	32332
32333	LaTeX~has~been~asked~to~create~a~separation~color~space,~	32333
32334	but~the~model~given~as\\ \\	32334
32335	\iow_indent:n { alternative-model~==<model> }	32335
32336	\\ \\	32336
32337	is~unknown.	32337
32338	}	32338
32339	\msg_new:nnnn { color } { unknown-export-format }	32339
32340	{ Unknown~export~format~'#1'. }	32340
32341	{	32341
32342	LaTeX~has~been~asked~to~export~a~color~in~format~'#1',~	32342
32343	but~this~has~never~been~defined.	32343
32344	}	32344
32345	\msg_new:nnnn { color } { unknown-CIELAB-illuminant }	32345
32346	{ Unknown~illuminant~model~'#1'. }	32346
32347	{	32347
32348	LaTeX~has~been~asked~to~use~create~a~color~space~using~CIELAB~	32348
32349	illuminant~'#1',~but~this~does~not~exist.	32349
32350	}	32350
32351	\msg_new:nnnn { color } { unknown-model }	32351
32352	{ Unknown~color~model~'#1'. }	32352
32353	{	32353
32354	LaTeX~has~been~asked~to~use~a~color~model~called~'#1',~	32354
32355	but~this~model~is~not~set~up.	32355
32356	}	32356
32357	\msg_new:nnnn { color } { unknown-model-type }	32357
32358	{ Unknown~color~model~type~'#1'. }	32358
32359	{	32359
32360	LaTeX~has~been~asked~to~create~a~new~color~model~called~'#1',~	32360
32361	but~this~type~of~model~was~never~set~up.	32361

```

32362 }
32363 \prop_gput:Nnn \g_msg_module_name_prop { color } { LaTeX }
32364 \prop_gput:Nnn \g_msg_module_type_prop { color } { }
32365 \msg_new:nnn { color } { show }
32366 {
32367   The~color~#1~
32368   \tl_if_empty:nTF {#2}
32369     { is~undefined. }
32370     { has~the~properties: #2 }
32371 }
32372 %% File: l3graphics.dtx
32373 \cs_if_exist:NT \@expl@finalise@setup@@
32374 {
32375   \tl_gput_right:Nn \@expl@finalise@setup@@
32376     { \declare@file@substitution { l3graphics.sty } { null.tex } }
32377 }
32378 \dim_new:N \l__graphics_internal_dim
32379 \ior_new:N \l__graphics_internal_ior
32380 \tl_new:N \l__graphics_internal_tl
32381 \scan_new:N \s__graphics_stop
32382 \tl_new:N \l__graphics_pagebox_tl
32383 \keys_define:nn { graphics }
32384 {
32385   decodearray .str_set:N =
32386     \l__graphics_decodearray_str ,
32387   draft .bool_set:N =
32388     \l__graphics_draft_bool ,
32389   interpolate .bool_set:N =
32390     \l__graphics_interpolate_bool ,
32391   pagebox .choices:nn =
32392     { art , bleed , crop , media , trim }
32393     {
32394       \tl_set:Ne \l__graphics_pagebox_tl
32395         { \l_keys_choice_tl box }
32396     } ,
32397   pagebox .initial:n =
32398     crop ,
32399   page .int_set:N =
32400     \l__graphics_page_int ,
32401   pdf-attr .str_set:N =
32402     \l__graphics_pdf_str ,
32403   type . str_set:N =
32404     \l__graphics_type_str
32405 }
32406 \dim_new:N \l__graphics_llx_dim
32407 \dim_new:N \l__graphics_lly_dim

```

```
32408 \dim_new:N \l__graphics_urx_dim 32408
32409 \dim_new:N \l__graphics_ury_dim 32409
32410 \cs_new_protected:Npn \__graphics_bb_save:n #1 32410
32411 { 32411
32412 \dim_if_exist:cTF { c__graphics_ #1 _urx_dim } 32412
32413 { \msg_error:nnn { graphic } { bb-already-cached } {#1} } 32413
32414 { 32414
32415 \dim_compare:nNnF \l__graphics_llx_dim = { Opt } 32415
32416 { \dim_const:cn { c__graphics_ #1 _llx_dim } { \l__graphics_llx_dim } } 32416
32417 \dim_compare:nNnF \l__graphics_lly_dim = { Opt } 32417
32418 { \dim_const:cn { c__graphics_ #1 _lly_dim } { \l__graphics_lly_dim } } 32418
32419 \dim_const:cn { c__graphics_ #1 _urx_dim } { \l__graphics_urx_dim } 32419
32420 \dim_const:cn { c__graphics_ #1 _ury_dim } { \l__graphics_ury_dim } 32420
32421 } 32421
32422 } 32422
32423 \cs_generate_variant:Nn \__graphics_bb_save:n { e } 32423
32424 \cs_new_protected:Npn \__graphics_bb_restore:nF #1#2 32424
32425 { 32425
32426 \dim_if_exist:cTF { c__graphics_ #1 _urx_dim } 32426
32427 { 32427
32428 \dim_set_eq:Nc \l__graphics_urx_dim { c__graphics_ #1 _urx_dim } 32428
32429 \dim_set_eq:Nc \l__graphics_ury_dim { c__graphics_ #1 _ury_dim } 32429
32430 \dim_if_exist:cTF { c__graphics_ #1 _llx_dim } 32430
32431 { \dim_set_eq:Nc \l__graphics_llx_dim { c__graphics_ #1 _llx_dim } } 32431
32432 { \dim_zero:N \l__graphics_llx_dim } 32432
32433 \dim_if_exist:cTF { c__graphics_ #1 _lly_dim } 32433
32434 { \dim_set_eq:Nc \l__graphics_lly_dim { c__graphics_ #1 _lly_dim } } 32434
32435 { \dim_zero:N \l__graphics_lly_dim } 32435
32436 } 32436
32437 {#2} 32437
32438 } 32438
32439 \cs_generate_variant:Nn \__graphics_bb_restore:nF { e } 32439
32440 \cs_new_protected:Npn \__graphics_extract_bb:n #1 32440
32441 { 32441
32442 \int_compare:nNnTF \l__graphics_page_int > 0 32442
32443 { \__graphics_extract_bb_auxi:Vn \l__graphics_page_int {#1} } 32443
32444 { \__graphics_extract_bb_auxii:nnn {#1} { } { } } 32444
32445 } 32445
32446 \cs_new_protected:Npn \__graphics_extract_bb_auxi:nn #1#2 32446
32447 { \__graphics_extract_bb_auxii:nnn {#2} { :P #1 } { -p~#1~ } } 32447
32448 \cs_generate_variant:Nn \__graphics_extract_bb_auxi:nn { Vn } 32448
32449 \cs_new_protected:Npn \__graphics_extract_bb_auxii:nnn #1#2#3 32449
32450 { 32450
32451 \tl_if_empty:NTF \l__graphics_pagebox_tl 32451
32452 { \__graphics_extract_bb_auxiv:nnn } 32452
32453 { \__graphics_extract_bb_auxiii:Vnnn \l__graphics_pagebox_tl } 32453
```



```
32454 {#1} {#2} {#3} 32454
32455 } 32455
32456 \cs_new_protected:Npn \__graphics_extract_bb_auxiii:nnnn #1#2#3#4 32456
32457 { \__graphics_extract_bb_auxiv:nnn {#2} { : #1 #3 } { #4 -B~#1~ } } 32457
32458 \cs_generate_variant:Nn \__graphics_extract_bb_auxiii:nnnn { V } 32458
32459 \cs_new_protected:Npn \__graphics_extract_bb_auxiv:nnn #1#2#3 32459
32460 { 32460
32461 \__graphics_read_bb_auxi:nnnn {#1} {#2} 32461
32462 { \ior_shell_open:Nn \l__graphics_internal_ior { extractbb~#3-0~#1 } } 32462
32463 { pipe-failed } 32463
32464 } 32464
32465 \cs_new_protected:Npn \__graphics_read_bb:n #1 32465
32466 { 32466
32467 \__graphics_read_bb_auxi:nnnn {#1} { } 32467
32468 { \ior_open:Nn \l__graphics_internal_ior {#1} } 32468
32469 { graphic-not-found } 32469
32470 } 32470
32471 \cs_new_protected:Npn \__graphics_read_bb_auxi:nnnn #1#2#3#4 32471
32472 { 32472
32473 \__graphics_bb_restore:nF {#1#2} 32473
32474 { \__graphics_read_bb_auxii:nnnn {#3} {#4} {#1} {#2} } 32474
32475 } 32475
32476 \cs_new_protected:Npe \__graphics_read_bb_auxii:nnnn #1#2#3#4 32476
32477 { 32477
32478 #1 32478
32479 \exp_not:N \ior_if_eof:NTF \exp_not:N \l__graphics_internal_ior 32479
32480 { \msg_error:nnn { graphics } {#2} {#3} } 32480
32481 { 32481
32482 \ior_str_map_inline:Nn \exp_not:N \l__graphics_internal_ior 32482
32483 { 32483
32484 \exp_not:N \__graphics_read_bb_auxiii:w 32484
32485 ##1 ~ \c_colon_str \s__graphics_stop 32485
32486 } 32486
32487 \__graphics_bb_save:n {#3#4} 32487
32488 } 32488
32489 \ior_close:N \exp_not:N \l__graphics_internal_ior 32489
32490 } 32490
32491 \use:e 32491
32492 { 32492
32493 \cs_new_protected:Npn \exp_not:N \__graphics_read_bb_auxiii:w 32493
32494 #1 \c_colon_str #2 \s__graphics_stop 32494
32495 { 32495
32496 \exp_not:N \str_if_eq:nnT 32496
32497 { \c_percent_str \c_percent_str BoundingBox } 32497
32498 {#1} 32498
32499 { \exp_not:N \__graphics_read_bb_auxiv:w #2 ( ) \s__graphics_stop } 32499
```

```

32500     }
32501 }
32502 \cs_new_protected:Npn \__graphics_read_bb_auxiv:w #1 ( #2 ) #3 \s__graphics_stop
32503 {
32504     \str_if_eq:nnF {#2} { atend }
32505     {
32506         \tl_set_rescan:Nne \l__graphics_internal_tl
32507         {
32508             \char_set_catcode_space:n { 9 }
32509             \char_set_catcode_space:n { 32 }
32510         }
32511         { \use:n #1 }
32512         \exp_after:wN \__graphics_read_bb_auxv:w \l__graphics_internal_tl
32513     }
32514 }
32515 \cs_new_protected:Npn \__graphics_read_bb_auxv:w #1~#2~#3~#4~#5 \s__graphics_stop
32516 {
32517     \dim_set:Nn \l__graphics_llx_dim { #1 bp }
32518     \dim_set:Nn \l__graphics_lly_dim { #2 bp }
32519     \dim_set:Nn \l__graphics_urx_dim { #3 bp }
32520     \dim_set:Nn \l__graphics_ury_dim { #4 bp }
32521     \ior_map_break:
32522 }
32523 \str_new:N \l__graphics_final_name_str
32524 \str_new:N \l__graphics_full_name_str
32525 \box_new:N \l__graphics_internal_box
32526 \str_new:N \l__graphics_dir_str
32527 \str_new:N \l__graphics_name_str
32528 \str_new:N \l__graphics_ext_str
32529 \seq_new:N \l__graphics_search_path_seq
32530 \seq_new:N \l__graphics_search_ext_seq
32531 \prop_new:N \l__graphics_ext_type_prop
32532 \prop_put:Nnn \l__graphics_ext_type_prop { .ps } { eps }
32533 \seq_new:N \g__graphics_record_seq
32534 \cs_new_protected:Npn \graphics_include:nn #1#2
32535 {
32536     \group_begin:
32537     \keys_set:nn { graphics } {#1}
32538     \seq_set_eq:NN \l_file_search_path_seq \l__graphics_search_path_seq
32539     \file_get_full_name:nNTF {#2} \l__graphics_full_name_str
32540     {
32541         \str_if_eq:eeTF { \l__graphics_full_name_str } { #2 .tex }
32542         { \msg_error:nnn { graphics } { graphic-not-found } {#2} }
32543         { \__graphics_include: }
32544     }

```

```
32543 { \msg_error:nnn { graphics } { graphic-not-found } {#2} } 32543
32544 \group_end: 32544
32545 } 32545
32546 \cs_generate_variant:Nn \graphics_include:nn { nV } 32546
32547 \cs_new_protected:Npn \__graphics_include: 32547
32548 { 32548
32549 \str_if_empty:NTF \l__graphics_type_str 32549
32550 { 32550
32551 \file_parse_full_name:VNNN \l__graphics_full_name_str 32551
32552 \l__graphics_dir_str \l__graphics_name_str \l__graphics_ext_str 32552
32553 \__graphics_include_auxi:e 32553
32554 { 32554
32555 \exp_args:Ne \str_tail:n 32555
32556 { \str_casefold:V \l__graphics_ext_str } 32556
32557 } 32557
32558 } 32558
32559 { \__graphics_include_auxi:e { \l__graphics_type_str } } 32559
32560 } 32560
32561 \cs_new_protected:Npn \__graphics_include_auxi:n #1 32561
32562 { 32562
32563 \prop_get:NnNF \l_graphics_ext_type_prop { .#1 } \l__graphics_internal_tl 32563
32564 { \tl_set:Nn \l__graphics_internal_tl {#1} } 32564
32565 \exp_args:NV \__graphics_include_auxii:n \l__graphics_internal_tl 32565
32566 } 32566
32567 \cs_generate_variant:Nn \__graphics_include_auxi:n { e } 32567
32568 \cs_new_protected:Npn \__graphics_include_auxii:n #1 32568
32569 { 32569
32570 \mode_leave_vertical: 32570
32571 \cs_if_exist:cTF { __graphics_backend_include_ #1 :n } 32571
32572 { 32572
32573 \tl_set_eq:NN \l__graphics_final_name_str \l__graphics_full_name_str 32573
32574 \str_set:Ne \l__graphics_full_name_str 32574
32575 { \exp_args:NV \__kernel_file_name_quote:n \l__graphics_full_name_str } 32575
32576 \exp_args:NnV \use:c { __graphics_backend_getbb_ #1 :n } 32576
32577 \l__graphics_full_name_str 32577
32578 \seq_gput_right:NV \g__graphics_record_seq \l__graphics_final_name_str 32578
32579 \clist_if_exist:NT \@filelist 32579
32580 { \exp_args:NV \@addtofilelist \l__graphics_final_name_str } 32580
32581 \bool_if:NTF \l__graphics_draft_bool 32581
32582 { \__graphics_include_auxiii:n } 32582
32583 { \__graphics_include_auxiv:n } 32583
32584 {#1} 32584
32585 } 32585
32586 { \msg_error:nnn { graphics } { unsupported-graphic-type } {#1} } 32586
32587 } 32587
32588 \cs_new_protected:Npn \__graphics_include_auxiii:n #1 32588
32589 } 32589
32590 \cs_new_protected:Npn \__graphics_include_auxiii:n #1 32590
```

```

32591 { 32591
32592 \hbox_to_wd:nn { \l__graphics_urx_dim - \l__graphics_llx_dim } 32592
32593 { 32593
32594 \tex_vrule:D 32594
32595 \tex_hss:D 32595
32596 \vbox_to_ht:nn 32596
32597 { \l__graphics_ury_dim - \l__graphics_lly_dim } 32597
32598 { 32598
32599 \tex_hrulerule:D width 32599
32600 \dim_eval:n { \l__graphics_urx_dim - \l__graphics_llx_dim } 32600
32601 \tex_vss:D 32601
32602 \hbox_to_wd:nn 32602
32603 { \l__graphics_urx_dim - \l__graphics_llx_dim } 32603
32604 { 32604
32605 \ttfamily 32605
32606 \tex_hss:D \l__graphics_full_name_str \tex_hss:D 32606
32607 } 32607
32608 \tex_vss:D 32608
32609 \tex_hrulerule:D 32609
32610 } 32610
32611 \tex_hss:D 32611
32612 \tex_vrule:D 32612
32613 } 32613
32614 } 32614
32615 \cs_new_protected:Npn \__graphics_include_auxiv:n #1 32615
32616 { 32616
32617 \hbox_set:Nn \l__graphics_internal_box 32617
32618 { 32618
32619 \exp_args:NnV \use:c { __graphics_backend_include_ #1 :n } 32619
32620 \l__graphics_full_name_str 32620
32621 } 32621
32622 \box_set_dp:Nn \l__graphics_internal_box { Opt } 32622
32623 \box_set_ht:Nn \l__graphics_internal_box 32623
32624 { \l__graphics_ury_dim - \l__graphics_lly_dim } 32624
32625 \box_set_wd:Nn \l__graphics_internal_box 32625
32626 { \l__graphics_urx_dim - \l__graphics_llx_dim } 32626
32627 \box_use_drop:N \l__graphics_internal_box 32627
32628 } 32628
32629 \cs_new_protected:Npn \graphics_show_list: { \__graphics_list:N \msg_show:nneeee } 32629
32630 \cs_new_protected:Npn \graphics_log_list: { \__graphics_list:N \msg_log:nneeee } 32630
32631 \cs_new_protected:Npn \__graphics_list:N #1 32631
32632 { 32632
32633 \seq_remove_duplicates:N \g__graphics_record_seq 32633
32634 #1 { kernel } { file-list } 32634
32635 { \seq_map_function:NN \g__graphics_record_seq \__graphics_list_aux:n } 32635
32636 { } { } { } { } 32636

```

```
32637 } 32637
32638 \cs_new:Npn \__graphics_list_aux:n #1 { \iow_newline: #1 } 32638
32639 \cs_new_protected:Npn \graphics_get_full_name:nN #1#2 32639
32640 { 32640
32641 \graphics_get_full_name:nNF {#1} #2 32641
32642 { \tl_set:Nn #2 { \q_no_value } } 32642
32643 } 32643
32644 \prg_new_protected_conditional:Npnn \graphics_get_full_name:nN #1#2 32644
32645 { T , F , TF } 32645
32646 { 32646
32647 \group_begin: 32647
32648 \seq_set_eq:NN \l_file_search_path_seq \l_graphics_search_path_seq 32648
32649 \file_get_full_name:nNTF {#1} \l__graphics_full_name_str 32649
32650 { 32650
32651 \str_if_eq:eeTF { \l__graphics_full_name_str } { #1 .tex } 32651
32652 { \__graphics_get_full_name:n {#1} } 32652
32653 { 32653
32654 \file_parse_full_name:VNNN \l__graphics_full_name_str 32654
32655 \l__graphics_dir_str \l__graphics_name_str \l__graphics_ext_str 32655
32656 \seq_map_inline:Nn \l_graphics_search_ext_seq 32656
32657 { 32657
32658 \str_if_eq:nVT {##1} \l__graphics_ext_str 32658
32659 { \seq_map_break:n { \use_none:nn } } 32659
32660 } 32660
32661 \__graphics_get_full_name:n {#1} 32661
32662 } 32662
32663 } 32663
32664 { \__graphics_get_full_name:n {#1} } 32664
32665 \exp_args:NNNV \group_end: 32665
32666 \tl_set:Nn #2 \l__graphics_full_name_str 32666
32667 \tl_if_empty:NTF #2 32667
32668 { \prg_return_false: } 32668
32669 { \prg_return_true: } 32669
32670 } 32670
32671 \cs_new_protected:Npn \__graphics_get_full_name:n #1 32671
32672 { 32672
32673 \str_clear:N \l__graphics_full_name_str 32673
32674 \seq_map_inline:Nn \l_graphics_search_ext_seq 32674
32675 { 32675
32676 \file_get_full_name:nNT { #1 ##1 } \l__graphics_full_name_str 32676
32677 { \seq_map_break:n { \use_none:nn } } 32677
32678 } 32678
32679 \use:n 32679
32680 { \str_clear:N \l__graphics_full_name_str } 32680
32681 } 32681
32682 \cs_new_protected:Npn \graphics_get_pagecount:nN #1#2 32682
```

```

32683 {
32684 \group_begin:
32685 \seq_set_eq:NN \l_file_search_path_seq \l_graphics_search_path_seq
32686 \file_get_full_name:nNTF {#1} \l__graphics_full_name_str
32687 {
32688 \int_if_exist:cF { c__graphics_ \l__graphics_full_name_str _pages_int }
32689 {
32690 \exp_args:NV \__graphics_backend_get_pagecount:n
32691 \l__graphics_full_name_str
32692 }
32693 \tl_set:Nv #2 { c__graphics_ \l__graphics_full_name_str _pages_int }
32694 }
32695 {
32696 \tl_set:Nn #2 { 0 }
32697 \msg_error:nnn { graphics } { graphic-not-found } {#1}
32698 }
32699 \exp_args:NNNV \group_end:
32700 \tl_set:Nn #2 #2
32701 }
32702 \cs_new_protected:Npe \__graphics_get_pagecount:n #1
32703 {
32704 \exp_not:N \ior_shell_open:Nn \exp_not:N \l__graphics_internal_ior
32705 { extractbb~-0~#1 }
32706 \exp_not:N \ior_if_eof:NTF \exp_not:N \l__graphics_internal_ior
32707 { \msg_error:nnn { graphics } { pipe-failed } }
32708 {
32709 \ior_str_map_inline:Nn \exp_not:N \l__graphics_internal_ior
32710 {
32711 \exp_not:N \__graphics_get_pagecount:nw {#1}
32712 ##1 ~ \c_colon_str \c_colon_str \s__graphics_stop
32713 }
32714 \exp_not:N \int_if_exist:cF { c__graphics_ #1 _pages_int }
32715 { \int_const:cn { c__graphics_ #1 _pages_int } { 1 } }
32716 }
32717 \ior_close:N \exp_not:N \l__graphics_internal_ior
32718 }
32719 \use:e
32720 {
32721 \cs_new_protected:Npn \exp_not:N \__graphics_get_pagecount:nw
32722 #1#2 \c_colon_str #3 \c_colon_str #4 \s__graphics_stop
32723 {
32724 \exp_not:N \str_if_eq:nnT
32725 { \c_percent_str \c_percent_str Pages }
32726 {#2}
32727 {
32728 \int_const:cn { c__graphics_ #1 _pages_int } {#3}

```



```

32729 \exp_not:N \ior_map_break:
32730 }
32731 }
32732 }
32733 \msg_new:nnnn { graphics } { graphic-not-found }
32734 { Image~file~'#1'~not~found. }
32735 {
32736 LaTeX~tried~to~open~graphic~file~'#1',~
32737 but~the~file~could~not~be~read.
32738 }
32739 \msg_new:nnnn { graphics } { pipe-failed }
32740 { Cannot~run~piped~system~commands. }
32741 {
32742 LaTeX~tried~to~call~a~system~process~but~this~was~not~possible.\\
32743 Try~the~"--shell-escape"~(or~"--enable-pipes")~option.
32744 }
32745 \msg_new:nnnn { graphics } { unsupported-graphic-type }
32746 { Image~type~'#1'~not~supported~by~current~driver. }
32747 {
32748 LaTeX~was~asked~to~include~an~graphic~of~type~'#1',~
32749 but~this~is~not~supported~by~the~current~driver~(production~route).
32750 }
32751 %% File: l3opacity.dtx
32752 \cs_if_exist:NT \@expl@finalise@setup@@
32753 {
32754 \tl_gput_right:Nn \@expl@finalise@setup@@
32755 { \declare@file@substitution { l3opacity.sty } { null.tex } }
32756 }
32757 \fp_new:N \l__opacity_tmp_fp
32758 \cs_new_protected:Npn \opacity_select:n #1
32759 { \__opacity_select:nN {#1} \__opacity_backend_select:n }
32760 \cs_new_protected:Npn \opacity_fill:n #1
32761 { \__opacity_select:nN {#1} \__opacity_backend_fill:n }
32762 \cs_new_protected:Npn \opacity_stroke:n #1
32763 { \__opacity_select:nN {#1} \__opacity_backend_stroke:n }
32764 \cs_new_protected:Npn \__opacity_select:nN #1#2
32765 {
32766 \fp_set:Nn \l__opacity_tmp_fp { #1 }
32767 \bool_lazy_or:nnTF
32768 { \fp_compare_p:nNn \l__opacity_tmp_fp < \c_zero_fp }
32769 { \fp_compare_p:nNn \l__opacity_tmp_fp > \c_one_fp }
32770 { \msg_error:nnn { opacity } { out-of-range } {#1} }
32771 { \exp_args:Ne #2 { \fp_use:N \l__opacity_tmp_fp } }
32772 }
32773 \msg_new:nnnn { opacity } { out-of-range }
32774 { Opacity~value~out~of~range. }

```



```
32775 { 32775
32776 LaTeX~was~asked~to~set~opacity~of~#1,~but~only~values~in~the~range~ 32776
32777 0~to~1~are~supported. 32777
32778 } 32778
32779 %% File: l3pdf.dtx 32779
32780 \scan_new:N \s__pdf_stop 32780
32781 \bool_new:N \g__pdf_init_bool 32781
32782 \bool_lazy_and:nnT 32782
32783 { \str_if_eq_p:Vn \fmtname { LaTeX2e } } 32783
32784 { \tl_if_exist_p:N \@expl@finalise@setup@@ } 32784
32785 { 32785
32786 \tl_gput_right:Nn \@expl@finalise@setup@@ 32786
32787 { 32787
32788 \tl_gput_right:Nn \@kernel@after@begindocument 32788
32789 { \bool_gset_true:N \g__pdf_init_bool } 32789
32790 } 32790
32791 } 32791
32792 \cs_new_protected:Npn \pdf_uncompress: 32792
32793 { 32793
32794 \bool_if:NF \g__pdf_init_bool 32794
32795 { 32795
32796 \__pdf_backend_compresslevel:n { 0 } 32796
32797 \__pdf_backend_compress_objects:n { \c_false_bool } 32797
32798 } 32798
32799 } 32799
32800 \int_new:N \g__pdf_backend_object_int 32800
32801 \cs_new_protected:Npn \pdf_object_new:n #1 32801
32802 { 32802
32803 \__pdf_backend_object_new: 32803
32804 \__pdf_object_record:nN {#1} \g__pdf_backend_object_int 32804
32805 } 32805
32806 \cs_new_protected:Npn \pdf_object_write:nnn #1#2#3 32806
32807 { 32807
32808 \exp_args:Ne \__pdf_backend_object_write:nnn 32808
32809 { \__pdf_object_retrieve:n {#1} } {#2} {#3} 32809
32810 \bool_gset_true:N \g__pdf_init_bool 32810
32811 } 32811
32812 \cs_generate_variant:Nn \pdf_object_write:nnn { nne , nnx } 32812
32813 \cs_new:Npn \pdf_object_ref:n #1 32813
32814 { 32814
32815 \exp_args:Ne \__pdf_backend_object_ref:n 32815
32816 { \__pdf_object_retrieve:n {#1} } 32816
32817 } 32817
32818 \cs_new:Npn \__kernel_pdf_object_id:n #1 32818
32819 { 32819
32820 \exp_args:Ne \__pdf_backend_object_id:n 32820
```

```

32821 { \__pdf_object_retrieve:n {#1} }
32822 }
32823 \sys_if_engine luatex:F
32824 {
32825   \cs_new_protected:Npn \__pdf_object_record:nN #1#2
32826   {
32827     \int_const:cn
32828     { c__pdf_object_ #1 _int } {#2}
32829   }
32830   \cs_new:Npn \__pdf_object_retrieve:n #1
32831   {
32832     \int_if_exist:cTF { c__pdf_object_ #1 _int }
32833     {
32834       \int_use:c
32835       { c__pdf_object_ #1 _int }
32836     }
32837     { 0 }
32838   }
32839 }
32840 \prg_new_conditional:Npnn \pdf_object_if_exist:n #1 { p , T , F , TF }
32841 {
32842   \int_compare:nNnTF { \__pdf_object_retrieve:n {#1} } = 0
32843   \prg_return_false:
32844   \prg_return_true:
32845 }
32846 \cs_new_protected:Npn \pdf_object_new_indexed:nn #1#2
32847 {
32848   \__pdf_backend_object_new:
32849   \__pdf_object_record:neN {#1}
32850   { \int_eval:n {#2} } \g__pdf_backend_object_int
32851 }
32852 \cs_new_protected:Npn \pdf_object_write_indexed:nnnn #1#2#3#4
32853 {
32854   \exp_args:Ne \__pdf_backend_object_write:nnn
32855   { \__pdf_object_retrieve:ne {#1} { \int_eval:n {#2} } } {#3} {#4}
32856   \bool_gset_true:N \g__pdf_init_bool
32857 }
32858 \cs_generate_variant:Nn \pdf_object_write_indexed:nnnn { nnne }
32859 \cs_new:Npn \pdf_object_ref_indexed:nn #1#2
32860 {
32861   \exp_args:Ne \__pdf_backend_object_ref:n
32862   { \__pdf_object_retrieve:ne {#1} { \int_eval:n {#2} } }
32863 }
32864 \cs_new:Npn \__kernel_pdf_object_id_indexed:nn #1#2
32865 {
32866   \exp_args:Ne \__pdf_backend_object_id:n

```

```

32867 { \_pdf_object_retrieve:ne {#1} { \int_eval:n {#2} } } 32867
32868 } 32868
32869 \sys_if_engine luatex:F 32869
32870 { 32870
32871 \cs_new_protected:Npn \_pdf_object_record:nnN #1#2#3 32871
32872 { 32872
32873 \use:e 32873
32874 { 32874
32875 \_pdf_object_record:NnN 32875
32876 \_pdf_object_index_split:nn {#1} {#2} 32876
32877 \exp_not:N #3 32877
32878 } 32878
32879 } 32879
32880 \cs_new_protected:Npn \_pdf_object_record:NnN #1#2#3 32880
32881 { 32881
32882 \intarray_if_exist:NF #1 32882
32883 { \intarray_new:Nn #1 \c_pdf_object_block_size_int } 32883
32884 \intarray_gset:Nnn #1 {#2} #3 32884
32885 } 32885
32886 \cs_new:Npn \_pdf_object_retrieve:nn #1#2 32886
32887 { 32887
32888 \use:e 32888
32889 { 32889
32890 \exp_not:N \_pdf_object_retrieve:Nn 32890
32891 \_pdf_object_index_split:nn {#1} {#2} 32891
32892 } 32892
32893 } 32893
32894 \cs_new:Npn \_pdf_object_retrieve:Nn #1#2 32894
32895 { \intarray_item:Nn #1 {#2} } 32895
32896 \cs_new:Npn \_pdf_object_index_split:nn #1#2 32896
32897 { 32897
32898 \exp_not:c 32898
32899 { 32899
32900 g__pdf_object_ #1 _ 32900
32901 \int_eval:n 32901
32902 { 32902
32903 \int_div_truncate:nn { #2 - 1 } 32903
32904 \c_pdf_object_block_size_int + 1 32904
32905 } 32905
32906 _intarray 32906
32907 } 32907
32908 { 32908
32909 \int_eval:n 32909
32910 { \int_mod:nn { #2 - 1 } \c_pdf_object_block_size_int + 1 } 32910
32911 } 32911
32912 } 32912

```

```
32913 \int_const:Nn \c__pdf_object_block_size_int { 10000 } 32913
32914 } 32914
32915 \cs_generate_variant:Nn \__pdf_object_record:nnN { ne } 32915
32916 \cs_generate_variant:Nn \__pdf_object_retrieve:nn { ne } 32916
32917 \cs_new_protected:Npn \pdf_object_unnamed_write:nn #1#2 32917
32918 { 32918
32919 \exp_args:Ne \__pdf_backend_object_now:nn {#1} {#2} 32919
32920 \bool_gset_true:N \g__pdf_init_bool 32920
32921 } 32921
32922 \cs_generate_variant:Nn \pdf_object_unnamed_write:nn { ne , nx } 32922
32923 \cs_new:Npn \pdf_object_ref_last: { \__pdf_backend_object_last: } 32923
32924 \cs_new:Npn \pdf_pageobject_ref:n #1 32924
32925 { \exp_args:Ne \__pdf_backend_pageobject_ref:n {#1} } 32925
32926 \prg_new_conditional:Npnn \pdf_version_compare:Nn #1#2 { p , T , F , TF } 32926
32927 { \use:c { __pdf_version_compare_ #1 :w } #2 . . \s__pdf_stop } 32927
32928 \cs_new:cpn { __pdf_version_compare_=:w } #1 . #2 . #3 \s__pdf_stop 32928
32929 { 32929
32930 \bool_lazy_and:nnTF 32930
32931 { \int_compare_p:nNn \__pdf_backend_version_major: = {#1} } 32931
32932 { \int_compare_p:nNn \__pdf_backend_version_minor: = {#2} } 32932
32933 { \prg_return_true: } 32933
32934 { \prg_return_false: } 32934
32935 } 32935
32936 \cs_new:cpn { __pdf_version_compare<:w } #1 . #2 . #3 \s__pdf_stop 32936
32937 { 32937
32938 \bool_lazy_or:nnTF 32938
32939 { \int_compare_p:nNn \__pdf_backend_version_major: < {#1} } 32939
32940 { 32940
32941 \bool_lazy_and_p:nn 32941
32942 { \int_compare_p:nNn \__pdf_backend_version_major: = {#1} } 32942
32943 { \int_compare_p:nNn \__pdf_backend_version_minor: < {#2} } 32943
32944 } 32944
32945 { \prg_return_true: } 32945
32946 { \prg_return_false: } 32946
32947 } 32947
32948 \cs_new:cpn { __pdf_version_compare>:w } #1 . #2 . #3 \s__pdf_stop 32948
32949 { 32949
32950 \bool_lazy_or:nnTF 32950
32951 { \int_compare_p:nNn \__pdf_backend_version_major: > {#1} } 32951
32952 { 32952
32953 \bool_lazy_and_p:nn 32953
32954 { \int_compare_p:nNn \__pdf_backend_version_major: = {#1} } 32954
32955 { \int_compare_p:nNn \__pdf_backend_version_minor: > {#2} } 32955
32956 } 32956
32957 { \prg_return_true: } 32957
32958 { \prg_return_false: } 32958
```

```

32959 }
32960 \cs_new_protected:Npn \pdf_version_gset:n #1
32961 { \__pdf_version_gset:w #1 . . \s__pdf_stop }
32962 \cs_new_protected:Npn \pdf_version_min_gset:n #1
32963 {
32964 \pdf_version_compare:NnT < {#1}
32965 { \__pdf_version_gset:w #1 . . \s__pdf_stop }
32966 }
32967 \cs_new_protected:Npn \__pdf_version_gset:w #1 . #2 . #3\s__pdf_stop
32968 {
32969 \bool_if:NF \g__pdf_init_bool
32970 {
32971 \__pdf_backend_version_major_gset:n {#1}
32972 \__pdf_backend_version_minor_gset:n {#2}
32973 }
32974 }
32975 \cs_new:Npn \pdf_version:
32976 { \__pdf_backend_version_major: . \__pdf_backend_version_minor: }
32977 \cs_new:Npn \pdf_version_major: { \__pdf_backend_version_major: }
32978 \cs_new:Npn \pdf_version_minor: { \__pdf_backend_version_minor: }
32979 \cs_new_protected:Npn \pdf_pagesize_gset:nn #1#2
32980 { \__pdf_backend_pagesize_gset:nn {#1} {#2} }
32981 \cs_new_protected:Npn \pdf_destination:nn #1#2
32982 { \__pdf_backend_destination:nn {#1} {#2} }
32983 \cs_new_protected:Npn \pdf_destination:nnnn #1#2#3#4
32984 {
32985 \hbox_to_zero:n
32986 { \__pdf_backend_destination:nnnn {#1} {#2} {#3} {#4} }
32987 }
32988 \cs_if_exist:NT \@kernel@before@begindocument
32989 {
32990 \tl_gput_right:Nn \@kernel@before@begindocument
32991 {
32992 \bool_lazy_all:nT
32993 {
32994 { \cs_if_exist_p:N \stockheight }
32995 { \cs_if_exist_p:N \stockwidth }
32996 { \cs_if_exist_p:N \IfDocumentMetadataTF }
32997 { \IfDocumentMetadataTF { \c_true_bool } { \c_false_bool } }
32998 { \int_compare_p:nNn \tex_mag:D = { 1000 } }
32999 }
33000 {
33001 \bool_lazy_and:nnTF
33002 { \dim_compare_p:nNn \stockheight > { 0pt } }
33003 { \dim_compare_p:nNn \stockwidth > { 0pt } }
33004 {

```

```

33005         \__pdf_backend_pagesize_gset:nn
33006         \stockwidth \stockheight
33007     }
33008     {
33009         \bool_lazy_or:nnF
33010         { \dim_compare_p:nNn \stockheight < { 0pt } }
33011         { \dim_compare_p:nNn \stockwidth < { 0pt } }
33012         {
33013             \bool_lazy_and:nnT
33014             { \dim_compare_p:nNn \paperheight > { 0pt } }
33015             { \dim_compare_p:nNn \paperwidth > { 0pt } }
33016             {
33017                 \__pdf_backend_pagesize_gset:nn
33018                 \paperwidth \paperheight
33019             }
33020         }
33021     }
33022 }
33023 }
33024 }
33025 %% File: l3coffins.dtx
33026 \box_new:N \l__coffin_internal_box
33027 \dim_new:N \l__coffin_internal_dim
33028 \tl_new:N \l__coffin_internal_tl
33029 \prop_const_from_keyval:Nn \c__coffin_corners_prop
33030 {
33031     tl = { 0pt } { 0pt } ,
33032     tr = { 0pt } { 0pt } ,
33033     bl = { 0pt } { 0pt } ,
33034     br = { 0pt } { 0pt } ,
33035 }
33036 \prop_const_from_keyval:Nn \c__coffin_poles_prop
33037 {
33038     l  = { 0pt } { 0pt } { 0pt } { 1000pt } ,
33039     hc = { 0pt } { 0pt } { 0pt } { 1000pt } ,
33040     r  = { 0pt } { 0pt } { 0pt } { 1000pt } ,
33041     b  = { 0pt } { 0pt } { 1000pt } { 0pt } ,
33042     vc = { 0pt } { 0pt } { 1000pt } { 0pt } ,
33043     t  = { 0pt } { 0pt } { 1000pt } { 0pt } ,
33044     B  = { 0pt } { 0pt } { 1000pt } { 0pt } ,
33045     H  = { 0pt } { 0pt } { 1000pt } { 0pt } ,
33046     T  = { 0pt } { 0pt } { 1000pt } { 0pt } ,
33047 }
33048 \fp_new:N \l__coffin_slope_A_fp
33049 \fp_new:N \l__coffin_slope_B_fp
33050 \bool_new:N \l__coffin_error_bool

```

33051	\dim_new:N \l__coffin_offset_x_dim	33051
33052	\dim_new:N \l__coffin_offset_y_dim	33052
33053	\tl_new:N \l__coffin_pole_a_tl	33053
33054	\tl_new:N \l__coffin_pole_b_tl	33054
33055	\dim_new:N \l__coffin_x_dim	33055
33056	\dim_new:N \l__coffin_y_dim	33056
33057	\dim_new:N \l__coffin_x_prime_dim	33057
33058	\dim_new:N \l__coffin_y_prime_dim	33058
33059	\cs_new_eq:NN __coffin_to_value:N \tex_number:D	33059
33060	\prg_new_conditional:Npnn \coffin_if_exist:N #1 { p , T , F , TF }	33060
33061	{	33061
33062	\cs_if_exist:NTF #1	33062
33063	{	33063
33064	\cs_if_exist:cTF { coffin ~ __coffin_to_value:N #1 ~ poles }	33064
33065	{ \prg_return_true: }	33065
33066	{ \prg_return_false: }	33066
33067	}	33067
33068	{ \prg_return_false: }	33068
33069	}	33069
33070	\prg_generate_conditional_variant:Nnn \coffin_if_exist:N	33070
33071	{ c } { p , T , F , TF }	33071
33072	\cs_new_protected:Npn __coffin_if_exist:NT #1#2	33072
33073	{	33073
33074	\coffin_if_exist:NTF #1	33074
33075	{ #2 }	33075
33076	{	33076
33077	\msg_error:nne { coffin } { unknown }	33077
33078	{ \token_to_str:N #1 }	33078
33079	}	33079
33080	}	33080
33081	\cs_new_protected:Npn \coffin_clear:N #1	33081
33082	{	33082
33083	__coffin_if_exist:NT #1	33083
33084	{	33084
33085	\box_clear:N #1	33085
33086	__coffin_reset_structure:N #1	33086
33087	}	33087
33088	}	33088
33089	\cs_generate_variant:Nn \coffin_clear:N { c }	33089
33090	\cs_new_protected:Npn \coffin_gclear:N #1	33090
33091	{	33091
33092	__coffin_if_exist:NT #1	33092
33093	{	33093
33094	\box_gclear:N #1	33094
33095	__coffin_greset_structure:N #1	33095
33096	}	33096


```
33097 } 33097
33098 \cs_generate_variant:Nn \coffin_gclear:N { c } 33098
33099 \cs_new_protected:Npn \coffin_new:N #1 33099
33100 { 33100
33101 \box_new:N #1 33101
33102 \debug_suspend: 33102
33103 \prop_gclear_new:c { coffin ~ \__coffin_to_value:N #1 ~ corners } 33103
33104 \prop_gclear_new:c { coffin ~ \__coffin_to_value:N #1 ~ poles } 33104
33105 \prop_gset_eq:cN { coffin ~ \__coffin_to_value:N #1 ~ corners } 33105
33106 \c__coffin_corners_prop 33106
33107 \prop_gset_eq:cN { coffin ~ \__coffin_to_value:N #1 ~ poles } 33107
33108 \c__coffin_poles_prop 33108
33109 \debug_resume: 33109
33110 } 33110
33111 \cs_generate_variant:Nn \coffin_new:N { c } 33111
33112 \cs_new_protected:Npn \hcoffin_set:Nn #1#2 33112
33113 { 33113
33114 \__coffin_if_exist:NT #1 33114
33115 { 33115
33116 \hbox_set:Nn #1 33116
33117 { 33117
33118 \color_ensure_current: 33118
33119 #2 33119
33120 } 33120
33121 \coffin_reset_poles:N #1 33121
33122 } 33122
33123 } 33123
33124 \cs_generate_variant:Nn \hcoffin_set:Nn { c } 33124
33125 \cs_new_protected:Npn \hcoffin_gset:Nn #1#2 33125
33126 { 33126
33127 \__coffin_if_exist:NT #1 33127
33128 { 33128
33129 \hbox_gset:Nn #1 33129
33130 { 33130
33131 \color_ensure_current: 33131
33132 #2 33132
33133 } 33133
33134 \coffin_greset_poles:N #1 33134
33135 } 33135
33136 } 33136
33137 \cs_generate_variant:Nn \hcoffin_gset:Nn { c } 33137
33138 \cs_new_protected:Npn \vcoffin_set:Nnn #1#2#3 33138
33139 { 33139
33140 \__coffin_set_vertical:NnnNNN #1 {#2} {#3} 33140
33141 \vbox_set:Nn \coffin_reset_poles:N \__coffin_set_pole:Nnn 33141
33142 } 33142
```

```

33143 \cs_generate_variant:Nn \vcoffin_set:Nnn { c } 33143
33144 \cs_new_protected:Npn \vcoffin_gset:Nnn #1#2#3 33144
33145 { 33145
33146   \__coffin_set_vertical:NnnNNN #1 {#2} {#3} 33146
33147   \vbox_gset:Nn \coffin_greset_poles:N \__coffin_gset_pole:Nnn 33147
33148 } 33148
33149 \cs_generate_variant:Nn \vcoffin_gset:Nnn { c } 33149
33150 \cs_new_protected:Npn \__coffin_set_vertical:NnnNNN #1#2#3#4#5#6 33150
33151 { 33151
33152   \__coffin_if_exist:NT #1 33152
33153   { 33153
33154     #4 #1 33154
33155     { 33155
33156       \dim_set:Nn \tex_hsize:D {#2} 33156
33157       \__coffin_set_vertical_aux: 33157
33158       #3 33158
33159     } 33159
33160     #5 #1 33160
33161     \vbox_set_top:Nn \l__coffin_internal_box { \vbox_unpack:N #1 } 33161
33162     #6 #1 { T } 33162
33163     { 33163
33164       { Opt } 33164
33165       { 33165
33166         \dim_eval:n 33166
33167         { \box_ht:N #1 - \box_ht:N \l__coffin_internal_box } 33167
33168       } 33168
33169       { 1000pt } 33169
33170       { Opt } 33170
33171     } 33171
33172     \box_clear:N \l__coffin_internal_box 33172
33173   } 33173
33174 } 33174
33175 \cs_new_protected:Npe \__coffin_set_vertical_aux: 33175
33176 { 33176
33177   \bool_lazy_and:nnT 33177
33178   { \cs_if_exist_p:N \fmtname } 33178
33179   { \str_if_eq_p:Vn \fmtname { LaTeX2e } } 33179
33180   { 33180
33181     \dim_set_eq:NN \exp_not:N \linewidth \tex_hsize:D 33181
33182     \dim_set_eq:NN \exp_not:N \columnwidth \tex_hsize:D 33182
33183   } 33183
33184 } 33184
33185 \cs_new_protected:Npn \hcoffin_set:Nw #1 33185
33186 { 33186
33187   \__coffin_if_exist:NT #1 33187
33188   { 33188

