

The uniquecounter package

Heiko Oberdiek*
<heiko.oberdiek at gmail.com>

2016/05/16 v1.3

Abstract

This package provides a kind of counter that provides unique number values. Several counters can be created by different names. The numeric values are not limited.

Contents

1	Documentation	2
1.1	Example	2
2	Implementation	2
2.1	Reload check and package identification	2
2.2	Catcodes	4
3	Test	6
3.1	Catcode checks for loading	6
3.2	Macro tests	8
3.2.1	Test with L ^A T _E X	8
3.2.2	Test with plain-T _E X	9
4	Installation	10
4.1	Download	10
4.2	Bundle installation	11
4.3	Package installation	11
4.4	Refresh file name databases	11
4.5	Some details for the interested	11
5	Catalogue	12
6	History	12
	[2009/09/11 v1.0]	12
	[2009/12/18 v1.1]	12
	[2011/01/30 v1.2]	12
	[2016/05/16 v1.3]	13
7	Index	13

*Please report any issues at <https://github.com/ho-tex/oberdiek/issues>

1 Documentation

`\UniqueCounterNew {⟨name⟩}`

Macro `\UniqueCounterNew` creates a new unique counter `⟨name⟩`. An error is thrown, if the counter already exists.

`\UniqueCounterCall {⟨name⟩} {⟨code⟩}`

Macro `\UniqueCounterCall` calls the given `⟨code⟩` with a new value of counter `⟨name⟩` as argument.

`\UniqueCounterIncrement {⟨name⟩}`

Macro `\UniqueCounterIncrement` generates a new value for the counter `⟨name⟩` by incrementing by one (globally).

`\UniqueCounterGet {⟨name⟩}`

Expandable macro `\UniqueCounterGet` returns the current value of counter `⟨name⟩`

1.1 Example

```
1 ⟨*example⟩
2 \documentclass{minimal}
3 \usepackage{uniquecounter}
4 \UniqueCounterNew{anchor}
5 \makeatletter
6 \newcommand*{\DefNewAnchorName}[2]{%
7   % #1 is unique counter value
8   % #2 is name of anchor
9   \@namedef{anchor@#2}{a#1}%
10 }
11 \newcommand*{\NewAnchorName}[1]{%
12   \UniqueCounterCall{anchor}\DefNewAnchorName{#1}%
13 }
14 \newcommand*{\PrintAnchorName}[1]{%
15   \@nameuse{anchor@#1}%
16 }
17 \begin{document}
18   \NewAnchorName{Top}%
19   \NewAnchorName{Left}%
20   \noindent
21   Top: \PrintAnchorName{Top}\\\\%
22   Left: \PrintAnchorName{Left}%
23 \end{document}
24 ⟨/example⟩
```

2 Implementation

```
25 ⟨*package⟩
```

2.1 Reload check and package identification

Reload check, especially if the package is not used with \LaTeX .

```
26 \begingroup\catcode61\catcode48\catcode32=10\relax%
27 \catcode13=5 % ^^M
28 \endlinechar=13 %
```

```

29 \catcode35=6 % #
30 \catcode39=12 % '
31 \catcode44=12 % ,
32 \catcode45=12 % -
33 \catcode46=12 % .
34 \catcode58=12 % :
35 \catcode64=11 % @
36 \catcode123=1 % {
37 \catcode125=2 % }
38 \expandafter\let\expandafter\x\csname ver@uniquecounter.sty\endcsname
39 \ifx\x\relax % plain-Tex, first loading
40 \else
41   \def\empty{}%
42   \ifx\x\empty % LaTeX, first loading,
43     % variable is initialized, but \ProvidesPackage not yet seen
44   \else
45     \expandafter\ifx\csname PackageInfo\endcsname\relax
46       \def\x#1#2{%
47         \immediate\write-1{Package #1 Info: #2.}%
48       }%
49     \else
50       \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
51     \fi
52     \x{uniquecounter}{The package is already loaded}%
53   \aftergroup\endinput
54 \fi
55 \fi
56 \endgroup%

