

# conseq — continued equalities\*

Joachim Breitner<sup>†</sup>

Released 2013/05/23

## 1 Introduction

There are various ways to layout an continued equality that spans multiple lines and has explanations along some steps. Often, the best layout is not clear beforehand, as it depends on the sizes of the various elements, or implementing it adds too much noise to the actual formulas. This package provides an environment for continued equalities (or inequalities) that allows you to easily define and switch layouts.

## 2 Usage

`conseq` Our running example is the following continued equality:

```
\begin{conseq}
e^{\pi\cdot i} \\
= -1 & Euler's formula \\
< 0 & this is an inequality \\
< \sqrt{3} \\
= \int e^{-x^2} dx & this is due to Gauss.
\end{conseq}
```

As you can see, the expressions of the continued equality are separated by `\\`, with equality signs (or other relations) at the beginning of all lines but the first. Some equalities also have explanations.

The result of the above code is

$$\begin{array}{ll} e^{\pi \cdot i} = -1 & \{ \text{Euler's formula} \} \\ < 0 & \{ \text{this is an inequality} \} \\ < \sqrt{3} & \\ = \int e^{-x^2} dx & \{ \text{this is due to Gauss.} \} \end{array}$$

---

\*This file describes version v0.1, last revised 2013/05/23.

<sup>†</sup>E-mail: mail@joachim-breitner.de

The expressions are typeset in math mode, while the explanation is assumed to be regular text. The curly braces around the explanations come from the default `\ConteqExplanationLayout`.

There are other layouts available, which you select with an optional parameter to `\begin{conseq}`, e.g.

```
\begin{conseq}[explline]
[.]
\end{conseq}
```

The layouts defined by `conseq` are:

**plain** The default layout, shown above.

**explline** A layout that puts the explanations below the right-hand side of the equality it is explaining. To be used when the explanations and right-hand sides are long.

$$\begin{aligned}
 e^{\pi i} &= -1 && \{ \text{Euler's formula} \} \\
 &< 0 && \{ \text{this is an inequality} \} \\
 &< \sqrt{3} \\
 &= \int e^{-x^2} dx && \{ \text{this is due to Gauss.} \}
 \end{aligned}$$

**headline** Like plain, but the first expression is also vertically aligned with the right-hand sides.

$$\begin{aligned}
 &e^{\pi i} \\
 = &-1 && \{ \text{Euler's formula} \} \\
 &< 0 && \{ \text{this is an inequality} \} \\
 &< \sqrt{3} \\
 = &\int e^{-x^2} dx && \{ \text{this is due to Gauss.} \}
 \end{aligned}$$

**onecolumn** A combination of **explline** and **headline** that puts everything in one column, for maximum space efficiency.

$$\begin{aligned}
 & e^{\pi \cdot i} \\
 & = -1 \\
 & \quad \{ \text{Euler's formula} \} \\
 & < 0 \\
 & \quad \{ \text{this is an inequality} \} \\
 & < \sqrt{3} \\
 & = \int e^{-x^2} dx \\
 & \quad \{ \text{this is due to Gauss.} \}
 \end{aligned}$$

**oneline** A layout, more for demonstrational purposes, that ignores the explanations and puts everything in one line.

$$e^{\pi \cdot i} = -1 < 0 < \sqrt{3} = \int e^{-x^2} dx$$

`\ConteqSetDefaultLayout` You can change the default layout using `\ConteqSetDefaultLayout{\langle layout \rangle}`

### 3 Defining layouts

`\ConteqDefineLayout` To define a new layout you use `\ConteqDefineLayout`, which takes seven arguments:

1. The name of the layout,
2. What to put before the continued equalities,
3. the left-hand side of the first equality,
4. the right-hand side of equalities without explanation,
5. the right-hand side of equalities with explanation,
6. the line separator and
7. what to put after the continued equalities.

For example the existing layouts are defined using

```

1
2 \ConteqDefineLayout
3   {plain}
4   {\begin{align*}}
5   { #1 }
6   { & #1 }
7   { & #1 && #2 }