```

33189	\hbox_set:Nw #1 \color_ensure_current:	33189
33190	\cs_set_protected:Npn \hcoffin_set_end:	33190
33191	{	33191
33192	\hbox_set_end:	33192
33193	\coffin_reset_poles:N #1	33193
33194	}	33194
33195	}	33195
33196	}	33196
33197	\cs_generate_variant:Nn \hcoffin_set:Nw { c }	33197
33198	\cs_new_protected:Npn \hcoffin_gset:Nw #1	33198
33199	{	33199
33200	__coffin_if_exist:NT #1	33200
33201	{	33201
33202	\hbox_gset:Nw #1 \color_ensure_current:	33202
33203	\cs_set_protected:Npn \hcoffin_gset_end:	33203
33204	{	33204
33205	\hbox_gset_end:	33205
33206	\coffin_greset_poles:N #1	33206
33207	}	33207
33208	}	33208
33209	}	33209
33210	\cs_generate_variant:Nn \hcoffin_gset:Nw { c }	33210
33211	\cs_new_protected:Npn \hcoffin_set_end: { }	33211
33212	\cs_new_protected:Npn \hcoffin_gset_end: { }	33212
33213	\cs_new_protected:Npn \vcoffin_set:Nnw #1#2	33213
33214	{	33214
33215	__coffin_set_vertical:NnNNNNNw #1 {#2} \vbox_set:Nw	33215
33216	\vcoffin_set_end:	33216
33217	\vbox_set_end: \coffin_reset_poles:N __coffin_set_pole:Nnn	33217
33218	}	33218
33219	\cs_generate_variant:Nn \vcoffin_set:Nnw { c }	33219
33220	\cs_new_protected:Npn \vcoffin_gset:Nnw #1#2	33220
33221	{	33221
33222	__coffin_set_vertical:NnNNNNNw #1 {#2} \vbox_gset:Nw	33222
33223	\vcoffin_gset_end:	33223
33224	\vbox_gset_end: \coffin_greset_poles:N __coffin_gset_pole:Nnn	33224
33225	}	33225
33226	\cs_generate_variant:Nn \vcoffin_gset:Nnw { c }	33226
33227	\cs_new_protected:Npn __coffin_set_vertical:NnNNNNNw #1#2#3#4#5#6#7	33227
33228	{	33228
33229	__coffin_if_exist:NT #1	33229
33230	{	33230
33231	#3 #1	33231
33232	\dim_set:Nn \tex_hsize:D {#2}	33232
33233	__coffin_set_vertical_aux:	33233
33234	\cs_set_protected:Npn #4	33234

```

33235 {
33236     #5
33237     #6 #1
33238     \vbox_set_top:Nn \l__coffin_internal_box { \vbox_unpack:N #1 }
33239     #7 #1 { T }
33240     {
33241         { Opt }
33242         {
33243             \dim_eval:n
33244             { \box_ht:N #1 - \box_ht:N \l__coffin_internal_box }
33245         }
33246         { 1000pt }
33247         { Opt }
33248     }
33249     \box_clear:N \l__coffin_internal_box
33250 }
33251 }
33252 }
33253 \cs_new_protected:Npn \vcoffin_set_end: { }
33254 \cs_new_protected:Npn \vcoffin_gset_end: { }
33255 \cs_new_protected:Npn \coffin_set_eq:NN #1#2
33256 {
33257     \__coffin_if_exist:NT #2
33258     {
33259         \box_set_eq:NN #1 #2
33260         \prop_set_eq:cc { coffin ~ \__coffin_to_value:N #1 ~ corners }
33261         { coffin ~ \__coffin_to_value:N #2 ~ corners }
33262         \prop_set_eq:cc { coffin ~ \__coffin_to_value:N #1 ~ poles }
33263         { coffin ~ \__coffin_to_value:N #2 ~ poles }
33264     }
33265 }
33266 \cs_generate_variant:Nn \coffin_set_eq:NN { c , Nc , cc }
33267 \cs_new_protected:Npn \coffin_gset_eq:NN #1#2
33268 {
33269     \__coffin_if_exist:NT #2
33270     {
33271         \box_gset_eq:NN #1 #2
33272         \prop_gset_eq:cc { coffin ~ \__coffin_to_value:N #1 ~ corners }
33273         { coffin ~ \__coffin_to_value:N #2 ~ corners }
33274         \prop_gset_eq:cc { coffin ~ \__coffin_to_value:N #1 ~ poles }
33275         { coffin ~ \__coffin_to_value:N #2 ~ poles }
33276     }
33277 }
33278 \cs_generate_variant:Nn \coffin_gset_eq:NN { c , Nc , cc }
33279 \coffin_new:N \c_empty_coffin
33280 \coffin_new:N \l__coffin_aligned_coffin

```

```
33281 \coffin_new:N \l__coffin_aligned_internal_coffin 33281
33282 \coffin_new:N \l_tmpa_coffin 33282
33283 \coffin_new:N \l_tmpb_coffin 33283
33284 \coffin_new:N \g_tmpa_coffin 33284
33285 \coffin_new:N \g_tmpb_coffin 33285
33286 \cs_new_eq:NN \coffin_dp:N \box_dp:N 33286
33287 \cs_new_eq:NN \coffin_dp:c \box_dp:c 33287
33288 \cs_new_eq:NN \coffin_ht:N \box_ht:N 33288
33289 \cs_new_eq:NN \coffin_ht:c \box_ht:c 33289
33290 \cs_new_eq:NN \coffin_ht_plus_dp:N \box_ht_plus_dp:N 33290
33291 \cs_new_eq:NN \coffin_ht_plus_dp:c \box_ht_plus_dp:c 33291
33292 \cs_new_eq:NN \coffin_wd:N \box_wd:N 33292
33293 \cs_new_eq:NN \coffin_wd:c \box_wd:c 33293
33294 \cs_new_protected:Npn \__coffin_get_pole:NnN #1#2#3 33294
33295 { 33295
33296 \prop_get:cnNF 33296
33297 { coffin ~ \__coffin_to_value:N #1 ~ poles } {#2} #3 33297
33298 { 33298
33299 \msg_error:nnee { coffin } { unknown-pole } 33299
33300 { \exp_not:n {#2} } { \token_to_str:N #1 } 33300
33301 \tl_set:Nn #3 { { Opt } { Opt } { Opt } { Opt } } 33301
33302 } 33302
33303 } 33303
33304 \cs_new_protected:Npn \__coffin_reset_structure:N #1 33304
33305 { 33305
33306 \prop_set_eq:cN { coffin ~ \__coffin_to_value:N #1 ~ corners } 33306
33307 \c__coffin_corners_prop 33307
33308 \prop_set_eq:cN { coffin ~ \__coffin_to_value:N #1 ~ poles } 33308
33309 \c__coffin_poles_prop 33309
33310 } 33310
33311 \cs_new_protected:Npn \__coffin_greset_structure:N #1 33311
33312 { 33312
33313 \prop_gset_eq:cN { coffin ~ \__coffin_to_value:N #1 ~ corners } 33313
33314 \c__coffin_corners_prop 33314
33315 \prop_gset_eq:cN { coffin ~ \__coffin_to_value:N #1 ~ poles } 33315
33316 \c__coffin_poles_prop 33316
33317 } 33317
33318 \cs_new_protected:Npn \coffin_set_horizontal_pole:Nnn #1#2#3 33318
33319 { \__coffin_set_horizontal_pole:NnnN #1 {#2} {#3} \prop_put:cne } 33319
33320 \cs_generate_variant:Nn \coffin_set_horizontal_pole:Nnn { c } 33320
33321 \cs_new_protected:Npn \coffin_gset_horizontal_pole:Nnn #1#2#3 33321
33322 { \__coffin_set_horizontal_pole:NnnN #1 {#2} {#3} \prop_gput:cne } 33322
33323 \cs_generate_variant:Nn \coffin_gset_horizontal_pole:Nnn { c } 33323
33324 \cs_new_protected:Npn \__coffin_set_horizontal_pole:NnnN #1#2#3#4 33324
33325 { 33325
33326 \__coffin_if_exist:NT #1 33326
```

```

33327 {
33328     #4 { coffin ~ \__coffin_to_value:N #1 ~ poles }
33329     {#2}
33330     {
33331         { Opt } { \dim_eval:n {#3} }
33332         { 1000pt } { Opt }
33333     }
33334 }
33335 }
33336 \cs_new_protected:Npn \coffin_set_vertical_pole:Nnn #1#2#3
33337 { \__coffin_set_vertical_pole:NnnN #1 {#2} {#3} \prop_put:cne }
33338 \cs_generate_variant:Nn \coffin_set_vertical_pole:Nnn { c }
33339 \cs_new_protected:Npn \coffin_gset_vertical_pole:Nnn #1#2#3
33340 { \__coffin_set_vertical_pole:NnnN #1 {#2} {#3} \prop_gput:cne }
33341 \cs_generate_variant:Nn \coffin_gset_vertical_pole:Nnn { c }
33342 \cs_new_protected:Npn \__coffin_set_vertical_pole:NnnN #1#2#3#4
33343 {
33344     \__coffin_if_exist:NT #1
33345     {
33346         #4 { coffin ~ \__coffin_to_value:N #1 ~ poles }
33347         {#2}
33348         {
33349             { \dim_eval:n {#3} } { Opt }
33350             { Opt } { 1000pt }
33351         }
33352     }
33353 }
33354 \cs_new_protected:Npn \__coffin_set_pole:Nnn #1#2#3
33355 {
33356     \prop_put:cne { coffin ~ \__coffin_to_value:N #1 ~ poles }
33357     {#2} {#3}
33358 }
33359 \cs_new_protected:Npn \__coffin_gset_pole:Nnn #1#2#3
33360 {
33361     \prop_gput:cne { coffin ~ \__coffin_to_value:N #1 ~ poles }
33362     {#2} {#3}
33363 }
33364 \cs_new_protected:Npn \coffin_reset_poles:N #1
33365 {
33366     \__coffin_reset_structure:N #1
33367     \__coffin_update_corners:N #1
33368     \__coffin_update_poles:N #1
33369 }
33370 \cs_new_protected:Npn \coffin_greset_poles:N #1
33371 {
33372     \__coffin_greset_structure:N #1

```

```

33373 \__coffin_gupdate_corners:N #1
33374 \__coffin_gupdate_poles:N #1
33375 }
33376 \cs_new_protected:Npn \__coffin_update_corners:N #1
33377 { \__coffin_update_corners:NN #1 \prop_put:Nne }
33378 \cs_new_protected:Npn \__coffin_gupdate_corners:N #1
33379 { \__coffin_update_corners:NN #1 \prop_gput:Nne }
33380 \cs_new_protected:Npn \__coffin_update_corners:NN #1#2
33381 {
33382 \exp_args:Nc \__coffin_update_corners:NNN
33383 { coffin ~ \__coffin_to_value:N #1 ~ corners }
33384 #1 #2
33385 }
33386 \cs_new_protected:Npn \__coffin_update_corners:NNN #1#2#3
33387 {
33388 #3 #1
33389 { tl }
33390 { { Opt } { \dim_eval:n { \box_ht:N #2 } } } }
33391 #3 #1
33392 { tr }
33393 {
33394 { \dim_eval:n { \box_wd:N #2 } }
33395 { \dim_eval:n { \box_ht:N #2 } }
33396 }
33397 #3 #1
33398 { bl }
33399 { { Opt } { \dim_eval:n { -\box_dp:N #2 } } } }
33400 #3 #1
33401 { br }
33402 {
33403 { \dim_eval:n { \box_wd:N #2 } }
33404 { \dim_eval:n { -\box_dp:N #2 } }
33405 }
33406 }
33407 \cs_new_protected:Npn \__coffin_update_poles:N #1
33408 { \__coffin_update_poles:NN #1 \prop_put:Nne }
33409 \cs_new_protected:Npn \__coffin_gupdate_poles:N #1
33410 { \__coffin_update_poles:NN #1 \prop_gput:Nne }
33411 \cs_new_protected:Npn \__coffin_update_poles:NN #1#2
33412 {
33413 \exp_args:Nc \__coffin_update_poles:NNN
33414 { coffin ~ \__coffin_to_value:N #1 ~ poles }
33415 #1 #2
33416 }
33417 \cs_new_protected:Npn \__coffin_update_poles:NNN #1#2#3
33418 {

```



```

33419 #3 #1 { hc }
33420 {
33421     { \dim_eval:n { 0.5 \box_wd:N #2 } }
33422     { Opt } { Opt } { 1000pt }
33423 }
33424 #3 #1 { r }
33425 {
33426     { \dim_eval:n { \box_wd:N #2 } }
33427     { Opt } { Opt } { 1000pt }
33428 }
33429 #3 #1 { vc }
33430 {
33431     { Opt }
33432     { \dim_eval:n { ( \box_ht:N #2 - \box_dp:N #2 ) / 2 } }
33433     { 1000pt }
33434     { Opt }
33435 }
33436 #3 #1 { t }
33437 {
33438     { Opt }
33439     { \dim_eval:n { \box_ht:N #2 } }
33440     { 1000pt }
33441     { Opt }
33442 }
33443 #3 #1 { b }
33444 {
33445     { Opt }
33446     { \dim_eval:n { -\box_dp:N #2 } }
33447     { 1000pt }
33448     { Opt }
33449 }
33450 }
33451 \cs_new_protected:Npn \__coffin_calculate_intersection:Nnn #1#2#3
33452 {
33453     \__coffin_get_pole:NnN #1 {#2} \l__coffin_pole_a_tl
33454     \__coffin_get_pole:NnN #1 {#3} \l__coffin_pole_b_tl
33455     \bool_set_false:N \l__coffin_error_bool
33456     \exp_last_two_unbraced:Noo
33457         \__coffin_calculate_intersection:nnnnnnnn
33458         \l__coffin_pole_a_tl \l__coffin_pole_b_tl
33459     \bool_if:NT \l__coffin_error_bool
33460     {
33461         \msg_error:nn { coffin } { no-pole-intersection }
33462         \dim_zero:N \l__coffin_x_dim
33463         \dim_zero:N \l__coffin_y_dim
33464     }

```

```
33465 } 33465
33466 \cs_new_protected:Npn \__coffin_calculate_intersection:nnnnnnnn 33466
33467 #1#2#3#4#5#6#7#8 33467
33468 { 33468
33469 \dim_compare:nNnTF {#3} = \c_zero_dim 33469
33470 { 33470
33471 \dim_set:Nn \l__coffin_x_dim {#1} 33471
33472 \dim_compare:nNnTF {#7} = \c_zero_dim 33472
33473 { \bool_set_true:N \l__coffin_error_bool } 33473
33474 { 33474
33475 \dim_set:Nn \l__coffin_y_dim 33475
33476 { 33476
33477 \dim_compare:nNnTF {#8} = \c_zero_dim 33477
33478 {#6} 33478
33479 { 33479
33480 \fp_to_dim:n 33480
33481 { 33481
33482 ( \dim_to_fp:n {#8} / \dim_to_fp:n {#7} ) 33482
33483 * ( \dim_to_fp:n {#1} - \dim_to_fp:n {#5} ) 33483
33484 + \dim_to_fp:n {#6} 33484
33485 } 33485
33486 } 33486
33487 } 33487
33488 } 33488
33489 } 33489
33490 { 33490
33491 \dim_compare:nNnTF {#4} = \c_zero_dim 33491
33492 { 33492
33493 \dim_set:Nn \l__coffin_y_dim {#2} 33493
33494 \dim_compare:nNnTF {#8} = { \c_zero_dim } 33494
33495 { \bool_set_true:N \l__coffin_error_bool } 33495
33496 { 33496
33497 \dim_set:Nn \l__coffin_x_dim 33497
33498 { 33498
33499 \dim_compare:nNnTF {#7} = \c_zero_dim 33499
33500 {#5} 33500
33501 { 33501
33502 \fp_to_dim:n 33502
33503 { 33503
33504 ( \dim_to_fp:n {#7} / \dim_to_fp:n {#8} ) 33504
33505 * ( \dim_to_fp:n {#4} - \dim_to_fp:n {#6} ) 33505
33506 + \dim_to_fp:n {#5} 33506
33507 } 33507
33508 } 33508
33509 } 33509
33510 } 33510
```

```

33511     }
33512     {
33513         \use:e
33514         {
33515             \__coffin_calculate_intersection:nnnnnn
33516             { \dim_to_fp:n {#4} / \dim_to_fp:n {#3} }
33517             { \dim_to_fp:n {#8} / \dim_to_fp:n {#7} }
33518         }
33519         {#1} {#2} {#5} {#6}
33520     }
33521 }
33522 }
33523 \cs_new_protected:Npn \__coffin_calculate_intersection:nnnnnn #1#2#3#4#5#6
33524 {
33525     \fp_compare:nNnTF {#1} = {#2}
33526     { \bool_set_true:N \l__coffin_error_bool }
33527     {
33528         \dim_set:Nn \l__coffin_x_dim
33529         {
33530             \fp_to_dim:n
33531             {
33532                 (
33533                     #1 * \dim_to_fp:n {#3}
33534                     - #2 * \dim_to_fp:n {#5}
33535                     - \dim_to_fp:n {#4}
33536                     + \dim_to_fp:n {#6}
33537                 )
33538                 /
33539                 ( #1 - #2 )
33540             }
33541         }
33542         \dim_set:Nn \l__coffin_y_dim
33543         {
33544             \fp_to_dim:n
33545             {
33546                 #1 * ( \l__coffin_x_dim - \dim_to_fp:n {#3} )
33547                 + \dim_to_fp:n {#4}
33548             }
33549         }
33550     }
33551 }
33552 \fp_new:N \l__coffin_sin_fp
33553 \fp_new:N \l__coffin_cos_fp
33554 \prop_new:N \l__coffin_bounding_prop
33555 \prop_new:N \l__coffin_corners_prop
33556 \prop_new:N \l__coffin_poles_prop

```

```
33557 \dim_new:N \l__coffin_bounding_shift_dim 33557
33558 \dim_new:N \l__coffin_left_corner_dim 33558
33559 \dim_new:N \l__coffin_right_corner_dim 33559
33560 \dim_new:N \l__coffin_bottom_corner_dim 33560
33561 \dim_new:N \l__coffin_top_corner_dim 33561
33562 \cs_new_protected:Npn \coffin_rotate:Nn #1#2 33562
33563 { \__coffin_rotate:Nnnnn #1 {#2} \box_rotate:Nn \prop_set_eq:cN \hbox_set:Nn } 33563
33564 \cs_generate_variant:Nn \coffin_rotate:Nn { c } 33564
33565 \cs_new_protected:Npn \coffin_grotate:Nn #1#2 33565
33566 { \__coffin_rotate:Nnnnn #1 {#2} \box_grotate:Nn \prop_gset_eq:cN \hbox_gset:Nn } 33566
33567 \cs_generate_variant:Nn \coffin_grotate:Nn { c } 33567
33568 \cs_new_protected:Npn \__coffin_rotate:Nnnnn #1#2#3#4#5 33568
33569 { 33569
33570 \fp_set:Nn \l__coffin_sin_fp { sind ( #2 ) } 33570
33571 \fp_set:Nn \l__coffin_cos_fp { cosd ( #2 ) } 33571
33572 \prop_set_eq:Nc \l__coffin_corners_prop 33572
33573 { coffin ~ \__coffin_to_value:N #1 ~ corners } 33573
33574 \prop_set_eq:Nc \l__coffin_poles_prop 33574
33575 { coffin ~ \__coffin_to_value:N #1 ~ poles } 33575
33576 \prop_map_inline:Nn \l__coffin_corners_prop 33576
33577 { \__coffin_rotate_corner:Nnnn #1 {##1} ##2 } 33577
33578 \prop_map_inline:Nn \l__coffin_poles_prop 33578
33579 { \__coffin_rotate_pole:Nnnnnn #1 {##1} ##2 } 33579
33580 \__coffin_set_bounding:N #1 33580
33581 \prop_map_inline:Nn \l__coffin_bounding_prop 33581
33582 { \__coffin_rotate_bounding:nnn {##1} ##2 } 33582
33583 \__coffin_find_corner_maxima:N #1 33583
33584 \__coffin_find_bounding_shift: 33584
33585 #3 #1 {#2} 33585
33586 \hbox_set:Nn \l__coffin_internal_box 33586
33587 { 33587
33588 \__kernel_kern:n 33588
33589 { \l__coffin_bounding_shift_dim - \l__coffin_left_corner_dim } 33589
33590 \box_move_down:nn { \l__coffin_bottom_corner_dim } 33590
33591 { \box_use:N #1 } 33591
33592 } 33592
33593 \box_set_ht:Nn \l__coffin_internal_box 33593
33594 { \l__coffin_top_corner_dim - \l__coffin_bottom_corner_dim } 33594
33595 \box_set_dp:Nn \l__coffin_internal_box { Opt } 33595
33596 \box_set_wd:Nn \l__coffin_internal_box 33596
33597 { \l__coffin_right_corner_dim - \l__coffin_left_corner_dim } 33597
33598 #5 #1 { \box_use_drop:N \l__coffin_internal_box } 33598
33599 \prop_map_inline:Nn \l__coffin_corners_prop 33599
33600 { \__coffin_shift_corner:Nnnn #1 {##1} ##2 } 33600
33601 \prop_map_inline:Nn \l__coffin_poles_prop 33601
33602 { \__coffin_shift_pole:Nnnnnn #1 {##1} ##2 } 33602
```

```
33603 #4 { coffin ~ \__coffin_to_value:N #1 ~ corners } 33603
33604 \l__coffin_corners_prop 33604
33605 #4 { coffin ~ \__coffin_to_value:N #1 ~ poles } 33605
33606 \l__coffin_poles_prop 33606
33607 } 33607
33608 \cs_new_protected:Npn \__coffin_set_bounding:N #1 33608
33609 { 33609
33610 \prop_put:Nne \l__coffin_bounding_prop { tl } 33610
33611 { { Opt } { \dim_eval:n { \box_ht:N #1 } } } 33611
33612 \prop_put:Nne \l__coffin_bounding_prop { tr } 33612
33613 { 33613
33614 { \dim_eval:n { \box_wd:N #1 } } 33614
33615 { \dim_eval:n { \box_ht:N #1 } } 33615
33616 } 33616
33617 \dim_set:Nn \l__coffin_internal_dim { -\box_dp:N #1 } 33617
33618 \prop_put:Nne \l__coffin_bounding_prop { bl } 33618
33619 { { Opt } { \dim_use:N \l__coffin_internal_dim } } 33619
33620 \prop_put:Nne \l__coffin_bounding_prop { br } 33620
33621 { 33621
33622 { \dim_eval:n { \box_wd:N #1 } } 33622
33623 { \dim_use:N \l__coffin_internal_dim } 33623
33624 } 33624
33625 } 33625
33626 \cs_new_protected:Npn \__coffin_rotate_bounding:nnn #1#2#3 33626
33627 { 33627
33628 \__coffin_rotate_vector:nnNN {#2} {#3} \l__coffin_x_dim \l__coffin_y_dim 33628
33629 \prop_put:Nne \l__coffin_bounding_prop {#1} 33629
33630 { { \dim_use:N \l__coffin_x_dim } { \dim_use:N \l__coffin_y_dim } } 33630
33631 } 33631
33632 \cs_new_protected:Npn \__coffin_rotate_corner:Nnnn #1#2#3#4 33632
33633 { 33633
33634 \__coffin_rotate_vector:nnNN {#3} {#4} \l__coffin_x_dim \l__coffin_y_dim 33634
33635 \prop_put:Nne \l__coffin_corners_prop {#2} 33635
33636 { { \dim_use:N \l__coffin_x_dim } { \dim_use:N \l__coffin_y_dim } } 33636
33637 } 33637
33638 \cs_new_protected:Npn \__coffin_rotate_pole:Nnnnnn #1#2#3#4#5#6 33638
33639 { 33639
33640 \__coffin_rotate_vector:nnNN {#3} {#4} \l__coffin_x_dim \l__coffin_y_dim 33640
33641 \__coffin_rotate_vector:nnNN {#5} {#6} 33641
33642 \l__coffin_x_prime_dim \l__coffin_y_prime_dim 33642
33643 \prop_put:Nne \l__coffin_poles_prop {#2} 33643
33644 { 33644
33645 { \dim_use:N \l__coffin_x_dim } { \dim_use:N \l__coffin_y_dim } 33645
33646 { \dim_use:N \l__coffin_x_prime_dim } 33646
33647 { \dim_use:N \l__coffin_y_prime_dim } 33647
33648 } 33648
```

```

33649 } 33649
33650 \cs_new_protected:Npn \__coffin_rotate_vector:nnNN #1#2#3#4 33650
33651 { 33651
33652 \dim_set:Nn #3 33652
33653 { 33653
33654 \fp_to_dim:n 33654
33655 { 33655
33656 \dim_to_fp:n {#1} * \l__coffin_cos_fp 33656
33657 - \dim_to_fp:n {#2} * \l__coffin_sin_fp 33657
33658 } 33658
33659 } 33659
33660 \dim_set:Nn #4 33660
33661 { 33661
33662 \fp_to_dim:n 33662
33663 { 33663
33664 \dim_to_fp:n {#1} * \l__coffin_sin_fp 33664
33665 + \dim_to_fp:n {#2} * \l__coffin_cos_fp 33665
33666 } 33666
33667 } 33667
33668 } 33668
33669 \cs_new_protected:Npn \__coffin_find_corner_maxima:N #1 33669
33670 { 33670
33671 \dim_set:Nn \l__coffin_top_corner_dim { -\c_max_dim } 33671
33672 \dim_set:Nn \l__coffin_right_corner_dim { -\c_max_dim } 33672
33673 \dim_set:Nn \l__coffin_bottom_corner_dim { \c_max_dim } 33673
33674 \dim_set:Nn \l__coffin_left_corner_dim { \c_max_dim } 33674
33675 \prop_map_inline:Nn \l__coffin_corners_prop 33675
33676 { \__coffin_find_corner_maxima_aux:nn ##2 } 33676
33677 } 33677
33678 \cs_new_protected:Npn \__coffin_find_corner_maxima_aux:nn #1#2 33678
33679 { 33679
33680 \dim_set:Nn \l__coffin_left_corner_dim 33680
33681 { \dim_min:nn { \l__coffin_left_corner_dim } {#1} } 33681
33682 \dim_set:Nn \l__coffin_right_corner_dim 33682
33683 { \dim_max:nn { \l__coffin_right_corner_dim } {#1} } 33683
33684 \dim_set:Nn \l__coffin_bottom_corner_dim 33684
33685 { \dim_min:nn { \l__coffin_bottom_corner_dim } {#2} } 33685
33686 \dim_set:Nn \l__coffin_top_corner_dim 33686
33687 { \dim_max:nn { \l__coffin_top_corner_dim } {#2} } 33687
33688 } 33688
33689 \cs_new_protected:Npn \__coffin_find_bounding_shift: 33689
33690 { 33690
33691 \dim_set:Nn \l__coffin_bounding_shift_dim { \c_max_dim } 33691
33692 \prop_map_inline:Nn \l__coffin_bounding_prop 33692
33693 { \__coffin_find_bounding_shift_aux:nn ##2 } 33693
33694 } 33694

```

```
33695 \cs_new_protected:Npn \__coffin_find_bounding_shift_aux:nn #1#2 33695
33696 { 33696
33697     \dim_set:Nn \l__coffin_bounding_shift_dim 33697
33698     { \dim_min:nn { \l__coffin_bounding_shift_dim } {#1} } 33698
33699 } 33699
33700 \cs_new_protected:Npn \__coffin_shift_corner:Nnnn #1#2#3#4 33700
33701 { 33701
33702     \prop_put:Nne \l__coffin_corners_prop {#2} 33702
33703     { 33703
33704         { \dim_eval:n { #3 - \l__coffin_left_corner_dim } } 33704
33705         { \dim_eval:n { #4 - \l__coffin_bottom_corner_dim } } 33705
33706     } 33706
33707 } 33707
33708 \cs_new_protected:Npn \__coffin_shift_pole:Nnnnnn #1#2#3#4#5#6 33708
33709 { 33709
33710     \prop_put:Nne \l__coffin_poles_prop {#2} 33710
33711     { 33711
33712         { \dim_eval:n { #3 - \l__coffin_left_corner_dim } } 33712
33713         { \dim_eval:n { #4 - \l__coffin_bottom_corner_dim } } 33713
33714         {#5} {#6} 33714
33715     } 33715
33716 } 33716
33717 \fp_new:N \l__coffin_scale_x_fp 33717
33718 \fp_new:N \l__coffin_scale_y_fp 33718
33719 \dim_new:N \l__coffin_scaled_total_height_dim 33719
33720 \dim_new:N \l__coffin_scaled_width_dim 33720
33721 \cs_new_protected:Npn \coffin_resize:Nnn #1#2#3 33721
33722 { 33722
33723     \__coffin_resize:NnnNN #1 {#2} {#3} 33723
33724     \box_resize_to_wd_and_ht_plus_dp:Nnn 33724
33725     \prop_set_eq:cN 33725
33726 } 33726
33727 \cs_generate_variant:Nn \coffin_resize:Nnn { c } 33727
33728 \cs_new_protected:Npn \coffin_gresize:Nnn #1#2#3 33728
33729 { 33729
33730     \__coffin_resize:NnnNN #1 {#2} {#3} 33730
33731     \box_gresize_to_wd_and_ht_plus_dp:Nnn 33731
33732     \prop_gset_eq:cN 33732
33733 } 33733
33734 \cs_generate_variant:Nn \coffin_gresize:Nnn { c } 33734
33735 \cs_new_protected:Npn \__coffin_resize:NnnNN #1#2#3#4#5 33735
33736 { 33736
33737     \fp_set:Nn \l__coffin_scale_x_fp 33737
33738     { \dim_to_fp:n {#2} / \dim_to_fp:n { \coffin_wd:N #1 } } 33738
33739     \fp_set:Nn \l__coffin_scale_y_fp 33739
33740     { 33740
```



```

33741         \dim_to_fp:n {#3}
33742         / \dim_to_fp:n { \coffin_ht:N #1 + \coffin_dp:N #1 }
33743     }
33744     #4 #1 {#2} {#3}
33745     \__coffin_resize_common:NnnN #1 {#2} {#3} #5
33746 }
33747 \cs_new_protected:Npn \__coffin_resize_common:NnnN #1#2#3#4
33748 {
33749     \prop_set_eq:Nc \l__coffin_corners_prop
33750     { coffin ~ \__coffin_to_value:N #1 ~ corners }
33751     \prop_set_eq:Nc \l__coffin_poles_prop
33752     { coffin ~ \__coffin_to_value:N #1 ~ poles }
33753     \prop_map_inline:Nn \l__coffin_corners_prop
33754     { \__coffin_scale_corner:Nnnn #1 {##1} ##2 }
33755     \prop_map_inline:Nn \l__coffin_poles_prop
33756     { \__coffin_scale_pole:Nnnnnn #1 {##1} ##2 }
33757     \fp_compare:nNnT \l__coffin_scale_x_fp < \c_zero_fp
33758     {
33759         \prop_map_inline:Nn \l__coffin_corners_prop
33760         { \__coffin_x_shift_corner:Nnnn #1 {##1} ##2 }
33761         \prop_map_inline:Nn \l__coffin_poles_prop
33762         { \__coffin_x_shift_pole:Nnnnnn #1 {##1} ##2 }
33763     }
33764     #4 { coffin ~ \__coffin_to_value:N #1 ~ corners }
33765     \l__coffin_corners_prop
33766     #4 { coffin ~ \__coffin_to_value:N #1 ~ poles }
33767     \l__coffin_poles_prop
33768 }
33769 \cs_new_protected:Npn \coffin_scale:Nnn #1#2#3
33770 { \__coffin_scale:NnnNN #1 {#2} {#3} \box_scale:Nnn \prop_set_eq:cN }
33771 \cs_generate_variant:Nn \coffin_scale:Nnn { c }
33772 \cs_new_protected:Npn \coffin_gscale:Nnn #1#2#3
33773 { \__coffin_scale:NnnNN #1 {#2} {#3} \box_gscale:Nnn \prop_gset_eq:cN }
33774 \cs_generate_variant:Nn \coffin_gscale:Nnn { c }
33775 \cs_new_protected:Npn \__coffin_scale:NnnNN #1#2#3#4#5
33776 {
33777     \fp_set:Nn \l__coffin_scale_x_fp {#2}
33778     \fp_set:Nn \l__coffin_scale_y_fp {#3}
33779     #4 #1 { \l__coffin_scale_x_fp } { \l__coffin_scale_y_fp }
33780     \dim_set:Nn \l__coffin_internal_dim
33781     { \coffin_ht:N #1 + \coffin_dp:N #1 }
33782     \dim_set:Nn \l__coffin_scaled_total_height_dim
33783     { \fp_abs:n { \l__coffin_scale_y_fp } \l__coffin_internal_dim }
33784     \dim_set:Nn \l__coffin_scaled_width_dim
33785     { -\fp_abs:n { \l__coffin_scale_x_fp } \coffin_wd:N #1 }
33786     \__coffin_resize_common:NnnN #1

```

```
33787 { \l__coffin_scaled_width_dim } { \l__coffin_scaled_total_height_dim } 33787
33788 #5 33788
33789 } 33789
33790 \cs_new_protected:Npn \__coffin_scale_vector:nnNN #1#2#3#4 33790
33791 { 33791
33792 \dim_set:Nn #3 33792
33793 { \fp_to_dim:n { \dim_to_fp:n {#1} * \l__coffin_scale_x_fp } } 33793
33794 \dim_set:Nn #4 33794
33795 { \fp_to_dim:n { \dim_to_fp:n {#2} * \l__coffin_scale_y_fp } } 33795
33796 } 33796
33797 \cs_new_protected:Npn \__coffin_scale_corner:Nnnn #1#2#3#4 33797
33798 { 33798
33799 \__coffin_scale_vector:nnNN {#3} {#4} \l__coffin_x_dim \l__coffin_y_dim 33799
33800 \prop_put:Nne \l__coffin_corners_prop {#2} 33800
33801 { { \dim_use:N \l__coffin_x_dim } { \dim_use:N \l__coffin_y_dim } } 33801
33802 } 33802
33803 \cs_new_protected:Npn \__coffin_scale_pole:Nnnnnn #1#2#3#4#5#6 33803
33804 { 33804
33805 \__coffin_scale_vector:nnNN {#3} {#4} \l__coffin_x_dim \l__coffin_y_dim 33805
33806 \prop_put:Nne \l__coffin_poles_prop {#2} 33806
33807 { 33807
33808 { \dim_use:N \l__coffin_x_dim } { \dim_use:N \l__coffin_y_dim } 33808
33809 {#5} {#6} 33809
33810 } 33810
33811 } 33811
33812 \cs_new_protected:Npn \__coffin_x_shift_corner:Nnnn #1#2#3#4 33812
33813 { 33813
33814 \prop_put:Nne \l__coffin_corners_prop {#2} 33814
33815 { 33815
33816 { \dim_eval:n { #3 + \box_wd:N #1 } } {#4} 33816
33817 } 33817
33818 } 33818
33819 \cs_new_protected:Npn \__coffin_x_shift_pole:Nnnnnn #1#2#3#4#5#6 33819
33820 { 33820
33821 \prop_put:Nne \l__coffin_poles_prop {#2} 33821
33822 { 33822
33823 { \dim_eval:n { #3 + \box_wd:N #1 } } {#4} 33823
33824 {#5} {#6} 33824
33825 } 33825
33826 } 33826
33827 \cs_new_protected:Npn \coffin_join:NnnNnnnn #1#2#3#4#5#6#7#8 33827
33828 { 33828
33829 \__coffin_join:NnnNnnnnN #1 {#2} {#3} #4 {#5} {#6} {#7} {#8} 33829
33830 \coffin_set_eq:NN 33830
33831 } 33831
33832 \cs_generate_variant:Nn \coffin_join:NnnNnnnn { c , Nnnc , cnnc } 33832
```

```
33833 \cs_new_protected:Npn \coffin_gjoin:NnnNnnnn #1#2#3#4#5#6#7#8 33833
33834 { 33834
33835 \__coffin_join:NnnNnnnnN #1 {#2} {#3} #4 {#5} {#6} {#7} {#8} 33835
33836 \coffin_gset_eq:NN 33836
33837 } 33837
33838 \cs_generate_variant:Nn \coffin_gjoin:NnnNnnnn { c , Nnnc , cnnc } 33838
33839 \cs_new_protected:Npn \__coffin_join:NnnNnnnnN #1#2#3#4#5#6#7#8#9 33839
33840 { 33840
33841 \__coffin_align:NnnNnnnnN 33841
33842 #1 {#2} {#3} #4 {#5} {#6} {#7} {#8} \l__coffin_aligned_coffin 33842
33843 \hbox_set:Nn \l__coffin_aligned_coffin 33843
33844 { 33844
33845 \dim_compare:nNnT { \l__coffin_offset_x_dim } < \c_zero_dim 33845
33846 { \__kernel_kern:n { -\l__coffin_offset_x_dim } } 33846
33847 \hbox_unpack:N \l__coffin_aligned_coffin 33847
33848 \dim_set:Nn \l__coffin_internal_dim 33848
33849 { \l__coffin_offset_x_dim - \box_wd:N #1 + \box_wd:N #4 } 33849
33850 \dim_compare:nNnT \l__coffin_internal_dim < \c_zero_dim 33850
33851 { \__kernel_kern:n { -\l__coffin_internal_dim } } 33851
33852 } 33852
33853 \__coffin_reset_structure:N \l__coffin_aligned_coffin 33853
33854 \prop_clear:c 33854
33855 { 33855
33856 coffin ~ \__coffin_to_value:N \l__coffin_aligned_coffin 33856
33857 \c_space_tl corners 33857
33858 } 33858
33859 \__coffin_update_poles:N \l__coffin_aligned_coffin 33859
33860 \dim_compare:nNnTF \l__coffin_offset_x_dim < \c_zero_dim 33860
33861 { 33861
33862 \__coffin_offset_poles:Nnn #1 { -\l__coffin_offset_x_dim } { Opt } 33862
33863 \__coffin_offset_poles:Nnn #4 { Opt } { \l__coffin_offset_y_dim } 33863
33864 \__coffin_offset_corners:Nnn #1 { -\l__coffin_offset_x_dim } { Opt } 33864
33865 \__coffin_offset_corners:Nnn #4 { Opt } { \l__coffin_offset_y_dim } 33865
33866 } 33866
33867 { 33867
33868 \__coffin_offset_poles:Nnn #1 { Opt } { Opt } 33868
33869 \__coffin_offset_poles:Nnn #4 33869
33870 { \l__coffin_offset_x_dim } { \l__coffin_offset_y_dim } 33870
33871 \__coffin_offset_corners:Nnn #1 { Opt } { Opt } 33871
33872 \__coffin_offset_corners:Nnn #4 33872
33873 { \l__coffin_offset_x_dim } { \l__coffin_offset_y_dim } 33873
33874 } 33874
33875 \__coffin_update_vertical_poles:NNN #1 #4 \l__coffin_aligned_coffin 33875
33876 #9 #1 \l__coffin_aligned_coffin 33876
33877 } 33877
33878 \cs_new_protected:Npn \coffin_attach:NnnNnnnn #1#2#3#4#5#6#7#8 33878
```

```
33879 { 33879
33880 \__coffin_attach:NnnNnnnnN #1 {#2} {#3} #4 {#5} {#6} {#7} {#8} 33880
33881 \coffin_set_eq:NN 33881
33882 } 33882
33883 \cs_generate_variant:Nn \coffin_attach:NnnNnnnn { c , Nnnc , cnnc } 33883
33884 \cs_new_protected:Npn \coffin_gattach:NnnNnnnn #1#2#3#4#5#6#7#8 33884
33885 { 33885
33886 \__coffin_attach:NnnNnnnnN #1 {#2} {#3} #4 {#5} {#6} {#7} {#8} 33886
33887 \coffin_gset_eq:NN 33887
33888 } 33888
33889 \cs_generate_variant:Nn \coffin_gattach:NnnNnnnn { c , Nnnc , cnnc } 33889
33890 \cs_new_protected:Npn \__coffin_attach:NnnNnnnnN #1#2#3#4#5#6#7#8#9 33890
33891 { 33891
33892 \__coffin_align:NnnNnnnnN 33892
33893 #1 {#2} {#3} #4 {#5} {#6} {#7} {#8} \l__coffin_aligned_coffin 33893
33894 \box_set_ht:Nn \l__coffin_aligned_coffin { \box_ht:N #1 } 33894
33895 \box_set_dp:Nn \l__coffin_aligned_coffin { \box_dp:N #1 } 33895
33896 \box_set_wd:Nn \l__coffin_aligned_coffin { \box_wd:N #1 } 33896
33897 \__coffin_reset_structure:N \l__coffin_aligned_coffin 33897
33898 \prop_set_eq:cc 33898
33899 { 33899
33900 coffin ~ \__coffin_to_value:N \l__coffin_aligned_coffin 33900
33901 \c_space_tl corners 33901
33902 } 33902
33903 { coffin ~ \__coffin_to_value:N #1 ~ corners } 33903
33904 \__coffin_update_poles:N \l__coffin_aligned_coffin 33904
33905 \__coffin_offset_poles:Nnn #1 { Opt } { Opt } 33905
33906 \__coffin_offset_poles:Nnn #4 33906
33907 { \l__coffin_offset_x_dim } { \l__coffin_offset_y_dim } 33907
33908 \__coffin_update_vertical_poles:NNN #1 #4 \l__coffin_aligned_coffin 33908
33909 #9 #1 \l__coffin_aligned_coffin 33909
33910 } 33910
33911 \cs_new_protected:Npn \__coffin_attach_mark:NnnNnnnn #1#2#3#4#5#6#7#8 33911
33912 { 33912
33913 \__coffin_align:NnnNnnnnN 33913
33914 #1 {#2} {#3} #4 {#5} {#6} {#7} {#8} \l__coffin_aligned_coffin 33914
33915 \box_set_ht:Nn \l__coffin_aligned_coffin { \box_ht:N #1 } 33915
33916 \box_set_dp:Nn \l__coffin_aligned_coffin { \box_dp:N #1 } 33916
33917 \box_set_wd:Nn \l__coffin_aligned_coffin { \box_wd:N #1 } 33917
33918 \box_set_eq:NN #1 \l__coffin_aligned_coffin 33918
33919 } 33919
33920 \cs_new_protected:Npn \__coffin_align:NnnNnnnnN #1#2#3#4#5#6#7#8#9 33920
33921 { 33921
33922 \__coffin_calculate_intersection:Nnn #4 {#5} {#6} 33922
33923 \dim_set:Nn \l__coffin_x_prime_dim { \l__coffin_x_dim } 33923
33924 \dim_set:Nn \l__coffin_y_prime_dim { \l__coffin_y_dim } 33924
```

```

33925 \__coffin_calculate_intersection:Nnn #1 {#2} {#3} 33925
33926 \dim_set:Nn \l__coffin_offset_x_dim 33926
33927 { \l__coffin_x_dim - \l__coffin_x_prime_dim + #7 } 33927
33928 \dim_set:Nn \l__coffin_offset_y_dim 33928
33929 { \l__coffin_y_dim - \l__coffin_y_prime_dim + #8 } 33929
33930 \hbox_set:Nn \l__coffin_aligned_internal_coffin 33930
33931 { 33931
33932 \box_use:N #1 33932
33933 \__kernel_kern:n { -\box_wd:N #1 } 33933
33934 \__kernel_kern:n { \l__coffin_offset_x_dim } 33934
33935 \box_move_up:nn { \l__coffin_offset_y_dim } { \box_use:N #4 } 33935
33936 } 33936
33937 \coffin_set_eq:NN #9 \l__coffin_aligned_internal_coffin 33937
33938 } 33938
33939 \cs_new_protected:Npn \__coffin_offset_poles:Nnn #1#2#3 33939
33940 { 33940
33941 \prop_map_inline:cn { coffin ~ \__coffin_to_value:N #1 ~ poles } 33941
33942 { \__coffin_offset_pole:Nnnnnnn #1 {##1} ##2 {#2} {#3} } 33942
33943 } 33943
33944 \cs_new_protected:Npn \__coffin_offset_pole:Nnnnnnn #1#2#3#4#5#6#7#8 33944
33945 { 33945
33946 \dim_set:Nn \l__coffin_x_dim { #3 + #7 } 33946
33947 \dim_set:Nn \l__coffin_y_dim { #4 + #8 } 33947
33948 \tl_if_in:nnTF {#2} { - } 33948
33949 { \tl_set:Nn \l__coffin_internal_tl { {#2} } } 33949
33950 { \tl_set:Nn \l__coffin_internal_tl { { #1 - #2 } } } 33950
33951 \exp_last_unbraced:NNo \__coffin_set_pole:Nnn \l__coffin_aligned_coffin 33951
33952 { \l__coffin_internal_tl } 33952
33953 { 33953
33954 { \dim_use:N \l__coffin_x_dim } { \dim_use:N \l__coffin_y_dim } 33954
33955 {#5} {#6} 33955
33956 } 33956
33957 } 33957
33958 \cs_new_protected:Npn \__coffin_offset_corners:Nnn #1#2#3 33958
33959 { 33959
33960 \prop_map_inline:cn { coffin ~ \__coffin_to_value:N #1 ~ corners } 33960
33961 { \__coffin_offset_corner:Nnnnn #1 {##1} ##2 {#2} {#3} } 33961
33962 } 33962
33963 \cs_new_protected:Npn \__coffin_offset_corner:Nnnnn #1#2#3#4#5#6 33963
33964 { 33964
33965 \prop_put:cne 33965
33966 { 33966
33967 coffin ~ \__coffin_to_value:N \l__coffin_aligned_coffin 33967
33968 \c_space_tl corners 33968
33969 } 33969
33970 { #1 - #2 } 33970

