

# The `texassert` package\*

Hanson Char  
`hanson.char@gmail.com`

November 9, 2024

## Abstract

An assertion library for unit testing in plain TeX.

## 1 Introduction

This package emerged from a desire to explore `l3build` and literate programming. It provides a collection of Plain TeX macros that I originally used for unit testing, now converted into a `.dtx` file, allowing for regeneration of the original source files from the literate code.

All `.tex` files in this package are written in Plain TeX, offering a simple mechanism for performing assertions in unit testing Plain TeX macros. I hope you find it useful. *Profitez!*

## 2 Usage Examples

### 2.1 Length Assertions

To unit test the `\lengthof` macro in this library, for example, we can create a file `length-tests.tex` with something like:

```
% Import the necessary macros
\input import \import{lengthof} \import{assert}

% Length of an empty string is zero
\lengthof{} \asserteq\the\length=0

% Length of '0' is one
\lengthof{0} \asserteq\the\length=1

% Length of '12.3456' is seven
\lengthof{12.3456} \asserteq\the\length=7

% Summary of the assertions made so far
\assertionsummary

\bye
```

Compile it with a TeX engine, e.g. `pdftex length-tests.tex`, we get an output file `length-tests.pdf` with something like:

Assertion Summary: 3/3 assertions passed i.e. 0/3 assertions failed.

---

\*This document corresponds to `texassert` v0.0.2, dated 2024/11/07.

## 2.2 (More Examples ...)

TODO

## 3 Source Repository

<https://github.com/hansonchar/texassert>

## 4 Useful Resources

Not so much related to the library provided by this package per se, but some commands and external resources which I found directly useful or necessary for the purpose of *constructing* this package per se.

1. [Examples](#) in the `l3build` repository. The `simple-tree` example in particular.
2. `texdoc l3build` – information directly related to `l3build`.
3. `texdoc doc` – the `doc` package used by `l3build` implicitly.
4. `texdoc docstrip` – the `docstrip` package used by `l3build` implicitly.
5. `texdoc source2e` – information related to various macros that are or can be used in a `.dtx` file.
6. `texdoc dtxtut` – Scott Pakin. *How to Package Your L<sup>A</sup>T<sub>E</sub>X Package*. January 21, 2024. (I had lots of *Aha!* moments in reading this.)
7. Michel Gossens, Frank Mittelbach, and Alexander Samarin. *The L<sup>A</sup>T<sub>E</sub>X Companion*. Addison Wesley, Reading, Massachusetts, October 1, 1994. ISBN 0-201-54199-8.

## 5 Implementation

`import.tex` Contains `\import`.

`\import` Used to prevent the same file from being `\input` more than once.

```
1 \def\import#1{%
2   \expandafter\ifx\csname import:#1\endcsname\relax
3     \input #1
4     \expandafter\gdef\csname import:#1\endcsname{}%
5     % Imported #1\par
6   \fi
7 }
```