```

Package identification:

```

57 \begingroup\catcode61\catcode48\catcode32=10\relax%
58 \catcode13=5 % ^^M
59 \endlinechar=13 %
60 \catcode35=6 % #
61 \catcode39=12 % '
62 \catcode40=12 % (
63 \catcode41=12 % )
64 \catcode44=12 % ,
65 \catcode45=12 % -
66 \catcode46=12 % .
67 \catcode47=12 % /
68 \catcode58=12 % :
69 \catcode64=11 % @
70 \catcode91=12 % [
71 \catcode93=12 % ]
72 \catcode123=1 % {
73 \catcode125=2 % }
74 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
75   \def\x#1#2#3[#4]{\endgroup
76     \immediate\write-1{Package: #3 #4}%
77     \xdef#1{#4}%
78   }%
79 \else
80   \def\x#1#2[#3]{\endgroup
81     #2[#{#3}]%
82     \ifx#1\@undefined
83       \xdef#1{#3}%
84     \fi
85     \ifx#1\relax
86       \xdef#1{#3}%
87     \fi
88   }%
89 \fi

```

```

90 \expandafter\x\csname ver@uniquecounter.sty\endcsname
91 \ProvidesPackage{uniquecounter}%
92 [2016/05/16 v1.3 Provide unlimited unique counter (HO)]%

```

2.2 Catcodes

```

93 \begingroup\catcode61\catcode48\catcode32=10\relax%
94 \catcode13=5 % ^^M
95 \endlinechar=13 %
96 \catcode123=1 % {
97 \catcode125=2 % }
98 \catcode64=11 % @
99 \def\x{\endgroup
100 \expandafter\edef\csname uqc@AtEnd\endcsname{%
101 \endlinechar=\the\endlinechar\relax
102 \catcode13=\the\catcode13\relax
103 \catcode32=\the\catcode32\relax
104 \catcode35=\the\catcode35\relax
105 \catcode61=\the\catcode61\relax
106 \catcode64=\the\catcode64\relax
107 \catcode123=\the\catcode123\relax
108 \catcode125=\the\catcode125\relax
109 }%
110 }%
111 \x\catcode61\catcode48\catcode32=10\relax%
112 \catcode13=5 % ^^M
113 \endlinechar=13 %
114 \catcode35=6 % #
115 \catcode64=11 % @
116 \catcode123=1 % {
117 \catcode125=2 % }
118 \def\TMP@EnsureCode#1#2{%
119 \edef\uqc@AtEnd{%
120 \uqc@AtEnd
121 \catcode#1=\the\catcode#1\relax
122 }%
123 \catcode#1=#2\relax
124 }
125 \TMP@EnsureCode{33}{12}% !
126 \TMP@EnsureCode{39}{12}% '
127 \TMP@EnsureCode{42}{12}% *
128 \TMP@EnsureCode{43}{12}% +
129 \TMP@EnsureCode{46}{12}% .
130 \TMP@EnsureCode{47}{12}% /
131 \TMP@EnsureCode{91}{12}% [
132 \TMP@EnsureCode{93}{12}% ]
133 \TMP@EnsureCode{96}{12}% '
134 \edef\uqc@AtEnd{\uqc@AtEnd\noexpand\endinput}

135 \begingroup\expandafter\expandafter\expandafter\endgroup
136 \expandafter\ifx\csname RequirePackage\endcsname\relax
137 \def\TMP@RequirePackage#1[#2]{%
138 \begingroup\expandafter\expandafter\expandafter\endgroup
139 \expandafter\ifx\csname ver@#1.sty\endcsname\relax
140 \input #1.sty\relax
141 \fi
142 }%
143 \TMP@RequirePackage{bigintcalc}[2007/11/11]%
144 \TMP@RequirePackage{infwarerr}[2007/09/09]%
145 \else
146 \RequirePackage{bigintcalc}[2007/11/11]%
147 \RequirePackage{infwarerr}[2007/09/09]%
148 \fi