```

```

33971 {
33972     { \dim_eval:n { #3 + #5 } }
33973     { \dim_eval:n { #4 + #6 } }
33974 }
33975 }
33976 \cs_new_protected:Npn \__coffin_update_vertical_poles:NNN #1#2#3
33977 {
33978     \__coffin_get_pole:NnN #3 { #1 -T } \l__coffin_pole_a_tl
33979     \__coffin_get_pole:NnN #3 { #2 -T } \l__coffin_pole_b_tl
33980     \exp_last_two_unbraced:Noo \__coffin_update_T:nnnnnnnnN
33981         \l__coffin_pole_a_tl \l__coffin_pole_b_tl #3
33982     \__coffin_get_pole:NnN #3 { #1 -B } \l__coffin_pole_a_tl
33983     \__coffin_get_pole:NnN #3 { #2 -B } \l__coffin_pole_b_tl
33984     \exp_last_two_unbraced:Noo \__coffin_update_B:nnnnnnnnN
33985         \l__coffin_pole_a_tl \l__coffin_pole_b_tl #3
33986 }
33987 \cs_new_protected:Npn \__coffin_update_T:nnnnnnnnN #1#2#3#4#5#6#7#8#9
33988 {
33989     \dim_compare:nNnTF {#2} < {#6}
33990     {
33991         \__coffin_set_pole:Nnn #9 { T }
33992         { { Opt } {#6} { 1000pt } { Opt } }
33993     }
33994     {
33995         \__coffin_set_pole:Nnn #9 { T }
33996         { { Opt } {#2} { 1000pt } { Opt } }
33997     }
33998 }
33999 \cs_new_protected:Npn \__coffin_update_B:nnnnnnnnN #1#2#3#4#5#6#7#8#9
34000 {
34001     \dim_compare:nNnTF {#2} < {#6}
34002     {
34003         \__coffin_set_pole:Nnn #9 { B }
34004         { { Opt } {#2} { 1000pt } { Opt } }
34005     }
34006     {
34007         \__coffin_set_pole:Nnn #9 { B }
34008         { { Opt } {#6} { 1000pt } { Opt } }
34009     }
34010 }
34011 \coffin_new:N \c__coffin_empty_coffin
34012 \tex_setbox:D \c__coffin_empty_coffin = \tex_hbox:D { }
34013 \cs_new_protected:Npn \coffin_typeset:Nnnnn #1#2#3#4#5
34014 {
34015     \mode_leave_vertical:
34016     \__coffin_align:NnnNnnnnN \c__coffin_empty_coffin { H } { l }

```


34017	#1 {#2} {#3} {#4} {#5} \l__coffin_aligned_coffin	34017
34018	\box_use_drop:N \l__coffin_aligned_coffin	34018
34019	}	34019
34020	\cs_generate_variant:Nn \coffin_typeset:Nnnnn { c }	34020
34021	\coffin_new:N \l__coffin_display_coffin	34021
34022	\coffin_new:N \l__coffin_display_coord_coffin	34022
34023	\coffin_new:N \l__coffin_display_pole_coffin	34023
34024	\prop_new:N \l__coffin_display_handles_prop	34024
34025	\prop_put:Nnn \l__coffin_display_handles_prop { tl }	34025
34026	{ { b } { r } { -1 } { 1 } }	34026
34027	\prop_put:Nnn \l__coffin_display_handles_prop { thc }	34027
34028	{ { b } { hc } { 0 } { 1 } }	34028
34029	\prop_put:Nnn \l__coffin_display_handles_prop { tr }	34029
34030	{ { b } { l } { 1 } { 1 } }	34030
34031	\prop_put:Nnn \l__coffin_display_handles_prop { vcl }	34031
34032	{ { vc } { r } { -1 } { 0 } }	34032
34033	\prop_put:Nnn \l__coffin_display_handles_prop { vhc }	34033
34034	{ { vc } { hc } { 0 } { 0 } }	34034
34035	\prop_put:Nnn \l__coffin_display_handles_prop { vcr }	34035
34036	{ { vc } { l } { 1 } { 0 } }	34036
34037	\prop_put:Nnn \l__coffin_display_handles_prop { bl }	34037
34038	{ { t } { r } { -1 } { -1 } }	34038
34039	\prop_put:Nnn \l__coffin_display_handles_prop { bhc }	34039
34040	{ { t } { hc } { 0 } { -1 } }	34040
34041	\prop_put:Nnn \l__coffin_display_handles_prop { br }	34041
34042	{ { t } { l } { 1 } { -1 } }	34042
34043	\prop_put:Nnn \l__coffin_display_handles_prop { Tl }	34043
34044	{ { t } { r } { -1 } { -1 } }	34044
34045	\prop_put:Nnn \l__coffin_display_handles_prop { Thc }	34045
34046	{ { t } { hc } { 0 } { -1 } }	34046
34047	\prop_put:Nnn \l__coffin_display_handles_prop { Tr }	34047
34048	{ { t } { l } { 1 } { -1 } }	34048
34049	\prop_put:Nnn \l__coffin_display_handles_prop { Hl }	34049
34050	{ { vc } { r } { -1 } { 1 } }	34050
34051	\prop_put:Nnn \l__coffin_display_handles_prop { Hhc }	34051
34052	{ { vc } { hc } { 0 } { 1 } }	34052
34053	\prop_put:Nnn \l__coffin_display_handles_prop { Hr }	34053
34054	{ { vc } { l } { 1 } { 1 } }	34054
34055	\prop_put:Nnn \l__coffin_display_handles_prop { Bl }	34055
34056	{ { b } { r } { -1 } { -1 } }	34056
34057	\prop_put:Nnn \l__coffin_display_handles_prop { Bhc }	34057
34058	{ { b } { hc } { 0 } { -1 } }	34058
34059	\prop_put:Nnn \l__coffin_display_handles_prop { Br }	34059
34060	{ { b } { l } { 1 } { -1 } }	34060
34061	\dim_new:N \l__coffin_display_offset_dim	34061
34062	\dim_set:Nn \l__coffin_display_offset_dim { 2pt }	34062


```

34063 \dim_new:N \l__coffin_display_x_dim 34063
34064 \dim_new:N \l__coffin_display_y_dim 34064
34065 \prop_new:N \l__coffin_display_poles_prop 34065
34066 \tl_new:N \l__coffin_display_font_tl 34066
34067 \bool_lazy_and:nnT 34067
34068 { \cs_if_exist_p:N \fmtname } 34068
34069 { \str_if_eq_p:Vn \fmtname { LaTeX2e } } 34069
34070 { 34070
34071 \tl_set:Nn \l__coffin_display_font_tl 34071
34072 { \sffamily \tiny } 34072
34073 } 34073
34074 \cs_new_protected:Npn \__coffin_rule:nn #1#2 34074
34075 { 34075
34076 \mode_leave_vertical: 34076
34077 \hbox:n { \tex_vrule:D width #1 height #2 \scan_stop: } 34077
34078 } 34078
34079 \cs_new_protected:Npn \coffin_mark_handle:Nnnn #1#2#3#4 34079
34080 { 34080
34081 \hcoffin_set:Nn \l__coffin_display_pole_coffin 34081
34082 { 34082
34083 \color_select:n {#4} 34083
34084 \__coffin_rule:nn { 1pt } { 1pt } 34084
34085 } 34085
34086 \__coffin_attach_mark:NnnNnnnn #1 {#2} {#3} 34086
34087 \l__coffin_display_pole_coffin { hc } { vc } { Opt } { Opt } 34087
34088 \hcoffin_set:Nn \l__coffin_display_coord_coffin 34088
34089 { 34089
34090 \color_select:n {#4} 34090
34091 \l__coffin_display_font_tl 34091
34092 ( \tl_to_str:n { #2 , #3 } ) 34092
34093 } 34093
34094 \prop_get:NnN \l__coffin_display_handles_prop 34094
34095 { #2 #3 } \l__coffin_internal_tl 34095
34096 \quark_if_no_value:NTF \l__coffin_internal_tl 34096
34097 { 34097
34098 \prop_get:NnN \l__coffin_display_handles_prop 34098
34099 { #3 #2 } \l__coffin_internal_tl 34099
34100 \quark_if_no_value:NTF \l__coffin_internal_tl 34100
34101 { 34101
34102 \__coffin_attach_mark:NnnNnnnn #1 {#2} {#3} 34102
34103 \l__coffin_display_coord_coffin { l } { vc } 34103
34104 { 1pt } { Opt } 34104
34105 } 34105
34106 { 34106
34107 \exp_last_unbraced:No \__coffin_mark_handle_aux:nnnnNnn 34107
34108 \l__coffin_internal_tl #1 {#2} {#3} 34108

```

```
34109         }
34110     }
34111     {
34112         \exp_last_unbraced:No \__coffin_mark_handle_aux:nnnnNnn
34113         \l__coffin_internal_tl #1 {#2} {#3}
34114     }
34115 }
34116 \cs_new_protected:Npn \__coffin_mark_handle_aux:nnnnNnn #1#2#3#4#5#6#7
34117 {
34118     \__coffin_attach_mark:NnnNnnnn #5 {#6} {#7}
34119     \l__coffin_display_coord_coffin {#1} {#2}
34120     { #3 \l__coffin_display_offset_dim }
34121     { #4 \l__coffin_display_offset_dim }
34122 }
34123 \cs_generate_variant:Nn \coffin_mark_handle:Nnnn { c }
34124 \cs_new_protected:Npn \coffin_display_handles:Nn #1#2
34125 {
34126     \hcoffin_set:Nn \l__coffin_display_pole_coffin
34127     {
34128         \color_select:n {#2}
34129         \__coffin_rule:nn { 1pt } { 1pt }
34130     }
34131     \prop_set_eq:Nc \l__coffin_display_poles_prop
34132     { coffin ~ \__coffin_to_value:N #1 ~ poles }
34133     \__coffin_get_pole:NnN #1 { H } \l__coffin_pole_a_tl
34134     \__coffin_get_pole:NnN #1 { T } \l__coffin_pole_b_tl
34135     \tl_if_eq:NNT \l__coffin_pole_a_tl \l__coffin_pole_b_tl
34136     { \prop_remove:Nn \l__coffin_display_poles_prop { T } }
34137     \__coffin_get_pole:NnN #1 { B } \l__coffin_pole_b_tl
34138     \tl_if_eq:NNT \l__coffin_pole_a_tl \l__coffin_pole_b_tl
34139     { \prop_remove:Nn \l__coffin_display_poles_prop { B } }
34140     \coffin_set_eq:NN \l__coffin_display_coffin #1
34141     \prop_map_inline:Nn \l__coffin_display_poles_prop
34142     {
34143         \prop_remove:Nn \l__coffin_display_poles_prop {##1}
34144         \__coffin_display_handles_aux:nnnnnn {##1} ##2 {#2}
34145     }
34146     \box_use_drop:N \l__coffin_display_coffin
34147 }
34148 \cs_new_protected:Npn \__coffin_display_handles_aux:nnnnnn #1#2#3#4#5#6
34149 {
34150     \prop_map_inline:Nn \l__coffin_display_poles_prop
34151     {
34152         \bool_set_false:N \l__coffin_error_bool
34153         \__coffin_calculate_intersection:nnnnnnnn {#2} {#3} {#4} {#5} ##2
34154         \bool_if:NF \l__coffin_error_bool
```

```

34155 {
34156     \dim_set:Nn \l__coffin_display_x_dim { \l__coffin_x_dim }
34157     \dim_set:Nn \l__coffin_display_y_dim { \l__coffin_y_dim }
34158     \__coffin_display_attach:Nnnnn
34159     \l__coffin_display_pole_coffin { hc } { vc }
34160     { Opt } { Opt }
34161     \hcoffin_set:Nn \l__coffin_display_coord_coffin
34162     {
34163         \color_select:n {#6}
34164         \l__coffin_display_font_tl
34165         ( \tl_to_str:n { #1 , ##1 } )
34166     }
34167     \prop_get:NnN \l__coffin_display_handles_prop
34168     { #1 ##1 } \l__coffin_internal_tl
34169     \quark_if_no_value:NTF \l__coffin_internal_tl
34170     {
34171         \prop_get:NnN \l__coffin_display_handles_prop
34172         { ##1 #1 } \l__coffin_internal_tl
34173         \quark_if_no_value:NTF \l__coffin_internal_tl
34174         {
34175             \__coffin_display_attach:Nnnnn
34176             \l__coffin_display_coord_coffin { l } { vc }
34177             { 1pt } { Opt }
34178         }
34179         {
34180             \exp_last_unbraced:No
34181             \__coffin_display_handles_aux:nnnn
34182             \l__coffin_internal_tl
34183         }
34184     }
34185     {
34186         \exp_last_unbraced:No \__coffin_display_handles_aux:nnnn
34187         \l__coffin_internal_tl
34188     }
34189 }
34190 }
34191 }
34192 \cs_new_protected:Npn \__coffin_display_handles_aux:nnnn #1#2#3#4
34193 {
34194     \__coffin_display_attach:Nnnnn
34195     \l__coffin_display_coord_coffin {#1} {#2}
34196     { #3 \l__coffin_display_offset_dim }
34197     { #4 \l__coffin_display_offset_dim }
34198 }
34199 \cs_generate_variant:Nn \coffin_display_handles:Nn { c }
34200 \cs_new_protected:Npn \__coffin_display_attach:Nnnnn #1#2#3#4#5

```

```

34201 {
34202 \__coffin_calculate_intersection:Nnn #1 {#2} {#3}
34203 \dim_set:Nn \l__coffin_x_prime_dim { \l__coffin_x_dim }
34204 \dim_set:Nn \l__coffin_y_prime_dim { \l__coffin_y_dim }
34205 \dim_set:Nn \l__coffin_offset_x_dim
34206 { \l__coffin_display_x_dim - \l__coffin_x_prime_dim + #4 }
34207 \dim_set:Nn \l__coffin_offset_y_dim
34208 { \l__coffin_display_y_dim - \l__coffin_y_prime_dim + #5 }
34209 \hbox_set:Nn \l__coffin_aligned_coffin
34210 {
34211 \box_use:N \l__coffin_display_coffin
34212 \__kernel_kern:n { -\box_wd:N \l__coffin_display_coffin }
34213 \__kernel_kern:n { \l__coffin_offset_x_dim }
34214 \box_move_up:nn { \l__coffin_offset_y_dim } { \box_use:N #1 }
34215 }
34216 \box_set_ht:Nn \l__coffin_aligned_coffin
34217 { \box_ht:N \l__coffin_display_coffin }
34218 \box_set_dp:Nn \l__coffin_aligned_coffin
34219 { \box_dp:N \l__coffin_display_coffin }
34220 \box_set_wd:Nn \l__coffin_aligned_coffin
34221 { \box_wd:N \l__coffin_display_coffin }
34222 \box_set_eq:NN \l__coffin_display_coffin \l__coffin_aligned_coffin
34223 }
34224 \cs_new_protected:Npn \coffin_show_structure:N
34225 { \__coffin_show_structure:NN \msg_show:nneeee }
34226 \cs_generate_variant:Nn \coffin_show_structure:N { c }
34227 \cs_new_protected:Npn \coffin_log_structure:N
34228 { \__coffin_show_structure:NN \msg_log:nneeee }
34229 \cs_generate_variant:Nn \coffin_log_structure:N { c }
34230 \cs_new_protected:Npn \__coffin_show_structure:NN #1#2
34231 {
34232 \__coffin_if_exist:NT #2
34233 {
34234 #1 { coffin } { show }
34235 { \token_to_str:N #2 }
34236 {
34237 \iow_newline: >~ ht ~~~ \dim_eval:n { \coffin_ht:N #2 }
34238 \iow_newline: >~ dp ~~~ \dim_eval:n { \coffin_dp:N #2 }
34239 \iow_newline: >~ wd ~~~ \dim_eval:n { \coffin_wd:N #2 }
34240 }
34241 {
34242 \prop_map_function:cN
34243 { coffin ~ \__coffin_to_value:N #2 ~ poles }
34244 \msg_show_item_unbraced:nn
34245 }
34246 { }

```

```

34247 }
34248 }
34249 \cs_new_protected:Npn \coffin_show:N #1
34250 { \coffin_show:Nnn #1 \c_max_int \c_max_int }
34251 \cs_generate_variant:Nn \coffin_show:N { c }
34252 \cs_new_protected:Npn \coffin_log:N #1
34253 { \coffin_log:Nnn #1 \c_max_int \c_max_int }
34254 \cs_generate_variant:Nn \coffin_log:N { c }
34255 \cs_new_protected:Npn \coffin_show:Nnn
34256 { \__coffin_show:NNNnn \msg_term:nneeee \box_show:Nnn }
34257 \cs_generate_variant:Nn \coffin_show:Nnn { c }
34258 \cs_new_protected:Npn \coffin_log:Nnn
34259 { \__coffin_show:NNNnn \msg_log:nneeee \box_show:Nnn }
34260 \cs_generate_variant:Nn \coffin_log:Nnn { c }
34261 \cs_new_protected:Npn \__coffin_show:NNNnn #1#2#3#4#5
34262 {
34263     \__coffin_if_exist:NT #3
34264     {
34265         \__coffin_show_structure:NN #1 #3
34266         #2 #3 {#4} {#5}
34267     }
34268 }
34269 \msg_new:nnnn { coffin } { no-pole-intersection }
34270 { No~intersection~between~coffin~poles. }
34271 {
34272     LaTeX~was~asked~to~find~the~intersection~between~two~poles,~
34273     but~they~do~not~have~a~unique~meeting~point:~
34274     the~value~(0pt,~0pt)~will~be~used.
34275 }
34276 \msg_new:nnnn { coffin } { unknown }
34277 { Unknown~coffin~'#1'. }
34278 { The~coffin~'#1'~was~never~defined. }
34279 \msg_new:nnnn { coffin } { unknown-pole }
34280 { Pole~'#1'~unknown~for~coffin~'#2'. }
34281 {
34282     LaTeX~was~asked~to~find~a~typesetting~pole~for~a~coffin,~
34283     but~either~the~coffin~does~not~exist~or~the~pole~name~is~wrong.
34284 }
34285 \msg_new:nnn { coffin } { show }
34286 {
34287     Size~of~coffin~#1 : #2 \\
34288     Poles~of~coffin~#1 : #3 .
34289 }
34290 %% File: l3luatex.dtx
34291 \cs_new_eq:NN \__lua_escape:n \tex_luaescapestring:D
34292 \cs_new_eq:NN \__lua_now:n \tex_directlua:D

```

34293	\cs_new_eq:NN __lua_shipout:n \tex_latelua:D	34293
34294	\cs_undefine:N \lua_escape:e	34294
34295	\cs_undefine:N \lua_now:e	34295
34296	\cs_new:Npn \lua_now:e #1 { __lua_now:n {#1} }	34296
34297	\cs_new:Npn \lua_now:n #1 { \lua_now:e { \exp_not:n {#1} } }	34297
34298	\cs_new_protected:Npn \lua_shipout_e:n #1 { __lua_shipout:n {#1} }	34298
34299	\cs_new_protected:Npn \lua_shipout:n #1	34299
34300	{ \lua_shipout_e:n { \exp_not:n {#1} } }	34300
34301	\cs_new:Npn \lua_escape:e #1 { __lua_escape:n {#1} }	34301
34302	\cs_new:Npn \lua_escape:n #1 { \lua_escape:e { \exp_not:n {#1} } }	34302
34303	\str_new:N \l__lua_err_msg_str	34303
34304	\cs_new_protected:Npn \lua_load_module:n #1	34304
34305	{	34305
34306	\bool_if:nF { __lua_load_module_p:n { #1 } }	34306
34307	{	34307
34308	\msg_error:nnnV	34308
34309	{ luatex } { module-not-found } { #1 } \l__lua_err_msg_str	34309
34310	}	34310
34311	}	34311
34312	\sys_if_engine luatex:F	34312
34313	{	34313
34314	\clist_map_inline:nn	34314
34315	{	34315
34316	\lua_escape:n , \lua_escape:e ,	34316
34317	\lua_now:n , \lua_now:e	34317
34318	}	34318
34319	{	34319
34320	\cs_gset:Npn #1 ##1	34320
34321	{	34321
34322	\msg_expandable_error:nnn	34322
34323	{ luatex } { luatex-required } { #1 }	34323
34324	}	34324
34325	}	34325
34326	\clist_map_inline:nn	34326
34327	{ \lua_shipout_e:n , \lua_shipout:n, \lua_load_module:n }	34327
34328	{	34328
34329	\cs_gset_protected:Npn #1 ##1	34329
34330	{	34330
34331	\msg_error:nnn	34331
34332	{ luatex } { luatex-required } { #1 }	34332
34333	}	34333
34334	}	34334
34335	}	34335
34336	\msg_new:nnnn { luatex } { luatex-required }	34336
34337	{ LuaTeX-engine~not~in~use!~Ignoring~#1. }	34337
34338	{	34338

```
34339 The~feature~you~are~using~is~only~available~ 34339
34340 with~the~LuaTeX~engine.~LaTeX3~ignored~'#1'. 34340
34341 } 34341
34342 \msg_new:nnnn { luatex } { module-not-found } 34342
34343 { Lua~module~'#1'~not~found. } 34343
34344 { 34344
34345 The~file~'#1.lua'~could~not~be~found.~Please~ensure~ 34345
34346 that~the~file~was~properly~installed~and~that~the~ 34346
34347 filename~database~is~current. \\ \\ 34347
34348 The~Lua~loader~provided~this~additional~information: \\ 34348
34349 #2 34349
34350 } 34350
34351 \prop_gput:Nnn \g_msg_module_name_prop { luatex } { LaTeX } 34351
34352 \prop_gput:Nnn \g_msg_module_type_prop { luatex } { } 34352
34353 %% File: l3unicode.dtx 34353
34354 \sys_if_engine_opentype:TF 34354
34355 { 34355
34356 \cs_new:Npn \codepoint_str_generate:n #1 34356
34357 { 34357
34358 \int_compare:nNnTF {#1} = { \_ } 34358
34359 { ~ } 34359
34360 { \char_generate:nn {#1} { 12 } } 34360
34361 } 34361
34362 \cs_new:Npn \codepoint_generate:nn #1#2 34362
34363 { 34363
34364 \int_compare:nNnTF {#1} = { \_ } 34364
34365 { ~ } 34365
34366 { 34366
34367 \__kernel_exp_not:w \exp_after:wN \exp_after:wN \exp_after:wN 34367
34368 { \char_generate:nn {#1} {#2} } 34368
34369 } 34369
34370 } 34370
34371 } 34371
34372 { 34372
34373 \cs_new:Npn \codepoint_str_generate:n #1 34373
34374 { 34374
34375 \int_compare:nNnTF {#1} = { \_ } 34375
34376 { ~ } 34376
34377 { 34377
34378 \use:e 34378
34379 { 34379
34380 \exp_not:N \__codepoint_str_generate:nnnn 34380
34381 \__kernel_codepoint_to_bytes:n {#1} 34381
34382 } 34382
34383 } 34383
34384 } 34384
```



```
34385 \cs_new:Npn \__codepoint_str_generate:nnnn #1#2#3#4 34385
34386 { 34386
34387   \char_generate:nn {#1} { 12 } 34387
34388   \tl_if_blank:nF {#2} 34388
34389   { 34389
34390     \char_generate:nn {#2} { 12 } 34390
34391     \tl_if_blank:nF {#3} 34391
34392     { 34392
34393       \char_generate:nn {#3} { 12 } 34393
34394       \tl_if_blank:nF {#4} 34394
34395       { \char_generate:nn {#4} { 12 } } 34395
34396     } 34396
34397   } 34397
34398 } 34398
34399 \cs_new:Npn \codepoint_generate:nn #1#2 34399
34400 { 34400
34401   \int_compare:nNnTF {#1} = { `\_ } 34401
34402   { ~ } 34402
34403   { 34403
34404     \int_compare:nNnTF {#1} < { "80 } 34404
34405     { 34405
34406       \__kernel_exp_not:w \exp_after:wN \exp_after:wN \exp_after:wN 34406
34407       { \char_generate:nn {#1} {#2} } 34407
34408     } 34408
34409     { 34409
34410       \use:e 34410
34411       { 34411
34412         \exp_not:N \__codepoint_generate:nnnn 34412
34413         \__kernel_codepoint_to_bytes:n {#1} 34413
34414       } 34414
34415     } 34415
34416   } 34416
34417 } 34417
34418 \cs_new:Npn \__codepoint_generate:nnnn #1#2#3#4 34418
34419 { 34419
34420   \__kernel_exp_not:w \exp_after:wN 34420
34421   { 34421
34422     \tex_expanded:D 34422
34423     { 34423
34424       \__codepoint_generate:n {#1} 34424
34425       \__codepoint_generate:n {#2} 34425
34426       \tl_if_blank:nF {#3} 34426
34427       { 34427
34428         \__codepoint_generate:n {#3} 34428
34429         \tl_if_blank:nF {#4} 34429
34430         { \__codepoint_generate:n {#4} } 34430
```

```

34431         }
34432     }
34433 }
34434 }
34435 \cs_new:Npn \__codepoint_generate:n #1
34436 {
34437     \__kernel_exp_not:w \exp_after:wN \exp_after:wN \exp_after:wN
34438     { \char_generate:nn {#1} { 13 } }
34439 }
34440 }
34441 \cs_new:Npn \__kernel_codepoint_to_bytes:n #1
34442 {
34443     \exp_args:Nf \__codepoint_to_bytes_auxi:n
34444     { \int_eval:n {#1} }
34445 }
34446 \cs_new:Npn \__codepoint_to_bytes_auxi:n #1
34447 {
34448     \if_int_compare:w #1 > "80 \exp_stop_f:
34449     \if_int_compare:w #1 < "800 \exp_stop_f:
34450         \__codepoint_to_bytes_outputi:nw
34451         { \__codepoint_to_bytes_auxii:Nnn C {#1} { 64 } }
34452         \__codepoint_to_bytes_outputii:nw
34453         { \__codepoint_to_bytes_auxiii:n {#1} }
34454     \else:
34455         \if_int_compare:w #1 < "10000 \exp_stop_f:
34456             \__codepoint_to_bytes_outputi:nw
34457             { \__codepoint_to_bytes_auxii:Nnn E {#1} { 64 * 64 } }
34458             \__codepoint_to_bytes_outputii:nw
34459             {
34460                 \__codepoint_to_bytes_auxiii:n
34461                 { \int_div_truncate:nn {#1} { 64 } }
34462             }
34463             \__codepoint_to_bytes_outputiii:nw
34464             { \__codepoint_to_bytes_auxiii:n {#1} }
34465         \else:
34466             \__codepoint_to_bytes_outputi:nw
34467             {
34468                 \__codepoint_to_bytes_auxii:Nnn F
34469                 {#1} { 64 * 64 * 64 }
34470             }
34471             \__codepoint_to_bytes_outputii:nw
34472             {
34473                 \__codepoint_to_bytes_auxiii:n
34474                 { \int_div_truncate:nn {#1} { 64 * 64 } }
34475             }
34476             \__codepoint_to_bytes_outputiii:nw

```

```

34477         {
34478             \__codepoint_to_bytes_auxiii:n
34479             { \int_div_truncate:nn {#1} { 64 } }
34480         }
34481         \__codepoint_to_bytes_outputiv:nw
34482         { \__codepoint_to_bytes_auxiii:n {#1} }
34483         \fi:
34484         \fi:
34485         \else:
34486             \__codepoint_to_bytes_outputi:nw {#1}
34487         \fi:
34488         \__codepoint_to_bytes_end: { } { } { } { }
34489     }
34490     \cs_new:Npn \__codepoint_to_bytes_auxii:Nnn #1#2#3
34491     { "#10 + \int_div_truncate:nn {#2} {#3} }
34492     \cs_new:Npn \__codepoint_to_bytes_auxiii:n #1
34493     { \int_mod:nn {#1} { 64 } + 128 }
34494     \cs_new:Npn \__codepoint_to_bytes_outputi:nw
34495     #1 #2 \__codepoint_to_bytes_end: #3
34496     { \__codepoint_to_bytes_output:fnn { \int_eval:n {#1} } { } {#2} }
34497     \cs_new:Npn \__codepoint_to_bytes_outputii:nw
34498     #1 #2 \__codepoint_to_bytes_end: #3#4
34499     { \__codepoint_to_bytes_output:fnn { \int_eval:n {#1} } { {#3} } {#2} }
34500     \cs_new:Npn \__codepoint_to_bytes_outputiii:nw
34501     #1 #2 \__codepoint_to_bytes_end: #3#4#5
34502     {
34503         \__codepoint_to_bytes_output:fnn
34504         { \int_eval:n {#1} } { {#3} {#4} } {#2}
34505     }
34506     \cs_new:Npn \__codepoint_to_bytes_outputiv:nw
34507     #1 #2 \__codepoint_to_bytes_end: #3#4#5#6
34508     {
34509         \__codepoint_to_bytes_output:fnn
34510         { \int_eval:n {#1} } { {#3} {#4} {#5} } {#2}
34511     }
34512     \cs_new:Npn \__codepoint_to_bytes_output:nnn #1#2#3
34513     {
34514         #3
34515         \__codepoint_to_bytes_end: #2 {#1}
34516     }
34517     \cs_generate_variant:Nn \__codepoint_to_bytes_output:nnn { f }
34518     \cs_new:Npn \__codepoint_to_bytes_end: { }
34519     \cs_new:Npn \codepoint_to_category:n #1
34520     {
34521         \cs:w
34522         c__codepoint_category_

```

```

34523 \tex_romannumeral:D 34523
34524 \__kernel_codepoint_data:nn { category } {#1} 34524
34525 _str 34525
34526 \cs_end: 34526
34527 } 34527
34528 \cs_new:Npn \codepoint_to_nfd:n #1 34528
34529 { \exp_args:Ne \__codepoint_to_nfd:n { \int_eval:n {#1} } } 34529
34530 \cs_new:Npn \__codepoint_to_nfd:n #1 34530
34531 { \__codepoint_to_nfd:nn {#1} { \char_value_catcode:n {#1} } } 34531
34532 \sys_if_engine_opentype:F 34532
34533 { 34533
34534 \cs_gset:Npn \__codepoint_to_nfd:n #1 34534
34535 { 34535
34536 \int_compare:nNnTF {#1} > { "80 } 34536
34537 { \__codepoint_to_nfd:nn {#1} { 12 } } 34537
34538 { \__codepoint_to_nfd:nn {#1} { \char_value_catcode:n {#1} } } 34538
34539 } 34539
34540 } 34540
34541 \cs_new:Npn \__codepoint_to_nfd:nn #1#2 34541
34542 { 34542
34543 \exp_args:Ne \__codepoint_to_nfd:nnn 34543
34544 { \__codepoint_nfd:n {#1} } {#1} {#2} 34544
34545 } 34545
34546 \cs_new:Npn \__codepoint_to_nfd:nnn #1#2#3 { \__codepoint_to_nfd:nnnn #1 {#2} {#3} } 34546
34547 \cs_new:Npn \__codepoint_to_nfd:nnnn #1#2#3#4 34547
34548 { 34548
34549 \int_compare:nNnTF {#1} = {#3} 34549
34550 { \codepoint_generate:nn {#1} {#4} } 34550
34551 { 34551
34552 \__codepoint_to_nfd:nn {#1} {#4} 34552
34553 \tl_if_blank:nF {#2} 34553
34554 { \__codepoint_to_nfd:nn {#2} {#4} } 34554
34555 } 34555
34556 } 34556
34557 \int_const:Nn \c__codepoint_block_size_int { 64 } 34557
34558 \ior_new:N \g__codepoint_data_ior 34558
34559 \group_begin: 34559
34560 \clist_map_inline:nn 34560
34561 { category , grapheme , lowercase , uppercase , wordbreak } 34561
34562 { 34562
34563 \cs_set_nopar:cpn { l__codepoint_ #1 _block_clist } { } 34563
34564 \cs_set_nopar:cpn { l__codepoint_ #1 _block_tl } { 1 } 34564
34565 \cs_set_nopar:cpn { l__codepoint_ #1 _pos_tl } { 0 } 34565
34566 \cs_set_nopar:cpn { l__codepoint_ #1 _next_tl } { 0 } 34566
34567 \intarray_new:cn { g__codepoint_ #1 _index_intarray } 34567
34568 { \int_div_truncate:nn { "110000 } \c__codepoint_block_size_int } 34568

```

```

34569 } 34569
34570 \cs_set_nopar:Npn \l__codepoint_matched_block_tl { 0 } 34570
34571 \cs_set_nopar:Npn \l__codepoint_uppercase_default_tl { 0 } 34571
34572 \cs_set_nopar:Npn \l__codepoint_lowercase_default_tl { 0 } 34572
34573 \cs_set_protected:Npn \__codepoint_data_auxi:w #1#2#3 34573
34574 { 34574
34575 \quark_if_recursion_tail_stop:n {#3} 34575
34576 \cs_set_nopar:cpn { l__codepoint_ #2 _ #3 _tl } {#1} 34576
34577 \str_const:cn { c__codepoint_ #2 _ \tex_romannumeral:D #1 _str } {#3} 34577
34578 \exp_args:Ne \__codepoint_data_auxi:w { \int_eval:n { #1 + 1 } } {#2} 34578
34579 } 34579
34580 \__codepoint_data_auxi:w { 1 } { category } 34580
34581 { Lu } { Ll } { Lt } { Lm } { Lo } 34581
34582 { Mn } { Me } { Mc } 34582
34583 { Nd } { Nl } { No } 34583
34584 { Zs } { Zl } { Zp } 34584
34585 { Cc } { Cf } { Co } { Cs } { Cn } 34585
34586 { Pd } { Ps } { Pe } { Pc } { Po } { Pi } { Pf } 34586
34587 { Sm } { Sc } { Sk } { So } 34587
34588 \q_recursion_tail 34588
34589 \q_recursion_stop 34589
34590 \cs_set_eq:NN \l__codepoint_category_default_tl \l__codepoint_category_Cn_tl 34590
34591 \__codepoint_data_auxi:w { 1 } { grapheme } 34591
34592 { Control } 34592
34593 { CR } { LF } { ZWJ } 34593
34594 { Extend } 34594
34595 { L } { LV } { LVT } { T } { V } 34595
34596 { Prepend } 34596
34597 { Regional_Indicator } 34597
34598 { SpacingMark } 34598
34599 { Other } 34599
34600 \q_recursion_tail 34600
34601 \q_recursion_stop 34601
34602 \cs_set_eq:NN \l__codepoint_grapheme_default_tl \l__codepoint_grapheme_Other_tl 34602
34603 \__codepoint_data_auxi:w { 1 } { wordbreak } 34603
34604 { Double_Quote } { Single_Quote } 34604
34605 { CR } { LF } { Newline } 34605
34606 { WSegSpace } { ZWJ } 34606
34607 { Extend } { ExtendNumLet } 34607
34608 { Regional_Indicator } 34608
34609 { Format } 34609
34610 { Katakana } 34610
34611 { ALetter } { MidLetter } { Hebrew_Letter } 34611
34612 { Numeric } { MidNum } { MidNumLet } 34612
34613 { Other } 34613
34614 \q_recursion_tail 34614