`common.tex` Contains common code and configuration used in this library.

```
8 \showboxdepth=\maxdimen \showboxbreadth=\maxdimen
9
10 \newtoks\result \newtoks\tokstemp
11 \newcount\n
12 \newcount\integer
```

```

13
14 \def\true{\let\bool=\iftrue}
15 \def\false{\let\bool=\iffalse}

\debug Writes a line of debug message immediately to the terminal and the log file when
debugging is enabled (via \debugtrue which is the default).
16 \newif\ifdebug
17 \debugtrue
18 \def\debug#1{\ifdebug \immediate\write16{[DEBUG] #1}\fi}

\ifEmpty Checks if the given parameter is empty.
19 \newif\ifempty
20 \def\checkifempty#1{\expandafter\def\expandafter\input\expandafter{#1}%
21   \global\ifx\input\empty \emptytrue\else\emptyfalse\fi}
22
23 % Assigning \iffalse to \then and use as a parameter delimiter is critical
24 % in making the if-macros skippable.
25 % Source: https://tug.org/TUGboat/tb45-1/tb139weremuth-isint.pdf
26 \let\then=\iffalse
27 \def\ifEmpty#1\then{%
28   \checkifempty{#1}\ifempty
29 }

\ifUndefined Checks if the given control sequence is undefined.
30 \long\def\ifUndefined#1\then{%
31   \edef\x{\meaning#1}%
32   \let\e=\escapechar \escapechar--1
33   \edef\y{\string\undefined}\escapechar=\e
34   \def\true{\iftrue}\def\false{\iffalse}%
35   \def\next{\expandafter\expandafter\expandafter
36     \aftergroup\ifx\x\y\true\else\false\fi}\next}}

\ifDefined Checks if the given control sequence is defined.
37 \long\def\not#1#2\then{#1#2\then \false \else \true \fi \bool}
38 \long\def\ifDefined#1\then{\ifUndefined#1\then \false \else \true\fi \bool}

lengthof.tex Contains the code used to find out the length of a given string.

\lengthof Computes the length of the given string parameter.
39 \input import \import{common}
40
41 \newcount\length
42 \edef\temp{\the\catcode'\@}\catcode'\@=11
43
44 \def\lengthof#1{\length=0 %
45   \bgroup
46     \edef\lengthof@input{#1}%
47     \ifEmpty\lengthof@input\then
48       \let\next=\relax
49     \else
50       \def\next{\expandafter\lengthofA\lengthof@input\eof}%
51     \fi
52     \next
53   \egroup

```

```

54 }
55 \def\lengthofA#1#2\eut{\global\advance\length by1 %
56 \ifEmpty#2\then
57 \let\next=\relax
58 \else
59 \def\next{\lengthofA#2\eut}%
60 \fi
61 \next
62 }
63
64 \catcode'\@=\temp % restore the original catcode for @

```

`checkeq.tex` Contains the code used to check if two given strings are equal.

`\checkeq` Used to check if two given strings are equal. Assume no space in the strings.

```

65 \input import \import{lengthof}
66
67 \newif\ifeq
68 \edef\temp{\the\catcode'\@}\catcode'\@=11
69
70 \global\eqtrue
71 % Assume no spaces
72 \def\checkeq#1#2{%
73 \edef\checkeq@fstparam{#1}%
74 \edef\checkeq@sndparam{#2}%
75 \lengthof\checkeq@fstparam \edef\lena{\number\length}%
76 \lengthof\checkeq@sndparam \edef\lenb{\number\length}%
77 \ifx\lena\lenb
78 \ifnum\length=0
79 \global\eqtrue \let\next=\relax
80 \else
81 \expandafter\expandafter\expandafter
82 \def\expandafter\expandafter\expandafter
83 \next\expandafter\expandafter\expandafter
84 {\expandafter\expandafter\expandafter
85 \checkeqA\expandafter\checkeq@fstparam
86 \expandafter\eut\checkeq@sndparam\eut}%
87 \fi
88 \else
89 \global\eqfalse \let\next=\relax
90 \fi
91 \next
92 }}
93 \def\checkeqA#1#2\eut#3#4\eut{%
94 \if#1#3{}% the trailing '{}' is necessary to avoid adding extra spaces
95 \ifx\relax#2\relax
96 \global\eqtrue \let\next=\relax
97 \else
98 \def\next{\checkeqA#2\eut#4\eut}%
99 \fi
100 \else
101 \global\eqfalse \let\next=\relax
102 \fi
103 \next