```

\uqc@IncNum

```
149 \begingroup\expandafter\expandafter\expandafter\endgroup
150 \expandafter\ifx\csname numexpr\endcsname\relax
151 \def\uqc@IncNum#1{%
152   \begingroup
153     \count@=\csname uqc@cnt@#1\endcsname\relax
154     \advance\count@\@ne
155     \expandafter\xdef\csname uqc@cnt@#1\endcsname{%
156       \number\count@
157     }%
158     \ifnum\count@=2147483647 %
159       \global\expandafter\let\csname uqc@inc@#1\endcsname
160         \uqc@IncBig
161     \fi
162   \endgroup
163 }%
164 \else
165 \def\uqc@IncNum#1{%
166   \expandafter\xdef\csname uqc@cnt@#1\endcsname{%
167     \number\numexpr\csname uqc@cnt@#1\endcsname+1%
168   }%
169   \ifnum\csname uqc@cnt@#1\endcsname=2147483647 %
170     \global\expandafter\let\csname uqc@inc@#1\endcsname
171       \uqc@IncBig
172   \fi
173 }%
174 \fi
```

\uqc@IncBig

```
175 \def\uqc@IncBig#1{%
176   \expandafter\xdef\csname uqc@cnt@#1\endcsname{%
177     \expandafter\expandafter\expandafter
178     \BigIntCalcInc\csname uqc@cnt@#1\endcsname!%
179   }%
180 }
```

\uqc@Def

```
181 \begingroup\expandafter\expandafter\expandafter\endgroup
182 \expandafter\ifx\csname newcommand\endcsname\relax
183 \def\uqc@Def#1{\def#1##1}%
184 \else
185 \def\uqc@Def#1{\newcommand*{#1}[1]}%
186 \fi
```

\UniqueCounterNew

```
187 \uqc@Def\UniqueCounterNew{%
188   \expandafter\ifx\csname uqc@cnt@#1\endcsname\relax
189     \expandafter\xdef\csname uqc@cnt@#1\endcsname{0}%
190     \global\expandafter\let\csname uqc@inc@#1\endcsname\uqc@IncNum
191     \@PackageInfo{uniquecounter}{New unique counter '#1'}%
192   \else
193     \@PackageError{uniquecounter}{Unique counter '#1' is already defined}\@ehc
194   \fi
195 }
```

\UniqueCounterIncrement

```
196 \uqc@Def\UniqueCounterIncrement{%
197   \expandafter\ifx\csname uqc@cnt@#1\endcsname\relax
198     \@PackageError{uniquecounter}{Unique counter '#1' is undefined}\@ehc
199   \else
200     \csname uqc@inc@#1\endcsname{#1}%
201   \fi
202 }
```

\UniqueCounterGet

```
203 \uqc@Def\UniqueCounterGet{%
204   \csname uqc@cnt@#1\endcsname
205 }
```

\UniqueCounterCall

```
206 \uqc@Def\UniqueCounterCall{%
207   \expandafter\ifx\csname uqc@cnt@#1\endcsname\relax
208   \@@PackageError{uniquecounter}{Unique counter ‘#1’ is undefined}\@ehc
209   \expandafter\uqc@Call\expandafter0%
210   \else
211     \UniqueCounterIncrement{#1}%
212     \expandafter\expandafter\expandafter\uqc@Call
213     \expandafter\expandafter\expandafter{%
214       \csname uqc@cnt@#1\expandafter\endcsname\expandafter
215     }%
216   \fi
217 }
```

\uqc@Call

```
218 \long\def\uqc@Call#1#2{#2{#1}}%

219 \uqc@AtEnd%

220 \</package>
```