```

```

34615 \q_recursion_stop 34615
34616 \cs_set_eq:NN \l__codepoint_wordbreak_default_tl \l__codepoint_wordbreak_Other_tl 34616
34617 \cs_set_protected:Npn \__codepoint_data_auxi:w #1 ;~ #2 ~ #3 \q_stop 34617
34618 { \__codepoint_data_auxii:w #1 .. \q_stop {#2} } 34618
34619 \cs_set_protected:Npn \__codepoint_data_auxii:w #1 .. #2 \q_stop 34619
34620 { \__codepoint_data_auxiii:w #1 ~ .. #2 ~ \q_stop } 34620
34621 \cs_set_protected:Npn \__codepoint_data_auxiii:w #1 ~ #2 .. #3 ~ #4 \q_stop #5#6 34621
34622 { 34622
34623 \cs_set_nopar:cpe { l__codepoint_ #6 _ \tex_romannumeral:D "#1 _tl } 34623
34624 { 34624
34625 {#3} 34625
34626 { \use:c { l__codepoint_ #6 _ #5 _tl } } 34626
34627 } 34627
34628 } 34628
34629 \cs_set_protected:Npn \__codepoint_data_auxvi:w #1#2 34629
34630 { 34630
34631 \ior_open:Nn \g__codepoint_data_ior {#1} 34631
34632 \ior_str_map_inline:Nn \g__codepoint_data_ior 34632
34633 { 34633
34634 \str_if_eq:eeF { \tl_head:w ##1 \c_hash_str \q_stop } { \c_hash_str } 34634
34635 { 34635
34636 \tl_if_blank:nF {##1} 34636
34637 { \__codepoint_data_auxi:w ##1 \q_stop {#2} } 34637
34638 } 34638
34639 } 34639
34640 \ior_close:N \g__codepoint_data_ior 34640
34641 } 34641
34642 \__codepoint_data_auxvi:w { GraphemeBreakProperty.txt } { grapheme } 34642
34643 \__codepoint_data_auxvi:w { WordBreakProperty.txt } { wordbreak } 34643
34644 \cs_set_protected:Npn \__codepoint_data_property:nnnn #1#2#3#4 34644
34645 { 34645
34646 \int_compare:nNnT {#3} > { \use:c { l__codepoint_ #4 _next_tl } } 34646
34647 { 34647
34648 \__codepoint_range:nnv {#3} {#4} 34648
34649 { l__codepoint_ #4 _default_tl } 34649
34650 } 34650
34651 \__codepoint_add:nn {#4} {#2} 34651
34652 \tl_set:ce { l__codepoint_ #4 _next_tl } { \int_eval:n { #3 + 1 } } 34652
34653 \tl_if_blank:nF {#1} 34653
34654 { 34654
34655 \__codepoint_range:nnn {"#1} {#4} {#2} 34655
34656 \__codepoint_add:nn {#4} {#2} 34656
34657 \tl_set:ce { l__codepoint_ #4 _next_tl } { \int_eval:n { "#1 + 1 } } 34657
34658 } 34658
34659 } 34659
34660 \cs_set_protected:Npn \__codepoint_data_auxi:w 34660

```

```

34661 #1 ; #2 ; #3 ; #4 ; #5 ; #6 ; #7 ; #8 ; #9 ; 34661
34662 { 34662
34663 \tl_if_blank:nF {#6} 34663
34664 { 34664
34665 \tl_if_head_eq_charcode:nNF {#6} < % > 34665
34666 { \__codepoint_data_auxii:w #1 ; #6 ~ \q_stop } 34666
34667 } 34667
34668 \__codepoint_data_auxiii:w #1 ; #2 ; #3 ; 34668
34669 } 34669
34670 \cs_set_protected:Npn \__codepoint_data_auxii:w #1 ; #2 ~ #3 \q_stop 34670
34671 { 34671
34672 \tl_const:ce 34672
34673 { c__codepoint_nfd_ \codepoint_str_generate:n {"#1} _tl } 34673
34674 { 34674
34675 {"#2} 34675
34676 { \tl_if_blank:nF {#3} {"#3} } 34676
34677 } 34677
34678 } 34678
34679 \cs_set_protected:Npn \__codepoint_data_auxiii:w 34679
34680 #1 ; #2 ; #3 ; #4 ; #5 ; #6 ; #7 ; #8 ; #9 ~ \q_stop 34680
34681 { 34681
34682 \use:e 34682
34683 { 34683
34684 \__codepoint_data_auxiv:w 34684
34685 #1 ; #2 ; 34685
34686 \__codepoint_data_category:n {#3} ; 34686
34687 \__codepoint_data_offset:nn {#1} {#7} ; 34687
34688 \__codepoint_data_offset:nn {#1} {#8} ; 34688
34689 #9; 34689
34690 } 34690
34691 } 34691
34692 \cs_set:Npn \__codepoint_data_category:n #1 34692
34693 { \use:c { l__codepoint_category_ #1 _tl } } 34693
34694 \cs_set:Npn \__codepoint_data_offset:nn #1#2 34694
34695 { 34695
34696 \tl_if_blank:nTF {#2} 34696
34697 { 0 } 34697
34698 { \int_eval:n { "#2 - "#1 } } 34698
34699 } 34699
34700 \cs_set_protected:Npn \__codepoint_data_auxiv:w #1 ; #2 ; #3 ; #4 ; #5 ; #6 ; 34700
34701 { 34701
34702 \int_compare:nNnT {"#1} > \l__codepoint_category_next_tl 34702
34703 { 34703
34704 \__codepoint_data_auxv:nnnw {#1} {#3} {#4} {#5} 34704
34705 #2 Last> \q_stop 34705
34706 } 34706

```



```

34707 \__codepoint_add:nn { category } {#3}
34708 \__codepoint_add:nn { uppercase } {#4}
34709 \__codepoint_add:nn { lowercase } {#5}
34710 \int_compare:nNnF {#4} = { \__codepoint_data_offset:nn {#1} {#6} }
34711 {
34712   \tl_const:ce
34713     { c__codepoint_titlecase_ \codepoint_str_generate:n {"#1} _tl }
34714     { {"#6} { } { } }
34715   }
34716 \__codepoint_data_auxvi:nn { grapheme } {"#1}
34717 \__codepoint_data_auxvi:nn { wordbreak } {"#1}
34718 \int_compare:nNnT {"#1} = { "AC00 }
34719 {
34720   \int_step_inline:nnn { "AC01 } { "D7A2 }
34721   { \__codepoint_data_auxvi:nn { grapheme } {#1} }
34722 }
34723 \tl_set:Nc \l__codepoint_category_next_tl
34724   { \int_eval:n { "#1 + 1 } }
34725 \tl_set_eq:NN \l__codepoint_lowercase_next_tl \l__codepoint_category_next_tl
34726 \tl_set_eq:NN \l__codepoint_uppercase_next_tl \l__codepoint_category_next_tl
34727 }
34728 \cs_set_protected:Npn \__codepoint_add:nn #1#2
34729 {
34730   \clist_put_right:cn { l__codepoint_ #1 _block_clist } {#2}
34731   \int_compare:nNnT { \clist_count:c { l__codepoint_ #1 _block_clist } }
34732     = \c__codepoint_block_size_int
34733     { \__codepoint_save_blocks:nn {#1} { 1 } }
34734 }
34735 \cs_set_protected:Npn \__codepoint_data_auxv:nnnnw #1#2#3#4#5 Last> #6 \q_stop
34736 {
34737   \tl_if_blank:nTF {#6}
34738   {
34739     \__codepoint_range:nno {"#1} { category }
34740     \l__codepoint_category_default_tl
34741     \__codepoint_range:nnn {"#1} { uppercase } { 0 }
34742     \__codepoint_range:nnn {"#1} { lowercase } { 0 }
34743   }
34744   {
34745     \__codepoint_range:nnn {"#1} { category } {#2}
34746     \__codepoint_range:nnn {"#1} { uppercase } {#3}
34747     \__codepoint_range:nnn {"#1} { lowercase } {#4}
34748   }
34749 }
34750 \cs_set_protected:Npn \__codepoint_data_auxvi:nn #1#2
34751 {
34752   \cs_if_exist:cT

```

```

34753 { l__codepoint_ #1 _ \tex_romannumeral:D #2 _tl } 34753
34754 { 34754
34755 \exp_after:wN \exp_after:wN \exp_after:wN \__codepoint_data_property:nnnn 34755
34756 \cs:w 34756
34757 l__codepoint_ #1 _ \tex_romannumeral:D #2 _tl 34757
34758 \cs_end: {#2} {#1} 34758
34759 } 34759
34760 } 34760
34761 \cs_set_protected:Npn \__codepoint_range:nnn #1#2 34761
34762 { 34762
34763 \exp_args:Nf \__codepoint_range_aux:nnn 34763
34764 { \int_eval:n { #1 - \use:c { l__codepoint_ #2 _next_tl } } } 34764
34765 {#2} 34765
34766 } 34766
34767 \cs_set_protected:Npn \__codepoint_range:nno { \exp_args:Nnno \__codepoint_range:nnn } 34767
34768 \cs_set_protected:Npn \__codepoint_range:nnv { \exp_args:Nnnv \__codepoint_range:nnn } 34768
34769 \cs_set_protected:Npn \__codepoint_range_aux:nnn #1#2 34769
34770 { 34770
34771 \exp_args:Nf \__codepoint_range:nnnn 34771
34772 { 34772
34773 \int_min:nn 34773
34774 {#1} 34774
34775 { 34775
34776 \c__codepoint_block_size_int 34776
34777 - \clist_count:c { l__codepoint_ #2 _block_clist } 34777
34778 } 34778
34779 } 34779
34780 {#1} {#2} 34780
34781 } 34781
34782 \cs_set_protected:Npn \__codepoint_range:nnnn #1#2#3#4 34782
34783 { 34783
34784 \prg_replicate:nn {#1} 34784
34785 { \clist_put_right:cn { l__codepoint_ #3 _block_clist } {#4} } 34785
34786 \int_compare:nNnT { \clist_count:c { l__codepoint_ #3 _block_clist } } 34786
34787 = \c__codepoint_block_size_int 34787
34788 { \__codepoint_save_blocks:nn {#3} { 1 } } 34788
34789 \int_compare:nNnF 34789
34790 { \int_div_truncate:nn { #2 - #1 } \c__codepoint_block_size_int } = 0 34790
34791 { 34791
34792 \tl_set:ce { l__codepoint_ #3 _block_clist } 34792
34793 { 34793
34794 \exp_args:NNe \use:nn \use_none:n 34794
34795 { \prg_replicate:nn { \c__codepoint_block_size_int } { , #4 } } 34795
34796 } 34796
34797 \__codepoint_save_blocks:nn {#3} 34797
34798 { \int_div_truncate:nn { (#2 - #1) } \c__codepoint_block_size_int } 34798

```

```

34799     }
34800     \prg_replicate:nn
34801     { \int_mod:nn { #2 - #1 } \c__codepoint_block_size_int }
34802     { \clist_put_right:ce { l__codepoint_ #3 _block_clist } {#4} }
34803 }
34804 \cs_set_protected:Npn \__codepoint_save_blocks:nn #1#2
34805 {
34806     \tl_set_eq:Nc \l__codepoint_matched_block_tl { l__codepoint_ #1 _block_tl }
34807     \int_step_inline:nn { \tl_use:c { l__codepoint_ #1 _block_tl } - 1 }
34808     {
34809         \tl_if_eq:ccT { l__codepoint_ #1 _block_clist }
34810         { l__codepoint_ #1 _block_ ##1 _clist }
34811         { \tl_set:Nn \l__codepoint_matched_block_tl {##1} }
34812     }
34813     \int_compare:nNnT
34814     { \tl_use:c { l__codepoint_ #1 _block_tl } } = \l__codepoint_matched_block_tl
34815     {
34816         \clist_set_eq:cc
34817         {
34818             l__codepoint_ #1 _block_
34819             \tl_use:c { l__codepoint_ #1 _block_tl } _clist
34820         }
34821         { l__codepoint_ #1 _block_clist }
34822         \tl_set:ce { l__codepoint_ #1 _block_tl }
34823         { \int_eval:n { \tl_use:c { l__codepoint_ #1 _block_tl } + 1 } }
34824     }
34825     \int_step_inline:nnn
34826     { \tl_use:c { l__codepoint_ #1 _pos_tl } + 1 }
34827     { \tl_use:c { l__codepoint_ #1 _pos_tl } + #2 }
34828     {
34829         \exp_args:Nc \__kernel_intarray_gset:Nnn
34830         { g__codepoint_ #1 _index_intarray }
34831         {##1}
34832         \l__codepoint_matched_block_tl
34833     }
34834     \tl_set:ce { l__codepoint_ #1 _pos_tl }
34835     { \int_eval:n { \tl_use:c { l__codepoint_ #1 _pos_tl } + #2 } }
34836     \clist_clear:c { l__codepoint_ #1 _block_clist }
34837 }
34838 \cs_set_protected:Npn \__codepoint_finalise_blocks:n #1
34839 {
34840     \clist_map_inline:nn {#1}
34841     {
34842         \exp_args:Nnnv \__codepoint_range:nnn { "110000 } {##1}
34843         { l__codepoint_ ##1 _default_tl }
34844         \__codepoint_finalise_blocks_aux:n {##1}

```

```

34845     }
34846 }
34847 \cs_set_protected:Npn \__codepoint_finalise_blocks_aux:n #1
34848 {
34849     \cs_gset_eq:cc { c__codepoint_ #1 _index_intarray } { g__codepoint_ #1
34850 _index_intarray }
34851     \cs_undefine:c { g__codepoint_ #1 _index_intarray }
34852     \intarray_new:cn { g__codepoint_ #1 _blocks_intarray }
34853     { ( \tl_use:c { l__codepoint_ #1 _block_tl } - 1 ) *
34854 \c__codepoint_block_size_int }
34855     \int_step_inline:nn { \tl_use:c { l__codepoint_ #1 _block_tl } - 1 }
34856 {
34857     \exp_args:Nv \__codepoint_finalise_blocks:nnn
34858     { l__codepoint_ #1 _block_ ##1 _clist }
34859     {##1} {#1}
34860 }
34861 \cs_gset_eq:cc { c__codepoint_ #1 _blocks_intarray }
34862 { g__codepoint_ #1 _blocks_intarray }
34863 \cs_undefine:c { g__codepoint_ #1 _blocks_intarray }
34864 }
34865 \cs_set_protected:Npn \__codepoint_finalise_blocks:nnn #1#2#3
34866 {
34867     \exp_args:Nnf \__codepoint_finalise_blocks:nnnw { 1 }
34868     { \int_eval:n { ( #2 - 1 ) * \c__codepoint_block_size_int } }
34869     {#3}
34870     #1 , \q_recursion_tail , \q_recursion_stop
34871 }
34872 \cs_set_protected:Npn \__codepoint_finalise_blocks:nnnw #1#2#3#4 ,
34873 {
34874     \quark_if_recursion_tail_stop:n {#4}
34875     \intarray_gset:cn { g__codepoint_ #3 _blocks_intarray }
34876     { #1 + #2 }
34877     {#4}
34878     \exp_args:Nf \__codepoint_finalise_blocks:nnnw
34879     { \int_eval:n { #1 + 1 } } {#2} {#3}
34880 }
34881 \ior_open:Nn \g__codepoint_data_ior { UnicodeData.txt }
34882 \char_set_catcode_space:n { \_ }%
34883 \ior_map_variable:NNn \g__codepoint_data_ior \l__codepoint_tmpa_tl
34884 {%
34885     \if_meaning:w \l__codepoint_tmpa_tl \c_space_tl
34886     \exp_after:wN \ior_map_break:
34887     \fi:
34888     \exp_after:wN \__codepoint_data_auxi:w \l__codepoint_tmpa_tl \q_stop
34889 }%
34890 \char_set_catcode_ignore:n { \_ }%

```

```

34889 \__codepoint_finalise_blocks:n
34890 { category , grapheme , lowercase , uppercase , wordbreak }
34891 \group_end:
34892 \cs_new:Npn \__kernel_codepoint_data:nn #1#2
34893 {
34894 \exp_args:Nf \__codepoint_data:nnn
34895 {
34896 \int_eval:n
34897 {
34898 \c__codepoint_block_size_int *
34899 (
34900 \intarray_item:cn { c__codepoint_ #1 _index_intarray }
34901 {
34902 \int_div_truncate:nn {#2}
34903 \c__codepoint_block_size_int
34904 + 1
34905 }
34906 - 1
34907 )
34908 }
34909 }
34910 {#2} {#1}
34911 }
34912 \cs_new:Npn \__codepoint_data:nnn #1#2#3
34913 {
34914 \intarray_item:cn { c__codepoint_ #3 _blocks_intarray }
34915 { #1 + \int_mod:nn {#2} \c__codepoint_block_size_int + 1 }
34916 }
34917 \group_begin:
34918 \ior_open:Nn \g__codepoint_data_ior { CaseFolding.txt }
34919 \cs_set_protected:Npn \__codepoint_data_auxi:w #1 ;~ #2 ;~ #3 ; #4 \q_stop
34920 {
34921 \if:w \tl_head:n { #2 ? } C
34922 \reverse_if:N \if_int_compare:w
34923 \int_eval:n { \__kernel_codepoint_data:nn { lowercase } {"#1} + "#1 }
34924 = "#3 ~
34925 \tl_const:ce
34926 { c__codepoint_casefold_ \codepoint_str_generate:n {"#1} _tl }
34927 { {"#3} { } { } }
34928 \fi:
34929 \else:
34930 \if:w \tl_head:n { #2 ? } F
34931 \__codepoint_data_auxii:w #1 ~ #3 ~ \q_stop
34932 \fi:
34933 \fi:
34934 }

```

```
34935 \cs_set_protected:Npn \__codepoint_data_auxii:w #1 ~ #2 ~ #3 ~ #4 \q_stop 34935
34936 { 34936
34937     \tl_const:ce { c__codepoint_casefold_ \codepoint_str_generate:n {"#1} _tl } 34937
34938     { 34938
34939         {"#2} 34939
34940         {"#3} 34940
34941         { \tl_if_blank:nF {#4} { " \int_to_Hex:n {"#4} } } 34941
34942     } 34942
34943 } 34943
34944 \ior_str_map_inline:Nn \g__codepoint_data_ior 34944
34945 { 34945
34946     \reverse_if:N \if:w \c_hash_str \tl_head:w #1 \c_hash_str \q_stop 34946
34947     \__codepoint_data_auxi:w #1 \q_stop 34947
34948     \fi: 34948
34949 } 34949
34950 \ior_close:N \g__codepoint_data_ior 34950
34951 \ior_open:Nn \g__codepoint_data_ior { SpecialCasing.txt } 34951
34952 \cs_set_protected:Npn \__codepoint_data_auxi:w 34952
34953     #1 ;~ #2 ;~ #3 ;~ #4 ; #5 \q_stop 34953
34954     { 34954
34955         \use:n { \__codepoint_data_auxii:w #1 ~ lower ~ #2 ~ } ~ \q_stop 34955
34956         \use:n { \__codepoint_data_auxii:w #1 ~ upper ~ #4 ~ } ~ \q_stop 34956
34957         \str_if_eq:nnF {#3} {#4} 34957
34958         { \use:n { \__codepoint_data_auxii:w #1 ~ title ~ #3 ~ } ~ \q_stop } 34958
34959     } 34959
34960 \cs_set_protected:Npn \__codepoint_data_auxii:w 34960
34961     #1 ~ #2 ~ #3 ~ #4 ~ #5 \q_stop 34961
34962     { 34962
34963         \tl_if_empty:nF {#4} 34963
34964         { 34964
34965             \tl_const:ce { c__codepoint_ #2 case_ \codepoint_str_generate:n {"#1} _tl } 34965
34966             { 34966
34967                 {"#3} 34967
34968                 {"#4} 34968
34969                 { \tl_if_blank:nF {#5} {"#5} } 34969
34970             } 34970
34971         } 34971
34972     } 34972
34973 \ior_str_map_inline:Nn \g__codepoint_data_ior 34973
34974 { 34974
34975     \str_if_eq:eeTF { \tl_head:w #1 \c_hash_str \q_stop } { \c_hash_str } 34975
34976     { 34976
34977         \str_if_eq:eeT 34977
34978         {#1} 34978
34979         { \c_hash_str \c_space_tl Conditional~Mappings } 34979
34980         { \ior_map_break: } 34980
```

```

34981     }
34982     { \__codepoint_data_auxi:w #1 \q_stop }
34983 }
34984 \ior_close:N \g__codepoint_data_ior
34985 \group_end:
34986 \cs_new:Npn \__kernel_codepoint_case:nn #1#2
34987 {
34988     \exp_args:Ne \__codepoint_case:nnn
34989     { \codepoint_str_generate:n {#2} } {#1} {#2}
34990 }
34991 \cs_new:Npn \__codepoint_case:nnn #1#2#3
34992 {
34993     \cs_if_exist:cTF { c__codepoint_ #2 _ #1 _tl }
34994     {
34995         \tl_use:c
34996         { c__codepoint_ #2 _ #1 _tl }
34997     }
34998     { \use:c { __codepoint_ #2 :n } {#3} }
34999 }
35000 \cs_new:Npn \__codepoint_uppercase:n { \__codepoint_case:nn { uppercase } }
35001 \cs_new:Npn \__codepoint_lowercase:n { \__codepoint_case:nn { lowercase } }
35002 \cs_new:Npn \__codepoint_titlecase:n { \__codepoint_case:nn { uppercase } }
35003 \cs_new:Npn \__codepoint_casefold:n { \__codepoint_case:nn { lowercase } }
35004 \cs_new:Npn \__codepoint_case:nn #1#2
35005 {
35006     { \int_eval:n { \__kernel_codepoint_data:nn {#1} {#2} + #2 } }
35007     { }
35008     { }
35009 }
35010 \cs_new:Npn \__kernel_codepoint_to_grapheme_class:n #1
35011 {
35012     \cs:w
35013     c__codepoint_grapheme_
35014     \tex_romannumeral:D
35015     \__kernel_codepoint_data:nn { grapheme } {#1}
35016     _str
35017     \cs_end:
35018 }
35019 \cs_new:Npn \__kernel_codepoint_to_wordbreak_class:n #1
35020 {
35021     \cs:w
35022     c__codepoint_wordbreak_
35023     \tex_romannumeral:D
35024     \__kernel_codepoint_data:nn { wordbreak } {#1}
35025     _str
35026     \cs_end:

```


35027	}	35027
35028	\cs_new:Npn __codepoint_nfd:n #1	35028
35029	{ \exp_args:Ne __codepoint_nfd:nn { \codepoint_str_generate:n {#1} } {#1} }	35029
35030	\cs_new:Npn __codepoint_nfd:nn #1#2	35030
35031	{	35031
35032	\tl_if_exist:cTF { c__codepoint_nfd_ #1 _tl }	35032
35033	{ \tl_use:c { c__codepoint_nfd_ #1 _tl } }	35033
35034	{ {#2} { } }	35034
35035	}	35035
35036	%% File: l3text.dtx	35036
35037	\cs_generate_variant:Nn \tl_if_head_eq_meaning_p:nN { o }	35037
35038	\scan_new:N \s__text_stop	35038
35039	\quark_new:N \q__text_nil	35039
35040	__kernel_quark_new_conditional:Nn __text_quark_if_nil:n { TF }	35040
35041	\quark_new:N \q__text_recursion_tail	35041
35042	\quark_new:N \q__text_recursion_stop	35042
35043	\cs_new:Npn __text_use_i_delimit_by_q_recursion_stop:nw	35043
35044	#1 #2 \q__text_recursion_stop {#1}	35044
35045	__kernel_quark_new_test:N __text_if_q_recursion_tail_stop_do:Nn	35045
35046	__kernel_quark_new_test:N __text_if_q_recursion_tail_stop_do:nn	35046
35047	\scan_new:N \s__text_recursion_tail	35047
35048	\scan_new:N \s__text_recursion_stop	35048
35049	\cs_new:Npn __text_use_i_delimit_by_s_recursion_stop:nw	35049
35050	#1 #2 \s__text_recursion_stop {#1}	35050
35051	\cs_new:Npn __text_if_s_recursion_tail_stop_do:Nn #1	35051
35052	{	35052
35053	\bool_lazy_and:nnTF	35053
35054	{ \cs_if_eq_p:NN \s__text_recursion_tail #1 }	35054
35055	{ \str_if_eq_p:nn { \s__text_recursion_tail } {#1} }	35055
35056	{ __text_use_i_delimit_by_s_recursion_stop:nw }	35056
35057	{ \use_none:n }	35057
35058	}	35058
35059	\cs_new_eq:NN __text_sep: __kernel_int_sep:	35059
35060	\group_begin:	35060
35061	\char_set_catcode_active:n { 0 }	35061
35062	\cs_new:Npn __text_token_to_explicit:N #1	35062
35063	{	35063
35064	\if_catcode:w \exp_not:N #1	35064
35065	\if_catcode:w \scan_stop: \exp_not:N #1	35065
35066	\scan_stop:	35066
35067	\else:	35067
35068	\exp_not:N ^^@	35068
35069	\fi:	35069
35070	\exp_after:wN __text_token_to_explicit_cs:N	35070
35071	\else:	35071
35072	\exp_after:wN __text_token_to_explicit_char:N	35072

```

35073     \fi:
35074     #1
35075 }
35076 \group_end:
35077 \cs_new:Npn \__text_token_to_explicit_cs:N #1
35078 {
35079     \exp_after:wN \if_meaning:w \exp_not:N #1 #1
35080     \exp_after:wN \use:nn \exp_after:wN
35081         \__text_token_to_explicit_cs_aux:N
35082 \else:
35083     \exp_after:wN \exp_not:n
35084 \fi:
35085     {#1}
35086 }
35087 \cs_new:Npn \__text_token_to_explicit_cs_aux:N #1
35088 {
35089     \bool_lazy_or:nnTF
35090     { \token_if_chardef_p:N #1 }
35091     { \token_if_mathchardef_p:N #1 }
35092     {
35093         \char_generate:nn {#1}
35094         {
35095             \if_int_compare:w \char_value_catcode:n {#1} = 10 \exp_stop_f:
35096             10
35097             \else:
35098             12
35099             \fi:
35100         }
35101     }
35102     {#1}
35103 }
35104 \cs_new:Npn \__text_token_to_explicit_char:N #1
35105 {
35106     \if:w
35107         \if_catcode:w ^ \exp_args:No \str_tail:n { \token_to_str:N #1 } ^
35108         \token_to_str:N #1 #1
35109     \else:
35110         AB
35111     \fi:
35112     \exp_after:wN \exp_not:n
35113 \else:
35114     \exp_after:wN \__text_token_to_explicit:n
35115 \fi:
35116     {#1}
35117 }
35118 \cs_new:Npn \__text_token_to_explicit:n #1

```

```

35119 {
35120 \exp_after:wN \__text_token_to_explicit_auxi:w
35121 \int_value:w
35122 \if_catcode:w \c_group_begin_token #1 1 \else:
35123 \if_catcode:w \c_group_end_token #1 2 \else:
35124 \if_catcode:w \c_math_toggle_token #1 3 \else:
35125 \if_catcode:w ## #1 6 \else:
35126 \if_catcode:w ^ #1 7 \else:
35127 \if_catcode:w \c_math_subscript_token #1 8 \else:
35128 \if_catcode:w \c_space_token #1 10 \else:
35129 \if_catcode:w A #1 11 \else:
35130 \if_catcode:w + #1 12 \else:
35131 4 \fi: \fi: \fi: \fi: \fi: \fi: \fi: \fi: \fi:
35132 \exp_after:wN \__text_sep:
35133 \token_to_meaning:N #1 \s__text_stop
35134 }
35135 \cs_new:Npn \__text_token_to_explicit_auxi:w #1 \__text_sep: #2 \s__text_stop
35136 {
35137 \char_generate:nn
35138 {
35139 \if_int_compare:w #1 < 9 \exp_stop_f:
35140 \exp_after:wN \__text_token_to_explicit_auxii:w
35141 \else:
35142 \exp_after:wN \__text_token_to_explicit_auxiii:w
35143 \fi:
35144 #2
35145 }
35146 {#1}
35147 }
35148 \exp_last_unbraced:NNNN \cs_new:Npn \__text_token_to_explicit_auxii:w
35149 #1 { \tl_to_str:n { character ~ } } { ` }
35150 \cs_new:Npn \__text_token_to_explicit_auxiii:w #1 ~ #2 ~ { ` }
35151 \cs_new:Npn \__text_char_catcode:N #1
35152 {
35153 \if_catcode:w \exp_not:N #1 \c_math_toggle_token
35154 3
35155 \else:
35156 \if_catcode:w \exp_not:N #1 \c_alignment_token
35157 4
35158 \else:
35159 \if_catcode:w \exp_not:N #1 \c_math_superscript_token
35160 7
35161 \else:
35162 \if_catcode:w \exp_not:N #1 \c_math_subscript_token
35163 8
35164 \else:

```

```

35165         \if_catcode:w \exp_not:N #1 \c_space_token 35165
35166         10 35166
35167     \else: 35167
35168         \if_catcode:w \exp_not:N #1 \c_catcode_letter_token 35168
35169         11 35169
35170     \else: 35170
35171         \if_catcode:w \exp_not:N #1 \c_catcode_other_token 35171
35172         12 35172
35173     \else: 35173
35174         13 35174
35175     \fi: 35175
35176     \fi: 35176
35177     \fi: 35177
35178     \fi: 35178
35179     \fi: 35179
35180     \fi: 35180
35181     \fi: 35181
35182 } 35182
35183 \prg_new_conditional:Npnn \__text_if_expandable:N #1 { T , F , TF } 35183
35184 { 35184
35185     \token_if_expandable:NTF #1 35185
35186     { 35186
35187         \bool_lazy_any:nTF 35187
35188         { 35188
35189             { \token_if_protected_macro_p:N #1 } 35189
35190             { \token_if_protected_long_macro_p:N #1 } 35190
35191             { \token_if_eq_meaning_p:NN \q__text_recursion_tail #1 } 35191
35192         } 35192
35193         { \prg_return_false: } 35193
35194         { \prg_return_true: } 35194
35195     } 35195
35196     { \prg_return_false: } 35196
35197 } 35197
35198 \sys_if_engine_opentype:TF 35198
35199 { 35199
35200     \cs_new:Npn \__text_codepoint_process:nN #1#2 { #1 {#2} } 35200
35201 } 35201
35202 { 35202
35203     \cs_new:Npe \__text_codepoint_process:nN #1#2 35203
35204     { 35204
35205         \exp_not:N \int_compare:nNnTF {`#2} > { "80 } 35205
35206         { 35206
35207             \sys_if_engine_pdftex:TF 35207
35208             { \exp_not:N \__text_codepoint_process_aux:nN } 35208
35209             { 35209
35210                 \exp_not:N \int_compare:nNnTF {`#2} > { "FF } 35210

```

```
35211         { \exp_not:N \use:n } 35211
35212         { \exp_not:N \__text_codepoint_process_aux:nN } 35212
35213     } 35213
35214 } 35214
35215 { \exp_not:N \use:n } 35215
35216 {#1} #2 35216
35217 } 35217
35218 \cs_new:Npn \__text_codepoint_process_aux:nN #1#2 35218
35219 { 35219
35220     \int_compare:nNnTF { `#2 } < { "E0 } 35220
35221     { \__text_codepoint_process:nNN } 35221
35222     { 35222
35223         \int_compare:nNnTF { `#2 } < { "F0 } 35223
35224         { \__text_codepoint_process:nNNN } 35224
35225         { \__text_codepoint_process:nNNNN } 35225
35226     } 35226
35227     {#1} #2 35227
35228 } 35228
35229 \cs_new:Npn \__text_codepoint_process:nNN #1#2#3 35229
35230 { #1 {#2#3} } 35230
35231 \cs_new:Npn \__text_codepoint_process:nNNN #1#2#3#4 35231
35232 { #1 {#2#3#4} } 35232
35233 \cs_new:Npn \__text_codepoint_process:nNNNN #1#2#3#4#5 35233
35234 { #1 {#2#3#4#5} } 35234
35235 } 35235
35236 \sys_if_engine_opentype:TF 35236
35237 { 35237
35238     \prg_new_conditional:Npnn 35238
35239     \__text_codepoint_compare:nNn #1#2#3 { TF , p } 35239
35240     { 35240
35241         \int_compare:nNnTF { `#1 } #2 {#3} 35241
35242         \prg_return_true: \prg_return_false: 35242
35243     } 35243
35244     \cs_new:Npn \__text_codepoint_from_chars:Nw #1 {`#1} 35244
35245 } 35245
35246 { 35246
35247     \prg_new_conditional:Npnn 35247
35248     \__text_codepoint_compare:nNn #1#2#3 { TF , p } 35248
35249     { 35249
35250         \int_compare:nNnTF { \__text_codepoint_from_chars:Nw #1 } 35250
35251         #2 {#3} 35251
35252         \prg_return_true: \prg_return_false: 35252
35253     } 35253
35254     \cs_new:Npe \__text_codepoint_from_chars:Nw #1 35254
35255     { 35255
35256         \exp_not:N \if_int_compare:w `#1 > "80 \exp_not:N \exp_stop_f: 35256
```

```

35257 \sys_if_engine_pdfTeX:TF
35258 {
35259     \exp_not:N \exp_after:wN
35260     \exp_not:N \__text_codepoint_from_chars_aux:Nw
35261 }
35262 {
35263     \exp_not:N \if_int_compare:w `#1 > "FF \exp_not:N \exp_stop_f:
35264     \exp_not:N \exp_after:wN \exp_not:N \exp_after:wN
35265     \exp_not:N \exp_after:wN
35266     \exp_not:N \__text_codepoint_from_chars:N
35267 \exp_not:N \else:
35268     \exp_not:N \exp_after:wN \exp_not:N \exp_after:wN
35269     \exp_not:N \exp_after:wN
35270     \exp_not:N \__text_codepoint_from_chars_aux:Nw
35271 \exp_not:N \fi:
35272 }
35273 \exp_not:N \else:
35274     \exp_not:N \exp_after:wN \exp_not:N \__text_codepoint_from_chars:N
35275 \exp_not:N \fi:
35276 #1
35277 }
35278 \cs_new:Npn \__text_codepoint_from_chars_aux:Nw #1
35279 {
35280     \if_int_compare:w `#1 < "E0 \exp_stop_f:
35281     \exp_after:wN \__text_codepoint_from_chars:NN
35282 \else:
35283     \if_int_compare:w `#1 < "F0 \exp_stop_f:
35284     \exp_after:wN \exp_after:wN \exp_after:wN
35285     \__text_codepoint_from_chars:NNN
35286 \else:
35287     \exp_after:wN \exp_after:wN \exp_after:wN
35288     \__text_codepoint_from_chars:NNNN
35289 \fi:
35290 \fi:
35291 #1
35292 }
35293 \cs_new:Npn \__text_codepoint_from_chars:N #1 {`#1}
35294 \cs_new:Npn \__text_codepoint_from_chars:NN #1#2
35295 { (`#1 - "C0) * "40 + `#2 - "80 }
35296 \cs_new:Npn \__text_codepoint_from_chars:NNN #1#2#3
35297 { (`#1 - "E0) * "1000 + (`#2 - "80) * "40 + `#3 - "80 }
35298 \cs_new:Npn \__text_codepoint_from_chars:NNNN #1#2#3#4
35299 {
35300     (`#1 - "F0) * "40000
35301 + (`#2 - "80) * "1000
35302 + (`#3 - "80) * "40

```

```

35303 + `#4 - "80
35304 }
35305 }
35306 \tl_new:N \l_text_accents_tl
35307 \tl_new:N \l_text_letterlike_tl
35308 \tl_new:N \l_text_case_exclude_arg_tl
35309 \tl_set:Nc \l_text_case_exclude_arg_tl
35310 {
35311   \exp_not:n { \begin \cite \end \label \ref }
35312   \exp_not:c { cite ~ }
35313   \exp_not:n { \babelshorthand }
35314 }
35315 \tl_new:N \l_text_math_arg_tl
35316 \tl_set:Nn \l_text_math_arg_tl { \ensuremath }
35317 \tl_new:N \l_text_math_delims_tl
35318 \tl_set:Nn \l_text_math_delims_tl { $ $ \ ( \ ) }
35319 \tl_new:N \l_text_expand_exclude_tl
35320 \tl_set:Nn \l_text_expand_exclude_tl
35321 { \begin \cite \end \label \ref }
35322 \bool_lazy_and:nnT
35323 { \str_if_eq_p:Vn \fmtname { LaTeX2e } }
35324 { \tl_if_exist_p:N \@expl@finalise@setup@@ }
35325 {
35326   \tl_gput_right:Nn \@expl@finalise@setup@@
35327   {
35328     \tl_gput_right:Nn \@kernel@after@begindocument
35329     {
35330       \group_begin:
35331       \cs_set_protected:Npn \__text_tmp:w #1
35332       {
35333         \tl_clear:N \l_text_expand_exclude_tl
35334         \tl_map_inline:nn {#1}
35335         {
35336           \bool_lazy_any:nF
35337           {
35338             { \token_if_protected_macro_p:N ##1 }
35339             { \token_if_protected_long_macro_p:N ##1 }
35340             {
35341               \str_if_eq_p:ee
35342               { \cs_replacement_spec:N ##1 }
35343               { \exp_not:n { \protect ##1 } \c_space_tl }
35344             }
35345           }
35346           { \tl_put_right:Nn \l_text_expand_exclude_tl {##1} }
35347         }
35348       }

```



```
35349 \exp_args:NV \__text_tmp:w \l_text_expand_exclude_tl 35349
35350 \exp_args:NNNV \group_end: 35350
35351 \tl_set:Nn \l_text_expand_exclude_tl \l_text_expand_exclude_tl 35351
35352 } 35352
35353 } 35353
35354 } 35354
35355 \tl_new:N \l__text_math_mode_tl 35355
35356 \tex_global:D \tex_chardef:D \c__text_chardef_space_token = `\_ % 35356
35357 \tex_global:D \tex_mathchardef:D \c__text_mathchardef_space_token = `\_ % 35357
35358 \tex_global:D \tex_chardef:D \c__text_chardef_group_begin_token = `{ % `} 35358
35359 \tex_global:D \tex_mathchardef:D \c__text_mathchardef_group_begin_token = `{ % `} `{\ 35359
35360 \tex_global:D \tex_chardef:D \c__text_chardef_group_end_token = `}% `{\ 35360
35361 \tex_global:D \tex_mathchardef:D \c__text_mathchardef_group_end_token = `}% 35361
35362 \cs_new:Npn \text_expand:n #1 35362
35363 { 35363
35364 \__kernel_exp_not:w \exp_after:wN 35364
35365 { 35365
35366 \exp:w 35366
35367 \__text_expand:n {#1} 35367
35368 } 35368
35369 } 35369
35370 \cs_new:Npn \__text_expand:n #1 35370
35371 { 35371
35372 \group_align_safe_begin: 35372
35373 \__text_expand_loop:w #1 35373
35374 \s__text_recursion_tail \s__text_recursion_stop 35374
35375 \__text_expand_result:n { } 35375
35376 } 35376
35377 \cs_new:Npn \__text_expand_store:n #1 35377
35378 { \__text_expand_store:nw {#1} } 35378
35379 \cs_generate_variant:Nn \__text_expand_store:n { o } 35379
35380 \cs_new:Npn \__text_expand_store:nw #1#2 \__text_expand_result:n #3 35380
35381 { #2 \__text_expand_result:n { #3 #1 } } 35381
35382 \cs_new:Npn \__text_expand_end:w #1 \__text_expand_result:n #2 35382
35383 { 35383
35384 \group_align_safe_end: 35384
35385 \exp_end: 35385
35386 #2 35386
35387 } 35387
35388 \cs_new:Npn \__text_expand_loop:w #1 \s__text_recursion_stop 35388
35389 { 35389
35390 \tl_if_head_is_N_type:nTF {#1} 35390
35391 { \__text_expand_N_type:N } 35391
35392 { 35392
35393 \tl_if_head_is_group:nTF {#1} 35393
35394 { \__text_expand_group:n } 35394
```

```

35395         { \_text_expand_space:w }
35396     }
35397     #1 \s__text_recursion_stop
35398 }
35399 \cs_new:Npn \_text_expand_group:n #1
35400 {
35401     \_text_expand_store:o
35402     {
35403         \exp_after:wN
35404         {
35405             \exp:w
35406             \_text_expand:n {#1}
35407         }
35408     }
35409     \_text_expand_loop:w
35410 }
35411 \exp_last_unbraced:NNo \cs_new:Npn \_text_expand_space:w \c_space_tl
35412 {
35413     \_text_expand_store:n { ~ }
35414     \_text_expand_loop:w
35415 }
35416 \cs_new:Npn \_text_expand_N_type:N #1
35417 {
35418     \_text_if_s_recursion_tail_stop_do:Nn #1
35419     { \_text_expand_end:w }
35420     \exp_after:wN \_text_expand_math_search:NNN
35421     \exp_after:wN #1 \l_text_math_delims_tl
35422     \q_text_recursion_tail \q_text_recursion_tail
35423     \q__text_recursion_stop
35424 }
35425 \cs_new:Npn \_text_expand_math_search:NNN #1#2#3
35426 {
35427     \_text_if_q_recursion_tail_stop_do:Nn #2
35428     { \_text_expand_explicit:N #1 }
35429     \token_if_eq_meaning:NNTF #1 #2
35430     {
35431         \_text_use_i_delimit_by_q_recursion_stop:nw
35432         {
35433             \_text_expand_store:n {#1}
35434             \_text_expand_math_loop:Nw #3
35435         }
35436     }
35437     { \_text_expand_math_search:NNN #1 }
35438 }
35439 \cs_new:Npn \_text_expand_math_loop:Nw #1#2 \s__text_recursion_stop
35440 {

```

```

35441 \tl_if_head_is_N_type:nTF {#2}
35442 { \_text_expand_math_N_type:NN }
35443 {
35444 \tl_if_head_is_group:nTF {#2}
35445 { \_text_expand_math_group:Nn }
35446 { \_text_expand_math_space:Nw }
35447 }
35448 #1#2 \s__text_recursion_stop
35449 }
35450 \cs_new:Npn \_text_expand_math_N_type:NN #1#2
35451 {
35452 \_text_if_s_recursion_tail_stop_do:Nn #2
35453 { \_text_expand_end:w }
35454 \token_if_eq_meaning:NNTF #2 \exp_not:N
35455 { \_text_expand_store:n {#2} }
35456 \token_if_eq_meaning:NNTF #2 #1
35457 { \_text_expand_loop:w }
35458 { \_text_expand_math_loop:Nw #1 }
35459 }
35460 \cs_new:Npn \_text_expand_math_group:Nn #1#2
35461 {
35462 \_text_expand_store:n { {#2} }
35463 \_text_expand_math_loop:Nw #1
35464 }
35465 \exp_after:wN \cs_new:Npn \exp_after:wN \_text_expand_math_space:Nw
35466 \exp_after:wN # \exp_after:wN 1 \c_space_tl
35467 {
35468 \_text_expand_store:n { ~ }
35469 \_text_expand_math_loop:Nw #1
35470 }
35471 \cs_new:Npn \_text_expand_explicit:N #1
35472 {
35473 \token_if_cs:NNTF #1
35474 { \_text_expand_exclude:N #1 }
35475 {
35476 \bool_lazy_and:nnTF
35477 { \token_if_active_p:N #1 }
35478 {
35479 ! \bool_lazy_any_p:n
35480 {
35481 { \token_if_protected_macro_p:N #1 }
35482 { \token_if_protected_long_macro_p:N #1 }
35483 { \tl_if_head_eq_meaning_p:oN {#1} \UTFviii@two@octets }
35484 { \tl_if_head_eq_meaning_p:oN {#1} \UTFviii@three@octets }
35485 { \tl_if_head_eq_meaning_p:oN {#1} \UTFviii@four@octets }
35486 { \tl_if_head_eq_meaning_p:oN {#1} \active@prefix }

