Le package dcolumn*

David Carlisle

Traduction française par Jean-Pierre Drucbert†

1999/07/05

Résumé

Ce package permet de définir des entrées de colonnes dans un environnement `array` ou `tabular` qui doivent être alignés sur un « point décimal ».

Ce package définit un spécificateur de colonne `D` qui prend trois arguments.
`D{`⟨*sep.tex*⟩`}{`⟨*sep.dvi*⟩`}{`⟨*nombre de décimales*⟩`}`

- L'argument ⟨*sep.tex*⟩ doit être un unique caractère, qui sera utilisé comme séparateur dans le fichier `.tex`. Ce sera en général « `.` » ou « `,` ».
- L'argument ⟨*sep.dvi*⟩ est utilisé comme séparateur dans la sortie, il peut être identique au premier argument, mais peut être toute expression en mode mathématique, comme `\cdot`. Il faut noter que `dcolumn` utilise toujours le mode mathématique pour écrire les chiffres, ainsi que le séparateur.
- L'argument ⟨*nombre de décimales*⟩ doit être le nombre maximal de chiffres placés après la virgule dans la colonne. Si ce nombre est négatif, un nombre quelconque de chiffres pourra être utilisé dans la colonne, et toutes les entrées seront centrées sur le séparateur. Notez que ceci peut rendre la colonne trop large, comme c'est le cas pour la première colonne de l'exemple ci-dessous. Si cet argument est positif, la colonne utilise des macros équivalentes à `\rightdots` et `\endrightdots` de `array.sty`, sinon les macros sont essentiellement équivalentes à `\centerdots` et `\endcenterdots`.

Vous pouvez ne pas désirer utiliser toutes les trois entrées dans le préambule de `array` ou `tabular`, et vous pouvez définir vos propres spécificateurs de préambule en utilisant `\newcolumntype`.

Par exemple, nous pouvons écrire :
`\newcolumntype{d}[1]{D{.}{\cdot}{#1}}`

`d` prend un seul argument, qui spécifie le nombre de chiffres décimaux, et le fichier `.tex` utilisera un « `.` », tandis que « $\cdot$ » sera utilisé dans la sortie.
`\newcolumntype{.}{D{.}{.}{-1}}`

« `.` » spécifie une colonne d'entrées à centrer sur le point « . ».
`\newcolumntype{,}{D{,}{,}{2}}`

« `,` » spécifie qu'il faut prendre une colonne d'entrées avec au plus deux décimales après la virgule « , ».

*Ce fichier a le numéro de version v1.05, révisé le 1999/07/05.
†Dernière mise à jour le 19/01/2000

Avec ces définitions, la table décrite par :

```
\begin{center}
\begin{tabular}{|d{-1}|d{2}|.|,|}
1.2   & 1.2   &1.2    &1,2    \\
1.23  & 1.23  &12.5   &300,2  \\
1121.2& 1121.2&861.20 &674,29 \\
184   & 184   &10     &69     \\
.4    & .4    &       &,4     \\
      &       &.4     &
\end{tabular}
\end{center}
```

donne :

1·2	1·2	1.2	1,2
1·23	1·23	12.5	300,2
1121·2	1121·2	861.20	674,29
184	184	10	69
·4	·4		,4
		.4	

Notez que la première colonne, qui a un ⟨*nombre de décimales*⟩ négatif (premier argument du spécificateur `d`), est plus large que la seconde colonne, et donc le point décimal apparaît au milieu de la colonne.

Notez aussi que les entrées sans point décimal, sans partie décimale ou sans partie entière sont correctement traitées, ainsi que les entrées vides.

Si vous avez des cellules avec en-têtes, entrées avec `\multicolumn{1}{c}{..}` pour remplacer le type de colonne `D`), alors il se peut qu'aucune des deux formes « centrées » et « alignées à droite » ne soit satisfaisante.

titre	titre	titre
1.2	1.2	1.2
11212.2	11212.2	11212.2
.4	.4	.4

en-tête large	en-tête large	en-tête large
1.2	1.2	1.2
.4	.4	.4

Dans chacune de ces deux tables, la première colonne est définie par `D{.}{.}{-1}` pour afficher une colonne centrée sur le « `.` », et la seconde colonne est définie par `D{.}{.}{1}` pour afficher une colonne cadrée à droite.