3 Test

3.1 Catcode checks for loading

```
221 (*test1)

222 \catcode'\{=1 %
223 \catcode'\}=2 %
224 \catcode'\#=6 %
225 \catcode'\@=11 %
226 \expandafter\ifx\csname count@\endcsname\relax
227   \countdef\count@=255 %
228 \fi
229 \expandafter\ifx\csname @gobble\endcsname\relax
230   \long\def\@gobble#1{%
231   \fi
232 \expandafter\ifx\csname @firstofone\endcsname\relax
233   \long\def\@firstofone#1{#1}%
234 \fi
235 \expandafter\ifx\csname loop\endcsname\relax
236   \expandafter\@firstofone
237 \else
238   \expandafter\@gobble
239 \fi
240 {%
241   \def\loop#1\repeat{%
242     \def\body{#1}%
243     \iterate
244   }%
245   \def\iterate{%
246     \body
247     \let\next\iterate
248   \else
249     \let\next\relax
250   \fi
251   \next
252 }%
```

```

253 \let\repeat=\fi
254 }%
255 \def\RestoreCatcodes{}
256 \count@=0 %
257 \loop
258 \edef\RestoreCatcodes{%
259   \RestoreCatcodes
260   \catcode\the\count@=\the\catcode\count@\relax
261 }%
262 \ifnum\count@<255 %
263   \advance\count@ 1 %
264 \repeat
265
266 \def\RangeCatcodeInvalid#1#2{%
267   \count@=#1\relax
268   \loop
269     \catcode\count@=15 %
270   \ifnum\count@<#2\relax
271     \advance\count@ 1 %
272   \repeat
273 }
274 \def\RangeCatcodeCheck#1#2#3{%
275   \count@=#1\relax
276   \loop
277     \ifnum#3=\catcode\count@
278   \else
279     \errmessage{%
280       Character \the\count@\space
281       with wrong catcode \the\catcode\count@\space
282       instead of \number#3%
283     }%
284   \fi
285   \ifnum\count@<#2\relax
286     \advance\count@ 1 %
287   \repeat
288 }
289 \def\space{ }
290 \expandafter\ifx\csname LoadCommand\endcsname\relax
291 \def\LoadCommand{\input uniquecounter.sty\relax}%
292 \fi
293 \def\Test{%
294   \RangeCatcodeInvalid{0}{47}%
295   \RangeCatcodeInvalid{58}{64}%
296   \RangeCatcodeInvalid{91}{96}%
297   \RangeCatcodeInvalid{123}{255}%
298   \catcode'\@=12 %
299   \catcode'\=0 %
300   \catcode'\%=14 %
301   \LoadCommand
302   \RangeCatcodeCheck{0}{36}{15}%
303   \RangeCatcodeCheck{37}{37}{14}%
304   \RangeCatcodeCheck{38}{47}{15}%
305   \RangeCatcodeCheck{48}{57}{12}%
306   \RangeCatcodeCheck{58}{63}{15}%
307   \RangeCatcodeCheck{64}{64}{12}%
308   \RangeCatcodeCheck{65}{90}{11}%
309   \RangeCatcodeCheck{91}{91}{15}%
310   \RangeCatcodeCheck{92}{92}{0}%
311   \RangeCatcodeCheck{93}{96}{15}%
312   \RangeCatcodeCheck{97}{122}{11}%
313   \RangeCatcodeCheck{123}{255}{15}%
314   \RestoreCatcodes

```

```

315 }
316 \Test
317 \csname @@end\endcsname
318 \end
319 </test1>

```

3.2 Macro tests

3.2.1 Test with L^AT_EX

```

320 <*test2>
321 \NeedsTeXFormat{LaTeX2e}
322 \nofiles
323 \documentclass{minimal}
324 \usepackage{uniquecounter}[2016/05/16]
325 \usepackage{qstest}
326 \IncludeTests{*}
327 \LogTests{log}{*}{*}
328
329 \newcommand*{\CheckValue}[2]{%
330   \Expect*{#2}*{\UniqueCounterGet{#1}}}%
331 }
332 \newcommand*{\CheckSpace}[1]{%
333   \sbox0{#1}%
334   \Expect{0.0pt}*{\the\wd0}%
335 }
336
337 \begin{qstest}{creation}{creation}
338   \CheckSpace{%
339     \UniqueCounterNew{test}%
340   }%
341   \CheckValue{test}{0}%
342 \end{qstest}
343
344 \begin{qstest}{increment}{increment}
345   \CheckSpace{%
346     \UniqueCounterIncrement{test}%
347   }%
348   \CheckValue{test}{1}%
349   \makeatletter
350   \def\uqc@cnc@test{2147483645}%
351   \CheckValue{test}{2147483645}%
352   \CheckSpace{%
353     \UniqueCounterIncrement{test}%
354   }%
355   \CheckValue{test}{2147483646}%
356   \CheckSpace{%
357     \UniqueCounterIncrement{test}%
358   }%
359   \Expect{true}*{\ifx\uqc@inc\uqc@NumInc true\else false\fi}%
360   \CheckValue{test}{2147483647}%
361   \CheckSpace{%
362     \UniqueCounterIncrement{test}%
363   }%
364   \CheckValue{test}{2147483648}%
365   \CheckSpace{%
366     \UniqueCounterIncrement{test}%
367   }%
368   \CheckValue{test}{2147483649}%
369 \end{qstest}
370
371 \begin{qstest}{call}{call}
372   \def\CheckCall#1#2{%