```

```

8      { \ }
9      {\end{align*}}
10 \ConteqSetDefaultLayout{plain}
11
12 \ConteqDefineLayout
13   {explline}
14   {\begin{align*}}
15   { #1 }
16   { & #1 }
17   { & #1 \ \ & \mathrel{\phantom{=}} #2 }
18   { \ }
19   {\end{align*}}
20
21 \ConteqDefineLayout
22   {headline}
23   {\begin{align*}}
24   { & \mathrel{\phantom{=}} #1 \ \ }
25   { & #1 }
26   { & #1 && #2 }
27   { \ }
28   {\end{align*}}
29
30 \ConteqDefineLayout
31   {onecolumn}
32   {\begin{align*}}
33   { & \mathrel{\phantom{=}} #1 \ \ }
34   { & #1 }
35   { & #1 \ \ & \mathrel{\phantom{=}} #2 }
36   { \ }
37   {\end{align*}}
38
39 \ConteqDefineLayout
40   {oneline}
41   {\[ ]}
42   { #1 }
43   { #1 }
44   { #1 }
45   {}
46   {\]}
47

```

## 4 Changing the explanation style

`\ConteqExplStyle` The explanation is formatted using the macro `\ConteqExplStyle{explanation}`, which is by default defined as

```

48 \newcommand{\ConteqExplStyle}[1]{\{-#1-\}}

```

To change the style, simply redefine this macro using, for example:

```

\renewcommand{\ConteqExplStyle}{\textit{-- #1}}

```

## 5 Future work

This package is, at the time of writing, very new, so please let me know about problems you are having or features you are missing.

One feature that I am considering is an auto-selection of layouts, so when you specify `\begin{coneq}[plain,explline,onecolumn]... \end{coneq}`, it will analyze the table and select, from your list of layouts, the first one that is “ok”, where “ok” would be some layout-specific heuristic taking the size of the expressions and explanations as well as the current `\linewidth` into account. If you think that this would be useful to you, please let me know.

## 6 Implementation

### 6.1 Package loading

```
49 \RequirePackage{amsmath}
50 \RequirePackage{environ}
```

### 6.2 Defining Messages

```
51 \msg_new:nnn
52   {coneq}
53   {empty}
54   { \msg_warning_text:n{coneq}:~Empty~coneq~environment~\msg_line_context: }
55
56 \msg_new:nnn
57   {coneq}
58   {ignoreddata}
59   { \msg_warning_text:n{coneq}:~Ignored~text~\msg_line_context: }
60
61 \msg_new:nnn
62   {coneq}
63   {undefinedlayout}
64   { \msg_critical_text:n{coneq}:~Undefined~layout~‘#1’~\msg_line_context: }
```

### 6.3 Declaring local variables

```
65 \tl_new:N \l_coneq_default_layout_tl
66 \tl_new:N \l_coneq_layout_tl
67 \tl_new:N \l_coneq_body_tl
68 \tl_new:N \l_coneq_lines_seq
69 \tl_new:N \l_coneq_cells_seq
70 \tl_new:N \l_coneq_head_tl
71 \tl_new:N \l_coneq_lastline_tl
72 \tl_new:N \l_coneq_rhs_tl
73 \tl_new:N \l_coneq_expl_tl
74 \tl_new:N \l_coneq_result_tl
```

## 6.4 Layouts

`\ConteqSetDefaultLayout` The code to set the default layout.

```

75 \cs_new_protected:Nn \conteq_set_default_layout:n {
76   \tl_set:Nn \l_conteq_default_layout_tl {#1}
77 }
78 \cs_new_eq:NN \ConteqSetDefaultLayout \conteq_set_default_layout:n

```

`\ConteqDefineLayout` The code to define new layouts.

```

79 \cs_new_protected:Nn \conteq_define_layout:nnnnnnn {
80   \cs_new_protected:cn {conteq_#1_begin:} { #2 }
81   \cs_new_protected:cn {conteq_#1_lhs:n} { #3 }
82   \cs_new_protected:cn {conteq_#1_rhs_only:n} { #4 }
83   \cs_new_protected:cn {conteq_#1_rhs_expl:nn} { #5 }
84   \cs_new_protected:cn {conteq_#1_nl:} { #6 }
85   \cs_new_protected:cn {conteq_#1_end:} { #7 }
86 }
87 \cs_new_eq:NN \ConteqDefineLayout \conteq_define_layout:nnnnnnn

```

## 6.5 Expansion utility function

`\conteq_args_once:Nn` For ease of debugging(?) we construct a token list that contains exactly the tokens  
`\conteq_args_once:NV` that a programmer would enter to create the layout manually. For that we need some  
`\conteq_args_once:Nnn` fine-grained control over expansion.  
`\conteq_args_once:NVV`

```

88 \cs_new:Npn \conteq_args_once:Nn #1#2 {
89   \exp_not:o {#1 {#2}}
90 }
91 \cs_generate_variant:Nn \conteq_args_once:Nn {NV}
92
93 \cs_new:Npn \conteq_args_once:Nnn #1#2#3 {
94   \exp_not:o {#1 {#2}{#3}}
95 }
96 \cs_generate_variant:Nn \conteq_args_once:Nnn {NVV}