```

```
35487         }
35488     }
35489     { \exp_after:wN \__text_expand_loop:w #1 }
35490     {
35491         \__text_expand_store:n {#1}
35492         \__text_expand_loop:w
35493     }
35494 }
35495 }
35496 \cs_new:Npn \__text_expand_exclude:N #1
35497 {
35498     \cs_if_eq:NNTF #1 \text_case_switch:nnnn
35499     { \__text_expand_exclude_switch:Nnnnn #1 }
35500     {
35501         \exp_args:Ne \__text_expand_exclude:nN
35502         {
35503             \exp_not:V \l_text_math_arg_tl
35504             \exp_not:V \l_text_expand_exclude_tl
35505             \exp_not:V \l_text_case_exclude_arg_tl
35506         }
35507         #1
35508     }
35509 }
35510 \cs_new:Npn \__text_expand_exclude_switch:Nnnnn #1#2#3#4#5
35511 {
35512     \__text_expand_store:n { #1 {#2} {#3} {#4} {#5} }
35513     \__text_expand_loop:w
35514 }
35515 \cs_new:Npn \__text_expand_exclude:nN #1#2
35516 {
35517     \__text_expand_exclude:NN #2 #1
35518     \q__text_recursion_tail \q__text_recursion_stop
35519 }
35520 \cs_new:Npn \__text_expand_exclude:NN #1#2
35521 {
35522     \__text_if_q_recursion_tail_stop_do:Nn #2
35523     { \__text_expand_accent:N #1 }
35524     \str_if_eq:nnTF {#1} {#2}
35525     {
35526         \__text_use_i_delimit_by_q_recursion_stop:nw
35527         { \__text_expand_exclude:Nw #1 }
35528     }
35529     { \__text_expand_exclude:NN #1 }
35530 }
35531 \cs_new:Npn \__text_expand_exclude:Nw #1#2#
35532 { \__text_expand_exclude:Nnn #1 {#2} }
```

```
35533 \cs_new:Npn \__text_expand_exclude:Nnn #1#2#3
35534 {
35535     \__text_expand_store:n { #1#2 {#3} }
35536     \__text_expand_loop:w
35537 }
35538 \cs_new:Npn \__text_expand_accent:N #1
35539 {
35540     \exp_after:wN \__text_expand_accent:NN \exp_after:wN
35541     #1 \l_text_accents_tl
35542     \q__text_recursion_tail \q__text_recursion_stop
35543 }
35544 \cs_new:Npn \__text_expand_accent:NN #1#2
35545 {
35546     \__text_if_q_recursion_tail_stop_do:Nn #2
35547     { \__text_expand_letterlike:N #1 }
35548     \cs_if_eq:NNTF #2 #1
35549     {
35550         \__text_use_i_delimit_by_q_recursion_stop:nw
35551         {
35552             \__text_expand_store:n {#1}
35553             \__text_expand_loop:w
35554         }
35555     }
35556     { \__text_expand_accent:NN #1 }
35557 }
35558 \cs_new:Npn \__text_expand_letterlike:N #1
35559 {
35560     \exp_after:wN \__text_expand_letterlike:NN \exp_after:wN
35561     #1 \l_text_letterlike_tl
35562     \q__text_recursion_tail \q__text_recursion_stop
35563 }
35564 \cs_new:Npn \__text_expand_letterlike:NN #1#2
35565 {
35566     \__text_if_q_recursion_tail_stop_do:Nn #2
35567     { \__text_expand_cs:N #1 }
35568     \cs_if_eq:NNTF #2 #1
35569     {
35570         \__text_use_i_delimit_by_q_recursion_stop:nw
35571         {
35572             \__text_expand_store:n {#1}
35573             \__text_expand_loop:w
35574         }
35575     }
35576     { \__text_expand_letterlike:NN #1 }
35577 }
35578 \cs_new:Npe \__text_expand_cs:N #1
```

```
35579 { 35579
35580 \exp_not:N \str_if_eq:nnTF {#1} { \exp_not:N \protect } 35580
35581 { \exp_not:N \__text_expand_protect:w } 35581
35582 { 35582
35583 \bool_lazy_and:nnTF 35583
35584 { \cs_if_exist_p:N \fmtname } 35584
35585 { \str_if_eq_p:Vn \fmtname { LaTeX2e } } 35585
35586 { \exp_not:N \__text_expand_testopt:N #1 } 35586
35587 { \exp_not:N \__text_expand_replace:N #1 } 35587
35588 } 35588
35589 } 35589
35590 \cs_new:Npn \__text_expand_protect:w #1 \s__text_recursion_stop 35590
35591 { 35591
35592 \tl_if_head_is_N_type:nTF {#1} 35592
35593 { \__text_expand_protect:N } 35593
35594 { 35594
35595 \__text_expand_store:n { \protect } 35595
35596 \__text_expand_loop:w 35596
35597 } 35597
35598 #1 \s__text_recursion_stop 35598
35599 } 35599
35600 \cs_new:Npn \__text_expand_protect:N #1 35600
35601 { 35601
35602 \__text_if_s_recursion_tail_stop_do:Nn #1 35602
35603 { 35603
35604 \__text_expand_store:n { \protect } 35604
35605 \__text_expand_end:w 35605
35606 } 35606
35607 \exp_args:Ne \__text_expand_protect:nN 35607
35608 { \cs_to_str:N #1 } #1 35608
35609 } 35609
35610 \cs_new:Npn \__text_expand_protect:nN #1#2 35610
35611 { \__text_expand_protect:Nw #2 #1 \q__text_nil #1 ~ \q__text_nil \q__text_nil ✓ 35611
35612 \s__text_stop } 35611
35612 \cs_new:Npn \__text_expand_protect:Nw #1 #2 ~ \q__text_nil #3 \q__text_nil #4 ✓ 35612
35613 \s__text_stop 35612
35614 { 35613
35615 \__text_quark_if_nil:nTF {#4} 35614
35616 { 35615
35617 \cs_if_exist:cTF {#2} 35616
35618 { \exp_args:Ne \__text_expand_store:n { \exp_not:c {#2} } } 35617
35619 { \__text_expand_store:n { \protect #1 } } 35618
35620 } 35619
35621 { \__text_expand_store:n { \protect #1 } } 35620
35622 \__text_expand_loop:w 35621
35623 } 35622
```

```
35623 \cs_new:Npn \__text_expand_testopt:N #1 35623
35624 { 35624
35625     \token_if_eq_meaning:NNTF #1 \@protected@testopt 35625
35626     { \__text_expand_testopt:NNn } 35626
35627     { \__text_expand_encoding:N #1 } 35627
35628 } 35628
35629 \cs_new:Npn \__text_expand_testopt:NNn #1#2#3 35629
35630 { 35630
35631     \__text_expand_store:n {#1} 35631
35632     \__text_expand_loop:w 35632
35633 } 35633
35634 \cs_new:Npn \__text_expand_encoding:N #1 35634
35635 { 35635
35636     \bool_lazy_or:nnTF 35636
35637     { \cs_if_eq_p:NN #1 \@current@cmd } 35637
35638     { \cs_if_eq_p:NN #1 \@changed@cmd } 35638
35639     { \exp_after:wN \__text_expand_loop:w \__text_expand_encoding_escape:NN } 35639
35640     { \__text_expand_replace:N #1 } 35640
35641 } 35641
35642 \cs_new:Npn \__text_expand_encoding_escape:NN #1#2 { \exp_not:n {#1} } 35642
35643 \cs_new:Npn \__text_expand_replace:N #1 35643
35644 { 35644
35645     \bool_lazy_and:nnTF 35645
35646     { \cs_if_exist_p:c { l__text_expand_ \token_to_str:N #1 _tl } } 35646
35647     { 35647
35648         \bool_lazy_or_p:nn 35648
35649         { \token_if_cs_p:N #1 } 35649
35650         { \token_if_active_p:N #1 } 35650
35651     } 35651
35652     { 35652
35653         \exp_args:Nv \__text_expand_replace:n 35653
35654         { l__text_expand_ \token_to_str:N #1 _tl } 35654
35655     } 35655
35656     { \__text_expand_cs_expand:N #1 } 35656
35657 } 35657
35658 \cs_new:Npn \__text_expand_replace:n #1 { \__text_expand_loop:w #1 } 35658
35659 \cs_new:Npn \__text_expand_cs_expand:N #1 35659
35660 { 35660
35661     \__text_if_expandable:NNTF #1 35661
35662     { 35662
35663         \token_if_eq_meaning:NNTF #1 \exp_not:n 35663
35664         { \__text_expand_unexpanded:w } 35664
35665         { \exp_after:wN \__text_expand_loop:w #1 } 35665
35666     } 35666
35667     { 35667
35668         \__text_expand_store:n {#1} 35668
```



```

35669 \__text_expand_loop:w
35670 }
35671 }
35672 \cs_new:Npn \__text_expand_unexpanded:w
35673 {
35674 \exp_after:wN \__text_expand_unexpanded_test:w
35675 \exp:w \exp_end_continue_f:w
35676 }
35677 \cs_new:Npn \__text_expand_unexpanded_test:w #1 \s__text_recursion_stop
35678 {
35679 \tl_if_head_is_group:nTF {#1}
35680 { \__text_expand_unexpanded:n }
35681 {
35682 \__text_expand_unexpanded:w
35683 \tl_if_head_is_N_type:nT {#1} { \__text_expand_unexpanded:N }
35684 }
35685 #1 \s__text_recursion_stop
35686 }
35687 \cs_new:Npn \__text_expand_unexpanded:N #1
35688 {
35689 \exp_after:wN \if_meaning:w \exp_not:N #1 #1
35690 \else:
35691 \exp_after:wN #1
35692 \fi:
35693 }
35694 \cs_new:Npn \__text_expand_unexpanded:n #1
35695 {
35696 \__text_expand_store:n {#1}
35697 \__text_expand_loop:w
35698 }
35699 \cs_new_protected:Npn \text_declare_expand_equivalent:Nn #1#2
35700 {
35701 \tl_clear_new:c { l__text_expand_ \token_to_str:N #1 _tl }
35702 \tl_set:cn { l__text_expand_ \token_to_str:N #1 _tl } {#2}
35703 }
35704 \cs_generate_variant:Nn \text_declare_expand_equivalent:Nn { c }
35705 \tl_map_inline:nn
35706 { \` \' \^ \~ \= \u \. \" \r \H \v \d \c \k \b \t }
35707 { \text_declare_expand_equivalent:Nn #1 { \exp_not:n {#1} } }
35708 \tl_map_inline:nn
35709 {
35710 \AA \aa
35711 \AE \ae
35712 \DH \dh
35713 \DJ \dj
35714 \IJ \ij

```

```

35715      \L \l
35716      \NG \ng
35717      \O \o
35718      \OE \oe
35719      \SS \ss
35720      \TH \th
35721  }
35722  { \text_declare_expand_equivalent:Nn #1 { \exp_not:n {#1} } }
35723 %% File: l3text-case.dtx
35724 \bool_new:N \l_text_titlecase_check_letter_bool
35725 \bool_set_true:N \l_text_titlecase_check_letter_bool
35726 \cs_new:Npn \text_lowercase:n #1
35727   { \__text_change_case:nnn { lower } { } {#1} }
35728 \cs_new:Npn \text_uppercase:n #1
35729   { \__text_change_case:nnn { upper } { } {#1} }
35730 \cs_new:Npn \text_titlecase_all:n #1
35731   { \__text_change_case:nnn { title } { } {#1} }
35732 \cs_new:Npn \text_titlecase_first:n #1
35733   { \__text_change_case:nnnn { title } { break } { } {#1} }
35734 \cs_new:Npn \text_lowercase:nn #1#2
35735   { \__text_change_case:nnn { lower } {#1} {#2} }
35736 \cs_new:Npn \text_uppercase:nn #1#2
35737   { \__text_change_case:nnn { upper } {#1} {#2} }
35738 \cs_new:Npn \text_titlecase_all:nn #1#2
35739   { \__text_change_case:nnn { title } {#1} {#2} }
35740 \cs_new:Npn \text_titlecase_first:nn #1#2
35741   { \__text_change_case:nnnn { title } { break } {#1} {#2} }
35742 \cs_new:Npn \__text_change_case:nnn #1#2#3
35743   { \__text_change_case:nnnn {#1} {#1} {#2} {#3} }
35744 \cs_new:Npn \__text_change_case:nnnn #1#2#3#4
35745   {
35746     \__kernel_exp_not:w \exp_after:wN
35747     {
35748       \exp:w
35749       \exp_args:Ne \__text_change_case_auxi:nnnn
35750         { \text_expand:n {#4} }
35751         {#1} {#2} {#3}
35752     }
35753   }
35754 \cs_new:Npn \__text_change_case_auxi:nnnn #1#2#3#4
35755   {
35756     \exp_args:No \__text_change_case_BCP:nnnn
35757     { \tl_to_str:n {#4} } {#1} {#2} {#3}
35758   }
35759 \cs_new:Npe \__text_change_case_BCP:nnnn #1#2#3#4
35760   {

```

```
35761 \exp_not:N \__text_change_case_BCP:nnnw 35761
35762 {#2} {#3} {#4} #1 \tl_to_str:n { -x- -x- } \exp_not:N \q__text_stop 35762
35763 } 35763
35764 \use:e 35764
35765 { 35765
35766 \cs_new:Npn \exp_not:N \__text_change_case_BCP:nnnw 35766
35767 #1#2#3#4 \tl_to_str:n { -x- } #5 \tl_to_str:n { -x- } #6 35767
35768 \exp_not:N \q__text_stop 35768
35769 } 35769
35770 { \__text_change_case_BCP:nnnnnw {#1} {#2} {#3} {#5} {#4} #4 - \q__text_stop } 35770
35771 \cs_new:Npn \__text_change_case_BCP:nnnnnw #1#2#3#4#5#6 - #7 \q__text_stop 35771
35772 { 35772
35773 \bool_lazy_or:nnTF 35773
35774 { \cs_if_exist_p:c { __text_change_case_ #2 _ #6 -x- #4 :nnnnn } } 35774
35775 { \tl_if_exist_p:c { l__text_ #2 case_special_ #6 -x- #4 _tl } } 35775
35776 { \__text_change_case_auxii:nnnn {#1} {#2} {#3} { #6 -x- #4 } } 35776
35777 { 35777
35778 \cs_if_exist:cTF { __text_change_case_ #2 _ #6 :nnnnn } 35778
35779 { \__text_change_case_auxii:nnnn {#1} {#2} {#3} {#6} } 35779
35780 { \__text_change_case_auxii:nnnn {#1} {#2} {#3} {#5} } 35780
35781 } 35781
35782 } 35782
35783 \cs_new:Npn \__text_change_case_auxii:nnnn #1#2#3#4 35783
35784 { 35784
35785 \group_align_safe_begin: 35785
35786 \cs_if_exist_use:c { __text_change_case_boundary_ #2 _ #4 :Nnnnw } 35786
35787 \__text_change_case_loop:nnnw {#2} {#3} {#4} #1 35787
35788 \q__text_recursion_tail \q__text_recursion_stop 35788
35789 \__text_change_case_result:n { } 35789
35790 } 35790
35791 \cs_new:Npn \__text_change_case_store:n #1 35791
35792 { \__text_change_case_store:nw {#1} } 35792
35793 \cs_generate_variant:Nn \__text_change_case_store:n { o , e , V , v } 35793
35794 \cs_new:Npn \__text_change_case_store:nw #1#2 \__text_change_case_result:n #3 35794
35795 { #2 \__text_change_case_result:n { #3 #1 } } 35795
35796 \cs_new:Npn \__text_change_case_end:w #1 \__text_change_case_result:n #2 35796
35797 { 35797
35798 \group_align_safe_end: 35798
35799 \exp_end: 35799
35800 #2 35800
35801 } 35801
35802 \cs_new:Npn \__text_change_case_loop:nnnw #1#2#3#4 \q__text_recursion_stop 35802
35803 { 35803
35804 \tl_if_head_is_N_type:nTF {#4} 35804
35805 { \__text_change_case_N_type:nnnN } 35805
35806 { 35806
```

```
35807         \tl_if_head_is_group:nTF {#4} 35807
35808         { \use:c { __text_change_case_group_ #1 :nnnn } } 35808
35809         { \__text_change_case_space:nnnw } 35809
35810     } 35810
35811     {#1} {#2} {#3} #4 \q__text_recursion_stop 35811
35812 } 35812
35813 \cs_new:Npn \__text_change_case_break:w 35813
35814 { \__text_change_case_break_aux:w \prg_do_nothing: } 35814
35815 \cs_new:Npn \__text_change_case_break_aux:w 35815
35816 #1 \q__text_recursion_tail \q__text_recursion_stop 35816
35817 { 35817
35818     \__text_change_case_store:o {#1} 35818
35819     \__text_change_case_end:w 35819
35820 } 35820
35821 \cs_new:Npn \__text_change_case_group_lower:nnnn #1#2#3#4 35821
35822 { 35822
35823     \__text_change_case_store:o 35823
35824     { 35824
35825         \exp_after:wN 35825
35826         { 35826
35827             \exp:w 35827
35828             \__text_change_case_auxii:nnnn {#4} {#1} {#2} {#3} 35828
35829         } 35829
35830     } 35830
35831     \__text_change_case_loop:nnnw {#1} {#2} {#3} 35831
35832 } 35832
35833 \cs_new_eq:NN \__text_change_case_group_upper:nnnn 35833
35834 \__text_change_case_group_lower:nnnn 35834
35835 \cs_new:Npn \__text_change_case_group_title:nnnn #1#2#3#4 35835
35836 { 35836
35837     \__text_change_case_store:o 35837
35838     { 35838
35839         \exp_after:wN 35839
35840         { 35840
35841             \exp:w 35841
35842             \__text_change_case_auxii:nnnn {#4} {#1} {#2} {#3} 35842
35843         } 35843
35844     } 35844
35845     \__text_change_case_skip:nnw {#2} {#3} 35845
35846 } 35846
35847 \use:e 35847
35848 { 35848
35849     \cs_new:Npn \exp_not:N \__text_change_case_space:nnnw #1#2#3 \c_space_tl 35849
35850 } 35850
35851 { 35851
35852     \__text_change_case_store:n { ~ } 35852
```

```
35853 \cs_if_exist_use:cF { __text_change_case_space_ #2 :nnn } 35853
35854 { 35854
35855 \cs_if_exist_use:c { __text_change_case_boundary_ #1 _ #3 :Nnnnw } 35855
35856 \__text_change_case_loop:nnnw 35856
35857 } 35857
35858 {#2} {#2} {#3} 35858
35859 } 35859
35860 \cs_new:Npn \__text_change_case_space_break:nnn #1#2#3 35860
35861 { \__text_change_case_break:w } 35861
35862 \cs_new:Npn \__text_change_case_N_type:nnnN #1#2#3#4 35862
35863 { 35863
35864 \__text_if_q_recursion_tail_stop_do:Nn #4 35864
35865 { \__text_change_case_end:w } 35865
35866 \__text_change_case_N_type_aux:nnnN {#1} {#2} {#3} #4 35866
35867 } 35867
35868 \cs_new:Npn \__text_change_case_N_type_aux:nnnN #1#2#3#4 35868
35869 { 35869
35870 \exp_args:NV \__text_change_case_N_type:nnnnN 35870
35871 \l_text_math_delims_tl {#1} {#2} {#3} #4 35871
35872 } 35872
35873 \cs_new:Npn \__text_change_case_N_type:nnnnN #1#2#3#4#5 35873
35874 { 35874
35875 \__text_change_case_math_search:nnnNNN {#2} {#3} {#4} #5 #1 35875
35876 \q__text_recursion_tail \q__text_recursion_tail 35876
35877 \q__text_recursion_stop 35877
35878 } 35878
35879 \cs_new:Npn \__text_change_case_math_search:nnnNNN #1#2#3#4#5#6 35879
35880 { 35880
35881 \__text_if_q_recursion_tail_stop_do:Nn #5 35881
35882 { \__text_change_case_cs_check:nnnN {#1} {#2} {#3} #4 } 35882
35883 \token_if_eq_meaning:NNTF #4 #5 35883
35884 { 35884
35885 \__text_use_i_delimit_by_q_recursion_stop:nw 35885
35886 { 35886
35887 \__text_change_case_store:n {#4} 35887
35888 \__text_change_case_math_loop:nnnNw {#1} {#2} {#3} #6 35888
35889 } 35889
35890 } 35890
35891 { \__text_change_case_math_search:nnnNNN {#1} {#2} {#3} #4 } 35891
35892 } 35892
35893 \cs_new:Npn \__text_change_case_math_loop:nnnNw #1#2#3#4#5 \q__text_recursion_stop 35893
35894 { 35894
35895 \tl_if_head_is_N_type:nTF {#5} 35895
35896 { \__text_change_case_math_N_type:nnnNN } 35896
35897 { 35897
35898 \tl_if_head_is_group:nTF {#5} 35898
```

```
35899         { \_text_change_case_math_group:nnnNn } 35899
35900         { \_text_change_case_math_space:nnnNw } 35900
35901     } 35901
35902     {#1} {#2} {#3} #4 #5 \q_text_recursion_stop 35902
35903 } 35903
35904 \cs_new:Npn \_text_change_case_math_N_type:nnnNN #1#2#3#4#5 35904
35905 { 35905
35906     \_text_if_q_recursion_tail_stop_do:Nn #5 35906
35907     { \_text_change_case_end:w } 35907
35908     \_text_change_case_store:n {#5} 35908
35909     \token_if_eq_meaning:NNTF #5 #4 35909
35910     { \_text_change_case_loop:nnnw {#1} {#2} {#3} } 35910
35911     { \_text_change_case_math_loop:nnnNw {#1} {#2} {#3} #4 } 35911
35912 } 35912
35913 \cs_new:Npn \_text_change_case_math_group:nnnNn #1#2#3#4#5 35913
35914 { 35914
35915     \_text_change_case_store:n { {#5} } 35915
35916     \_text_change_case_math_loop:nnnNw {#1} {#2} {#3} #4 35916
35917 } 35917
35918 \use:e 35918
35919 { 35919
35920     \cs_new:Npn \exp_not:N \_text_change_case_math_space:nnnNw #1#2#3#4 35920
35921     \c_space_tl 35921
35922 } 35922
35923 { 35923
35924     \_text_change_case_store:n { ~ } 35924
35925     \_text_change_case_math_loop:nnnNw {#1} {#2} {#3} #4 35925
35926 } 35926
35927 \cs_new:Npn \_text_change_case_cs_check:nnnN #1#2#3#4 35927
35928 { 35928
35929     \token_if_cs:NNTF #4 35929
35930     { \_text_change_case_exclude:nnnN {#1} {#2} {#3} } 35930
35931     { 35931
35932         \_text_codepoint_process:nN 35932
35933         { \use:c { \_text_change_case_custom_ #1 :nnnn } {#1} {#2} {#3} } 35933
35934     } 35934
35935     #4 35935
35936 } 35936
35937 \cs_new:Npn \_text_change_case_exclude:nnnN #1#2#3#4 35937
35938 { 35938
35939     \exp_args:Ne \_text_change_case_exclude:nnnnN 35939
35940     { 35940
35941         \exp_not:V \l_text_math_arg_tl 35941
35942         \exp_not:V \l_text_case_exclude_arg_tl 35942
35943     } 35943
35944     {#1} {#2} {#3} #4 35944
```

```
35945 } 35945
35946 \cs_new:Npn \__text_change_case_exclude:nnnnN #1#2#3#4#5 35946
35947 { 35947
35948 \__text_change_case_exclude:nnnNN {#2} {#3} {#4} #5 #1 35948
35949 \q__text_recursion_tail \q__text_recursion_stop 35949
35950 } 35950
35951 \cs_new:Npn \__text_change_case_exclude:nnnNN #1#2#3#4#5 35951
35952 { 35952
35953 \__text_if_q_recursion_tail_stop_do:Nn #5 35953
35954 { \__text_change_case_replace:nnnN {#1} {#2} {#3} #4 } 35954
35955 \str_if_eq:nnTF {#4} {#5} 35955
35956 { 35956
35957 \__text_use_i_delimit_by_q_recursion_stop:nw 35957
35958 { \__text_change_case_exclude:nnnNw {#1} {#2} {#3} #4 } 35958
35959 } 35959
35960 { \__text_change_case_exclude:nnnNN {#1} {#2} {#3} #4 } 35960
35961 } 35961
35962 \cs_new:Npn \__text_change_case_exclude:nnnNw #1#2#3#4#5# 35962
35963 { \__text_change_case_exclude:nnnNnn {#1} {#2} {#3} {#4} {#5} } 35963
35964 \cs_new:Npn \__text_change_case_exclude:nnnNnn #1#2#3#4#5#6 35964
35965 { 35965
35966 \tl_if_blank:nTF {#5} 35966
35967 { \__text_change_case_store:n { #4 {#6} } } 35967
35968 { 35968
35969 \__text_change_case_store:o 35969
35970 { 35970
35971 \exp_after:wN #4 35971
35972 \exp:w \__text_change_case_auxii:nnnn {#5} {#1} {#2} {#3} 35972
35973 {#6} 35973
35974 } 35974
35975 } 35975
35976 \__text_change_case_loop:nnnw {#1} {#2} {#3} 35976
35977 } 35977
35978 \cs_new:Npn \__text_change_case_replace:nnnN #1#2#3#4 35978
35979 { 35979
35980 \cs_if_exist:cTF { l__text_case_ \token_to_str:N #4 _tl } 35980
35981 { 35981
35982 \__text_change_case_replace:vn nn 35982
35983 { l__text_case_ \token_to_str:N #4 _tl } {#1} {#2} {#3} 35983
35984 } 35984
35985 { \__text_change_case_switch:nnnN {#1} {#2} {#3} #4 } 35985
35986 } 35986
35987 \cs_new:Npn \__text_change_case_replace:nnnn #1#2#3#4 35987
35988 { \__text_change_case_loop:nnnw {#2} {#3} {#4} #1 } 35988
35989 \cs_generate_variant:Nn \__text_change_case_replace:nnnn { v } 35989
35990 \cs_new:Npn \__text_change_case_switch:nnnN #1#2#3#4 35990
```



```
35991 { 35991
35992 \cs_if_eq:NNTF #4 \text_case_switch:nnnn 35992
35993 { \use:c { __text_change_case_switch_ #1 :nnnNnnnn } } 35993
35994 { \use:c { __text_change_case_letterlike_ #1 :nnnN } } 35994
35995 {#1} {#2} {#3} #4 35995
35996 } 35996
35997 \cs_new:Npn \__text_change_case_switch_lower:nnnNnnnn #1#2#3#4#5#6#7#8 35997
35998 { 35998
35999 \__text_change_case_store:n {#7} 35999
36000 \__text_change_case_loop:nnnw {#1} {#2} {#3} 36000
36001 } 36001
36002 \cs_new:Npn \__text_change_case_switch_upper:nnnNnnnn #1#2#3#4#5#6#7#8 36002
36003 { 36003
36004 \__text_change_case_store:n {#6} 36004
36005 \__text_change_case_loop:nnnw {#1} {#2} {#3} 36005
36006 } 36006
36007 \cs_new:Npn \__text_change_case_switch_title:nnnNnnnn #1#2#3#4#5#6#7#8 36007
36008 { 36008
36009 \__text_change_case_store:n {#8} 36009
36010 \__text_change_case_skip:nnw {#2} {#3} 36010
36011 } 36011
36012 \cs_new:Npn \__text_change_case_skip:nnw #1#2#3 \q__text_recursion_stop 36012
36013 { 36013
36014 \tl_if_head_is_N_type:nTF {#3} 36014
36015 { \__text_change_case_skip_N_type:nnN } 36015
36016 { 36016
36017 \tl_if_head_is_group:nTF {#3} 36017
36018 { \__text_change_case_skip_group:nnn } 36018
36019 { \__text_change_case_skip_space:nnw } 36019
36020 } 36020
36021 {#1} {#2} #3 \q__text_recursion_stop 36021
36022 } 36022
36023 \cs_new:Npn \__text_change_case_skip_N_type:nnN #1#2#3 36023
36024 { 36024
36025 \__text_if_q_recursion_tail_stop_do:Nn #3 36025
36026 { \__text_change_case_end:w } 36026
36027 \__text_change_case_store:n {#3} 36027
36028 \__text_change_case_skip:nnw {#1} {#2} 36028
36029 } 36029
36030 \cs_new:Npn \__text_change_case_skip_group:nnn #1#2#3 36030
36031 { 36031
36032 \__text_change_case_store:n { {#3} } 36032
36033 \__text_change_case_skip:nnw {#1} {#2} 36033
36034 } 36034
36035 \cs_new:Npn \__text_change_case_skip_space:nnw #1#2 36035
36036 { \__text_change_case_space:nnnw {#1} {#1} {#2} } 36036
```

```
36037 \cs_new:Npn \__text_change_case_letterlike_lower:nnnN #1#2#3#4 36037
36038 { \__text_change_case_letterlike:nnnnnN {#1} {#1} {#1} {#2} {#3} #4 } 36038
36039 \cs_new_eq:NN \__text_change_case_letterlike_upper:nnnN 36039
36040 \__text_change_case_letterlike_lower:nnnN 36040
36041 \cs_new:Npn \__text_change_case_letterlike_title:nnnN #1#2#3#4 36041
36042 { \__text_change_case_letterlike:nnnnnN { upper } { end } {#1} {#2} {#3} #4 } 36042
36043 \cs_new:Npn \__text_change_case_letterlike:nnnnnN #1#2#3#4#5#6 36043
36044 { 36044
36045 \cs_if_exist:cTF { c__text_ #1 case_ \token_to_str:N #6 _tl } 36045
36046 { 36046
36047 \__text_change_case_store:v 36047
36048 { c__text_ #1 case_ \token_to_str:N #6 _tl } 36048
36049 \use:c { __text_change_case_next_ #2 :nnn } {#2} {#4} {#5} 36049
36050 } 36050
36051 { 36051
36052 \__text_change_case_store:n {#6} 36052
36053 \cs_if_exist:cTF 36053
36054 { 36054
36055 c__text_ 36055
36056 \str_if_eq:nnTF {#1} { lower } { upper } { lower } 36056
36057 case_ \token_to_str:N #6 _tl 36057
36058 } 36058
36059 { \use:c { __text_change_case_next_ #2 :nnn } {#2} {#4} {#5} } 36059
36060 { \__text_change_case_loop:nnnw {#3} {#4} {#5} } 36060
36061 } 36061
36062 } 36062
36063 \cs_new:Npn \__text_change_case_custom_lower:nnnn #1#2#3#4 36063
36064 { 36064
36065 \__text_change_case_custom:nnnnnn {#1} {#1} {#2} {#3} {#4} 36065
36066 { \use:c { __text_change_case_codepoint_ #1 :nnnn } {#1} {#2} {#3} {#4} } 36066
36067 } 36067
36068 \cs_new_eq:NN \__text_change_case_custom_upper:nnnn 36068
36069 \__text_change_case_custom_lower:nnnn 36069
36070 \cs_new:Npn \__text_change_case_custom_title:nnnn #1#2#3#4 36070
36071 { 36071
36072 \__text_change_case_custom:nnnnnn { title } {#1} {#2} {#3} {#4} 36072
36073 { 36073
36074 \__text_change_case_custom:nnnnnn { upper } {#1} {#2} {#3} {#4} 36074
36075 { \use:c { __text_change_case_codepoint_ #1 :nnnn } {#1} {#2} {#3} {#4} } 36075
36076 } 36076
36077 } 36077
36078 \cs_new:Npn \__text_change_case_custom:nnnnnn #1#2#3#4#5#6 36078
36079 { 36079
36080 \tl_if_exist:cTF { l__text_ #1 case _ \tl_to_str:n {#5} _ #4 _tl } 36080
36081 { 36081
36082 \__text_change_case_replace:vnnn 36082
```

```
36083 { l__text_ #1 case _ \tl_to_str:n {#5} _ #4 _tl } {#2} {#3} {#4} 36083
36084 } 36084
36085 { 36085
36086 \tl_if_exist:cTF { l__text_ #1 case _ \tl_to_str:n {#5} _tl } 36086
36087 { 36087
36088 \__text_change_case_replace:vnbn 36088
36089 { l__text_ #1 case _ \tl_to_str:n {#5} _tl } {#2} {#3} {#4} 36089
36090 } 36090
36091 {#6} 36091
36092 } 36092
36093 } 36093
36094 \cs_new:Npn \__text_change_case_codepoint_lower:nnnn #1#2#3#4 36094
36095 { 36095
36096 \cs_if_exist_use:cF { __text_change_case_lower_ #3 :nnnnn } 36096
36097 { \__text_change_case_lower_sigma:nnnnn } 36097
36098 {#1} {#1} {#2} {#3} {#4} 36098
36099 } 36099
36100 \cs_new:Npn \__text_change_case_codepoint_upper:nnnn #1#2#3#4 36100
36101 { 36101
36102 \cs_if_exist_use:cF { __text_change_case_upper_ #3 :nnnnn } 36102
36103 { \__text_change_case_codepoint:nnnnn } 36103
36104 {#1} {#1} {#2} {#3} {#4} 36104
36105 } 36105
36106 \cs_new:Npn \__text_change_case_lower_sigma:nnnnn #1#2#3#4#5 36106
36107 { 36107
36108 \__text_codepoint_compare:nNnTF {#5} = { "03A3 } 36108
36109 { \__text_change_case_lower_sigma:nnnnw {#2} } 36109
36110 { \__text_change_case_codepoint:nnnnn {#1} {#2} } 36110
36111 {#3} {#4} {#5} 36111
36112 } 36112
36113 \cs_new:Npn \__text_change_case_lower_sigma:nnnnw #1#2#3#4#5 \q_text_recursion_stop 36113
36114 { 36114
36115 \tl_if_head_is_N_type:nTF {#5} 36115
36116 { \__text_change_case_lower_sigma:nnnnN {#4} } 36116
36117 { 36117
36118 \__text_change_case_store:e 36118
36119 { \codepoint_generate:nn { "03C2 } { \__text_char_catcode:N #4 } } 36119
36120 \__text_change_case_loop:nnnw 36120
36121 } 36121
36122 {#1} {#2} {#3} #5 \q_text_recursion_stop 36122
36123 } 36123
36124 \cs_new:Npn \__text_change_case_lower_sigma:nnnnN #1#2#3#4#5 36124
36125 { 36125
36126 \__text_change_case_store:e 36126
36127 { 36127
36128 \bool_lazy_or:nnTF 36128
```

```
36129         { \token_if_letter_p:N #5 }
36130     {
36131         \bool_lazy_and_p:nn
36132             { \token_if_active_p:N #5 }
36133             { \int_compare_p:nNn {\`#5} > { "80 } }
36134     }
36135     { \codepoint_generate:nn { "03C3 } { \__text_char_catcode:N #1 } }
36136     { \codepoint_generate:nn { "03C2 } { \__text_char_catcode:N #1 } }
36137 }
36138 \__text_change_case_loop:nnnw {#2} {#3} {#4} #5
36139 }
36140 \cs_new:Npn \__text_change_case_codepoint_title:nnnn #1#2#3#4
36141 {
36142     \bool_if:NTF \l_text_titlecase_check_letter_bool
36143     {
36144         \exp_args:Ne \__text_change_case_codepoint_title_auxi:nnnn
36145         {
36146             \codepoint_to_category:n
36147                 { \__text_codepoint_from_chars:Nw #4 }
36148         }
36149     }
36150     { \__text_change_case_codepoint_title:nnn }
36151     {#2} {#3} {#4}
36152 }
36153 \cs_new:Npn \__text_change_case_codepoint_title_auxi:nnnn #1#2#3#4
36154 {
36155     \tl_if_head_eq_charcode:nNTF {#1} { L }
36156     { \__text_change_case_codepoint_title:nnn }
36157     { \__text_change_case_codepoint_title_auxii:nnnn { title } }
36158     {#2} {#3} {#4}
36159 }
36160 \cs_new:Npn \__text_change_case_codepoint_title:nnn #1#2#3
36161     { \__text_change_case_codepoint_title_auxii:nnnn { end } {#1} {#2} {#3} }
36162 \cs_new:Npn \__text_change_case_codepoint_title_auxii:nnnn #1#2#3#4
36163 {
36164     \cs_if_exist_use:cF { __text_change_case_title_ #3 :nnnnn }
36165     {
36166         \cs_if_exist_use:cF { __text_change_case_upper_ #3 :nnnnn }
36167         { \__text_change_case_codepoint:nnnn }
36168     }
36169     { title } {#1} {#2} {#3} {#4}
36170 }
36171 \cs_new:Npn \__text_change_case_codepoint:nnnnn #1#2#3#4#5
36172 {
36173     \bool_lazy_and:nnTF
36174         { \tl_if_single_p:n {#5} }
```

```

36175 { \token_if_active_p:N #5 }
36176 { \__text_change_case_store:n {#5} }
36177 {
36178   \__text_change_case_store:e
36179   { \__text_change_case_codepoint:nn {#1} {#5} }
36180 }
36181 \use:c { \__text_change_case_next_ #2 :nnn } {#2} {#3} {#4}
36182 }
36183 \cs_new:Npn \__text_change_case_codepoint:nn #1#2
36184 {
36185   \__text_change_case_codepoint:fnn
36186   { \int_eval:n { \__text_codepoint_from_chars:Nw #2 } } {#1} {#2}
36187 }
36188 \cs_new:Npn \__text_change_case_codepoint:nnn #1#2#3
36189 {
36190   \exp_args:Ne \__text_change_case_codepoint_aux:nn
36191   { \__kernel_codepoint_case:nn { #2 case } {#1} } {#3}
36192 }
36193 \cs_generate_variant:Nn \__text_change_case_codepoint:nnn { f }
36194 \sys_if_engine_ptex:T
36195 {
36196   \cs_new_eq:NN \__text_change_case_codepoint_aux:nnn
36197   \__text_change_case_codepoint:nnn
36198   \cs_gset:Npn \__text_change_case_codepoint:nnn #1#2#3
36199   {
36200     \int_compare:nNnTF {#1} = { -1 }
36201     { \exp_not:n {#3} }
36202     { \__text_change_case_codepoint_aux:nnn {#1} {#2} {#3} }
36203   }
36204 }
36205 \cs_new:Npn \__text_change_case_codepoint_aux:nn #1#2
36206 {
36207   \use:e { \__text_change_case_codepoint_aux:nnnn #1 {#2} }
36208 }
36209 \cs_new:Npn \__text_change_case_codepoint_aux:nnnn #1#2#3#4
36210 {
36211   \__text_codepoint_compare:nNnTF {#4} = {#1}
36212   { \exp_not:n {#4} }
36213   {
36214     \codepoint_generate:nn {#1}
36215     { \__text_change_case_catcode:nn {#4} {#1} }
36216     \tl_if_blank:nF {#2}
36217     {
36218       \codepoint_generate:nn {#2}
36219       { \char_value_catcode:n {#2} }
36220       \tl_if_blank:nF {#3}

```

```

36221         {
36222             \codepoint_generate:nn {#3}
36223             { \char_value_catcode:n {#3} }
36224         }
36225     }
36226 }
36227 }
36228 \sys_if_engine_opentype:TF
36229 {
36230     \cs_new:Npn \__text_change_case_catcode:nn #1#2
36231     { \__text_char_catcode:N #1 }
36232 }
36233 {
36234     \cs_new:Npn \__text_change_case_catcode:nn #1#2
36235     {
36236         \__text_codepoint_compare:nNnTF {#1} < { "80 }
36237         { \__text_char_catcode:N #1 }
36238         {
36239             \int_compare:nNnTF {#2} < { "80 }
36240             { \char_value_catcode:n {#2} }
36241             { 13 }
36242         }
36243     }
36244 }
36245 \cs_new:Npn \__text_change_case_next_lower:nnn #1#2#3
36246 { \__text_change_case_loop:nnnw {#1} {#2} {#3} }
36247 \cs_new_eq:NN \__text_change_case_next_upper:nnn
36248 \__text_change_case_next_lower:nnn
36249 \cs_new_eq:NN \__text_change_case_next_title:nnn
36250 \__text_change_case_next_lower:nnn
36251 \cs_new:Npn \__text_change_case_next_end:nnn #1#2#3
36252 { \__text_change_case_skip:nnw {#2} {#3} }
36253 \cs_new_protected:Npn \text_declare_case_equivalent:Nn #1#2
36254 {
36255     \tl_clear_new:c { l__text_case_ \token_to_str:N #1 _tl }
36256     \tl_set:cn { l__text_case_ \token_to_str:N #1 _tl } {#2}
36257 }
36258 \cs_new_protected:Npn \text_declare_lowercase_mapping:nn #1#2
36259 { \__text_declare_case_mapping:nnn { lower } {#1} {#2} }
36260 \cs_new_protected:Npn \text_declare_titlecase_mapping:nn #1#2
36261 { \__text_declare_case_mapping:nnn { title } {#1} {#2} }
36262 \cs_new_protected:Npn \text_declare_uppercase_mapping:nn #1#2
36263 { \__text_declare_case_mapping:nnn { upper } {#1} {#2} }
36264 \cs_new_protected:Npn \__text_declare_case_mapping:nnn #1#2#3
36265 {
36266     \exp_args:Ne \__text_declare_case_mapping_aux:nnn