```



```

373   \Expect{#1}{#2}%
374 }%
375 \CheckSpace{%
376   \UniqueCounterNew{foo}%
377 }%
378 \CheckValue{foo}{0}%
379 \def\Check#1{%
380   \CheckSpace{%
381     \UniqueCounterCall{foo}{\CheckCall}{#1}%
382   }%
383   \CheckValue{foo}{#1}%
384 }%
385 \Check{1}%
386 \Check{2}%
387 \Check{3}%
388 \Check{4}%
389 \Check{5}%
390 \Check{6}%
391 \Check{7}%
392 \Check{8}%
393 \Check{9}%
394 \Check{10}%
395 \Check{11}%
396 \Check{12}%
397 \end{qstest}
398
399 \csname @@end\endcsname
400 /test2)

```

3.2.2 Test with plain-TeX

```

401 (*test3)
402 \input uniquecounter.sty\relax
403 \catcode'\@=11 %
404 \def\CheckValue#1#2{%
405   \begingroup
406     \edef\A{#2}%
407     \edef\B{\UniqueCounterGet{#1}}%
408     \ifx\A\B
409       \else
410         \@PackageError{TEST}{Failed: \A\space<> \B}\@ehc
411       \fi
412   \endgroup
413 }
414 \def\CheckSpace#1{%
415   \setbox0=\hbox{#1}%
416   \ifdim\wd0=\z@
417     \else
418       \@PackageError{TEST}{Failed: 0.0pt <> \the\wd0}\@ehc
419     \fi
420 }
421
422 \begingroup
423   \CheckSpace{%
424     \UniqueCounterNew{test}%
425   }%
426   \CheckValue{test}{0}%
427 \endgroup
428
429 \begingroup
430   \CheckSpace{%
431     \UniqueCounterIncrement{test}%
432   }%
433   \CheckValue{test}{1}%

```

```

434 \def\uqc@cnt@test{2147483645}%
435 \CheckValue{test}{2147483645}%
436 \CheckSpace{%
437   \UniqueCounterIncrement{test}%
438 }%
439 \CheckValue{test}{2147483646}%
440 \CheckSpace{%
441   \UniqueCounterIncrement{test}%
442 }%
443 \ifx\uqc@inc\uqc@NumInc
444 \else
445   \@PackageError{TEST}{Failed: wrong inc function}\@ehc
446 \fi
447 \CheckValue{test}{2147483647}%
448 \CheckSpace{%
449   \UniqueCounterIncrement{test}%
450 }%
451 \CheckValue{test}{2147483648}%
452 \CheckSpace{%
453   \UniqueCounterIncrement{test}%
454 }%
455 \CheckValue{test}{2147483649}%
456 \endgroup
457 \begingroup
458 \def\CheckCall#1#2{%
459   \begingroup
460     \def\A{#1}%
461     \def\B{#2}%
462     \ifx\A\B
463     \else
464       \@PackageError{TEST}{Failed: \A\space <> \B}\@ehc
465     \fi
466   \endgroup
467 }%
468 \CheckSpace{%
469   \UniqueCounterNew{foo}%
470 }%
471 \CheckValue{foo}{0}%
472 \CheckSpace{%
473   \UniqueCounterCall{foo}{\CheckCall}{1}%
474 }%
475 \CheckSpace{%
476   \UniqueCounterCall{foo}{\CheckCall}{2}%
477 }%
478 \CheckValue{foo}{2}%
479 \endgroup
480 \csname @@end\endcsname\end
481 /test3

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

[CTAN:macros/latex/contrib/oberdiek/uniquecounter.dtx](http://ctan.org/pkg/uniquecounter) The source file.

[CTAN:macros/latex/contrib/oberdiek/uniquecounter.pdf](http://ctan.org/pkg/uniquecounter) Documentation.

Bundle. All the packages of the bundle ‘oberdiek’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

¹<http://ctan.org/pkg/uniquecounter>

[CTAN:install/macros/latex/contrib/oberdiek.tds.zip](#)

TDS refers to the standard “A Directory Structure for T_EX Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

4.2 Bundle installation

Unpacking. Unpack the `oberdiek.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip oberdiek.tds.zip -d ~/texmf
```