```

```

104 }
105
106 \catcode'@=\temp % restore the original catcode for @

```

**assert.tex** Contains the code used for assertion purposes.

```

107 \input import \import{checked}
108
109 \ifDefined\ProvidesPackage\then
110   \ProvidesPackage{texassert}
111 \fi
112
113 \newcount\countassertions
114 \newcount\countassertionspassed
115 \newcount\countassertionsfailed
116 \newif\ifassertmessageonly
117 \edef\temp{\the\catcode'@}\catcode'@=11
118
119 \let\assertDone=\iffalse
120 \def\unexpected{\toks0={unexpected!}}
121 \def\expected{\toks0={expected}}
122 \def\assert{\asserteq\the\toks0={expected}}
123 \def\assertTrue#1\assertDone{#1\then \expected \else \unexpected\fi \assert}
124 \def\assertFalse#1\assertDone{#1\then \unexpected \else \expected\fi \assert}
125
126 \def\resetassertions{%
127   \countassertions=0
128   \countassertionspassed=0
129   \countassertionsfailed=0
130 }

```

**\asserteq** Asserts that the two given string are equal, taking catcode into account.

```

131 \def\asserteq#1=#2{%
132   \global\advance\countassertions by1
133   \edef\assert@a{#1}%
134   % \message{assert@a: [\meaning\assert@a]}%
135   \edef\assert@b{#2}%
136   % \message{assert@b: [\meaning\assert@b]}%
137   \ifx\assert@a\assert@b\relax\relax
138     \global\advance\countassertionspassed by1
139   \else
140     \global\advance\countassertionsfailed by1%
141     \message{...}%
142     \def\errmsg{*** assertion (\the\countassertions) failure:
143       '#1' not equal '#2' ***}%
144     \message{\errmsg}%
145     \ifassertmessageonly\else
146       \medbreak
147       \indent\indent{\errmsg}%
148       \medbreak\fi
149   \fi
150 }}

```

**\asserteqnocat** Asserts that the two given string are equal, disregarding any catcode differences.

```

151 \def\asserteqnocat#1=#2{%

```

```

152 \global\advance\countassertions by1
153 \edef\assert@a{#1}%
154 % \message{assert@a: [\meaning\assert@a]]}%
155 \edef\assert@b{#2}%
156 % \message{assert@b: [\meaning\assert@b]]}%
157 \checkeq\assert@a\assert@b
158 \ifeq
159   \global\advance\countassertionspassed by1
160 \else
161   \global\advance\countassertionsfailed by1
162   \message{...}%
163   \def\errmsg{*** assertion (\the\countassertions) failure:
164     '#1' not equal '#2' ***}%
165   \message{\errmsg}%
166   \ifassertmessageonly\else
167     \medbreak
168     \indent\indent{\errmsg}%
169     \medbreak\fi
170 \fi
171 }}

```

`\assertneq` Asserts that the two given string are not equal, taking catcode into account.

```

172 \def\assertneq#1=#2{%
173   \global\advance\countassertions by1
174   \edef\assert@a{#1}%
175   % \message{assert@a: [\meaning\assert@a]]}%
176   \edef\assert@b{#2}%
177   % \message{assert@b: [\meaning\assert@b]]}%
178   \ifx\assert@a\assert@b\relax\relax
179     \global\advance\countassertionsfailed by1%
180     \message{...}%
181     \def\errmsg{*** assertion (\the\countassertions) failure:
182       '#1' equal '#2' ***}%
183     \message{\errmsg}%
184     \ifassertmessageonly\else
185       \medbreak
186       \indent\indent{\errmsg}%
187       \medbreak\fi
188   \else
189     \global\advance\countassertionspassed by1
190   \fi
191 }}

```

`\assertionsummary` Typesets a summary of the assertions made. Then reset to a state as if no assertion has been made.

```

192 \def\assertionsummary{%
193   \def\sp{ }%
194   \def\summary{%
195     Assertion Summary: \the\countassertionspassed/\the\countassertions\sp
196     assertions passed i.e.
197     \the\countassertionsfailed/\the\countassertions\sp assertions failed.}%
198   \message{\summary}%
199   \ifassertmessageonly\else
200     \medbreak

```

```

201     \summary
202   \fi}\resetassertions}
203
204 \catcode'\@=\temp % restore the original catcode for @

```

texassert.sty Used for packaging purposes.

```

205 \input{assert}

```

## 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; numbers in *roman* refer to the code lines where the entry is used.