```



```
36267 { \codepoint_str_generate:n {#2} } {#1} {#3} 36267
36268 } 36268
36269 \cs_new_protected:Npn \__text_declare_case_mapping_aux:nnn #1#2#3 36269
36270 { 36270
36271     \tl_clear_new:c { l__text_ #2 case _ #1 _tl } 36271
36272     \tl_set:cn { l__text_ #2 case _ #1 _tl } {#3} 36272
36273 } 36273
36274 \cs_new_protected:Npn \text_declare_lowercase_mapping:nnn #1#2#3 36274
36275 { \__text_declare_case_mapping:nnnn { lower } {#1} {#2} {#3} } 36275
36276 \cs_new_protected:Npn \text_declare_titlecase_mapping:nnn #1#2#3 36276
36277 { \__text_declare_case_mapping:nnnn { title } {#1} {#2} {#3} } 36277
36278 \cs_new_protected:Npn \text_declare_uppercase_mapping:nnn #1#2#3 36278
36279 { \__text_declare_case_mapping:nnnn { upper } {#1} {#2} {#3} } 36279
36280 \cs_new_protected:Npn \__text_declare_case_mapping:nnnn #1#2#3#4 36280
36281 { 36281
36282     \exp_args:Ne \__text_declare_case_mapping_aux:nnnn 36282
36283     { \codepoint_str_generate:n {#3} } {#1} {#2} {#4} 36283
36284 } 36284
36285 \cs_new_protected:Npn \__text_declare_case_mapping_aux:nnnn #1#2#3#4 36285
36286 { 36286
36287     \tl_clear_new:c { l__text_ #2 case _ #1 _ #3 _tl } 36287
36288     \tl_set:cn { l__text_ #2 case _ #1 _ #3 _tl } {#4} 36288
36289     \tl_clear_new:c { l__text_ #2 case_special_ #3 _tl } 36289
36290 } 36290
36291 \cs_new:Npn \text_case_switch:nnnn #1#2#3#4 36291
36292 { 36292
36293     \__text_case_switch_marker: 36293
36294     #1 36294
36295 } 36295
36296 \cs_new:Npn \__text_case_switch_marker: { } 36296
36297 \cs_new:Npn \__text_change_case_generate:n #1 36297
36298 { \codepoint_generate:nn {#1} { \char_value_catcode:n {#1} } } 36298
36299 \cs_new:cpn { __text_change_case_upper_de-x-eszett:nnnnn } #1#2#3#4#5 36299
36300 { 36300
36301     \__text_codepoint_compare:nNnTF {#5} = { "00DF } 36301
36302     { 36302
36303         \__text_change_case_store:e 36303
36304         { 36304
36305             \codepoint_generate:nn { "1E9E } 36305
36306             { \__text_change_case_catcode:nn {#5} { "1E9E } } 36306
36307         } 36307
36308         \use:c { __text_change_case_next_ #2 :nnn } 36308
36309         {#2} {#3} {#4} 36309
36310     } 36310
36311     { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} } 36311
36312 } 36312
```



```
36313 \cs_new_eq:cc { __text_change_case_upper_de-alt:nnnnn } 36313
36314 { __text_change_case_upper_de-x-eszett:nnnnn } 36314
36315 \cs_new:Npn \__text_change_case_upper_el:nnnnn #1#2#3#4#5 36315
36316 { 36316
36317 \bool_lazy_and:nnTF 36317
36318 { \__text_change_case_if_greek_p:n {#5} } 36318
36319 { 36319
36320 ! \bool_lazy_or_p:nn 36320
36321 { \__text_codepoint_compare_p:nNn {#5} = { "0374 } } 36321
36322 { \__text_codepoint_compare_p:nNn {#5} = { "037E } } 36322
36323 } 36323
36324 { 36324
36325 \__text_change_case_if_greek_spacing_diacritic:nTF {#5} 36325
36326 { 36326
36327 \__text_change_case_store:n {#5} 36327
36328 \__text_change_case_loop:nnnw 36328
36329 } 36329
36330 { 36330
36331 \exp_args:Ne \__text_change_case_upper_el:nnnn 36331
36332 { 36332
36333 \codepoint_to_nfd:n 36333
36334 { \__text_codepoint_from_chars:Nw #5 } 36334
36335 } 36335
36336 } 36336
36337 {#2} {#3} {#4} 36337
36338 } 36338
36339 { 36339
36340 \__text_codepoint_compare:nNnTF {#5} = { "0345 } 36340
36341 { 36341
36342 \__text_change_case_store:e 36342
36343 { 36343
36344 \codepoint_generate:nn { "0399 } 36344
36345 { \char_value_catcode:n { "0399 } } 36345
36346 } 36346
36347 \__text_change_case_loop:nnnw {#2} {#3} {#4} 36347
36348 } 36348
36349 { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} } 36349
36350 } 36350
36351 } 36351
36352 \cs_new_eq:cN { __text_change_case_upper_el-x-iota:nnnnn } 36352
36353 \__text_change_case_upper_el:nnnnn 36353
36354 \cs_new:Npn \__text_change_case_upper_el:nnnn #1#2#3#4 36354
36355 { 36355
36356 \__text_codepoint_process:nN 36356
36357 { \__text_change_case_upper_el:nnnnw {#2} {#3} {#4} } #1 36357
36358 } 36358
```

```
36359 \cs_new:Npn \__text_change_case_upper_el:nnnnw #1#2#3#4#5 \q__text_recursion_stop 36359
36360 { 36360
36361 \tl_if_head_is_N_type:nTF {#5} 36361
36362 { \__text_change_case_upper_el:nnnnN {#4} } 36362
36363 { 36363
36364 \__text_change_case_store:e 36364
36365 { \__text_change_case_codepoint:nn { upper } {#4} } 36365
36366 \__text_change_case_loop:nnnw 36366
36367 } 36367
36368 {#1} {#2} {#3} #5 \q__text_recursion_stop 36368
36369 } 36369
36370 \cs_new:Npn \__text_change_case_upper_el:nnnnN #1#2#3#4#5 36370
36371 { 36371
36372 \token_if_cs:NTF #5 36372
36373 { 36373
36374 \__text_change_case_store:e 36374
36375 { \__text_change_case_codepoint:nn { upper } {#1} } 36375
36376 \__text_change_case_loop:nnnw {#2} {#3} {#4} #5 36376
36377 } 36377
36378 { 36378
36379 \__text_change_case_if_takes_ypogegrammeni:nTF {#1} 36379
36380 { 36380
36381 \__text_change_case_upper_el_ypogegrammeni:nnnnnnw 36381
36382 {#1} {#2} {#3} {#4} { } { } #5 36382
36383 } 36383
36384 { \__text_change_case_upper_el_aux:nnnnN {#1} {#2} {#3} {#4} #5 } 36384
36385 } 36385
36386 } 36386
36387 \cs_new:Npn \__text_change_case_upper_el_ypogegrammeni:nnnnnnw 36387
36388 #1#2#3#4#5#6#7 \q__text_recursion_stop 36388
36389 { 36389
36390 \tl_if_head_is_N_type:nTF {#7} 36390
36391 { 36391
36392 \__text_change_case_upper_el_ypogegrammeni:nnnnnnN 36392
36393 {#1} {#2} {#3} {#4} {#5} {#6} 36393
36394 } 36394
36395 { \__text_change_case_upper_el_aux:nnnnN {#1} {#2} {#3} {#4} #5#6 } 36395
36396 #7 \q__text_recursion_stop 36396
36397 } 36397
36398 \cs_new:Npn \__text_change_case_upper_el_ypogegrammeni:nnnnnnN #1#2#3#4#5#6#7 36398
36399 { 36399
36400 \token_if_cs:NTF #7 36400
36401 { \__text_change_case_upper_el_aux:nnnnN {#1} {#2} {#3} {#4} #5#6 } 36401
36402 { 36402
36403 \__text_codepoint_process:nN 36403
36404 { 36404
```

```
36405         \__text_change_case_upper_el_ypogegrammeni:nnnnnnn 36405
36406         {#1} {#2} {#3} {#4} {#5} {#6} 36406
36407     } 36407
36408 } 36408
36409 #7 36409
36410 } 36410
36411 \cs_new:Npn \__text_change_case_upper_el_ypogegrammeni:nnnnnnn #1#2#3#4#5#6#7 36411
36412 { 36412
36413     \__text_codepoint_compare:nNnTF {#7} = { "0345 } 36413
36414     { 36414
36415         \__text_change_case_upper_el_ypogegrammeni:nnnnnnnw 36415
36416         {#1} {#2} {#3} {#4} {#5} {#7} 36416
36417     } 36417
36418     { 36418
36419         \bool_lazy_or:nnTF 36419
36420         { \__text_change_case_if_greek_accent_p:n {#7} } 36420
36421         { \__text_change_case_if_greek_breathing_p:n {#7} } 36421
36422         { 36422
36423             \__text_change_case_upper_el_ypogegrammeni:nnnnnnw 36423
36424             {#1} {#2} {#3} {#4} {#5#7} {#6} 36424
36425         } 36425
36426         { \__text_change_case_upper_el_aux:nnnnN {#1} {#2} {#3} {#4} #5#6 #7 } 36426
36427     } 36427
36428 } 36428
36429 \cs_new:Npn \__text_change_case_upper_el_aux:nnnnN #1#2#3#4#5 36429
36430 { 36430
36431     \__text_codepoint_process:nN 36431
36432     { \__text_change_case_upper_el_aux:nnnnn {#1} {#2} {#3} {#4} } #5 36432
36433 } 36433
36434 \cs_new:Npn \__text_change_case_upper_el_aux:nnnnn #1#2#3#4#5 36434
36435 { 36435
36436     \__text_codepoint_compare:nNnTF {#5} = { "0308 } 36436
36437     { \__text_change_case_upper_el_dialytika:nnnn {#2} {#3} {#4} {#1} } 36437
36438     { 36438
36439         \__text_change_case_if_greek_accent:nTF {#5} 36439
36440         { \__text_change_case_upper_el_hiatus:nnnnw {#2} {#3} {#4} {#1} } 36440
36441         { 36441
36442             \__text_change_case_if_greek_breathing:nTF {#5} 36442
36443             { \__text_change_case_upper_el:nnnn {#1} {#2} {#3} {#4} } 36443
36444             { 36444
36445                 \__text_codepoint_compare:nNnTF {#5} = { "0345 } 36445
36446                 { 36446
36447                     \__text_change_case_store:e 36447
36448                     { \use:c { \__text_change_case_upper_ #4 _ypogegrammeni:n } {#1} } 36448
36449                     \__text_change_case_loop:nnnw {#2} {#3} {#4} 36449
36450                 } 36450
```

```
36451         {
36452             \__text_change_case_if_greek_stress:nTF {#5}
36453             {
36454                 \__text_change_case_store:e
36455                 { \__text_change_case_upper_el_stress:nn {#1} {#5} }
36456                 \__text_change_case_loop:nnnw {#2} {#3} {#4}
36457             }
36458             {
36459                 \__text_change_case_store:e
36460                 { \__text_change_case_codepoint:nn { upper } {#1} }
36461                 \__text_change_case_loop:nnnw {#2} {#3} {#4} #5
36462             }
36463         }
36464     }
36465 }
36466 }
36467 }
36468 \cs_new:Npn \__text_change_case_upper_el_dialytika:nnnn #1#2#3#4
36469 {
36470     \__text_change_case_if_takes_dialytika:nTF {#4}
36471     { \__text_change_case_upper_el_dialytika:n {#4} }
36472     {
36473         \__text_change_case_store:e
36474         { \__text_change_case_codepoint:nn { upper } {#4} }
36475     }
36476     \__text_change_case_upper_el_gobble:nnnw {#1} {#2} {#3}
36477 }
36478 \cs_new:Npn \__text_change_case_upper_el_dialytika:n #1
36479 {
36480     \__text_change_case_store:e
36481     {
36482         \bool_lazy_or:nnTF
36483         { \__text_codepoint_compare_p:nNn {#1} = { "0399 } }
36484         { \__text_codepoint_compare_p:nNn {#1} = { "03B9 } }
36485         {
36486             \codepoint_generate:nn { "03AA }
36487             { \__text_change_case_catcode:nn {#1} { "03AA } }
36488         }
36489         {
36490             \codepoint_generate:nn { "03AB }
36491             { \__text_change_case_catcode:nn {#1} { "03AB } }
36492         }
36493     }
36494 }
36495 \cs_new:Npn \__text_change_case_upper_el_hiatus:nnnnw
36496 #1#2#3#4#5 \q__text_recursion_stop
```

```
36497 { 36497
36498 \tl_if_head_is_N_type:nTF {#5} 36498
36499 { \__text_change_case_upper_el_hiatus:nnnnN {#4} } 36499
36500 { 36500
36501 \__text_change_case_store:e 36501
36502 { \__text_change_case_codepoint:nn { upper } {#4} } 36502
36503 \__text_change_case_loop:nnnw 36503
36504 } 36504
36505 {#1} {#2} {#3} #5 \q__text_recursion_stop 36505
36506 } 36506
36507 \cs_new:Npn \__text_change_case_upper_el_hiatus:nnnnN #1#2#3#4#5 36507
36508 { 36508
36509 \token_if_cs:NTF #5 36509
36510 { 36510
36511 \__text_change_case_store:e 36511
36512 { \__text_change_case_codepoint:nn { upper } {#1} } 36512
36513 \__text_change_case_loop:nnnw {#2} {#3} {#4} #5 36513
36514 } 36514
36515 { 36515
36516 \__text_codepoint_process:nN 36516
36517 { \__text_change_case_upper_el_hiatus:nnnnn {#1} {#2} {#3} {#4} } #5 36517
36518 } 36518
36519 } 36519
36520 \cs_new:Npn \__text_change_case_upper_el_hiatus:nnnnn #1#2#3#4#5 36520
36521 { 36521
36522 \__text_change_case_if_takes_dialytika:nTF {#5} 36522
36523 { 36523
36524 \__text_change_case_store:e 36524
36525 { \__text_change_case_codepoint:nn { upper } {#1} } 36525
36526 \__text_change_case_upper_el_dialytika:n {#5} 36526
36527 \__text_change_case_upper_el_gobble:nnnw {#2} {#3} {#4} 36527
36528 } 36528
36529 { \__text_change_case_upper_el:nnnn {#1} {#2} {#3} {#4} #5 } 36529
36530 } 36530
36531 \cs_new:Npn \__text_change_case_upper_el_ypogegrammeni:n #1 36531
36532 { 36532
36533 \exp_args:Ne \__text_change_case_generate:n 36533
36534 { 36534
36535 \int_case:nn 36535
36536 { \__text_codepoint_from_chars:Nw #1 } 36536
36537 { 36537
36538 { "0391 } { "1FBC } 36538
36539 { "03B1 } { "1FBC } 36539
36540 { "0397 } { "1FCC } 36540
36541 { "03B7 } { "1FCC } 36541
36542 { "03A9 } { "1FFC } 36542
```

```
36543         { "03C9 } { "1FFC }
36544     }
36545 }
36546 }
36547 \cs_new:cpn { __text_change_case_upper_el-x-iota_ypogegrammeni:n } #1
36548 {
36549     \__text_change_case_codepoint:nn { upper } {#1}
36550     \codepoint_generate:nn { "0399 }
36551     { \char_value_catcode:n { "0399 } }
36552 }
36553 \cs_new:Npn \__text_change_case_upper_el_stress:nn #1#2
36554 {
36555     \exp_args:Ne \__text_change_case_generate:n
36556     {
36557         \int_case:nn
36558         { \__text_codepoint_from_chars:Nw #2 }
36559         {
36560             { "0304 }
36561             {
36562                 \int_case:nn { \__text_codepoint_from_chars:Nw #1 }
36563                 {
36564                     { "0391 } { "1FB9 }
36565                     { "03B1 } { "1FB9 }
36566                     { "0399 } { "1FD9 }
36567                     { "03B9 } { "1FD9 }
36568                     { "03A5 } { "1FE9 }
36569                     { "03C5 } { "1FE9 }
36570                 }
36571             }
36572             { "0306 }
36573             {
36574                 \int_case:nn { \__text_codepoint_from_chars:Nw #1 }
36575                 {
36576                     { "0391 } { "1FB8 }
36577                     { "03B1 } { "1FB8 }
36578                     { "0399 } { "1FD8 }
36579                     { "03B9 } { "1FD8 }
36580                     { "03A5 } { "1FE8 }
36581                     { "03C5 } { "1FE8 }
36582                 }
36583             }
36584         }
36585     }
36586 }
36587 \cs_new:Npn \__text_change_case_upper_el_gobble:nnnw
36588     #1#2#3#4 \q__text_recursion_stop
```

```
36589 { 36589
36590 \tl_if_head_is_N_type:nTF {#4} 36590
36591 { \__text_change_case_upper_el_gobble:nnnN } 36591
36592 { \__text_change_case_loop:nnnw } 36592
36593 {#1} {#2} {#3} #4 \q__text_recursion_stop 36593
36594 } 36594
36595 \cs_new:Npn \__text_change_case_upper_el_gobble:nnnN #1#2#3#4 36595
36596 { 36596
36597 \token_if_cs:NTF #4 36597
36598 { \__text_change_case_loop:nnnw {#1} {#2} {#3} } 36598
36599 { 36599
36600 \__text_codepoint_process:nN 36600
36601 { \__text_change_case_upper_el_gobble:nnnn {#1} {#2} {#3} } 36601
36602 } 36602
36603 #4 36603
36604 } 36604
36605 \cs_new:Npn \__text_change_case_upper_el_gobble:nnnn #1#2#3#4 36605
36606 { 36606
36607 \bool_lazy_or:nnTF 36607
36608 { \__text_change_case_if_greek_accent_p:n {#4} } 36608
36609 { \__text_change_case_if_greek_breathing_p:n {#4} } 36609
36610 { \__text_change_case_upper_el_gobble:nnnw {#1} {#2} {#3} } 36610
36611 { \__text_change_case_loop:nnnw {#1} {#2} {#3} #4 } 36611
36612 } 36612
36613 \prg_new_conditional:Npnn \__text_change_case_if_greek:n #1 { p , TF } 36613
36614 { 36614
36615 \exp_args:Nf \__text_change_case_if_greek:n 36615
36616 { \int_eval:n { \__text_codepoint_from_chars:Nw #1 } } 36616
36617 } 36617
36618 \cs_new:Npn \__text_change_case_if_greek:n #1 36618
36619 { 36619
36620 \if_int_compare:w #1 < "0370 \exp_stop_f: 36620
36621 \prg_return_false: 36621
36622 \else: 36622
36623 \if_int_compare:w #1 > "03FF \exp_stop_f: 36623
36624 \if_int_compare:w #1 < "1F00 \exp_stop_f: 36624
36625 \prg_return_false: 36625
36626 \else: 36626
36627 \if_int_compare:w #1 > "1FFF \exp_stop_f: 36627
36628 \if_int_compare:w #1 = "2126 \exp_stop_f: 36628
36629 \prg_return_true: 36629
36630 \else: 36630
36631 \prg_return_false: 36631
36632 \fi: 36632
36633 \else: 36633
36634 \prg_return_true: 36634
```



```
36635         \fi: 36635
36636         \fi: 36636
36637     \else: 36637
36638         \prg_return_true: 36638
36639     \fi: 36639
36640 \fi: 36640
36641 } 36641
36642 \prg_new_conditional:Npnn \__text_change_case_if_greek_accent:n #1 { TF , p } 36642
36643 { 36643
36644     \exp_args:Nf \__text_change_case_if_greek_accent:n 36644
36645     { \int_eval:n { \__text_codepoint_from_chars:Nw #1 } } 36645
36646 } 36646
36647 \cs_new:Npn \__text_change_case_if_greek_accent:n #1 36647
36648 { 36648
36649     \if_int_compare:w #1 = "0300 \exp_stop_f: 36649
36650     \prg_return_true: 36650
36651 \else: 36651
36652     \if_int_compare:w #1 = "0301 \exp_stop_f: 36652
36653     \prg_return_true: 36653
36654 \else: 36654
36655     \if_int_compare:w #1 = "0342 \exp_stop_f: 36655
36656     \prg_return_true: 36656
36657 \else: 36657
36658     \if_int_compare:w #1 = "0302 \exp_stop_f: 36658
36659     \prg_return_true: 36659
36660 \else: 36660
36661     \if_int_compare:w #1 = "0303 \exp_stop_f: 36661
36662     \prg_return_true: 36662
36663 \else: 36663
36664     \if_int_compare:w #1 = "0311 \exp_stop_f: 36664
36665     \prg_return_true: 36665
36666     \else: 36666
36667     \prg_return_false: 36667
36668     \fi: 36668
36669     \fi: 36669
36670     \fi: 36670
36671     \fi: 36671
36672     \fi: 36672
36673 \fi: 36673
36674 } 36674
36675 \prg_new_conditional:Npnn \__text_change_case_if_greek_spacing_diacritic:n 36675
36676 #1 { TF } 36676
36677 { 36677
36678     \exp_args:Nf \__text_change_case_if_greek_spacing_diacritic:n 36678
36679     { \int_eval:n { \__text_codepoint_from_chars:Nw #1 } } 36679
36680 } 36680
```

36681	\cs_new:Npn __text_change_case_if_greek_spacing_diacritic:n #1	36681
36682	{	36682
36683	\if_int_compare:w #1 < "1FBD \exp_stop_f:	36683
36684	\if_int_compare:w #1 = "037A \exp_stop_f:	36684
36685	\prg_return_true:	36685
36686	\else:	36686
36687	\prg_return_false:	36687
36688	\fi:	36688
36689	\else:	36689
36690	\if_int_compare:w #1 = "1FBD \exp_stop_f:	36690
36691	\prg_return_true:	36691
36692	\else:	36692
36693	\if_int_compare:w #1 = "1FBF \exp_stop_f:	36693
36694	\prg_return_true:	36694
36695	\else:	36695
36696	\if_int_compare:w #1 = "1FC0 \exp_stop_f:	36696
36697	\prg_return_true:	36697
36698	\else:	36698
36699	\if_int_compare:w #1 = "1FC1 \exp_stop_f:	36699
36700	\prg_return_true:	36700
36701	\else:	36701
36702	\if_int_compare:w #1 = "1FCD \exp_stop_f:	36702
36703	\prg_return_true:	36703
36704	\else:	36704
36705	\if_int_compare:w #1 = "1FCE \exp_stop_f:	36705
36706	\prg_return_true:	36706
36707	\else:	36707
36708	\if_int_compare:w #1 = "1FCF \exp_stop_f:	36708
36709	\prg_return_true:	36709
36710	\else:	36710
36711	\if_int_compare:w #1 = "1FDD \exp_stop_f:	36711
36712	\prg_return_true:	36712
36713	\else:	36713
36714	\if_int_compare:w #1 = "1FDE \exp_stop_f:	36714
36715	\prg_return_true:	36715
36716	\else:	36716
36717	\if_int_compare:w #1 = "1FDF \exp_stop_f:	36717
36718	\prg_return_true:	36718
36719	\else:	36719
36720	\if_int_compare:w #1 = "1FED \exp_stop_f:	36720
36721	\prg_return_true:	36721
36722	\else:	36722
36723	\if_int_compare:w #1 = "1FEE \exp_stop_f:	36723
36724	\prg_return_true:	36724
36725	\else:	36725
36726	\if_int_compare:w #1 = "1FEF \exp_stop_f:	36726

```
36727         \prg_return_true: 36727
36728     \else: 36728
36729         \if_int_compare:w #1 = "1FFD \exp_stop_f: 36729
36730         \prg_return_true: 36730
36731     \else: 36731
36732         \if_int_compare:w #1 = "1FFE \exp_stop_f: 36732
36733         \prg_return_true: 36733
36734     \else: 36734
36735         \prg_return_false: 36735
36736     \fi: 36736
36737 \fi: 36737
36738 \fi: 36738
36739 \fi: 36739
36740 \fi: 36740
36741 \fi: 36741
36742 \fi: 36742
36743 \fi: 36743
36744 \fi: 36744
36745 \fi: 36745
36746 \fi: 36746
36747 \fi: 36747
36748 \fi: 36748
36749 \fi: 36749
36750 \fi: 36750
36751 \fi: 36751
36752 } 36752
36753 \prg_new_conditional:Npnn \__text_change_case_if_greek_breathing:n 36753
36754 #1 { TF , p } 36754
36755 { 36755
36756     \exp_args:Nf \__text_change_case_if_greek_breathing:n 36756
36757     { \int_eval:n { \__text_codepoint_from_chars:Nw #1 } } 36757
36758 } 36758
36759 \cs_new:Npn \__text_change_case_if_greek_breathing:n #1 36759
36760 { 36760
36761     \if_int_compare:w #1 = "0313 \exp_stop_f: 36761
36762     \prg_return_true: 36762
36763 \else: 36763
36764     \if_int_compare:w #1 = "0314 \exp_stop_f: 36764
36765     \prg_return_true: 36765
36766 \else: 36766
36767     \prg_return_false: 36767
36768 \fi: 36768
36769 \fi: 36769
36770 } 36770
36771 \prg_new_conditional:Npnn \__text_change_case_if_greek_stress:n 36771
36772 #1 { TF , p } 36772
```

```

36773 { 36773
36774 \exp_args:Nf \__text_change_case_if_greek_stress:n 36774
36775 { \int_eval:n { \__text_codepoint_from_chars:Nw #1 } } 36775
36776 } 36776
36777 \cs_new:Npn \__text_change_case_if_greek_stress:n #1 36777
36778 { 36778
36779 \if_int_compare:w #1 = "0304 \exp_stop_f: 36779
36780 \prg_return_true: 36780
36781 \else: 36781
36782 \if_int_compare:w #1 = "0306 \exp_stop_f: 36782
36783 \prg_return_true: 36783
36784 \else: 36784
36785 \prg_return_false: 36785
36786 \fi: 36786
36787 \fi: 36787
36788 } 36788
36789 \prg_new_conditional:Npnn \__text_change_case_if_takes_dialytika:n #1 { TF } 36789
36790 { 36790
36791 \exp_args:Nf \__text_change_case_if_takes_dialytika:n 36791
36792 { \int_eval:n { \__text_codepoint_from_chars:Nw #1 } } 36792
36793 } 36793
36794 \cs_new:Npn \__text_change_case_if_takes_dialytika:n #1 36794
36795 { 36795
36796 \if_int_compare:w #1 = "0399 \exp_stop_f: 36796
36797 \prg_return_true: 36797
36798 \else: 36798
36799 \if_int_compare:w #1 = "03B9 \exp_stop_f: 36799
36800 \prg_return_true: 36800
36801 \else: 36801
36802 \if_int_compare:w #1 = "03A5 \exp_stop_f: 36802
36803 \prg_return_true: 36803
36804 \else: 36804
36805 \if_int_compare:w #1 = "03C5 \exp_stop_f: 36805
36806 \prg_return_true: 36806
36807 \else: 36807
36808 \prg_return_false: 36808
36809 \fi: 36809
36810 \fi: 36810
36811 \fi: 36811
36812 \fi: 36812
36813 } 36813
36814 \prg_new_conditional:Npnn \__text_change_case_if_takes_ypogegrammeni:n #1 { TF } 36814
36815 { 36815
36816 \exp_args:Nf \__text_change_case_if_takes_ypogegrammeni:n 36816
36817 { \int_eval:n { \__text_codepoint_from_chars:Nw #1 } } 36817
36818 } 36818