Script installation. Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

4.3 Package installation

Unpacking. The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain T_EX:

```
tex uniquecounter.dtx
```

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

<code>uniquecounter.sty</code>	→ <code>tex/generic/oberdiek/uniquecounter.sty</code>
<code>uniquecounter.pdf</code>	→ <code>doc/latex/oberdiek/uniquecounter.pdf</code>
<code>uniquecounter-example.tex</code>	→ <code>doc/latex/oberdiek/uniquecounter-example.tex</code>
<code>test/uniquecounter-test1.tex</code>	→ <code>doc/latex/oberdiek/test/uniquecounter-test1.tex</code>
<code>test/uniquecounter-test2.tex</code>	→ <code>doc/latex/oberdiek/test/uniquecounter-test2.tex</code>
<code>test/uniquecounter-test3.tex</code>	→ <code>doc/latex/oberdiek/test/uniquecounter-test3.tex</code>
<code>uniquecounter.dtx</code>	→ <code>source/latex/oberdiek/uniquecounter.dtx</code>

If you have a `docstrip.cfg` that configures and enables `docstrip`’s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

4.4 Refresh file name databases

If your T_EX distribution (teT_EX, mikT_EX, ...) relies on file name databases, you must refresh these. For example, teT_EX users run `texhash` or `mktextlsr`.

4.5 Some details for the interested

Unpacking with L^AT_EX. The `.dtx` chooses its action depending on the format:

plain T_EX: Run `docstrip` and extract the files.

L^AT_EX: Generate the documentation.

If you insist on using L^AT_EX for `docstrip` (really, `docstrip` does not need L^AT_EX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{uniquecounter.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdfL^AT_EX:

```
pdflatex uniquecounter.dtx
makeindex -s gind.ist uniquecounter.idx
pdflatex uniquecounter.dtx
makeindex -s gind.ist uniquecounter.idx
pdflatex uniquecounter.dtx
```

5 Catalogue

The following XML file can be used as source for the [T_EX Catalogue](#). The elements `caption` and `description` are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is `uniquecounter.xml`.

```
482 (*catalogue)
483 <?xml version='1.0' encoding='us-ascii'?>
484 <!DOCTYPE entry SYSTEM 'catalogue.dtd'>
485 <entry datestamp='$Date$' modifier='$Author$' id='uniquecounter'>
486   <name>uniquecounter</name>
487   <caption>Provides unlimited unique counter.</caption>
488   <authorref id='auth:oberdiek'>/>
489   <copyright owner='Heiko Oberdiek' year='2009,2011'>/>
490   <license type='lppl1.3'>/>
491   <version number='1.3'>/>
492   <description>
493     This package provides a kind of counter that provides unique
494     number values. Several counters can be created with different names.
495     The numeric values are not limited.
496   <p/>
497   The package is part of the <xref refid='oberdiek'>oberdiek</xref>
498   bundle.
499 </description>
500 <documentation details='Package documentation'
501   href='ctan:/macros/latex/contrib/oberdiek/uniquecounter.pdf'>/>
502 <ctan file='true' path='/macros/latex/contrib/oberdiek/uniquecounter.dtx'>/>
503 <miktex location='oberdiek'>/>
504 <texlive location='oberdiek'>/>
505 <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'>/>
506 </entry>
507 </catalogue>
```

6 History

[2009/09/11 v1.0]

- First public version.