<b>A</b>		
\advance	55, 132, 138, 140, 152, 159, 161, 173, 179, 189	
\aftergroup	..... 36	
\assert	... 122, 123, 124	
\assert.tex	..... <u>107</u>	
\assert@a	. 133, 134, 137, 153, 154, 157, 174, 175, 178	
\assert@b	. 135, 136, 137, 155, 156, 157, 176, 177, 178	
\assertDone	119, 123, 124	
\asserteq	.... 122, <u>131</u>	
\asserteqnocat	.... <u>151</u>	
\assertFalse	..... 124	
\assertionssummary	. <u>192</u>	
\assertneq	..... <u>172</u>	
\assertTrue	..... 123	
<b>B</b>		
\bgroup	..... 45	
\bool	.... 14, 15, 37, 38	
<b>C</b>		
\catcode	..... 42, 64, 68, 106, 117, 204	
\checkeq	..... <u>65</u> , 157	
\checkeq.tex	..... <u>65</u>	
\checkeq@fstparam	. ..... 73, 75, 85	
\checkeq@sndparam	. ..... 74, 76, 86	
\checkeqA	... 85, 93, 98	
\checkifempty	... 20, 28	
\common.tex	..... <u>8</u>	
\countassertions	.. .. 113, 127, 132, 142, 152, 163, 173, 181, 195, 197	
\countassertionsfailed	.... 115, 129, 140, 161, 179, 197	
\countassertionspassed	.... 114, 128, 138, 159, 189, 195	
\csname	..... 2, 4	
<b>D</b>		
\debug	..... <u>16</u>	
\debugtrue	..... 17	
\def	1, 14, 15, 18, 20, 27, 30, 34, 35, 37, 38, 44, 50, 55, 59, 72, 82, 93, 98, 120, 121, 122, 123, 124, 126, 131, 142, 151, 163, 172, 181, 192, 193, 194	
\emptyfalse	..... 21	
\emptytrue	..... 21	
\endcsname	..... 2, 4	
\eot	50, 55, 59, 86, 93, 98	
\eqfalse	..... 89, 101	
\eqtrue	..... 70, 79, 96	
\errmsg	... 142, 144, 147, 163, 165, 168, 181, 183, 186	
\escapechar	..... 32, 33	
\expandafter	.... 2, 4, 20, 35, 50, 81, 82, 83, 84, 85, 86	
\expected	. 121, 123, 124	
<b>F</b>		
\false	. 15, 34, 36, 37, 38	
\fi	... 6, 18, 21, 36, 37, 38, 51, 60, 87, 90, 99, 102, 111, 123, 124, 148, 149, 169, 170, 187, 190, 202	
<b>G</b>		
\gdef	..... 4	
\global	.... 21, 55, 70, 79, 89, 96, 101, 132, 138, 140, 152, 159, 161, 173, 179, 189	
<b>E</b>		
\e	..... 32, 33	
\edef	31, 33, 42, 46, 68, 73, 74, 75, 76, 117, 133, 135, 153, 155, 174, 176	
\egroup	..... 53	
\else	21, 36, 37, 38, 49, 58, 80, 88, 97, 100, 123, 124, 139, 145, 160, 166, 184, 188, 199	
\empty	..... 21	
<b>I</b>		
\if	..... 94	
\ifassertmessageonly	..... 116, 145, 166, 184, 199	
\ifdebug	..... 16, 18	