La colonne centrée produit des colonnes qui sont plus larges que nécessaire pour s'assurer que le point décimal est centré. La deuxième colonne, qui est cadrée à droite, n'a pas cet inconvénient, mais sous un en-tête large, l'apparence visuelle est pas bonne avec peu de chiffres dans les colonnes.

Dans la version v1.03, une troisième alternative a été introduite. Le troisième argument, ⟨*nombre de décimales*⟩, peut définir à la fois le nombre de chiffres à gauche et à droite de la virgule. La troisième colonne est définie par `D{.}{.}{5.1}` dans la table de gauche ci-dessus, et par `D{.}{.}{1.1}` dans la table de droite, pour indiquer respectivement « cinq chiffres à gauche et un à droite » et « un chiffre à gauche et un chiffre à droite ». Le « `.` » peut être remplacé par « `,` » ou tout autre caractère dans cet argument. Les colonnes de chiffres sont alors positionnées de telle sorte qu'un nombre avec ce format soit centré dans la colonne.

1 The Macros

```
⟨*package⟩
```

First we load `array.sty` if it not already loaded.

```
\RequirePackage{array}
```

The basic ideas behind these macros are explained in the documentation for `array.sty`. However they use three tricks which may be useful in other contexts.

- The separator is surrounded in extra `{ }`, so that it is set with `\mathord` spacing, otherwise, for instance a ',' would have extra space after it.

- The separator is not given its special definition by making it active, as this would not work for an entry such as `& .5 &`, as the first token of an alignment entry is read *before* the preamble part, incase it is an `\omit`, in which case the preamble is to be omitted. Instead we switch the mathcode to (hex) 8000, which makes the token act as if it were active.

- Although `\mathcode‘.="8000` makes `.` act as if it were active, it is still not allowed in constructions such as `\def.{}`, even in math-mode, so we have to construct an active version of the separator, this is done by making it the uppercase of `~`, and then using the construct
  `\uppercase{\def~}{`⟨*definition*⟩`}`.
  Note that the ⟨*definition*⟩ is not uppercased, so the definition can refer to the standard, non-active use of the separator.

`\DC@` Set up uppercase tables as required, and then grab the first part of the numerical argument into `\count@`.

```
\def\DC@#1#2#3{%
  \uccode‘\~=‘#1\relax
  \m@th
  \afterassignment\DC@x\count@#3\relax{#1}{#2}}
```

`\DC@x` If `\count@` is negative, centre on the decimal point. If it is positive either `#1` will be empty in which case bad out decimal part to the number of digits specified by `\count@` or (new feature in v1.03) it is none empty in which case `\count@` contains the number of digits to the left of the point, and `#1` contains a junk token (probably `.`) followed by the number of digits to the right of the point. In either of these latter cases, `\DC@right` is used.

```
\def\DC@x#1\relax#2#3{%
  \ifnum\z@>\count@
    \expandafter\DC@centre
  \else
    \expandafter\DC@right
  \fi
  {#2}{#3}{#1}}
```

`\DC@centre` If centering on the decimal point, just need to box up the two halves.

```
\def\DC@centre#1#2#3{%
```

```
  \let\DC@end\DC@endcentre
  \uppercase{\def~}{$\egroup\setbox\tw@=\hbox\bgroup${#2}}%
  \setbox\tw@=\hbox{${\phantom{{#2}}}$}%
  \setbox\z@=\hbox\bgroup$\mathcode`#1="8000 }
```

\DC@endcentre and then pad out the smaller of the two boxes so there is the same amount of stuff either side of the point.

```
\def\DC@endcentre{$\egroup
    \ifdim \wd\z@>\wd\tw@
      \setbox\tw@=\hbox to\wd\z@{\unhbox\tw@\hfill}%
    \else
      \setbox\z@=\hbox to\wd\tw@{\hfill\unhbox\z@}\fi
    \box\z@\box\tw@}
```

\DC@right This deals with both the cases where a specified number of decimal places is given.

```
\def\DC@right#1#2#3{%
  \ifx\relax#3\relax
```

If `#3` is empty, add `\hfill` to right align the column, and Just set `\DC@rl` to begin a group, so nothing fancy is done with the whole number part.

```
    \hfill
    \let\DC@rl\bgroup
  \else
```