[2009/12/18 v1.1]

- Bug fix in `\UniqueCounterCall` for values > 9 (bug report of Lev Bishop).

[2011/01/30 v1.2]

- Already loaded package files are not input in plain T_EX.

- Documentation updates.

7 Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; plain numbers refer to the code lines where the entry is used.

Symbols	
<code>\#</code>	224
<code>\%</code>	300
<code>\@</code>	225, 298, 403
<code>\@PackageError</code>	193, 198, 208, 410, 418, 445, 464
<code>\@PackageInfo</code>	191
<code>\@ehc</code>	193, 198, 208, 410, 418, 445, 464
<code>\@firstofone</code>	233, 236
<code>\@gobble</code>	230, 238
<code>\@namedef</code>	9
<code>\@nameuse</code>	15
<code>\@ne</code>	154
<code>\@undefined</code>	82
<code>\%</code>	21, 299
<code>\{</code>	222
<code>\}</code>	223
A	
<code>\A</code>	406, 408, 410, 460, 462, 464
<code>\advance</code>	154, 263, 271, 286
<code>\aftergroup</code>	53
B	
<code>\B</code>	407, 408, 410, 461, 462, 464
<code>\begin</code>	17, 337, 344, 371
<code>\BigIntCalcInc</code>	178
<code>\body</code>	242, 246
C	
<code>\catcode</code>	26, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 57, 58, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 93, 94, 96, 97, 98, 102, 103, 104, 105, 106, 107, 108, 111, 112, 114, 115, 116, 117, 121, 123, 222, 223, 224, 225, 260, 269, 277, 281, 298, 299, 300, 403
<code>\Check</code>	379, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396
<code>\CheckCall</code>	372, 381, 458, 473, 476
<code>\CheckSpace</code>	332, 338, 345, 352, 356, 361, 365, 375, 380, 414, 423, 430, 436, 440, 448, 452, 468, 472, 475
<code>\CheckValue</code>	329, 341, 348, 351, 355, 360, 364, 368, 378, 383, 404, 426, 433, 435, 439, 447, 451, 455, 471, 478
<code>\count@</code>	153, 154, 156, 158, 227, 256, 260, 262, 263, 267, 269, 270, 271, 275, 277, 280, 281, 285, 286
<code>\countdef</code>	227
<code>\csname</code>	38, 45, 74, 90, 100, 136, 139, 150, 153, 155, 159, 166, 167, 169, 170, 176, 178, 182, 188, 189, 190, 197, 200, 204, 207, 214, 226, 229, 232, 235, 290, 317, 399, 480
D	
<code>\DefNewAnchorName</code>	6, 12
<code>\documentclass</code>	2, 323
E	
<code>\empty</code>	41, 42
<code>\end</code>	23, 318, 342, 369, 397, 480
<code>\endcsname</code>	38, 45, 74, 90, 100, 136, 139, 150, 153, 155, 159, 166, 167, 169, 170, 176, 178, 182, 188, 189, 190, 197, 200, 204, 207, 214, 226, 229, 232, 235, 290, 317, 399, 480
<code>\endinput</code>	53, 134
<code>\endlinechar</code>	28, 59, 95, 101, 113
<code>\errmessage</code>	279
<code>\Expect</code>	330, 334, 359, 373
H	
<code>\hbox</code>	415
I	
<code>\ifdim</code>	416
<code>\ifnum</code>	158, 169, 262, 270, 277, 285
<code>\ifx</code>	39, 42, 45, 74, 82, 85, 136, 139, 150, 182, 188, 197, 207, 226, 229, 232, 235, 290, 359, 408, 443, 462
<code>\immediate</code>	47, 76
<code>\IncludeTests</code>	326
<code>\input</code>	140, 291, 402
<code>\iterate</code>	243, 245, 247
L	
<code>\LoadCommand</code>	291, 301
<code>\LogTests</code>	327
<code>\loop</code>	241, 257, 268, 276
M	
<code>\makeatletter</code>	5, 349
N	
<code>\NeedsTeXFormat</code>	321
<code>\NewAnchorName</code>	11, 18, 19