<code>\ifDefined</code> . . . . .	<a href="#">37</a> , <a href="#">109</a>	<b>M</b>	<code>\result</code> . . . . .	<a href="#">10</a>
<code>\ifEmpty</code> . . . . .	<a href="#">19</a> , <a href="#">47</a> , <a href="#">56</a>	<code>\maxdimen</code> . . . . .	<a href="#">8</a>	
<code>\ifempty</code> . . . . .	<a href="#">19</a> , <a href="#">28</a>	<code>\meaning</code> <a href="#">31</a> , <a href="#">134</a> , <a href="#">136</a> ,	<b>S</b>	
<code>\ifeq</code> . . . . .	<a href="#">67</a> , <a href="#">158</a>	<a href="#">154</a> , <a href="#">156</a> , <a href="#">175</a> , <a href="#">177</a>	<code>\showboxbreadth</code> . . . . .	<a href="#">8</a>
<code>\iffalse</code> . . . . .	<a href="#">15</a> , <a href="#">23</a> , <a href="#">26</a> , <a href="#">34</a> , <a href="#">119</a>	<code>\medbreak</code> . . . . .	<code>\showboxdepth</code> . . . . .	<a href="#">8</a>
		<a href="#">146</a> , <a href="#">148</a> , <a href="#">167</a> ,	<code>\sp</code> . . . . .	<a href="#">193</a> , <a href="#">195</a> , <a href="#">197</a>
<code>\ifnum</code> . . . . .	<a href="#">78</a>	<a href="#">169</a> , <a href="#">185</a> , <a href="#">187</a> , <a href="#">200</a>	<code>\string</code> . . . . .	<a href="#">33</a>
<code>\iftrue</code> . . . . .	<a href="#">14</a> , <a href="#">34</a>	<code>\message</code> <a href="#">134</a> , <a href="#">136</a> , <a href="#">141</a> ,	<code>\summary</code> . . . . .	<a href="#">194</a> , <a href="#">198</a> , <a href="#">201</a>
<code>\ifUndefined</code> . . . . .	<a href="#">30</a> , <a href="#">38</a>	<a href="#">144</a> , <a href="#">154</a> , <a href="#">156</a> ,	<b>T</b>	
<code>\ifx</code> . . . . .	<a href="#">2</a> , <a href="#">21</a> ,	<a href="#">162</a> , <a href="#">165</a> , <a href="#">175</a> ,	<code>\temp</code> . . . . .	<a href="#">42</a> ,
	<a href="#">36</a> , <a href="#">77</a> , <a href="#">95</a> , <a href="#">137</a> , <a href="#">178</a>	<a href="#">177</a> , <a href="#">180</a> , <a href="#">183</a> , <a href="#">198</a>		<a href="#">64</a> , <a href="#">68</a> , <a href="#">106</a> , <a href="#">117</a> , <a href="#">204</a>
<code>\immediate</code> . . . . .	<a href="#">18</a>	<b>N</b>	<code>\texassert.sty</code> . . . . .	<a href="#">205</a>
<code>\import</code> . . . . .	<a href="#">1</a> , <a href="#">39</a> , <a href="#">65</a> , <a href="#">107</a>	<code>\n</code> . . . . .	<code>\the</code> . . . . .	<a href="#">42</a> , <a href="#">68</a> ,
<code>\import.tex</code> . . . . .	<a href="#">1</a>	<code>\newcount</code> . . . . .		<a href="#">117</a> , <a href="#">122</a> , <a href="#">142</a> ,
<code>\indent</code> . . . . .	<a href="#">147</a> , <a href="#">168</a> , <a href="#">186</a>	<a href="#">12</a> , <a href="#">41</a> , <a href="#">113</a> , <a href="#">114</a> , <a href="#">115</a>		<a href="#">163</a> , <a href="#">181</a> , <a href="#">195</a> , <a href="#">197</a>
<code>\input</code> . . . . .	<a href="#">3</a> , <a href="#">20</a> ,	<code>\newif</code> . . . . .	<code>\then</code> . . . . .	<a href="#">23</a> , <a href="#">26</a> ,
