

# The `colcell` Package

Martin Scharrer

[martin@scharrer-online.de](mailto:martin@scharrer-online.de)

<http://www.ctan.org/pkg/colcell/>

Version v0.1 – 2011/02/04

## Abstract

This small package provides macros which collect the cell content of a tabular and provide it to a macro as argument. It was inspired by the `\collect@body` macro defined by the `amsmath` or the `environ` package, which can be used to collect the body of an environment. Special care is taken to remove all aligning macros inserted by tabular from the cell content. The macros also work in the last column of a table. They do not support verbatim material inside the cells.

## 1 Usage

This package provides the macros `\collectcell` and `\endcollectcell` which are supposed to be used with the `>{ }` and `<{ }` tabular column declarations of the `array` package. This can be done either in the argument of tabular or using `\newcolumntype`.

The following code defines a ‘E’ column which passes the contents of its cell to `\usermacro` as an argument. The macro can then process the content as usual.

```
% Preamble:
\usepackage{array}
\usepackage{colcell}
% Preamble or document:
\newcolumntype{E}{>{\collectmacro\usermacro}c<{\endcollectmacro}}

% Document:
\begin{tabular}{lE}
  A & Example \\ % Same as \usermacro{Example}
  B & Text      \\ % Same as \usermacro{Text}
\end{tabular}
```

For example `\usermacro` could be `\fbox` and wrap the cell content in a frame box. More complicated macros are also supported as long they take one argument. This package was originally programmed to be used with the `\tikztiming` macro of the `tikz-timing` package. This macro takes some complex user input and draws a timing diagram from it.

Note that if such a cell contains a tabular environment by itself, the environment must be wrapped in braces ‘{ }’ to ensure proper operation.

## 2 Implementation

```
1 \newtoks\collect@cell@toks
```

`\collectcell`

```
2 \newenvironment{collectcell}{}{}
3 \def\collectcell#1#2\ignorespaces{%
4   \collect@cell@toks{}%
5   \let\collect@cell@spaces\empty
6   \def\collect@cell@end{%
7     \expandafter\scantokens\expandafter
8     {\expandafter#1\expandafter{\the\collect@cell@toks}}%
9   }
10  \def\collect@cell@next{\collect@cell@look}%
11  \collect@cell@next
12 }
```

`\endcollectcell`

Holds unique signature which will expand to nothing.

```
13 \def\endcollectcell{\@gobble{endcollectcell}}
```

`\collect@cell@look`

```
14 \def\collect@cell@look{%
15   \futurelet\collect@cell@lettoken\collect@cell@look@
16 }
```

`\collect@cell@eatSPACE`

```
17 \begingroup
18 \def\:\{\collect@cell@eatSPACE}
19 \expandafter\gdef\:\ {\collect@cell@look}
20 \endgroup
```

`\collect@cell@look@`

```

21 \def\collect@cell@look@{%
22   \ifx\collect@cell@lettoken\@sptoken
23     \edef\collect@cell@spaces{\collect@cell@spaces\✓
24       space}%
25   \def\collect@cell@next{\collect@cell@eatSPACE}%
26   \else
27     \ifx\collect@cell@lettoken\bgroup
28       \def\collect@cell@next{\collect@cell@group}%
29     \else
30       \def\collect@cell@next{\collect@cell@arg}%
31     \fi
32   \fi
33 }

```

`\collect@cell@group`

```

34 \def\collect@cell@group#1{%
35   \begingroup
36   \def\@tempa{#1}%
37   \def\@tempb{\bgroup}%
38   \ifx\@tempa\@tempb
39     \endgroup
40     \def\collect@cell@next{\collect@cell@arg\bgroup}%
41   \else
42     \endgroup
43     \def\collect@cell@next{\collect@cell@arg{{#1}}}%
44   \fi
45   \collect@cell@next
46 }

```

`\collect@cell@addarg`

```

47 \def\collect@cell@addarg#1{%
48   \expandafter\expandafter\expandafter\✓
49     collect@cell@toks
50   \expandafter\expandafter\expandafter
51     {\expandafter\the\expandafter\collect@cell@toks\✓
52       collect@cell@spaces#1}%
53   \let\collect@cell@spaces\empty
54 }

```

`\collect@cell@arg`

```
53 \def\collect@cell@arg#1{%
54   \ifx\collect@cell@lettoken\
55     \def\collect@cell@next{\collect@cell@end#1}%
56   \else
57     \ifx\collect@cell@lettoken\unskip
58       \def\collect@cell@next{%
59         \@ifnextchar\endcollectcell
60         {\collect@cell@end#1}%
61         {\collect@cell@addarg{#1}\collect@cell@look}
62       }%
63     \else
64       \collect@cell@addarg{#1}%
65       \def\collect@cell@next{\collect@cell@look}%
66     \fi
67   \fi
68   \collect@cell@next
69 }
```