```

## 6.6 Main code

`\conteq_print_line:nn` This macro splits and prints one line of the table. The second argument is either `\` or, for the last line of the table, empty.

```

97 \cs_new_protected:Npn \conteq_print_line:Nnn #1#2#3 {
98   \seq_set_split:Nnn \l_conteq_cells_seq { & } {#2}
99   \seq_pop_left:NN \l_conteq_cells_seq \l_conteq_rhs_tl
100  \tl_clear:N \l_conteq_expl_tl
101  \seq_if_empty:NF \l_conteq_cells_seq {
102    \seq_pop_left:NN \l_conteq_cells_seq \l_conteq_expl_tl
103  }
104  \tl_if_blank:VTF \l_conteq_expl_tl {
105    \tl_put_right:Nx #1 {
106      \conteq_args_once:NV \conteq_rhs_only:n \l_conteq_rhs_tl

```

```

107   }
108   }{
109     \tl_set:Nx \l_tmpa_tl {\exp_not:N \text {\exp_not:N \ConteqExplStyle {\exp_not:V \l_conteq
110     \tl_put_right:Nx #1 {
111       \conteq_args_once:NVV \conteq_rhs_expl:nn \l_conteq_rhs_tl \l_tmpa_tl
112     }
113   }
114   \seq_if_empty:NF \l_conteq_cells_seq {
115     \msg_warning:nn{conteq}{ignoreddata}
116   }
117   \tl_put_right:Nn #1 {#3}
118 }
119 \cs_generate_variant:Nn \conteq_print_line:Nnn {Nno}
120 \cs_generate_variant:Nn \conteq_print_line:Nnn {NVn}

```

**conteq** The main environment of the package.

```

121 \NewEnviron{conteq}[1][\l_conteq_default_layout_tl]{
122   \tl_set:NV \l_conteq_body_tl \BODY
123
124   \tl_if_blank:oT \l_conteq_body_tl{
125     \msg_warning:nn{conteq}{empty}
126   }

```

Figure out the layout to use...

```

127   \tl_set:Nn \l_conteq_layout_tl {#1}

```

and set the various functions accordingly, if the layout exists.

```

128   \cs_if_exist:cTF {conteq_ \l_conteq_layout_tl _ begin:}{
129     \clist_map_inline:nn{begin:, lhs:n,rhs_only:n, rhs_expl:nn, nl:, end:}{
130       \cs_set_eq:cc {conteq_##1}{conteq_ \l_conteq_layout_tl _##1}
131     }
132   }{
133     \msg_critical:nmx{conteq}{undefined layout}{\l_conteq_layout_tl}
134   }

```

Split the body into individual lines.

```

135   \seq_set_split:NnV \l_conteq_lines_seq { \\ } \l_conteq_body_tl
136
137   \tl_clear:N \l_conteq_result_tl

```

If there is only one line, simply print it.

```

138
139   \int_case:nnn {\seq_count:N \l_conteq_lines_seq}
140   {
141     {1} {
142       \tl_put_right:NV \l_conteq_result_tl \conteq_begin:
143       \tl_put_right:NV \l_conteq_result_tl \l_conteq_body_tl
144       \tl_put_right:NV \l_conteq_result_tl \conteq_end:
145     }
146   }{

```

Otherwise extract the head and the last line, and print each line using `\conteq_print_line:nn`

```
147 \seq_pop_left:NN \l_conteq_lines_seq \l_conteq_head_tl
148 \seq_pop_right:NN \l_conteq_lines_seq \l_conteq_lastline_tl
149
150 \tl_put_right:NV \l_conteq_result_tl \conteq_begin:
151 \tl_put_right:Nx \l_conteq_result_tl {
152   \conteq_args_once:NV \conteq_lhs:n \l_conteq_head_tl
153 }
154 \seq_map_inline:Nn \l_conteq_lines_seq {
155   \conteq_print_line:Nno \l_conteq_result_tl {##1} {\conteq_nl:}
156 }
157 \conteq_print_line:NVn \l_conteq_result_tl \l_conteq_lastline_tl {}
158 \tl_put_right:No \l_conteq_result_tl \conteq_end:
159 }
160 % Use this for debugging
161 %\tl_show:N \l_conteq_result_tl
162 \tl_use:N \l_conteq_result_tl
163 }
164 \endinput
```