	<a href="#">21</a> , <a href="#">39</a> , <a href="#">65</a> , <a href="#">107</a> , <a href="#">205</a>	<code>\newtoks</code> . . . . .		<a href="#">27</a> , <a href="#">30</a> , <a href="#">37</a> , <a href="#">38</a> ,
<code>\integer</code> . . . . .	<a href="#">12</a>	<code>\next</code> . . . . .		<a href="#">47</a> , <a href="#">56</a> , <a href="#">109</a> , <a href="#">123</a> , <a href="#">124</a>
		<a href="#">50</a> , <a href="#">52</a> , <a href="#">57</a> , <a href="#">59</a> ,	<code>\toks</code> . . . . .	<a href="#">120</a> , <a href="#">121</a> , <a href="#">122</a>
<b>L</b>		<a href="#">61</a> , <a href="#">79</a> , <a href="#">83</a> , <a href="#">89</a> ,	<code>\tokstemp</code> . . . . .	<a href="#">10</a>
<code>\lena</code> . . . . .	<a href="#">75</a> , <a href="#">77</a>	<a href="#">91</a> , <a href="#">96</a> , <a href="#">98</a> , <a href="#">101</a> , <a href="#">103</a>	<code>\true</code> . . . . .	<a href="#">14</a> , <a href="#">34</a> , <a href="#">36</a> , <a href="#">37</a> , <a href="#">38</a>
<code>\lenb</code> . . . . .	<a href="#">76</a> , <a href="#">77</a>	<code>\not</code> . . . . .	<b>U</b>	
<code>\length</code> . . . . .	<a href="#">41</a> ,	<code>\number</code> . . . . .		<a href="#">75</a> , <a href="#">76</a>
	<a href="#">44</a> , <a href="#">55</a> , <a href="#">75</a> , <a href="#">76</a> , <a href="#">78</a>	<b>P</b>	<code>\undefined</code> . . . . .	<a href="#">33</a>
<code>\lengthof</code> . . . . .	<a href="#">39</a> , <a href="#">75</a> , <a href="#">76</a>	<code>\par</code> . . . . .	<code>\unexpected</code> <a href="#">120</a> , <a href="#">123</a> , <a href="#">124</a>	
<code>\lengthof.tex</code> . . . . .	<a href="#">39</a>	<code>\ProvidesPackage</code> . . . . .	<b>W</b>	
<code>\lengthof@input</code> . . . . .	<a href="#">46</a> , <a href="#">47</a> , <a href="#">50</a>		<code>\write</code> . . . . .	<a href="#">18</a>
		<a href="#">109</a> , <a href="#">110</a>	<b>X</b>	
<code>\lengthofA</code> . . . . .	<a href="#">50</a> , <a href="#">55</a> , <a href="#">59</a>	<b>R</b>		
<code>\let</code> . . . . .	<a href="#">14</a> , <a href="#">15</a> ,	<code>\relax</code> <a href="#">2</a> , <a href="#">48</a> , <a href="#">57</a> , <a href="#">79</a> , <a href="#">89</a> ,	<code>\x</code> . . . . .	<a href="#">31</a> , <a href="#">36</a>
	<a href="#">26</a> , <a href="#">32</a> , <a href="#">48</a> , <a href="#">57</a> ,	<a href="#">95</a> , <a href="#">96</a> , <a href="#">101</a> , <a href="#">137</a> , <a href="#">178</a>	<b>Y</b>	
	<a href="#">79</a> , <a href="#">89</a> , <a href="#">96</a> , <a href="#">101</a> , <a href="#">119</a>	<code>\resetassertions</code> . . . . .		
<code>\long</code> . . . . .	<a href="#">30</a> , <a href="#">37</a> , <a href="#">38</a>		<code>\y</code> . . . . .	<a href="#">33</a> , <a href="#">36</a>
		<a href="#">126</a> , <a href="#">202</a>		

## Change History

v0.0.1 – 2024-11-05

General: Initial version . . . . . [1](#)

v0.0.2 – 2024-11-07

General: Migrate source files

to `texassert.dtx` . . . . . [1](#)