```

```
36819 \cs_new:Npn \__text_change_case_if_takes_ypogegrammeni:n #1 36819
36820 { 36820
36821 \if_int_compare:w #1 = "03B1 \exp_stop_f: 36821
36822 \prg_return_true: 36822
36823 \else: 36823
36824 \if_int_compare:w #1 = "03B7 \exp_stop_f: 36824
36825 \prg_return_true: 36825
36826 \else: 36826
36827 \if_int_compare:w #1 = "03C9 \exp_stop_f: 36827
36828 \prg_return_true: 36828
36829 \else: 36829
36830 \prg_return_false: 36830
36831 \fi: 36831
36832 \fi: 36832
36833 \fi: 36833
36834 } 36834
36835 \cs_new:Npn \__text_change_case_boundary_upper_el:Nnnnw 36835
36836 #1#2#3#4#5 \q__text_recursion_stop 36836
36837 { 36837
36838 \tl_if_head_is_N_type:nTF {#5} 36838
36839 { \__text_change_case_boundary_upper_el:nnnN } 36839
36840 { \__text_change_case_loop:nnnw } 36840
36841 {#2} {#3} {#4} #5 \q__text_recursion_stop 36841
36842 } 36842
36843 \cs_new_eq:cN { __text_change_case_boundary_upper_el-x-iota:Nnnnw } 36843
36844 \__text_change_case_boundary_upper_el:Nnnnw 36844
36845 \cs_new:Npn \__text_change_case_boundary_upper_el:nnnN #1#2#3#4 36845
36846 { 36846
36847 \token_if_cs:NTF #4 36847
36848 { \__text_change_case_loop:nnnw {#1} {#2} {#3} } 36848
36849 { 36849
36850 \__text_codepoint_process:nN 36850
36851 { \__text_change_case_boundary_upper_el:nnnn {#1} {#2} {#3} } 36851
36852 } 36852
36853 #4 36853
36854 } 36854
36855 \cs_new:Npn \__text_change_case_boundary_upper_el:nnnn #1#2#3#4 36855
36856 { 36856
36857 \bool_lazy_any:nTF 36857
36858 { 36858
36859 { \__text_codepoint_compare_p:nNn {#4} = { "0389 } } 36859
36860 { \__text_codepoint_compare_p:nNn {#4} = { "03AE } } 36860
36861 { \__text_codepoint_compare_p:nNn {#4} = { "1F22 } } 36861
36862 { \__text_codepoint_compare_p:nNn {#4} = { "1F2A } } 36862
36863 } 36863
36864 { \__text_change_case_boundary_upper_el:nnnnw {#1} {#2} {#3} {#4} } 36864
```

```
36865 { \_text_change_case_breathing:nnnn {#1} {#2} {#3} {#4} } 36865
36866 } 36866
36867 \cs_new:Npn \_text_change_case_boundary_upper_el:nnnnw 36867
36868 #1#2#3#4#5 \q__text_recursion_stop 36868
36869 { 36869
36870 \tl_if_head_is_N_type:nTF {#5} 36870
36871 { \_text_change_case_loop:nnnw {#1} {#2} {#3} #4 } 36871
36872 { 36872
36873 \_text_change_case_store:e 36873
36874 { 36874
36875 \codepoint_generate:nn { "0389 } 36875
36876 { \_text_change_case_catcode:nn {#4} { "0389 } } 36876
36877 } 36877
36878 \_text_change_case_loop:nnnw {#1} {#2} {#3} 36878
36879 } 36879
36880 #5 \q__text_recursion_stop 36880
36881 } 36881
36882 \cs_new:Npn \_text_change_case_breathing:nnnn #1#2#3#4 36882
36883 { 36883
36884 \_text_change_case_if_greek:nTF {#4} 36884
36885 { 36885
36886 \exp_args:Ne \_text_change_case_breathing:nnnnn 36886
36887 { 36887
36888 \codepoint_to_nfd:n 36888
36889 { \_text_codepoint_from_chars:Nw #4 } 36889
36890 } 36890
36891 {#1} {#2} {#3} {#4} 36891
36892 } 36892
36893 { \_text_change_case_loop:nnnw {#1} {#2} {#3} #4 } 36893
36894 } 36894
36895 \cs_new:Npn \_text_change_case_breathing:nnnnn #1#2#3#4#5 36895
36896 { 36896
36897 \_text_codepoint_process:nN 36897
36898 { \_text_change_case_breathing:nnnnnw {#2} {#3} {#4} {#5} } 36898
36899 #1 \q_mark 36899
36900 } 36900
36901 \cs_new:Npn \_text_change_case_breathing:nnnnnw #1#2#3#4#5#6 \q_mark 36901
36902 { 36902
36903 \tl_if_blank:nTF {#6} 36903
36904 { \_text_change_case_loop:nnnw {#1} {#2} {#3} #4 } 36904
36905 { 36905
36906 \_text_codepoint_process:nN 36906
36907 { \_text_change_case_breathing:nnnnnw {#1} {#2} {#3} {#4} {#5} } 36907
36908 #6 \q_mark 36908
36909 } 36909
36910 } 36910
```

```
36911 \cs_new:Npn \__text_change_case_breathing:nnnnnnw #1#2#3#4#5#6#7 \q_mark 36911
36912 { 36912
36913 \tl_if_blank:nTF {#7} 36913
36914 { 36914
36915 \__text_change_case_breathing_aux:nnnnnn 36915
36916 {#1} {#2} {#3} {#4} {#5} {#6} 36916
36917 } 36917
36918 { 36918
36919 \__text_codepoint_process:nN 36919
36920 { \__text_change_case_breathing:nnnnnnw {#1} {#2} {#3} {#4} {#5} } 36920
36921 #7 \q_mark 36921
36922 } 36922
36923 } 36923
36924 \cs_new:Npn \__text_change_case_breathing_aux:nnnnnn #1#2#3#4#5#6 36924
36925 { 36925
36926 \bool_lazy_or:nnTF 36926
36927 { \__text_codepoint_compare_p:nNn {#6} = { "0313 } } 36927
36928 { \__text_codepoint_compare_p:nNn {#6} = { "0314 } } 36928
36929 { \__text_change_case_breathing_aux:nnnnw {#1} {#2} {#3} {#5} } 36929
36930 { \__text_change_case_loop:nnnw {#1} {#2} {#3} #4 } 36930
36931 } 36931
36932 \cs_new:Npn \__text_change_case_breathing_aux:nnnnw #1#2#3#4#5 36932
36933 \q__text_recursion_stop 36933
36934 { 36934
36935 \__text_change_case_store:e 36935
36936 { \__text_change_case_codepoint:nn { upper } {#4} } 36936
36937 \tl_if_head_is_N_type:nTF {#5} 36937
36938 { \__text_change_case_breathing_aux:nnnN } 36938
36939 { \__text_change_case_loop:nnnw } 36939
36940 {#1} {#2} {#3} #5 \q__text_recursion_stop 36940
36941 } 36941
36942 \cs_new:Npn \__text_change_case_breathing_aux:nnnN #1#2#3#4 36942
36943 { 36943
36944 \__text_codepoint_process:nN 36944
36945 { \__text_change_case_breathing_dialytika:nnnn {#1} {#2} {#3} } #4 36945
36946 } 36946
36947 \cs_new:Npn \__text_change_case_breathing_dialytika:nnnn #1#2#3#4 36947
36948 { 36948
36949 \__text_change_case_if_takes_dialytika:nTF {#4} 36949
36950 { 36950
36951 \__text_change_case_upper_el_dialytika:n {#4} 36951
36952 \__text_change_case_loop:nnnw {#1} {#2} {#3} 36952
36953 } 36953
36954 { \__text_change_case_loop:nnnw {#1} {#2} {#3} #4 } 36954
36955 } 36955
36956 \cs_new:Npn \__text_change_case_title_el:nnnnn #1#2#3#4#5 36956
```



```
36957 { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} } 36957
36958 \cs_new:Npn \__text_change_case_upper_hy:nnnnn #1#2#3#4#5 36958
36959 { 36959
36960 \__text_codepoint_compare:nNnTF {#5} = { "0587 } 36960
36961 { 36961
36962 \__text_change_case_store:e 36962
36963 { 36963
36964 \codepoint_generate:nn { "0535 } 36964
36965 { \__text_change_case_catcode:nn {#5} { "0535 } } 36965
36966 \codepoint_generate:nn { "054E } 36966
36967 { \__text_change_case_catcode:nn {#5} { "054E } } 36967
36968 } 36968
36969 \use:c { __text_change_case_next_ #2 :nnn } 36969
36970 {#2} {#3} {#4} 36970
36971 } 36971
36972 { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} } 36972
36973 } 36973
36974 \cs_new:Npn \__text_change_case_title_hy:nnnnn #1#2#3#4#5 36974
36975 { 36975
36976 \__text_codepoint_compare:nNnTF {#5} = { "0587 } 36976
36977 { 36977
36978 \__text_change_case_store:e 36978
36979 { 36979
36980 \codepoint_generate:nn { "0535 } 36980
36981 { \__text_change_case_catcode:nn {#5} { "0535 } } 36981
36982 \codepoint_generate:nn { "057E } 36982
36983 { \__text_change_case_catcode:nn {#5} { "057E } } 36983
36984 } 36984
36985 \use:c { __text_change_case_next_ #2 :nnn } 36985
36986 {#2} {#3} {#4} 36986
36987 } 36987
36988 { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} } 36988
36989 } 36989
36990 \cs_new:cpn { __text_change_case_upper_hy-x-yiwn:nnnnn } #1#2#3#4#5 36990
36991 { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} } 36991
36992 \cs_new_eq:cc { __text_change_case_title_hy-x-yiwn:nnnnn } 36992
36993 { __text_change_case_upper_hy-x-yiwn:nnnnn } 36993
36994 \cs_new:cpn { __text_change_case_lower_la-x-medieval:nnnnn } #1#2#3#4#5 36994
36995 { 36995
36996 \__text_codepoint_compare:nNnTF {#5} = { "0056 } 36996
36997 { 36997
36998 \__text_change_case_store:e 36998
36999 { \char_generate:nn { "0075 } { \__text_char_catcode:N #5 } } 36999
37000 \use:c { __text_change_case_next_ #2 :nnn } 37000
37001 {#2} {#3} {#4} 37001
37002 } 37002
```

```
37003 { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} } 37003
37004 } 37004
37005 \cs_new:cpn { \__text_change_case_upper_la-x-medieval:nnnnn } #1#2#3#4#5 37005
37006 { 37006
37007 \__text_codepoint_compare:nNnTF {#5} = { "0075 } 37007
37008 { 37008
37009 \__text_change_case_store:e 37009
37010 { \char_generate:nn { "0056 } { \__text_char_catcode:N #5 } } 37010
37011 \use:c { \__text_change_case_next_ #2 :nnn } 37011
37012 {#2} {#3} {#4} 37012
37013 } 37013
37014 { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} } 37014
37015 } 37015
37016 \cs_new:Npn \__text_change_case_lower_lt:nnnnn #1#2#3#4#5 37016
37017 { 37017
37018 \exp_args:Ne \__text_change_case_lower_lt_auxi:nnnnn 37018
37019 { 37019
37020 \int_case:nn { \__text_codepoint_from_chars:Nw #5 } 37020
37021 { 37021
37022 { "00CC } { "0300 } 37022
37023 { "00CD } { "0301 } 37023
37024 { "0128 } { "0303 } 37024
37025 } 37025
37026 } 37026
37027 {#2} {#3} {#4} {#5} 37027
37028 } 37028
37029 \cs_new:Npn \__text_change_case_lower_lt_auxi:nnnnn #1#2#3#4#5 37029
37030 { 37030
37031 \tl_if_blank:nTF {#1} 37031
37032 { 37032
37033 \exp_args:Ne \__text_change_case_lower_lt_auxii:nnnnn 37033
37034 { 37034
37035 \int_case:nn { \__text_codepoint_from_chars:Nw #5 } 37035
37036 { 37036
37037 { "0049 } { "0069 } 37037
37038 { "004A } { "006A } 37038
37039 { "012E } { "012F } 37039
37040 } 37040
37041 } 37041
37042 {#2} {#3} {#4} {#5} 37042
37043 } 37043
37044 { 37044
37045 \__text_change_case_store:e 37045
37046 { 37046
37047 \codepoint_generate:nn { "0069 } 37047
37048 { \__text_change_case_catcode:nn {#5} { "0069 } } 37048
```

```
37049         \codepoint_generate:nn { "0307 } 37049
37050         { \__text_change_case_catcode:nn {#5} { "0307 } } 37050
37051     \codepoint_generate:nn {#1} 37051
37052     { \__text_change_case_catcode:nn {#5} {#1} } 37052
37053 } 37053
37054 \__text_change_case_loop:nnnw {#2} {#3} {#4} 37054
37055 } 37055
37056 } 37056
37057 \cs_new:Npn \__text_change_case_lower_lt_auxii:nnnnn #1#2#3#4#5 37057
37058 { 37058
37059     \tl_if_blank:nTF {#1} 37059
37060     { \__text_change_case_codepoint:nnnnn {#2} {#2} {#3} {#4} {#5} } 37060
37061     { 37061
37062         \__text_change_case_store:e 37062
37063         { 37063
37064             \codepoint_generate:nn {#1} 37064
37065             { \__text_change_case_catcode:nn {#5} {#1} } 37065
37066         } 37066
37067         \__text_change_case_lower_lt:nnnw {#2} {#3} {#4} 37067
37068     } 37068
37069 } 37069
37070 \cs_new:Npn \__text_change_case_lower_lt:nnnw #1#2#3#4 \q__text_recursion_stop 37070
37071 { 37071
37072     \tl_if_head_is_N_type:nTF {#4} 37072
37073     { \__text_change_case_lower_lt:nnnN } 37073
37074     { \__text_change_case_loop:nnnw } 37074
37075     {#1} {#2} {#3} #4 \q__text_recursion_stop 37075
37076 } 37076
37077 \cs_new:Npn \__text_change_case_lower_lt:nnnN #1#2#3#4 37077
37078 { 37078
37079     \__text_codepoint_process:nN 37079
37080     { \__text_change_case_lower_lt:nnnn {#1} {#2} {#3} } #4 37080
37081 } 37081
37082 \cs_new:Npn \__text_change_case_lower_lt:nnnn #1#2#3#4 37082
37083 { 37083
37084     \bool_lazy_and:nnT 37084
37085     { 37085
37086         \bool_lazy_or_p:nn 37086
37087         { ! \tl_if_single_p:n {#4} } 37087
37088         { ! \token_if_cs_p:N #4 } 37088
37089     } 37089
37090     { 37090
37091         \bool_lazy_any_p:n 37091
37092         { 37092
37093             { \__text_codepoint_compare_p:nNn {#4} = { "0300 } } 37093
37094             { \__text_codepoint_compare_p:nNn {#4} = { "0301 } } 37094
```

```
37095 { \_text_codepoint_compare_p:nNn {#4} = { "0303 } } 37095
37096 } 37096
37097 } 37097
37098 { 37098
37099 \_text_change_case_store:e 37099
37100 { 37100
37101 \codepoint_generate:nn { "0307 } 37101
37102 { \_text_change_case_catcode:nn {#4} { "0307 } } 37102
37103 } 37103
37104 } 37104
37105 \_text_change_case_loop:nnnw {#1} {#2} {#3} #4 37105
37106 } 37106
37107 \cs_new:Npn \_text_change_case_upper_lt:nnnnn #1#2#3#4#5 37107
37108 { 37108
37109 \exp_args:Ne \_text_change_case_upper_lt_aux:nnnnn 37109
37110 { 37110
37111 \int_case:nn { \_text_codepoint_from_chars:Nw #5 } 37111
37112 { 37112
37113 { "0069 } { "0049 } 37113
37114 { "006A } { "004A } 37114
37115 { "012F } { "012E } 37115
37116 } 37116
37117 } 37117
37118 {#2} {#3} {#4} {#5} 37118
37119 } 37119
37120 \cs_new:Npn \_text_change_case_upper_lt_aux:nnnnn #1#2#3#4#5 37120
37121 { 37121
37122 \tl_if_blank:nTF {#1} 37122
37123 { \_text_change_case_codepoint:nnnnn { upper } {#2} {#3} {#4} {#5} } 37123
37124 { 37124
37125 \_text_change_case_store:e 37125
37126 { 37126
37127 \codepoint_generate:nn {#1} 37127
37128 { \_text_change_case_catcode:nn {#5} {#1} } 37128
37129 } 37129
37130 \_text_change_case_upper_lt:nnnw {#2} {#3} {#4} 37130
37131 } 37131
37132 } 37132
37133 \cs_new:Npn \_text_change_case_upper_lt:nnnw #1#2#3#4 \q_text_recursion_stop 37133
37134 { 37134
37135 \tl_if_head_is_N_type:nTF {#4} 37135
37136 { \_text_change_case_upper_lt:nnnN } 37136
37137 { \use:c { \_text_change_case_next_ #1 :nnn } } 37137
37138 {#1} {#2} {#3} #4 \q_text_recursion_stop 37138
37139 } 37139
37140 \cs_new:Npn \_text_change_case_upper_lt:nnnN #1#2#3#4 37140
```

```
37141 { 37141
37142 \__text_codepoint_process:nN 37142
37143 { \__text_change_case_upper_lt:nnnn {#1} {#2} {#3} } #4 37143
37144 } 37144
37145 \cs_new:Npn \__text_change_case_upper_lt:nnnn #1#2#3#4 37145
37146 { 37146
37147 \bool_lazy_and:nnTF 37147
37148 { 37148
37149 \bool_lazy_or_p:nn 37149
37150 { ! \tl_if_single_p:n {#4} } 37150
37151 { ! \token_if_cs_p:N #4 } 37151
37152 } 37152
37153 { \__text_codepoint_compare_p:nNn {#4} = { "0307 } } 37153
37154 { \use:c { __text_change_case_next_ #1 :nnn } {#1} {#2} {#3} } 37154
37155 { \use:c { __text_change_case_next_ #1 :nnn } {#1} {#2} {#3} #4 } 37155
37156 } 37156
37157 \cs_new:Npn \__text_change_case_title_nl:nnnnn #1#2#3#4#5 37157
37158 { 37158
37159 \tl_if_single:nTF {#5} 37159
37160 { \__text_change_case_title_nl_aux:nnnnn } 37160
37161 { \__text_change_case_codepoint:nnnnn } 37161
37162 {#1} {#2} {#3} {#4} {#5} 37162
37163 } 37163
37164 \cs_new:Npn \__text_change_case_title_nl_aux:nnnnn #1#2#3#4#5 37164
37165 { 37165
37166 \bool_lazy_or:nnTF 37166
37167 { \int_compare_p:nNn {`#5} = { "0049 } } 37167
37168 { \int_compare_p:nNn {`#5} = { "0069 } } 37168
37169 { 37169
37170 \__text_change_case_store:e 37170
37171 { \char_generate:nn { "0049 } { \__text_char_catcode:N #5 } } 37171
37172 \__text_change_case_title_nl:nnnw {#2} {#3} {#4} 37172
37173 } 37173
37174 { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} } 37174
37175 } 37175
37176 \cs_new:Npn \__text_change_case_title_nl:nnnw #1#2#3#4 \q__text_recursion_stop 37176
37177 { 37177
37178 \tl_if_head_is_N_type:nTF {#4} 37178
37179 { \__text_change_case_title_nl:nnnN } 37179
37180 { \use:c { __text_change_case_next_ #1 :nnn } } 37180
37181 {#1} {#2} {#3} #4 \q__text_recursion_stop 37181
37182 } 37182
37183 \cs_new:Npn \__text_change_case_title_nl:nnnN #1#2#3#4 37183
37184 { 37184
37185 \bool_lazy_and:nnTF 37185
37186 { ! \token_if_cs_p:N #4 } 37186
```

```
37187 { 37187
37188 \bool_lazy_or_p:nn 37188
37189 { \int_compare_p:nNn {`#4} = { "004A } } 37189
37190 { \int_compare_p:nNn {`#4} = { "006A } } 37190
37191 } 37191
37192 { 37192
37193 \__text_change_case_store:e 37193
37194 { \char_generate:nn { "004A } { \__text_char_catcode:N #4 } } 37194
37195 \use:c { \__text_change_case_next_ #1 :nnn } {#1} {#2} {#3} 37195
37196 } 37196
37197 { \use:c { \__text_change_case_next_ #1 :nnn } {#1} {#2} {#3} #4 } 37197
37198 } 37198
37199 \cs_new:Npn \__text_change_case_lower_tr:nnnnn #1#2#3#4#5 37199
37200 { 37200
37201 \__text_codepoint_compare:nNnTF {#5} = { "0049 } 37201
37202 { \__text_change_case_lower_tr:nnnNw {#1} {#3} {#4} #5 } 37202
37203 { 37203
37204 \__text_codepoint_compare:nNnTF {#5} = { "0130 } 37204
37205 { 37205
37206 \__text_change_case_store:e 37206
37207 { 37207
37208 \codepoint_generate:nn { "0069 } 37208
37209 { \__text_change_case_catcode:nn {#5} { "0069 } } 37209
37210 } 37210
37211 \__text_change_case_loop:nnnw {#1} {#3} {#4} 37211
37212 } 37212
37213 { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} } 37213
37214 } 37214
37215 } 37215
37216 \cs_new:Npn \__text_change_case_lower_tr:nnnNw #1#2#3#4#5 \q__text_recursion_stop 37216
37217 { 37217
37218 \tl_if_head_is_N_type:nTF {#5} 37218
37219 { \__text_change_case_lower_tr:NnnnN #4 {#1} {#2} {#3} } 37219
37220 { 37220
37221 \__text_change_case_store:e 37221
37222 { 37222
37223 \codepoint_generate:nn { "0131 } 37223
37224 { \__text_change_case_catcode:nn {#4} { "0131 } } 37224
37225 } 37225
37226 \__text_change_case_loop:nnnw {#1} {#2} {#3} 37226
37227 } 37227
37228 #5 \q__text_recursion_stop 37228
37229 } 37229
37230 \cs_new:Npn \__text_change_case_lower_tr:NnnnN #1#2#3#4#5 37230
37231 { 37231
37232 \__text_codepoint_process:nN 37232
```

```
37233 { \__text_change_case_lower_tr:Nnnnn #1 {#2} {#3} {#4} } #5 37233
37234 } 37234
37235 \cs_new:Npn \__text_change_case_lower_tr:Nnnnn #1#2#3#4#5 37235
37236 { 37236
37237 \bool_lazy_or:nnTF 37237
37238 { 37238
37239 \bool_lazy_and_p:nn 37239
37240 { \tl_if_single_p:n {#5} } 37240
37241 { \token_if_cs_p:N #5 } 37241
37242 } 37242
37243 { ! \__text_codepoint_compare_p:nNn {#5} = { "0307 } } 37243
37244 { 37244
37245 \__text_change_case_store:e 37245
37246 { 37246
37247 \codepoint_generate:nn { "0131 } 37247
37248 { \__text_change_case_catcode:nn {#1} { "0131 } } 37248
37249 } 37249
37250 \__text_change_case_loop:nnnw {#2} {#3} {#4} #5 37250
37251 } 37251
37252 { 37252
37253 \__text_change_case_store:e 37253
37254 { 37254
37255 \codepoint_generate:nn { "0069 } 37255
37256 { \__text_change_case_catcode:nn {#1} { "0069 } } 37256
37257 } 37257
37258 \__text_change_case_loop:nnnw {#2} {#3} {#4} 37258
37259 } 37259
37260 } 37260
37261 \cs_new:Npn \__text_change_case_upper_tr:nnnnn #1#2#3#4#5 37261
37262 { 37262
37263 \__text_codepoint_compare:nNnTF {#5} = { "0069 } 37263
37264 { 37264
37265 \__text_change_case_store:e 37265
37266 { 37266
37267 \codepoint_generate:nn { "0130 } 37267
37268 { \__text_change_case_catcode:nn {#5} { "0130 } } 37268
37269 } 37269
37270 \use:c { __text_change_case_next_ #2 :nnn } {#2} {#3} {#4} 37270
37271 } 37271
37272 { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} } 37272
37273 } 37273
37274 \cs_new_eq:NN \__text_change_case_lower_az:nnnnn 37274
37275 \__text_change_case_lower_tr:nnnnn 37275
37276 \cs_new_eq:NN \__text_change_case_upper_az:nnnnn 37276
37277 \__text_change_case_upper_tr:nnnnn 37277
37278 \group_begin: 37278
```



```
37279 \cs_set_protected:Npn \__text_change_case_setup:NN #1#2 37279
37280 { 37280
37281 \quark_if_recursion_tail_stop:N #1 37281
37282 \tl_const:cn { c__text_lowercase_ \token_to_str:N #1 _tl } 37282
37283 { #2 } 37283
37284 \tl_const:cn { c__text_uppercase_ \token_to_str:N #2 _tl } 37284
37285 { #1 } 37285
37286 \__text_change_case_setup:NN 37286
37287 } 37287
37288 \__text_change_case_setup:NN 37288
37289 \AA \aa 37289
37290 \AE \ae 37290
37291 \DH \dh 37291
37292 \DJ \dj 37292
37293 \IJ \ij 37293
37294 \L \l 37294
37295 \NG \ng 37295
37296 \O \o 37296
37297 \OE \oe 37297
37298 \SS \ss 37298
37299 \TH \th 37299
37300 \q_recursion_tail ? 37300
37301 \q_recursion_stop 37301
37302 \tl_const:cn { c__text_uppercase_ \token_to_str:N \i _tl } { I } 37302
37303 \tl_const:cn { c__text_uppercase_ \token_to_str:N \j _tl } { J } 37303
37304 \group_end: 37304
37305 \tl_if_exist:NT \@expl@finalise@setup@@ 37305
37306 { 37306
37307 \tl_gput_right:Nn \@expl@finalise@setup@@ 37307
37308 { 37308
37309 \tl_gput_right:Nn \@kernel@after@begindocument 37309
37310 { 37310
37311 \group_begin: 37311
37312 \cs_set_protected:Npn \__text_change_case_setup:Nn #1#2 37312
37313 { 37313
37314 \quark_if_recursion_tail_stop:N #1 37314
37315 \tl_if_single_token:nT {#2} 37315
37316 { 37316
37317 \cs_if_exist:cF 37317
37318 { c__text_uppercase_ \token_to_str:N #1 _tl } 37318
37319 { 37319
37320 \tl_const:cn 37320
37321 { c__text_uppercase_ \token_to_str:N #1 _tl } 37321
37322 { #2 } 37322
37323 } 37323
37324 \cs_if_exist:cF 37324
```

```
37325         { c__text_lowercase_ \token_to_str:N #2 _t1 } 37325
37326     { 37326
37327         \tl_const:cn 37327
37328         { c__text_lowercase_ \token_to_str:N #2 _t1 } 37328
37329         { #1 } 37329
37330     } 37330
37331 } 37331
37332 \__text_change_case_setup:Nn 37332
37333 } 37333
37334 \exp_after:wN \__text_change_case_setup:Nn \@uclclist 37334
37335 \q_recursion_tail ? 37335
37336 \q_recursion_stop 37336
37337 \group_end: 37337
37338 } 37338
37339 } 37339
37340 } 37340
37341 \sys_if_engine_opentype:F 37341
37342 { 37342
37343     \text_declare_uppercase_mapping:nn { "01F0 } { \v { J } } 37343
37344 } 37344
37345 %% File: l3text-map.dtx 37345
37346 \cs_new:Npn \__text_map_function:nnN #1#2#3 37346
37347 { 37347
37348     \__text_map_loop:Nnnw #3 {#2} { } #1 37348
37349     \q__text_recursion_tail \q__text_recursion_stop 37349
37350     \prg_break_point:Nn \text_map_break: { } 37350
37351 } 37351
37352 \cs_generate_variant:Nn \__text_map_function:nnN { e } 37352
37353 \cs_new:Npn \__text_map_loop:Nnnw #1#2#3#4 \q__text_recursion_stop 37353
37354 { 37354
37355     \tl_if_head_is_N_type:nTF {#4} 37355
37356     { \__text_map_N_type:NnnN } 37356
37357     { 37357
37358         \tl_if_head_is_group:nTF {#4} 37358
37359         { \__text_map_group:Nnnn } 37359
37360         { \__text_map_space:Nnnw } 37360
37361     } 37361
37362     #1 {#2} {#3} #4 \q__text_recursion_stop 37362
37363 } 37363
37364 \cs_new:Npn \__text_map_group:Nnnn #1#2#3#4 37364
37365 { 37365
37366     \__text_map_output:Nn #1 {#3} 37366
37367     { 37367
37368         \__text_map_loop:Nnnw #1 {#2} { } #4 37368
37369         \q__text_recursion_tail \q__text_recursion_stop 37369
37370         \prg_break_point:Nn \text_map_break: { } 37370
```

```
37371     }
37372     \__text_map_loop:Nnnw #1 {#2} { }
37373 }
37374 \use:e
37375 { \cs_new:Npn \exp_not:N \__text_map_space:Nnnw #1#2#3 \c_space_tl }
37376 {
37377     \__text_map_output:Nn #1 {#3}
37378     #1 { ~ }
37379     \__text_map_loop:Nnnw #1 {#2} { }
37380 }
37381 \cs_new:Npn \__text_map_N_type:NnnN #1#2#3#4
37382 {
37383     \__text_if_q_recursion_tail_stop_do:Nn #4
37384     {
37385         \__text_map_output:Nn #1 {#3}
37386         \text_map_break:
37387     }
37388     \token_if_cs:NTF #4
37389     {
37390         \__text_map_output:Nn #1 {#3}
37391         #1 {#4}
37392         \__text_map_loop:Nnnw #1 {#2} { }
37393     }
37394     {
37395         \__text_codepoint_process:nN
37396         { \__text_map_codepoint:Nnnn #1 {#2} {#3} } #4
37397     }
37398 }
37399 \cs_new:Npn \__text_map_codepoint:Nnnn #1#2#3#4
37400 {
37401     \__text_codepoint_compare:nNnTF {#4} = { "000D }
37402     {
37403         \__text_map_output:Nn #1 {#3}
37404         \__text_map_CR:Nnnw #1 {#2} {#4}
37405     }
37406     {
37407         \__text_codepoint_compare:nNnTF {#4} = { "200D }
37408         { \__text_map_loop:Nnnw #1 {#2} {#3#4} }
37409         { \__text_map_class:Nnnn #1 {#2} {#3} {#4} }
37410     }
37411 }
37412 \cs_new:Npn \__text_map_CR:Nnnw #1#2#3#4 \q__text_recursion_stop
37413 {
37414     \tl_if_head_is_N_type:nTF {#4}
37415     { \__text_map_CR:NnnN #1 {#2} {#3} }
37416     {
```

```
37417         #1 {#3}
37418         \__text_map_loop:Nnnw #1 {#2} { }
37419     }
37420     #4 \q__text_recursion_stop
37421 }
37422 \cs_new:Npn \__text_map_CR:NnnN #1#2#3#4
37423 {
37424     \__text_if_q_recursion_tail_stop_do:Nn #4
37425     {
37426         #1 {#3}
37427         \text_map_break:
37428     }
37429     \bool_lazy_and:nnTF
37430     { ! \token_if_cs_p:N #4 }
37431     { \int_compare_p:nNn {`#4} = { "000A } }
37432     {
37433         \__text_map_output:Nn #1 {#3#4}
37434         \__text_map_loop:Nnnw #1 {#2} { }
37435     }
37436     { \__text_map_loop:Nnnw #1 {#2} { } #3 }
37437 }
37438 \cs_new:Npn \__text_map_class:Nnnn #1#2#3#4
37439 {
37440     \exp_args:Nnnne \__text_map_class:Nnnnn #1 {#2} {#3} {#4}
37441     {
37442         \use:c { __kernel_codepoint_to_ #2 _class:n }
37443         { \__text_codepoint_from_chars:Nw #4 }
37444     }
37445 }
37446 \cs_new:Npn \__text_map_class:Nnnnn #1#2#3#4#5
37447 {
37448     \cs_if_exist_use:cF { __text_map_ #5 :Nnnn }
37449     { \__text_map_Other:Nnnn }
37450     #1 {#2} {#3} {#4}
37451 }
37452 \cs_new:Npn \__text_map_lookahead:Nnnnnw #1#2#3#4#5#6 \q__text_recursion_stop
37453 {
37454     \tl_if_head_is_N_type:nTF {#6}
37455     { \__text_map_lookahead:NnnnnN #1 {#2} {#3} {#4} {#5} }
37456     { \__text_map_loop:Nnnw #1 {#2} {#3} #4 }
37457     #6 \q__text_recursion_stop
37458 }
37459 \cs_new:Npn \__text_map_lookahead:NnnnnN #1#2#3#4#5#6
37460 {
37461     \__text_if_q_recursion_tail_stop_do:Nn #6
37462     {
```

```
37463         #1 {#3} 37463
37464         \tl_if_blank:nF {#4} { #1 {#4} } 37464
37465     } 37465
37466     \token_if_cs:NTF #6 37466
37467     { 37467
37468         #1 {#3} 37468
37469         \__text_map_loop:Nnnw #1 {#2} { } #4 37469
37470     } 37470
37471     { \__text_codepoint_process:nN { #5 #1 {#2} {#3} {#4} } } 37471
37472     #6 37472
37473 } 37473
37474 \prg_new_conditional:Npnn \__text_map_if_ignorable:n #1 { TF } 37474
37475 { 37475
37476     \str_case:nnTF {#1} 37476
37477     { 37477
37478         { Extend } { } 37478
37479         { Format } { } 37479
37480         { ZWJ } { } 37480
37481     } 37481
37482     \prg_return_true: 37482
37483     \prg_return_false: 37483
37484 } 37484
37485 \cs_new:Npn \__text_map_output:Nn #1#2 37485
37486 { \tl_if_blank:nF {#2} { #1 {#2} } } 37486
37487 \cs_new:Npn \text_map_break: 37487
37488 { \prg_map_break:Nn \text_map_break: { } } 37488
37489 \cs_new:Npn \text_map_break:n 37489
37490 { \prg_map_break:Nn \text_map_break: } 37490
37491 \cs_new:Npn \__text_map_Control:Nnnn #1#2#3#4 37491
37492 { 37492
37493     \__text_map_output:Nn #1 {#3} 37493
37494     \__text_map_output:Nn #1 {#4} 37494
37495     \__text_map_loop:Nnnw #1 {#2} { } 37495
37496 } 37496
37497 \cs_new_eq:NN \__text_map_Newline:Nnnn \__text_map_Control:Nnnn 37497
37498 \cs_new:Npn \__text_map_Extend:Nnnn #1#2#3#4 37498
37499 { \__text_map_loop:Nnnw #1 {#2} {#3#4} } 37499
37500 \cs_new_eq:NN \__text_map_Format:Nnnn \__text_map_Extend:Nnnn 37500
37501 \cs_new_eq:NN \__text_map_SpacingMark:Nnnn \__text_map_Extend:Nnnn 37501
37502 \cs_new:Npn \__text_map_Other:Nnnn #1#2#3#4 37502
37503 { 37503
37504     \__text_map_output:Nn #1 {#3} 37504
37505     \__text_map_loop:Nnnw #1 {#2} {#4} 37505
37506 } 37506
37507 \cs_new:Npn \__text_map_Regional_Indicator:Nnnn #1#2#3#4 37507
37508 { 37508
```

```
37509 \__text_map_output:Nn #1 {#3} 37509
37510 \__text_map_lookahead:Nnnnw #1 {#2} {#4} { } 37510
37511 \__text_map_Regional_Indicator_aux:Nnnnn 37511
37512 } 37512
37513 \cs_new:Npn \__text_map_Regional_Indicator_aux:Nnnnn #1#2#3#4#5 37513
37514 { 37514
37515 \bool_lazy_or:nnTF 37515
37516 { \__text_codepoint_compare_p:nNn {#5} < { "1F1E6 } } 37516
37517 { \__text_codepoint_compare_p:nNn {#5} > { "1F1FF } } 37517
37518 { 37518
37519 \str_if_eq:nnTF {#2} { wordbreak } 37519
37520 { 37520
37521 \exp_args:Ne \__text_map_if_ignorable:nTF 37521
37522 { 37522
37523 \__kernel_codepoint_to_grapheme_class:n 37523
37524 { \__text_codepoint_from_chars:Nw #5 } 37524
37525 } 37525
37526 { 37526
37527 \__text_map_lookahead:Nnnnw #1 {#2} {#3#5} { } 37527
37528 \__text_map_Regional_Indicator_aux:Nnnnn 37528
37529 } 37529
37530 { \__text_map_loop:Nnnw #1 {#2} {#3} #5 } 37530
37531 } 37531
37532 { \__text_map_loop:Nnnw #1 {#2} {#3} #5 } 37532
37533 } 37533
37534 { \__text_map_loop:Nnnw #1 {#2} {#3#5} } 37534
37535 } 37535
37536 \cs_new:Npn \text_map_function:nN #1#2 37536
37537 { 37537
37538 \__text_map_function:enN { \text_expand:n {#1} } 37538
37539 { grapheme } #2 37539
37540 } 37540
37541 \cs_new:Npn \__text_map_Prepend:Nnnn #1#2#3#4 37541
37542 { 37542
37543 \__text_map_output:Nn #1 {#3} 37543
37544 \__text_map_lookahead:Nnnnw #1 { grapheme } {#4} { } 37544
37545 \__text_map_Prepend_aux:Nnnnn 37545
37546 } 37546
37547 \cs_new:Npn \__text_map_Prepend_aux:Nnnnn #1#2#3#4#5 37547
37548 { 37548
37549 \bool_lazy_or:nnTF 37549
37550 { \__text_codepoint_compare_p:nNn {#5} = { "000A } } 37550
37551 { \__text_codepoint_compare_p:nNn {#5} = { "000D } } 37551
37552 { 37552
37553 #1 {#3} 37553
37554 \__text_map_loop:Nnnw #1 { grapheme } {#5} 37554
```

```
37555 } 37555
37556 { \__text_map_Prepend:Nnn #1 {#3} {#5} } 37556
37557 } 37557
37558 \cs_new:Npn \__text_map_Prepend:Nnn #1#2#3 37558
37559 { 37559
37560 \str_if_eq:eeTF 37560
37561 { Control } 37561
37562 { 37562
37563 \__kernel_codepoint_to_grapheme_class:n 37563
37564 { \__text_codepoint_from_chars:Nw #3 } 37564
37565 } 37565
37566 { \__text_map_loop:Nnnw #1 { grapheme } {#2} #3 } 37566
37567 { \__text_map_loop:Nnnw #1 { grapheme } {#2#3} } 37567
37568 } 37568
37569 \cs_new:Npn \__text_map_L:Nnnn #1#2#3#4 37569
37570 { 37570
37571 \__text_map_output:Nn #1 {#3} 37571
37572 \__text_map_hangul:Nnnw 37572
37573 #1 {#4} { L ; V ; LV ; LVT } 37573
37574 } 37574
37575 \cs_new:Npn \__text_map_LV:Nnnn #1#2#3#4 37575
37576 { 37576
37577 \__text_map_output:Nn #1 {#3} 37577
37578 \__text_map_hangul:Nnnw 37578
37579 #1 {#4} { V ; T } 37579
37580 } 37580
37581 \cs_new_eq:NN \__text_map_V:Nnnn \__text_map_LV:Nnnn 37581
37582 \cs_new:Npn \__text_map_LVT:Nnnn #1#2#3#4 37582
37583 { 37583
37584 \__text_map_output:Nn #1 {#3} 37584
37585 \__text_map_hangul:Nnnw 37585
37586 #1 {#4} { T } 37586
37587 } 37587
37588 \cs_new_eq:NN \__text_map_T:Nnnn \__text_map_LVT:Nnnn 37588
37589 \cs_new:Npn \__text_map_hangul:Nnnw #1#2#3#4 \q__text_recursion_stop 37589
37590 { 37590
37591 \tl_if_head_is_N_type:nTF {#4} 37591
37592 { \__text_map_hangul:NnnN #1 {#2} {#3} } 37592
37593 { 37593
37594 #1 {#2} 37594
37595 \__text_map_loop:Nnnw #1 { grapheme } { } 37595
37596 } 37596
37597 #4 \q__text_recursion_stop 37597
37598 } 37598
37599 \cs_new:Npn \__text_map_hangul:NnnN #1#2#3#4 37599
37600 { 37600
```



```

37601 \__text_if_q_recursion_tail_stop_do:Nn #4
37602 {
37603     #1 {#2}
37604     \text_map_break:
37605 }
37606 \token_if_cs:NTF #4
37607 {
37608     #1 {#2}
37609     \__text_map_loop:Nnnw #1 { grapheme } { }
37610 }
37611 {
37612     \__text_codepoint_process:nN
37613     { \__text_map_hangul:Nnnn #1 {#2} {#3} } #4
37614 }
37615 }
37616 \exp_args_generate:n { Nnne }
37617 \cs_new:Npn \__text_map_hangul:Nnnn #1#2#3#4
37618 {
37619     \exp_args:NNnne \__text_map_hangul_aux:Nnnnw #1 {#2} {#4}
37620     {
37621         \__kernel_codepoint_to_grapheme_class:n
37622         { \__text_codepoint_from_chars:Nw #4 }
37623     }
37624     #3 ; \q_recursion_tail ; \q_recursion_stop
37625 }
37626 \cs_new:Npn \__text_map_hangul_aux:Nnnnw #1#2#3#4#5 ;
37627 {
37628     \quark_if_recursion_tail_stop_do:nn {#5}
37629     { \__text_map_loop:Nnnw #1 { grapheme } {#2} #3 }
37630     \__text_map_hangul:Nnnnnw #1 {#2} {#3} {#4} {#5}
37631 }
37632 \cs_generate_variant:Nn \__text_map_hangul_aux:Nnnnw { Nnne }
37633 \cs_new:Npn \__text_map_hangul:Nnnnnw #1#2#3#4#5#6 \q_recursion_stop
37634 {
37635     \str_if_eq:nnTF {#4} {#5}
37636     { \use:c { __text_map_hangul_ #5 :Nnn } #1 {#2} {#3} }
37637     { \__text_map_hangul_next:Nnnnn #1 {#2} {#3} {#4} {#6} }
37638 }
37639 \cs_new:Npn \__text_map_hangul_next:Nnnnn #1#2#3#4#5
37640 { \__text_map_hangul_aux:Nnnnw #1 {#2} {#3} {#4} #5 \q_recursion_stop }
37641 \cs_new:Npn \__text_map_hangul_end:nw #1#2 \q_text_recursion_stop {#1}
37642 \cs_new:Npn \__text_map_hangul_L:Nnn #1#2#3
37643 {
37644     \__text_map_hangul:Nnnw
37645     #1 {#2#3} { L V { LV } { LVT } }
37646 }

```

```
37647 \cs_new:Npn \__text_map_hangul_LV:Nnn #1#2#3 37647
37648 { 37648
37649 \__text_map_hangul:Nnnw 37649
37650 #1 {#2#3} { VT } 37650
37651 } 37651
37652 \cs_new_eq:NN \__text_map_hangul_V:Nnn \__text_map_hangul_LV:Nnn 37652
37653 \cs_new:Npn \__text_map_hangul_LVT:Nnn #1#2#3 37653
37654 { 37654
37655 \__text_map_hangul:Nnnw 37655
37656 #1 {#2#3} { T } 37656
37657 } 37657
37658 \cs_new_eq:NN \__text_map_hangul_T:Nnn \__text_map_hangul_LVT:Nnn 37658
37659 \cs_new:Npn \text_words_map_function:nN #1#2 37659
37660 { 37660
37661 \__text_map_function:enN { \text_expand:n {#1} } 37661
37662 { wordbreak } #2 37662
37663 } 37663
37664 \cs_new:Npn \__text_map_collect:Nnnnn #1#2#3#4#5 37664
37665 { 37665
37666 \__text_map_lookahead:Nnnnnw #1 { wordbreak } {#2} { } 37666
37667 { \__text_map_collect_auxi:nnnNnnnn {#3} {#4} {#5} } 37667
37668 } 37668
37669 \cs_new:Npn \__text_map_collect_auxi:nnnNnnnn #1#2#3#4#5#6#7#8 37669
37670 { 37670
37671 \exp_args:Ne \__text_map_collect_auxii:nNnnnnnn 37671
37672 { 37672
37673 \__kernel_codepoint_to_wordbreak_class:n 37673
37674 { \__text_codepoint_from_chars:Nw #8 } 37674
37675 } 37675
37676 #4 {#6} {#1} {#2} {#3} {#8} 37676
37677 } 37677
37678 \cs_new:Npn \__text_map_collect_auxii:nNnnnnnn #1#2#3#4#5#6#7 37678
37679 { 37679
37680 \str_case:neTF {#1} 37680
37681 { 37681
37682 \tl_map_function:eN 37682
37683 { 37683
37684 #4 37684
37685 \str_if_eq:nnF {#4} { { WSegSpace } } { { ExtendNumLet } } 37685
37686 } 37686
37687 \__text_map_collect_auxiii:n 37687
37688 } 37688
37689 { 37689
37690 \cs_if_exist_use:cF { __text_map_ #1 :Nnnn } 37690
37691 { \__text_map_Other:Nnnn } 37691
37692 #2 { wordbreak } { } {#3#7} 37692
```

```
37693 }
37694 {
37695     \__text_map_if_ignorable:nTF {#1}
37696     { \__text_map_collect:Nnnnn #2 {#3#7} {#4} {#5} {#6} }
37697     {
37698         \str_case:neTF {#1}
37699         { \tl_map_function:nN {#5} \__text_map_collect_auxiii:n }
37700         {
37701             \__text_map_lookahead:Nnnnnw #2 { wordbreak } {#3} {#7}
37702             { \__text_map_collect_auxiv:nnNnnnn {#5} {#6} }
37703         }
37704     }
37705     \__text_map_output:Nn #2 {#3}
37706     \__text_map_loop:Nnnw #2 { wordbreak } { } #7
37707 }
37708 }
37709 }
37710 }
37711 \cs_new:Npn \__text_map_collect_auxiii:n #1
37712 { \exp_not:n { {#1} { } } }
37713 \cs_new:Npn \__text_map_collect_auxiv:nnNnnnn #1#2#3#4#5#6#7
37714 {
37715     \exp_args:Ne \__text_map_collect_auxv:nNnnnnn
37716     {
37717         \__kernel_codepoint_to_wordbreak_class:n
37718         { \__text_codepoint_from_chars:Nw #7 }
37719     }
37720     #3 {#5} {#6} {#1} {#2} {#7}
37721 }
37722 \cs_new:Npn \__text_map_collect_auxv:nNnnnnn #1#2#3#4#5#6#7
37723 {
37724     \str_case:neTF {#1}
37725     { \tl_map_function:nN {#6} \__text_map_collect_auxiii:n }
37726     { \use:c { \__text_map_ #1 :Nnnn } #2 { wordbreak } { } {#3#4#7} }
37727     {
37728         \__text_map_if_ignorable:nTF {#1}
37729         {
37730             \__text_map_lookahead:Nnnnnw #2 { wordbreak } {#3} {#4#7}
37731             { \__text_map_collect_auxiv:nnNnnnn {#5} {#6} }
37732         }
37733     }
37734     \__text_map_output:Nn #2 {#3}
37735     \__text_map_loop:Nnnw #2 { wordbreak } { } #4#7
37736 }
37737 }
37738 }
```

```
37739 \cs_new:Npn \__text_map_ALetter:Nnnn #1#2#3#4 37739
37740 { 37740
37741 \__text_map_output:Nn #1 {#3} 37741
37742 \__text_map_collect:Nnnnn #1 {#4} 37742
37743 { { ALetter } { Hebrew_Letter } { Numeric } } 37743
37744 { { MidLetter } { MidNumLet } { Single_Quote } } 37744
37745 { { ALetter } { Hebrew_Letter } } 37745
37746 } 37746
37747 \cs_new:Npn \__text_map_Hebrew_Letter:Nnnn #1#2#3#4 37747
37748 { 37748
37749 \__text_map_output:Nn #1 {#3} 37749
37750 \__text_map_collect:Nnnnn #1 {#4} 37750
37751 { { ALetter } { Hebrew_Letter } { Numeric } { Single_Quote } } 37751
37752 { { MidLetter } { MidNumLet } { Double_Quote } } 37752
37753 { { Hebrew_Letter } } 37753
37754 } 37754
37755 \cs_new:Npn \__text_map_Katakana:Nnnn #1#2#3#4 37755
37756 { 37756
37757 \__text_map_output:Nn #1 {#3} 37757
37758 \__text_map_collect:Nnnnn #1 {#4} { { Katakana } } { } { } 37758
37759 } 37759
37760 \cs_new:Npn \__text_map_Numeric:Nnnn #1#2#3#4 37760
37761 { 37761
37762 \__text_map_output:Nn #1 {#3} 37762
37763 \__text_map_collect:Nnnnn #1 {#4} 37763
37764 { { ALetter } { Hebrew_Letter } { Numeric } } 37764
37765 { { MidNum } { MidNumLet } { Single_Quote } } 37765
37766 { { Numeric } } 37766
37767 } 37767
37768 \cs_new:Npn \__text_map_WSegSpace:Nnnn #1#2#3#4 37768
37769 { 37769
37770 \__text_map_output:Nn #1 {#3} 37770
37771 \__text_map_collect:Nnnnn #1 {#4} { { WSegSpace } } { } { } 37771
37772 } 37772
37773 \cs_new:Npn \__text_map_ExtendNumLet:Nnnn #1#2#3#4 37773
37774 { 37774
37775 \__text_map_output:Nn #1 {#3} 37775
37776 \__text_map_lookahead:Nnnnnw #1 { wordbreak } {#4} { } 37776
37777 \__text_map_ExtendNumLet_auxi:Nnnnn 37777
37778 } 37778
37779 \cs_new:Npn \__text_map_ExtendNumLet_auxi:Nnnnn #1#2#3#4#5 37779
37780 { 37780
37781 \exp_args:Ne \__text_map_ExtendNumLet_auxii:nNnn 37781
37782 { 37782
37783 \__kernel_codepoint_to_wordbreak_class:n 37783
37784 { \__text_codepoint_from_chars:Nw #5 } 37784
```

```

37785 }
37786 #1 {#3} {#5}
37787 }
37788 \cs_new:Npn \__text_map_ExtendNumLet_auxii:nNnn #1#2#3#4
37789 {
37790   \str_case:nnTF {#1}
37791   {
37792     { ALetter } { }
37793     { Hebrew_Letter } { }
37794     { Numeric } { }
37795     { Katakana } { }
37796     { ExtendNumLet } { }
37797   }
37798   {
37799     \cs_if_exist_use:cF { __text_map_ #1 :Nnnn } % TEMP?
37800     { \__text_map_Other:Nnnn }
37801     #2 { wordbreak } { } {#3#4}
37802   }
37803   {
37804     \__text_map_if_ignorable:nTF {#1}
37805     {
37806       \__text_map_lookahead:Nnnnnw #2 { wordbreak } {#3#4} { }
37807       \__text_map_ExtendNumLet_auxi:Nnnnn
37808     }
37809     {
37810       \__text_map_output:Nn #2 {#3}
37811       \__text_map_loop:Nnnw #2 { wordbreak } { } #4
37812     }
37813   }
37814 }
37815 \cs_new_protected:Npn \text_map_inline:nn #1#2
37816 {
37817   \int_gincr:N \g__kernel_prg_map_int
37818   \cs_gset_protected:cpn
37819   { __text_map_ \int_use:N \g__kernel_prg_map_int :w } ##1 {#2}
37820   \exp_args:Nnc \text_map_function:nN {#1}
37821   { __text_map_ \int_use:N \g__kernel_prg_map_int :w }
37822   \prg_break_point:Nn \text_map_break:
37823   { \int_gdecr:N \g__kernel_prg_map_int }
37824 }
37825 \cs_new_protected:Npn \text_words_map_inline:nn #1#2
37826 {
37827   \int_gincr:N \g__kernel_prg_map_int
37828   \cs_gset_protected:cpn
37829   { __text_map_ \int_use:N \g__kernel_prg_map_int :w } ##1 {#2}
37830   \exp_args:Nnc \text_words_map_function:nN {#1}

```

```
37831 { __text_map_ \int_use:N \g__kernel_prg_map_int :w } 37831
37832 \prg_break_point:Nn \text_map_break: 37832
37833 { \int_gdecr:N \g__kernel_prg_map_int } 37833
37834 } 37834
37835 %% File: l3text-purify.dtx 37835
37836 \__kernel_quark_new_test:N \__text_if_recursion_tail_stop:N 37836
37837 \cs_new:Npn \text_purify:n #1 37837
37838 { 37838
37839 \__kernel_exp_not:w \exp_after:wN 37839
37840 { 37840
37841 \exp:w 37841
37842 \exp_args:Ne \__text_purify:n 37842
37843 { \text_expand:n {#1} } 37843
37844 } 37844
37845 } 37845
37846 \cs_new:Npn \__text_purify:n #1 37846
37847 { 37847
37848 \group_align_safe_begin: 37848
37849 \__text_purify_loop:w #1 37849
37850 \q__text_recursion_tail \q__text_recursion_stop 37850
37851 \__text_purify_result:n { } 37851
37852 } 37852
37853 \cs_new:Npn \__text_purify_store:n #1 37853
37854 { \__text_purify_store:nw {#1} } 37854
37855 \cs_new:Npn \__text_purify_store:nw #1#2 \__text_purify_result:n #3 37855
37856 { #2 \__text_purify_result:n { #3 #1 } } 37856
37857 \cs_new:Npn \__text_purify_end:w #1 \__text_purify_result:n #2 37857
37858 { 37858
37859 \group_align_safe_end: 37859
37860 \exp_end: 37860
37861 #2 37861
37862 } 37862
37863 \cs_new:Npn \__text_purify_loop:w #1 \q__text_recursion_stop 37863
37864 { 37864
37865 \tl_if_head_is_N_type:nTF {#1} 37865
37866 { \__text_purify_N_type:N } 37866
37867 { 37867
37868 \tl_if_head_is_group:nTF {#1} 37868
37869 { \__text_purify_group:n } 37869
37870 { \__text_purify_space:w } 37870
37871 } 37871
37872 #1 \q__text_recursion_stop 37872
37873 } 37873
37874 \cs_new:Npn \__text_purify_group:n #1 { \__text_purify_loop:w #1 } 37874
37875 \exp_last_unbraced:NNo \cs_new:Npn \__text_purify_space:w \c_space_tl 37875
37876 { 37876
```

```
37877 \__text_purify_store:n { ~ } 37877
37878 \__text_purify_loop:w 37878
37879 } 37879
37880 \cs_new:Npn \__text_purify_N_type:N #1 37880
37881 { 37881
37882 \__text_if_q_recursion_tail_stop_do:Nn #1 { \__text_purify_end:w } 37882
37883 \__text_purify_N_type_aux:N #1 37883
37884 } 37884
37885 \cs_new:Npn \__text_purify_N_type_aux:N #1 37885
37886 { 37886
37887 \exp_after:wN \__text_purify_math_search:NNN 37887
37888 \exp_after:wN #1 \l_text_math_delims_tl 37888
37889 \q__text_recursion_tail ? 37889
37890 \q__text_recursion_stop 37890
37891 } 37891
37892 \cs_new:Npn \__text_purify_math_search:NNN #1#2#3 37892
37893 { 37893
37894 \__text_if_q_recursion_tail_stop_do:Nn #2 37894
37895 { \__text_purify_math_cmd:N #1 } 37895
37896 \token_if_eq_meaning:NNTF #1 #2 37896
37897 { 37897
37898 \__text_use_i_delimit_by_q_recursion_stop:nw 37898
37899 { \__text_purify_math_start:NNw #2 #3 } 37899
37900 } 37900
37901 { \__text_purify_math_search:NNN #1 } 37901
37902 } 37902
37903 \cs_new:Npn \__text_purify_math_start:NNw #1#2#3 \q__text_recursion_stop 37903
37904 { 37904
37905 \__text_purify_math_loop:NNw #1#2#3 \q__text_recursion_stop 37905
37906 \__text_purify_math_result:n { } 37906
37907 } 37907
37908 \cs_new:Npn \__text_purify_math_store:n #1 37908
37909 { \__text_purify_math_store:nw {#1} } 37909
37910 \cs_new:Npn \__text_purify_math_store:nw #1#2 \__text_purify_math_result:n #3 37910
37911 { #2 \__text_purify_math_result:n { #3 #1 } } 37911
37912 \cs_new:Npn \__text_purify_math_end:w #1 \__text_purify_math_result:n #2 37912
37913 { 37913
37914 \__text_purify_store:n { $ #2 $ } 37914
37915 \__text_purify_loop:w #1 37915
37916 } 37916
37917 \cs_new:Npn \__text_purify_math_stop:Nw #1 \__text_purify_math_result:n #2 37917
37918 { 37918
37919 \__text_purify_store:n {#1#2} 37919
37920 \__text_purify_end:w 37920
37921 } 37921
37922 \cs_new:Npn \__text_purify_math_loop:NNw #1#2#3 \q__text_recursion_stop 37922
```



```

37923 {
37924 \tl_if_head_is_N_type:nTF {#3}
37925 { \__text_purify_math_N_type:NNN }
37926 {
37927 \tl_if_head_is_group:nTF {#3}
37928 { \__text_purify_math_group:NNn }
37929 { \__text_purify_math_space:NNw }
37930 }
37931 #1#2#3 \q__text_recursion_stop
37932 }
37933 \cs_new:Npn \__text_purify_math_N_type:NNN #1#2#3
37934 {
37935 \__text_if_q_recursion_tail_stop_do:Nn #3
37936 { \__text_purify_math_stop:Nw #1 }
37937 \token_if_eq_meaning:NNTF #3 #2
37938 { \__text_purify_math_end:w }
37939 {
37940 \__text_purify_math_store:n {#3}
37941 \__text_purify_math_loop:NNw #1#2
37942 }
37943 }
37944 \cs_new:Npn \__text_purify_math_group:NNn #1#2#3
37945 {
37946 \__text_purify_math_store:n { {#3} }
37947 \__text_purify_math_loop:NNw #1#2
37948 }
37949 \exp_after:wN \cs_new:Npn \exp_after:wN \__text_purify_math_space:NNw
37950 \exp_after:wN # \exp_after:wN 1
37951 \exp_after:wN # \exp_after:wN 2 \c_space_tl
37952 {
37953 \__text_purify_math_store:n { ~ }
37954 \__text_purify_math_loop:NNw #1#2
37955 }
37956 \cs_new:Npn \__text_purify_math_cmd:N #1
37957 {
37958 \exp_after:wN \__text_purify_math_cmd:NN \exp_after:wN #1
37959 \l_text_math_arg_tl \q__text_recursion_tail \q__text_recursion_stop
37960 }
37961 \cs_new:Npn \__text_purify_math_cmd:NN #1#2
37962 {
37963 \__text_if_q_recursion_tail_stop_do:Nn #2
37964 { \__text_purify_replace:N #1 }
37965 \cs_if_eq:NNTF #2 #1
37966 {
37967 \__text_use_i_delimit_by_q_recursion_stop:nw
37968 { \__text_purify_math_cmd:n }