Otherwise set `\DC@rl` so that the whole number part is put in a box `\count@` times as wide as a digit. In order to share code with the other branch, then move `#3` (the number of decimal places) into `\count@` throwing away the '.' from the user syntax.

```
    \edef\DC@rl{to\the\count@\dimen@ii\bgroup\hss\hfill}%
    \count@\@gobble#3\relax
  \fi
  \let\DC@end\DC@endright
```

Box 2 contains the decimal part, set to `\dimen@` which is calculated below to be `\count@` times the width of a digit, plus the with of the 'decimal point'.

```
  \uppercase{\def~}{$\egroup\setbox\tw@\hbox to\dimen@\bgroup${#2}}%
   \setbox\z@\hbox{$1$}\dimen@ii\wd\z@
   \dimen@\count@\dimen@ii
   \setbox\z@\hbox{${#2}$}\advance\dimen@\wd\z@
   \setbox\tw@\hbox to\dimen@{}%
```

Box 0 contains the whole number part, either just at its natural size for right aligned columns, or set to (the old value of) `\count@` times the width of a digit. `\DC@rl` defined above determines the two cases.

```
   \setbox\z@\hbox\DC@rl$\mathcode`#1="8000 }
```

\DC@endright Just finish off the second box, and then put out both boxes.

```
\def\DC@endright{$\hfil\egroup\box\z@\box\tw@}
```

`D` The user interface, define the `D` column to take three arguments. For special purposes, you may need to directly access `\DC@` rather than the `D` column, eg to get a bold version you could use

```
\newcolumntype{E}[3]{>{\boldmath\DC@{#1}{#2}{#3}}c<{\DC@end}}
```

```
41 \newcolumntype{D}[3]{>{\DC@{#1}{#2}{#3}}c<{\DC@end}}
42 ⟨/package⟩
```

Change History

v1.00
- General: Version initiale 1

v1.01
- General: Ré-édition pour les nouveaux doc et docstrip. 1

v1.02
- General: Ré-édition pour LaTeX2e 1

v1.03
- General: Nouvelle fonctionnalité, après tools/2093 1
- `\DC@`: New feature, after tools/2093 3
- `\DC@endright`: Re-implemeted, after tools/2093 4
- `\DC@right`: Re-implemeted, after tools/2093 4
- `\DC@x`: Macro added 3

v1.04
- `\DC@right`: Add `\hfill` so integer part is still flush right if no decimal point used. 4

v1.05
- General: changements mineurs dans la doc latex/3058 1

Index

Numbers written in italic refer to the page where the corresponding entry is described, the ones underlined to the code line of the definition, the rest to the code lines where the entry is used.

Symbols

`\@gobble` 31
`\~` 4

A

`\advance` 37
`\afterassignment` . . . 6

B

`\bgroup` 16, 18, 28, 30, 34
`\box` 24, 40

C

`\count@` . 6, 8, 30, 31, 36

D

`\D` 41
`\DC@` 3, 41
`\DC@centre` 9, 14
`\DC@end` 15, 33, 41
`\DC@endcentre` . . . 15, 19
`\DC@endright` 33, 40
`\DC@right` 11, 25
`\DC@rl` 28, 30, 39
`\DC@x` 6, 7
`\def` 3, 7, 14, 16, 19, 25, 34, 40
`\dimen@` 34, 36–38
`\dimen@ii` . . . 30, 35, 36

E

`\edef` 30
`\egroup` . . 16, 19, 34, 40
`\else` 10, 22, 29
`\expandafter` 9, 11

F

`\fi` 12, 23, 32

H

`\hbox` 16–18, 21, 23, 34, 35, 37–39
`\hfil` 40
`\hfill` . . . 21, 23, 27, 30
`\hss` 30

I

`\ifdim` 20
`\ifnum` 8
`\ifx` 26

L

\let 15, 28, 33

M

\m@th 5

\mathcode 18, 39

N

\newcolumntype 41

P

\phantom 17

R

\relax . . . 4, 6, 7, 26, 31

\RequirePackage 2

S

\setbox . . 16–18, 21, 23, 34, 35, 37–39

T

\the 30

\tw@ . . 16, 17, 20, 21, 23, 24, 34, 38, 40

U

\uccode 4

\unhbox 21, 23

\uppercase 16, 34

W

\wd . . . 20, 21, 23, 35, 37

Z

\z@ 8, 18, 20, 21, 23, 24, 35, 37, 39, 40