

L'extension dcolumn*

David Carlisle

28/10/2014

Ce fichier est maintenu par l'équipe du « L^AT_EX Project ». Les rapports d'anomalie peuvent être envoyés en anglais à <http://latex-project.org/bugs.html> (catégorie `tools`).

Résumé

Cette extension définit des colonnes dans un tableau `array` ou `tabular` avec lesquelles l'alignement se fait sur le séparateur décimal.

Cette extension définit un spécificateur de colonne `D` ayant trois arguments `D{<sep.tex>}{<sep.dvi>}{<nombre de décimales>}` où :

- 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, comme il peut être toute expression en mode mathématique, comme `\cdot`. Notez que `dcolumn` utilise toujours le mode mathématique pour composer les chiffres et le séparateur.
- l'argument `<nombre de décimales>` est le nombre maximal de chiffres placés après la virgule dans la colonne. Avec un argument à valeur négative, un nombre quelconque de chiffres pourra être utilisé dans la colonne, et toutes les entrées seront centrées sur le (✖ **bord gauche du** ✖) séparateur : notez que ceci peut rendre la colonne trop large, comme dans le cas pour la première colonne de l'exemple ci-dessous. Avec un argument à valeur positive, la colonne utilise des commandes équivalentes à `\rightdots` et `\endrightdots` de l'extension `array`, sinon les commandes sont essentiellement équivalentes à `\centerdots` et `\endcenterdots`.

Vous pouvez préférer ne pas préciser l'ensemble de ces trois entrées dans le préambule de `array` ou `tabular`, et vous pouvez ici définir vos propres spécificateurs pour le préambule en utilisant `\newcolumnntype`.

Par exemple, vous pouvez écrire :

```
\newcolumnntype{d}[1]{D{.}{\cdot}{#1}}
```

*Ce fichier a pour numéro de version v1.06 et a été mis à jour le 28/10/2014. La première traduction, basée sur la version v1.05, a été publiée par Jean-Pierre Drucbert en 2000.

Ici, le spécificateur « d » prend un seul argument, qui indique le nombre de chiffres décimaux. Le fichier `.tex` utilisera un « . », tandis que la sortie utilisera un « · ».

```
\newcolumnntype{.}{D{.}{.}{-1}}
```

Le spécificateur « . » indique une colonne d'entrées centrées sur le point.

```
\newcolumnntype{,}{D{,}{,}{2}}
```

Le spécificateur « , » indique une colonne d'entrée avec au plus deux décimales après la virgule « , ».

Avec cela, une table débutant par `\begin{tabular}{|d{-1}|d{2}|.|.|}` pourra donner par exemple :

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. Le point décimal apparaît au milieu de la colonne. Notez aussi que les entrées sans point décimal, sans partie décimale ou sans partie entière tout comme les entrées vides sont correctement traitées.

Si vous avez des cellules avec entêtes (saisies 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.

entête	entête	entête		entête large	entête large	entête large
1.2	1.2	1.2		1.2	1.2	1.2
11212.2	11212.2	11212.2		.4	.4	.4
.4	.4	.4				

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

La colonne centrée produit une colonne qui plus large que nécessaire pour s'assurer que le point décimal est centré dans tous les cas. La deuxième colonne, qui est alignée à droite, n'a pas cet inconvénient mais, sous un entête large, l'apparence visuelle est assez peu satisfaisante lorsqu'il y a peu de chiffres dans les colonnes.

Dans la version v1.03, une troisième alternative a été introduite. Le troisième argument, le *nombre de décimales*, peut définir *à la fois* le nombre de chiffres à gauche et à droite de la virgule. La troisième colonne ainsi 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 « . », lui, peut être remplacé par « , » ou toute autre unité lexicale). Les colonnes de chiffres sont alors positionnées de telle sorte qu'un nombre avec le format indiqué soit centré dans la colonne.

Cette notation permet également de centrer les colonnes sur le milieu du séparateur plutôt que sur son bord gauche ; par exemple, `D{+}{\,\,\pm\,}{3,3}` donnera une belle mise en forme symétrique pour des nombres comportant jusqu'à trois chiffres de chaque côté d'un signe \pm .

1 Les commandes

```
1 \*package
```

Tout d'abord, nous chargeons l'extension `array` si ce n'est pas déjà le cas.

```
2 \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, in case 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~}{\langle definition \rangle}`.

Note that the `\langle definition \rangle` 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@`.

```
3 \def\DC@#1#2#3{%
4   \uccode'\~='#1\relax
5   \m@th
6   \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.

```
7 \def\DC@x#1\relax#2#3{%
8   \ifnum\z@>\count@
9     \expandafter\DC@centre
10  \else
11    \expandafter\DC@right
12  \fi
```

```

13   {#2}{#3}{#1}}

\DC@centre If centering on the decimal point, just need to box up the two halves.
14 \def\DC@centre#1#2#3{%
15   \let\DC@end\DC@endcentre
16   \uppercase{\def~}{$\egroup\setbox\tw@=\hbox\bgroup${#2}}}%
17   \setbox\tw@=\hbox{${\phantom{#{#2}}}$}%
18   \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.
19 \def\DC@endcentre{$\egroup
20   \ifdim \wd\z@>\wd\tw@
21     \setbox\tw@=\hbox to\wd\z@{\unhbox\tw@\hfill}%
22   \else
23     \setbox\z@=\hbox to\wd\tw@{\hfill\unhbox\z@}\fi
24   \box\z@\box\tw@}

\DC@right This deals with both the cases where a specified number of decimal places is given.
25 \def\DC@right#1#2#3{%
26   \ifx\relax#3\relax
27     \hfill
28     \let\DC@rl\bgroup
29   \else
30     \edef\DC@rl{to\the\count@\dimen@ii\bgroup\hss\hfill}%
31     \count@\@gobble#3\relax
32   \fi
33   \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 width of the 'decimal point'.
34 \uppercase{\def~}{$\egroup\setbox\tw@=\hbox to\dimen@\bgroup${#2}}}%
35 \setbox\z@=\hbox{${#1}$}\dimen@ii\wd\z@
36 \dimen@\count@\dimen@ii
37 \setbox\z@=\hbox{${#2}$}\advance\dimen@\wd\z@
38 \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.
39 \setbox\z@=\hbox\DC@rl$\mathcode'\#1="8000 }

```

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

```
40 \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

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

```
41 \newcolumnntype{D}[3]{>{\DC@{#1}{#2}{#3}}c<{\DC@end}}
```

```
42 \endpackage
```