```

```
37969 } 37969
37970 { \__text_purify_math_cmd:NN #1 } 37970
37971 } 37971
37972 \cs_new:Npn \__text_purify_math_cmd:n #1 37972
37973 { \__text_purify_math_end:w \__text_purify_math_result:n {#1} } 37973
37974 \cs_new:Npn \__text_purify_replace:N #1 37974
37975 { 37975
37976 \bool_lazy_and:nnTF 37976
37977 { \cs_if_exist_p:c { l__text_purify_ \token_to_str:N #1 _tl } } 37977
37978 { 37978
37979 \bool_lazy_or_p:nn 37979
37980 { \token_if_cs_p:N #1 } 37980
37981 { \token_if_active_p:N #1 } 37981
37982 } 37982
37983 { 37983
37984 \exp_args:Nv \__text_purify_replace_auxi:n 37984
37985 { l__text_purify_ \token_to_str:N #1 _tl } 37985
37986 } 37986
37987 { 37987
37988 \bool_lazy_or:nnTF 37988
37989 { \token_if_group_begin_p:N #1 } 37989
37990 { \token_if_group_end_p:N #1 } 37990
37991 { \__text_purify_loop:w } 37991
37992 { 37992
37993 \exp_args:Ne \__text_purify_replace_auxii:n 37993
37994 { \__text_token_to_explicit:N #1 } 37994
37995 } 37995
37996 } 37996
37997 } 37997
37998 \cs_new:Npn \__text_purify_replace_auxi:n #1 { \__text_purify_loop:w #1 } 37998
37999 \cs_new:Npn \__text_purify_replace_auxii:n #1 37999
38000 { 38000
38001 \token_if_cs:NTF #1 38001
38002 { \__text_purify_expand:N #1 } 38002
38003 { 38003
38004 \__text_purify_store:n {#1} 38004
38005 \__text_purify_loop:w 38005
38006 } 38006
38007 } 38007
38008 \cs_new:Npn \__text_purify_expand:N #1 38008
38009 { 38009
38010 \str_if_eq:nnTF {#1} { \protect } 38010
38011 { \__text_purify_protect:N } 38011
38012 { \__text_purify_encoding:N #1 } 38012
38013 } 38013
38014 \cs_new:Npn \__text_purify_protect:N #1 38014
```

```
38015 { 38015
38016 \_text_if_q_recursion_tail_stop_do:Nn #1 { \_text_purify_end:w } 38016
38017 \_text_purify_loop:w 38017
38018 } 38018
38019 \cs_new:Npn \_text_purify_encoding:N #1 38019
38020 { 38020
38021 \bool_lazy_or:nnTF 38021
38022 { \cs_if_eq_p:NN #1 \@current@cmd } 38022
38023 { \cs_if_eq_p:NN #1 \@changed@cmd } 38023
38024 { \_text_purify_encoding_escape:NN } 38024
38025 { 38025
38026 \_text_if_expandable:NTF #1 38026
38027 { \exp_after:wN \_text_purify_loop:w #1 } 38027
38028 { \_text_purify_loop:w } 38028
38029 } 38029
38030 } 38030
38031 \cs_new:Npn \_text_purify_encoding_escape:NN #1#2 38031
38032 { 38032
38033 \_text_purify_store:n {#1} 38033
38034 \_text_purify_loop:w 38034
38035 } 38035
38036 \cs_new_protected:Npn \text_declare_purify_equivalent:Nn #1#2 38036
38037 { 38037
38038 \tl_clear_new:c { l__text_purify_ \token_to_str:N #1 _tl } 38038
38039 \tl_set:cn { l__text_purify_ \token_to_str:N #1 _tl } {#2} 38039
38040 } 38040
38041 \cs_generate_variant:Nn \text_declare_purify_equivalent:Nn { Ne } 38041
38042 \tl_map_inline:nn 38042
38043 { 38043
38044 \fontencoding 38044
38045 \fontfamily 38045
38046 \fontseries 38046
38047 \fontshape 38047
38048 } 38048
38049 { \text_declare_purify_equivalent:Nn #1 { \use_none:n } } 38049
38050 \text_declare_purify_equivalent:Nn \fontsize { \use_none:nn } 38050
38051 \text_declare_purify_equivalent:Nn \selectfont { } 38051
38052 \text_declare_purify_equivalent:Nn \usefont { \use_none:nnnn } 38052
38053 \exp_args:Nc \text_declare_purify_equivalent:Nn 38053
38054 { @protected@testopt } { \use_none:nnn } 38054
38055 \text_declare_purify_equivalent:Nn \begin { \use:c } 38055
38056 \text_declare_purify_equivalent:Nn \end { \_text_end_env:n } 38056
38057 \cs_new:Npn \_text_end_env:n #1 { \cs:w end #1 \cs_end: } 38057
38058 \text_declare_purify_equivalent:Nn \\\ { } 38058
38059 \tl_map_inline:nn 38059
38060 { \{ \} \# \$ \% \_ } 38060
```

38061	{ \text_declare_purify_equivalent:Ne #1 { \cs_to_str:N #1 } }	38061
38062	\text_declare_purify_equivalent:Nn \label { \use_none:n }	38062
38063	\group_begin:	38063
38064	\char_set_catcode_active:N \~	38064
38065	\use:n	38065
38066	{	38066
38067	\group_end:	38067
38068	\text_declare_purify_equivalent:Ne ~ { \c_space_tl }	38068
38069	}	38069
38070	\text_declare_purify_equivalent:Nn \nobreakspace { ~ }	38070
38071	\text_declare_purify_equivalent:Nn _ { ~ }	38071
38072	\text_declare_purify_equivalent:Nn \, { ~ }	38072
38073	\cs_set_protected:Npn __text_loop:Nn #1#2	38073
38074	{	38074
38075	\quark_if_recursion_tail_stop:N #1	38075
38076	\text_declare_purify_equivalent:Ne #1	38076
38077	{	38077
38078	\codepoint_generate:nn {"#2}	38078
38079	{ \char_value_catcode:n {"#2} }	38079
38080	}	38080
38081	__text_loop:Nn	38081
38082	}	38082
38083	__text_loop:Nn	38083
38084	\AA { 00C5 }	38084
38085	\AE { 00C6 }	38085
38086	\DH { 00D0 }	38086
38087	\DJ { 0110 }	38087
38088	\IJ { 0132 }	38088
38089	\L { 0141 }	38089
38090	\NG { 014A }	38090
38091	\O { 00D8 }	38091
38092	\OE { 0152 }	38092
38093	\TH { 00DE }	38093
38094	\aa { 00E5 }	38094
38095	\ae { 00E6 }	38095
38096	\dh { 00F0 }	38096
38097	\dj { 0111 }	38097
38098	\i { 0131 }	38098
38099	\j { 0237 }	38099
38100	\ij { 0132 }	38100
38101	\l { 0142 }	38101
38102	\ng { 014B }	38102
38103	\o { 00F8 }	38103
38104	\oe { 0153 }	38104
38105	\ss { 00DF }	38105
38106	\th { 00FE }	38106

```
38107 \q_recursion_tail ? 38107
38108 \q_recursion_stop 38108
38109 \text_declare_purify_equivalent:Nn \SS { SS } 38109
38110 \cs_new:Npn \__text_purify_accent:NN #1#2 38110
38111 { 38111
38112 \cs_if_exist:cTF 38112
38113 { c__text_purify_ \token_to_str:N #1 _ \token_to_str:N #2 _tl } 38113
38114 { 38114
38115 \exp_not:v 38115
38116 { c__text_purify_ \token_to_str:N #1 _ \token_to_str:N #2 _tl } 38116
38117 } 38117
38118 { 38118
38119 \exp_not:n {#2} 38119
38120 \exp_not:v { c__text_purify_ \token_to_str:N #1 _tl } 38120
38121 } 38121
38122 } 38122
38123 \tl_map_inline:nn { \` \' \^ \~ \= \u \. \" \r \H \v \d \c \k \b \t } 38123
38124 { \text_declare_purify_equivalent:Nn #1 { \__text_purify_accent:NN #1 } } 38124
38125 \group_begin: 38125
38126 \cs_set_protected:Npn \__text_loop:Nn #1#2 38126
38127 { 38127
38128 \quark_if_recursion_tail_stop:N #1 38128
38129 \tl_const:ce { c__text_purify_ \token_to_str:N #1 _tl } 38129
38130 { \codepoint_generate:nn {"#2} { \char_value_catcode:n { "#2 } } } 38130
38131 \__text_loop:Nn 38131
38132 } 38132
38133 \__text_loop:Nn 38133
38134 \` { 0300 } 38134
38135 \' { 0301 } 38135
38136 \^ { 0302 } 38136
38137 \~ { 0303 } 38137
38138 \= { 0304 } 38138
38139 \u { 0306 } 38139
38140 \. { 0307 } 38140
38141 \" { 0308 } 38141
38142 \r { 030A } 38142
38143 \H { 030B } 38143
38144 \v { 030C } 38144
38145 \d { 0323 } 38145
38146 \c { 0327 } 38146
38147 \k { 0328 } 38147
38148 \b { 0331 } 38148
38149 \t { 0361 } 38149
38150 \q_recursion_tail { } 38150
38151 \q_recursion_stop 38151
38152 \cs_set_protected:Npn \__text_loop:NNn #1#2#3 38152
```

38153	{	38153
38154	\quark_if_recursion_tail_stop:N #1	38154
38155	\tl_const:ce	38155
38156	{ c__text_purify_ \token_to_str:N #1 _ \token_to_str:N #2 _tl }	38156
38157	{ \codepoint_generate:nn {"#3"} { \char_value_catcode:n { "#3" } } }	38157
38158	__text_loop:NNn	38158
38159	}	38159
38160	__text_loop:NNn	38160
38161	\` A { 00C0 }	38161
38162	\' A { 00C1 }	38162
38163	\^ A { 00C2 }	38163
38164	\~ A { 00C3 }	38164
38165	\" A { 00C4 }	38165
38166	\r A { 00C5 }	38166
38167	\c C { 00C7 }	38167
38168	\` E { 00C8 }	38168
38169	\' E { 00C9 }	38169
38170	\^ E { 00CA }	38170
38171	\" E { 00CB }	38171
38172	\` I { 00CC }	38172
38173	\' I { 00CD }	38173
38174	\^ I { 00CE }	38174
38175	\" I { 00CF }	38175
38176	\~ N { 00D1 }	38176
38177	\` O { 00D2 }	38177
38178	\' O { 00D3 }	38178
38179	\^ O { 00D4 }	38179
38180	\~ O { 00D5 }	38180
38181	\" O { 00D6 }	38181
38182	\` U { 00D9 }	38182
38183	\' U { 00DA }	38183
38184	\^ U { 00DB }	38184
38185	\" U { 00DC }	38185
38186	\' Y { 00DD }	38186
38187	\` a { 00E0 }	38187
38188	\' a { 00E1 }	38188
38189	\^ a { 00E2 }	38189
38190	\~ a { 00E3 }	38190
38191	\" a { 00E4 }	38191
38192	\r a { 00E5 }	38192
38193	\c c { 00E7 }	38193
38194	\` e { 00E8 }	38194
38195	\' e { 00E9 }	38195
38196	\^ e { 00EA }	38196
38197	\" e { 00EB }	38197
38198	\` i { 00EC }	38198

38199	\` \i	{ 00EC }	38199
38200	\' i	{ 00ED }	38200
38201	\' \i	{ 00ED }	38201
38202	\^ i	{ 00EE }	38202
38203	\^ \i	{ 00EE }	38203
38204	\" i	{ 00EF }	38204
38205	\" \i	{ 00EF }	38205
38206	\~ n	{ 00F1 }	38206
38207	\` o	{ 00F2 }	38207
38208	\' o	{ 00F3 }	38208
38209	\^ o	{ 00F4 }	38209
38210	\~ o	{ 00F5 }	38210
38211	\" o	{ 00F6 }	38211
38212	\` u	{ 00F9 }	38212
38213	\' u	{ 00FA }	38213
38214	\^ u	{ 00FB }	38214
38215	\" u	{ 00FC }	38215
38216	\' y	{ 00FD }	38216
38217	\" y	{ 00FF }	38217
38218	\= A	{ 0100 }	38218
38219	\= a	{ 0101 }	38219
38220	\u A	{ 0102 }	38220
38221	\u a	{ 0103 }	38221
38222	\k A	{ 0104 }	38222
38223	\k a	{ 0105 }	38223
38224	\' C	{ 0106 }	38224
38225	\' c	{ 0107 }	38225
38226	\^ C	{ 0108 }	38226
38227	\^ c	{ 0109 }	38227
38228	\. C	{ 010A }	38228
38229	\. c	{ 010B }	38229
38230	\v C	{ 010C }	38230
38231	\v c	{ 010D }	38231
38232	\v D	{ 010E }	38232
38233	\v d	{ 010F }	38233
38234	\= E	{ 0112 }	38234
38235	\= e	{ 0113 }	38235
38236	\u E	{ 0114 }	38236
38237	\u e	{ 0115 }	38237
38238	\. E	{ 0116 }	38238
38239	\. e	{ 0117 }	38239
38240	\k E	{ 0118 }	38240
38241	\k e	{ 0119 }	38241
38242	\v E	{ 011A }	38242
38243	\v e	{ 011B }	38243
38244	\^ G	{ 011C }	38244

38245	\^ g	{ 011D }	38245
38246	\u G	{ 011E }	38246
38247	\u g	{ 011F }	38247
38248	\. G	{ 0120 }	38248
38249	\. g	{ 0121 }	38249
38250	\c G	{ 0122 }	38250
38251	\c g	{ 0123 }	38251
38252	\^ H	{ 0124 }	38252
38253	\^ h	{ 0125 }	38253
38254	\~ I	{ 0128 }	38254
38255	\~ i	{ 0129 }	38255
38256	\~ \i	{ 0129 }	38256
38257	\= I	{ 012A }	38257
38258	\= i	{ 012B }	38258
38259	\= \i	{ 012B }	38259
38260	\u I	{ 012C }	38260
38261	\u i	{ 012D }	38261
38262	\u \i	{ 012D }	38262
38263	\k I	{ 012E }	38263
38264	\k i	{ 012F }	38264
38265	\k \i	{ 012F }	38265
38266	\. I	{ 0130 }	38266
38267	\^ J	{ 0134 }	38267
38268	\^ j	{ 0135 }	38268
38269	\^ \j	{ 0135 }	38269
38270	\c K	{ 0136 }	38270
38271	\c k	{ 0137 }	38271
38272	\' L	{ 0139 }	38272
38273	\' l	{ 013A }	38273
38274	\c L	{ 013B }	38274
38275	\c l	{ 013C }	38275
38276	\v L	{ 013D }	38276
38277	\v l	{ 013E }	38277
38278	\. L	{ 013F }	38278
38279	\. l	{ 0140 }	38279
38280	\' N	{ 0143 }	38280
38281	\' n	{ 0144 }	38281
38282	\c N	{ 0145 }	38282
38283	\c n	{ 0146 }	38283
38284	\v N	{ 0147 }	38284
38285	\v n	{ 0148 }	38285
38286	\= O	{ 014C }	38286
38287	\= o	{ 014D }	38287
38288	\u O	{ 014E }	38288
38289	\u o	{ 014F }	38289
38290	\H O	{ 0150 }	38290

38291	\H o	{ 0151 }	38291
38292	\' R	{ 0154 }	38292
38293	\' r	{ 0155 }	38293
38294	\c R	{ 0156 }	38294
38295	\c r	{ 0157 }	38295
38296	\v R	{ 0158 }	38296
38297	\v r	{ 0159 }	38297
38298	\' S	{ 015A }	38298
38299	\' s	{ 015B }	38299
38300	\^ S	{ 015C }	38300
38301	\^ s	{ 015D }	38301
38302	\c S	{ 015E }	38302
38303	\c s	{ 015F }	38303
38304	\v S	{ 0160 }	38304
38305	\v s	{ 0161 }	38305
38306	\c T	{ 0162 }	38306
38307	\c t	{ 0163 }	38307
38308	\v T	{ 0164 }	38308
38309	\v t	{ 0165 }	38309
38310	\~ U	{ 0168 }	38310
38311	\~ u	{ 0169 }	38311
38312	\= U	{ 016A }	38312
38313	\= u	{ 016B }	38313
38314	\u U	{ 016C }	38314
38315	\u u	{ 016D }	38315
38316	\r U	{ 016E }	38316
38317	\r u	{ 016F }	38317
38318	\H U	{ 0170 }	38318
38319	\H u	{ 0171 }	38319
38320	\k U	{ 0172 }	38320
38321	\k u	{ 0173 }	38321
38322	\^ W	{ 0174 }	38322
38323	\^ w	{ 0175 }	38323
38324	\^ Y	{ 0176 }	38324
38325	\^ y	{ 0177 }	38325
38326	\" Y	{ 0178 }	38326
38327	\' Z	{ 0179 }	38327
38328	\' z	{ 017A }	38328
38329	\. Z	{ 017B }	38329
38330	\. z	{ 017C }	38330
38331	\v Z	{ 017D }	38331
38332	\v z	{ 017E }	38332
38333	\v A	{ 01CD }	38333
38334	\v a	{ 01CE }	38334
38335	\v I	{ 01CF }	38335
38336	\v \i	{ 01D0 }	38336

38337	\v i { 01D0 }	38337
38338	\v O { 01D1 }	38338
38339	\v o { 01D2 }	38339
38340	\v U { 01D3 }	38340
38341	\v u { 01D4 }	38341
38342	\v G { 01E6 }	38342
38343	\v g { 01E7 }	38343
38344	\v K { 01E8 }	38344
38345	\v k { 01E9 }	38345
38346	\k O { 01EA }	38346
38347	\k o { 01EB }	38347
38348	\v \j { 01F0 }	38348
38349	\v j { 01F0 }	38349
38350	\' G { 01F4 }	38350
38351	\' g { 01F5 }	38351
38352	\` N { 01F8 }	38352
38353	\` n { 01F9 }	38353
38354	\' \AE { 01FC }	38354
38355	\' \ae { 01FD }	38355
38356	\' \O { 01FE }	38356
38357	\' \o { 01FF }	38357
38358	\v H { 021E }	38358
38359	\v h { 021F }	38359
38360	\. A { 0226 }	38360
38361	\. a { 0227 }	38361
38362	\c E { 0228 }	38362
38363	\c e { 0229 }	38363
38364	\. O { 022E }	38364
38365	\. o { 022F }	38365
38366	\= Y { 0232 }	38366
38367	\= y { 0233 }	38367
38368	\q_recursion_tail ? { }	38368
38369	\q_recursion_stop	38369
38370	\group_end:	38370
38371	%% File: l3legacy.dtx	38371
38372	\prg_new_conditional:Npnn \legacy_if:n #1 { p , T , F , TF }	38372
38373	{	38373
38374	\exp_after:wN \reverse_if:N	38374
38375	\cs:w if#1 \cs_end:	38375
38376	\prg_return_false:	38376
38377	\else:	38377
38378	\prg_return_true:	38378
38379	\fi:	38379
38380	}	38380
38381	\cs_new_protected:Npn \legacy_if_set_true:n #1	38381
38382	{ \cs_set_eq:cN { if#1 } \if_true: }	38382

```

38383 \cs_new_protected:Npn \legacy_if_set_false:n #1
38384 { \cs_set_eq:cN { if#1 } \if_false: }
38385 \cs_new_protected:Npn \legacy_if_gset_true:n #1
38386 { \cs_gset_eq:cN { if#1 } \if_true: }
38387 \cs_new_protected:Npn \legacy_if_gset_false:n #1
38388 { \cs_gset_eq:cN { if#1 } \if_false: }
38389 \cs_new_protected:Npn \legacy_if_set:nn #1#2
38390 {
38391     \bool_if:nTF {#2} \legacy_if_set_true:n \legacy_if_set_false:n
38392     {#1}
38393 }
38394 \cs_new_protected:Npn \legacy_if_gset:nn #1#2
38395 {
38396     \bool_if:nTF {#2} \legacy_if_gset_true:n \legacy_if_gset_false:n
38397     {#1}
38398 }
38399 %% File: l3deprecation.dtx
38400 \cs_new_protected:Npn \__kernel_patch_deprecation:nnNNpn #1#2#3#4#5#
38401 { \__deprecation_patch_aux:nnNNnn {#1} {#2} #3 #4 {#5} }
38402 \cs_new_protected:Npn \__deprecation_patch_aux:nnNNnn #1#2#3#4#5#6
38403 {
38404     \__kernel_deprecation_code:nn
38405     {
38406         \tex_let:D #4 \scan_stop:
38407         \__kernel_deprecation_error:Nnn #4 {#2} {#1}
38408     }
38409     { \tex_let:D #4 \scan_stop: }
38410 \cs_if_eq:NNTF #3 \cs_gset_protected:Npn
38411 { \__deprecation_warn_once:nnNnn {#1} {#2} #4 {#5} {#6} }
38412 { \__deprecation_patch_aux:Nn #3 { #4 #5 {#6} } }
38413 }
38414 \cs_new_protected:Npn \__deprecation_warn_once:nnNnn #1#2#3#4#5
38415 {
38416     \cs_gset_protected:Npe #3
38417     {
38418         \__kernel_if_debug:TF
38419         {
38420             \exp_not:N \msg_warning:nneee
38421             { deprecation } { deprecated-command }
38422             {#1}
38423             { \token_to_str:N #3 }
38424             { \tl_to_str:n {#2} }
38425         }
38426         { }
38427         \exp_not:n { \cs_gset_protected:Npn #3 #4 {#5} }
38428         \exp_not:N #3

```

```

38429 }
38430 \__kernel_deprecation_code:nn { }
38431 { \cs_set_protected:Npn #3 #4 {#5} }
38432 }
38433 \cs_new_protected:Npn \__deprecation_patch_aux:Nn #1#2
38434 {
38435 #1 #2
38436 \cs_if_eq:NNTF #1 \cs_gset_protected:Npn
38437 { \__kernel_deprecation_code:nn { } { \cs_set_protected:Npn #2 } }
38438 { \__kernel_deprecation_code:nn { } { \cs_set:Npn #2 } }
38439 }
38440 \cs_new_protected:Npn \__kernel_deprecation_error:Nnn #1#2#3
38441 {
38442 \tex_protected:D \tex_outer:D \tex_edef:D #1
38443 {
38444 \exp_not:N \msg_expandable_error:nnnnn
38445 { deprecation } { deprecated-command }
38446 { \tl_to_str:n {#3} } { \token_to_str:N #1 } { \tl_to_str:n {#2} }
38447 \exp_not:N \msg_error:nneee
38448 { deprecation } { deprecated-command }
38449 { \tl_to_str:n {#3} } { \token_to_str:N #1 } { \tl_to_str:n {#2} }
38450 }
38451 }
38452 \msg_new:nnn { deprecation } { deprecated-command }
38453 {
38454 \tl_if_blank:nF {#3} { Use~ \tl_trim_spaces:n {#3} ~not~ }
38455 #2~deprecated~on~#1.
38456 }
38457 \cs_new:Npn \cs_argument_spec:N { \cs_parameter_spec:N }
38458 \cs_new_protected:Npn \iow_shipout_x:Nn { \iow_shipout_e:Nn }
38459 \cs_generate_variant:Nn \iow_shipout_x:Nn { Nx , c, cx }
38460 \cs_new_protected:cpn { \c__keys_props_root_str .str_set_x:N } #1
38461 { \__keys_variable_set:NnnN #1 { str } { } x }
38462 \cs_new_protected:cpn { \c__keys_props_root_str .str_set_x:c } #1
38463 { \__keys_variable_set:cnnN {#1} { str } { } x }
38464 \cs_new_protected:cpn { \c__keys_props_root_str .str_gset_x:N } #1
38465 { \__keys_variable_set:NnnN #1 { str } { g } x }
38466 \cs_new_protected:cpn { \c__keys_props_root_str .str_gset_x:c } #1
38467 { \__keys_variable_set:cnnN {#1} { str } { g } x }
38468 \cs_new_protected:cpn { \c__keys_props_root_str .tl_set_x:N } #1
38469 { \__keys_variable_set:NnnN #1 { tl } { } x }
38470 \cs_new_protected:cpn { \c__keys_props_root_str .tl_set_x:c } #1
38471 { \__keys_variable_set:cnnN {#1} { tl } { } x }
38472 \cs_new_protected:cpn { \c__keys_props_root_str .tl_gset_x:N } #1
38473 { \__keys_variable_set:NnnN #1 { tl } { g } x }
38474 \cs_new_protected:cpn { \c__keys_props_root_str .tl_gset_x:c } #1

```

38475 { __keys_variable_set:cnnN {#1} { t1 } { g } x } 38475

38476 \cs_new_protected:Npn \keys_set_filter:nnn { \keys_set_exclude_groups:nnn } 38476

38477 \cs_generate_variant:Nn \keys_set_filter:nnn { nnV , nnv , nno } 38477

38478 \cs_new_protected:Npn \keys_set_filter:nnnN { \keys_set_exclude_groups:nnnN } 38478

38479 \cs_generate_variant:Nn \keys_set_filter:nnnN { nnV , nnv , nno } 38479

38480 \cs_new_protected:Npn \keys_set_filter:nnnnN { \keys_set_exclude_groups:nnnnN } 38480

38481 \cs_generate_variant:Nn \keys_set_filter:nnnnN { nnV , nnv , nno } 38481

38482 __kernel_patch_deprecation:nnNNpn { 2024-02-13 } { \msg_set:nnnn } 38482

38483 \cs_new_protected:Npn \msg_gset:nnnn { \msg_set:nnnn } 38483

38484 __kernel_patch_deprecation:nnNNpn { 2024-02-13 } { \msg_set:nnn } 38484

38485 \cs_new_protected:Npn \msg_gset:nnn { \msg_set:nnn } 38485

38486 \prop_new:N \g__pdf_object_prop 38486

38487 __kernel_patch_deprecation:nnNNpn { 2022-08-30 } { [\pdf_object_new:n] } 38487

38488 \cs_new_protected:Npn \pdf_object_new:nn #1#2 38488

38489 { 38489

38490 \prop_gput:Nnn \g__pdf_object_prop {#1} {#2} 38490

38491 \pdf_object_new:n {#1} 38491

38492 } 38492

38493 __kernel_patch_deprecation:nnNNpn { 2022-08-30 } { [\pdf_object_write:n] } 38493

38494 \cs_new_protected:Npn \pdf_object_write:nn #1#2 38494

38495 { 38495

38496 \exp_args:Nee __pdf_backend_object_write:nnn 38496

38497 { __pdf_object_retrieve:n {#1} } 38497

38498 { \prop_item:Nn \g__pdf_object_prop {#1} } {#2} 38498

38499 \bool_gset_true:N \g__pdf_init_bool 38499

38500 } 38500

38501 \cs_generate_variant:Nn \pdf_object_write:nn { nx } 38501

38502 __kernel_patch_deprecation:nnNNpn { 2023-05-03 } { \bool_case:n } 38502

38503 \cs_new:Npn \bool_case_true:n { \bool_case:n } 38503

38504 __kernel_patch_deprecation:nnNNpn { 2023-05-03 } { \bool_case:nT } 38504

38505 \cs_new:Npn \bool_case_true:nT { \bool_case:nT } 38505

38506 __kernel_patch_deprecation:nnNNpn { 2023-05-03 } { \bool_case:nF } 38506

38507 \cs_new:Npn \bool_case_true:nF { \bool_case:nF } 38507

38508 __kernel_patch_deprecation:nnNNpn { 2023-05-03 } { \bool_case:nTF } 38508

38509 \cs_new:Npn \bool_case_true:nTF { \bool_case:nTF } 38509

38510 __kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_lowercase:n } 38510

38511 \cs_new:Npn \str_lower_case:n { \str_lowercase:n } 38511

38512 __kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_lowercase:f } 38512

38513 \cs_new:Npn \str_lower_case:f { \str_lowercase:f } 38513

38514 __kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_uppercase:n } 38514

38515 \cs_new:Npn \str_upper_case:n { \str_uppercase:n } 38515

38516 __kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_uppercase:f } 38516

38517 \cs_new:Npn \str_upper_case:f { \str_uppercase:f } 38517

38518 __kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_casefold:n } 38518

38519 \cs_new:Npn \str_fold_case:n { \str_casefold:n } 38519

38520 __kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_casefold:V } 38520

38521 \cs_new:Npn \str_fold_case:V { \str_casefold:V } 38521
38522 __kernel_patch_deprecation:nnNNpn { 2020-10-17 } { \str_casefold:n } 38522
38523 \cs_new:Npn \str_foldcase:n { \str_casefold:n } 38523
38524 __kernel_patch_deprecation:nnNNpn { 2022-10-17 } { \str_casefold:V } 38524
38525 \cs_new:Npn \str_foldcase:V { \str_casefold:V } 38525
38526 __kernel_patch_deprecation:nnNNpn { 2020-08-20 } { } 38526
38527 \cs_new_protected:Npn \str_declare_eight_bit_encoding:nnn #1 38527
38528 { __str_declare_eight_bit_encoding:nnnn {#1} { 1114112 } } 38528
38529 __kernel_patch_deprecation:nnNNpn { 2020-06-18 } { \seq_map_indexed_inline:Nn } 38529
38530 \cs_new_protected:Npn \seq_indexed_map_inline:Nn { \seq_map_indexed_inline:Nn } 38530
38531 __kernel_patch_deprecation:nnNNpn { 2020-06-18 } { \seq_map_indexed_function:NN } 38531
38532 \cs_new:Npn \seq_indexed_map_function:NN { \seq_map_indexed_function:NN } 38532
38533 __kernel_patch_deprecation:nnNNpn { 2023-05-10 } { \seq_map_pairwise_function:NNN } 38533
38534 \cs_new:Npn \seq_mapthread_function:NNN { \seq_map_pairwise_function:NNN } 38534
38535 __kernel_patch_deprecation:nnNNpn { 2023-10-26 } { \seq_set_map_e:NNn } 38535
38536 \cs_new_protected:Npn \seq_set_map_x:NNn { \seq_set_map_e:NNn } 38536
38537 __kernel_patch_deprecation:nnNNpn { 2023-10-26 } { \seq_gset_map_e:NNn } 38537
38538 \cs_new_protected:Npn \seq_gset_map_x:NNn { \seq_gset_map_e:NNn } 38538
38539 __kernel_patch_deprecation:nnNNpn { 2021-01-11 } { (no~longer~required) } 38539
38540 \cs_new_protected:Npn \sys_load_deprecation: { } 38540
38541 __kernel_patch_deprecation:nnNNpn { 2025-03-26 } { (no~longer~required) } 38541
38542 \cs_new:Npn \sys_if_timer_exist:T #1 {#1} 38542
38543 __kernel_patch_deprecation:nnNNpn { 2025-03-26 } { (no~longer~required) } 38543
38544 \cs_new:Npn \sys_if_timer_exist:F #1 { } 38544
38545 __kernel_patch_deprecation:nnNNpn { 2025-03-26 } { (no~longer~required) } 38545
38546 \cs_new:Npn \sys_if_timer_exist:TF #1#2 {#1} 38546
38547 __kernel_patch_deprecation:nnNNpn { 2025-03-26 } { (no~longer~required) } 38547
38548 \cs_new:Npn \sys_if_timer_exist_p: { \c_true_bool } 38548
38549 __kernel_patch_deprecation:nnNNpn { 2023-07-08 } { \text_titlecase_first:n } 38549
38550 \cs_new:Npn \text_titlecase:n #1 38550
38551 { \text_titlecase_first:n { \text_lowercase:n {#1} } } 38551
38552 __kernel_patch_deprecation:nnNNpn { 2023-07-08 } { \text_titlecase_first:nn } 38552
38553 \cs_new:Npn \text_titlecase:nn #1#2 38553
38554 { \text_titlecase_first:nn {#1} { \text_lowercase:n {#2} } } 38554
38555 __kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \text_lowercase:n } 38555
38556 \cs_new:Npn \tl_lower_case:n #1 38556
38557 { \text_lowercase:n {#1} } 38557
38558 __kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \text_lowercase:nn } 38558
38559 \cs_new:Npn \tl_lower_case:nn #1#2 38559
38560 { \text_lowercase:nn {#1} {#2} } 38560
38561 __kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \text_uppercase:n } 38561
38562 \cs_new:Npn \tl_upper_case:n #1 38562
38563 { \text_uppercase:n {#1} } 38563
38564 __kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \text_uppercase:nn } 38564
38565 \cs_new:Npn \tl_upper_case:nn #1#2 38565
38566 { \text_uppercase:nn {#1} {#2} } 38566

38567	<code>__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \text_titlecase_first:n }</code>	38567
38568	<code>\cs_new:Npn \tl_mixed_case:n #1</code>	38568
38569	<code>{ \text_titlecase_first:n { \text_lowercase:n {#1} } }</code>	38569
38570	<code>__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \text_titlecase_first:nn }</code>	38570
38571	<code>\cs_new:Npn \tl_mixed_case:nn #1#2</code>	38571
38572	<code>{ \text_titlecase_first:nn {#1} { \text_lowercase:n {#2} } }</code>	38572
38573	<code>__kernel_patch_deprecation:nnNNpn { 2022-05-23 } { \token_case_meaning:Nn }</code>	38573
38574	<code>\cs_new:Npn \tl_case:Nn { \token_case_meaning:Nn }</code>	38574
38575	<code>__kernel_patch_deprecation:nnNNpn { 2022-05-23 } { \token_case_meaning:NnT }</code>	38575
38576	<code>\cs_new:Npn \tl_case:NnT { \token_case_meaning:NnT }</code>	38576
38577	<code>__kernel_patch_deprecation:nnNNpn { 2022-05-23 } { \token_case_meaning:NnF }</code>	38577
38578	<code>\cs_new:Npn \tl_case:NnF { \token_case_meaning:NnF }</code>	38578
38579	<code>__kernel_patch_deprecation:nnNNpn { 2022-05-23 } { \token_case_meaning:NnTF }</code>	38579
38580	<code>\cs_new:Npn \tl_case:NnTF { \token_case_meaning:NnTF }</code>	38580
38581	<code>\cs_generate_variant:Nn \tl_case:Nn { c }</code>	38581
38582	<code>\prg_generate_conditional_variant:Nnn \tl_case:Nn</code>	38582
38583	<code>{ c } { T , F , TF }</code>	38583
38584	<code>__kernel_patch_deprecation:nnNNpn { 2023-10-18 } { \tl_build_begin:N }</code>	38584
38585	<code>\cs_new_protected:Npn \tl_build_clear:N { \tl_build_begin:N }</code>	38585
38586	<code>__kernel_patch_deprecation:nnNNpn { 2023-10-18 } { \tl_build_gbegin:N }</code>	38586
38587	<code>\cs_new_protected:Npn \tl_build_gclear:N { \tl_build_gbegin:N }</code>	38587
38588	<code>__kernel_patch_deprecation:nnNNpn { 2023-10-25 } { \tl_build_get_intermediate:NN }</code>	38588
38589	<code>\cs_new_protected:Npn \tl_build_get:NN { \tl_build_get_intermediate:NN }</code>	38589
38590	<code>__kernel_patch_deprecation:nnNNpn { 2022-10-09 } { [\codepoint_generate:nn] }</code>	38590
38591	<code>\cs_new:Npn \char_to_utfviii_bytes:n { __kernel_codepoint_to_bytes:n }</code>	38591
38592	<code>__kernel_patch_deprecation:nnNNpn { 2022-10-09 } { \codepoint_to_nfd:n }</code>	38592
38593	<code>\cs_new:Npn \char_to_nfd:N #1 { \codepoint_to_nfd:n {`#1} }</code>	38593
38594	<code>__kernel_patch_deprecation:nnNNpn { 2022-10-09 } { \codepoint_to_nfd:n }</code>	38594
38595	<code>\cs_new:Npn \char_to_nfd:n { \codepoint_to_nfd:n }</code>	38595
38596	<code>__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \text_lowercase:n }</code>	38596
38597	<code>\cs_new:Npn \char_lower_case:N { \text_lowercase:n }</code>	38597
38598	<code>__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \text_uppercase:n }</code>	38598
38599	<code>\cs_new:Npn \char_upper_case:N { \text_uppercase:n }</code>	38599
38600	<code>__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \text_titlecase_first:n }</code>	38600
38601	<code>\cs_new:Npn \char_mixed_case:N { \text_titlecase_first:n }</code>	38601
38602	<code>__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_casefold:n }</code>	38602
38603	<code>\cs_new:Npn \char_fold_case:N { \str_casefold:n }</code>	38603
38604	<code>__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_lowercase:n }</code>	38604
38605	<code>\cs_new:Npn \char_str_lower_case:N { \str_lowercase:n }</code>	38605
38606	<code>__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_uppercase:n }</code>	38606
38607	<code>\cs_new:Npn \char_str_upper_case:N { \str_uppercase:n }</code>	38607
38608	<code>__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_titlecase:n }</code>	38608
38609	<code>\cs_new:Npn \char_str_mixed_case:N { \str_titlecase:n }</code>	38609
38610	<code>__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_casefold:n }</code>	38610
38611	<code>\cs_new:Npn \char_str_fold_case:N { \str_casefold:n }</code>	38611
38612	<code>__kernel_patch_deprecation:nnNNpn { 2022-10-17 } { \text_lowercase:n }</code>	38612

38613	\cs_new:Npn \char_lowercase:N { \text_lowercase:n }	38613
38614	__kernel_patch_deprecation:nnNNpn { 2022-10-17 } { \text_uppercase:n }	38614
38615	\cs_new:Npn \char_uppercase:N { \text_uppercase:n }	38615
38616	__kernel_patch_deprecation:nnNNpn { 2022-10-17 } { \text_titlecase_first:n }	38616
38617	\cs_new:Npn \char_titlecase:N { \text_titlecase_first:n }	38617
38618	__kernel_patch_deprecation:nnNNpn { 2022-10-17 } { \str_casefold:n }	38618
38619	\cs_new:Npn \char_foldcase:N { \str_casefold:n }	38619
38620	__kernel_patch_deprecation:nnNNpn { 2022-10-17 } { \str_lowercase:n }	38620
38621	\cs_new:Npn \char_str_lowercase:N { \str_lowercase:n }	38621
38622	__kernel_patch_deprecation:nnNNpn { 2022-10-17 }	38622
38623	{ \tl_to_str:e { \text_titlecase_first:n } }	38623
38624	\cs_new:Npn \char_str_titlecase:N #1	38624
38625	{ \tl_to_str:e { \text_titlecase_first:n {#1} } }	38625
38626	__kernel_patch_deprecation:nnNNpn { 2022-10-17 } { \str_uppercase:n }	38626
38627	\cs_new:Npn \char_str_uppercase:N { \str_uppercase:n }	38627
38628	__kernel_patch_deprecation:nnNNpn { 2022-10-17 } { \str_casefold:n }	38628
38629	\cs_new:Npn \char_str_foldcase:N { \str_casefold:n }	38629
38630	\tl_map_inline:nn	38630
38631	{	38631
38632	{ catcode } { catcode_remove }	38632
38633	{ charcode } { charcode_remove }	38633
38634	{ meaning } { meaning_remove }	38634
38635	}	38635
38636	{	38636
38637	\use:e	38637
38638	{	38638
38639	__kernel_patch_deprecation:nnNNpn { 2022-01-11 } { \peek_remove_spaces:n }	38639
38640	\cs_gset_protected:Npn \exp_not:c { peek_ #1 _ignore_spaces:NTF } ##1##2##3	38640
38641	{	38641
38642	\peek_remove_spaces:n	38642
38643	{ \exp_not:c { peek_ #1 :NTF } ##1 {##2} {##3} }	38643
38644	}	38644
38645	__kernel_patch_deprecation:nnNNpn { 2022-01-11 } { \peek_remove_spaces:n }	38645
38646	\cs_gset_protected:Npn \exp_not:c { peek_ #1 _ignore_spaces:NT } ##1##2	38646
38647	{	38647
38648	\peek_remove_spaces:n	38648
38649	{ \exp_not:c { peek_ #1 :NT } ##1 {##2} }	38649
38650	}	38650
38651	__kernel_patch_deprecation:nnNNpn { 2022-01-11 } { \peek_remove_spaces:n }	38651
38652	\cs_gset_protected:Npn \exp_not:c { peek_ #1 _ignore_spaces:NF } ##1##2	38652
38653	{	38653
38654	\peek_remove_spaces:n	38654
38655	{ \exp_not:c { peek_ #1 :NF } ##1 {##2} }	38655
38656	}	38656
38657	}	38657
38658	}	38658

```
38659 \cs_new_protected:Npn \prop_put_if_new:Nnn { \prop_put_if_not_in:Nnn } 38659
38660 \cs_new_protected:Npn \prop_gput_if_new:Nnn { \prop_gput_if_not_in:Nnn } 38660
38661 \cs_generate_variant:Nn \prop_put_if_new:Nnn 38661
38662 { NnV , NV , c , cnV , cV } 38662
38663 \cs_generate_variant:Nn \prop_gput_if_new:Nnn 38663
38664 { NnV , NV , c , cnV , cV } 38664
38665 %% 38665
38666 %% 38666
38667 %% End of file `expl3-code.tex'. 38667
```