

# The nccrules package\*

Alexander I. Rozhenko  
rozhenko@oapmg.sccc.ru

2005/05/13

This package implements `\dashrule` and `\dashrulefill` commands, that simplify composing of dashed lines and dashed multilines. Two kinds of footnote rule generation commands are also introduced: `\newfootnoterule` creates a footnote rule with an arbitrary contents and `\newfootnotedashrule` creates a footnote rule based on dash rule.

## Contents

<b>1</b>	<b>User Interface</b>	<b>1</b>
1.1	Dash Rules . . . . .	1
1.2	Custom Footnote Rules . . . . .	3
<b>2</b>	<b>The Implementation</b>	<b>4</b>

## 1 User Interface

### 1.1 Dash Rules

`\dashrule` The command

`\dashrule[raise]{h-pattern}{v-pattern}`

prepares a dash rule. Its syntax is quite similar to the `\rule` command except that *h-pattern* and *v-pattern* can contain a list of sizes delimited with spaces. List sizes are interpreted as follows: size, space, size, space, etc. In other words, every odd size is a size of rule part and every even size is a space between neighbour parts. If the last size in the list is even (means a space), its space value is divided by two and is added before the first rule part and after the last rule part. Units in sizes can be omitted. In this case, pt-units are supposed. Examples:

---

\*This file has version number v1.0, last revised 2005/05/13.



## 1.2 Custom Footnote Rules

In `manyfoot` package, you can separate footnote levels with custom footnote rules. In this package we provide two ways for creation such rules. All footnote rules should follow the following contract: they must be prepared as `\hrule` or `\hbox`; the total height of rule or box must be compensated with kerns. Usually a negative kern is inserted before a footnote rule and a nonnegative kern is inserted after the rule. The sum of both kerns and of total height of rule must be equal to zero. We follow this contract in footnote rules created with this package.

In `manyfoot` package, every footnote rule must have a name `\langle prefix \rangle footnote rule`. So, we need to specify the prefix only when a new footnote rule is constructed. Next that we need to specify is a rule width. If width is omitted, the default width of `0.4\columnwidth` is used. The footnote rule generation commands described below are used in the preamble only.

`\newfootnoterule`      The command

```
\newfootnoterule{\langle prefix \rangle}[\langle width \rangle]{\langle rule code \rangle}
```

produces a custom user-defined rule. The `\langle rule code \rangle` must be stretchable to the given width. The `\mboxfill` command from the `mboxfill` package is useful in it. Examples:

```
\newfootnoterule{starred}{\mboxfill[1.5\width][s]{*$*$}}
\newfootnoterule{dotted}{\mboxfill[2\width][s]{.}}
```

The `\starredfootnoterule` and `\dottedfootnoterule` commands are created here. They produce the following rules:

```
*****
.....
```

`\newfootnotedashrule`      The command

```
\newfootnotedashrule{\langle prefix \rangle}[\langle width \rangle]{\langle h-pattern \rangle}{\langle v-pattern \rangle}
```

produces a dash footnote rule. The special case, when `\langle h-pattern \rangle` is empty means the solid rule. Examples:

```
\newfootnotedashrule{double}{}{.4 1 .4}
\newfootnotedashrule{dashed}{3 3}{.4}
\newfootnotedashrule{dotdashed}{4 2 1 2 4 -4}{.6}
\newfootnotedashrule{ddotdashed}{6 2 1 2 1 2 6 -6}{.6}
```

The following rules are created here: `\doublefootnoterule`, `\dashedfootnoterule`, `\dotdashedfootnoterule`, and `\ddotdashedfootnoterule`. They produce the following rules:

```
=====
-----
-----
-----
```

## 2 The Implementation

The `mboxfill` package is required here:

```
1 <*package>
2 \RequirePackage{mboxfill}
```

```
\dashrule \dashrule[<raise>]{<h-pattern>}{<v-pattern>}
3 \newcommand\dashrule{}
4 \DeclareRobustCommand*\dashrule[3][\z@]{%
5 \setbox\@tempboxa\vbox{}%
6 \NCC@composedash{\NCC@vdash{#2}}{#3}%
7 \setlength\@tempdima{#1}\leavevmode
8 \raise\@tempdima\hbox{%
9 \vbox to\@tempdimc{\vss\unvbox\@tempboxa\vss}}%
10 }
```

```
\dashrulefill \dashrulefill[<raise>][<leader-type>]{<h-pattern>}{<v-pattern>}
```

This command is a composition of the `\mboxfill` and `\dashrule` commands. A trick is applied here: we decrease dash rule width on the value of last space in it and then apply `\mboxfill`.

```
11 \newcommand*\dashrulefill[1][\z@]{%
12 \@ifnextchar[{\NCC@dashfill{#1}}{\NCC@dashfill{#1}[]}%
13 }
14 \def\NCC@dashfill#1[#2]#3#4{%
15 \NCC@composedash{}{#3}%
16 \advance\@tempdimb-\@tempdimc
17 \setbox\@tempboxa\hb@xt@-\@tempdimb
18 {\hss\dashrule[#1]{#3}{#4}\hss}%
19 \mboxfill[\@tempdimc][#2]{\box\@tempboxa}%
20 }
```

```
\NCC@composedash \NCC@composedash{<action>}{<pattern>} parses the pattern and calls the action
when a rule size is parsed. At this point, the \@tempdima contains a size parsed
and \@tempdimb contains the previous space. In \@tempdimc, the whole rule size
is calculated.
```

```
21 \def\NCC@composedash#1#2{%
22 \@tempdimb\z@ % Last space
23 \@tempdimc\z@ % Accumulator
24 \@tempswatruue % True value means producing an entry
25 \NCC@parsedash#2 ! !\@nil{%
26 \advance\@tempdimc\@tempdima
27 \if@tempswa #1\@tempswafalse \else
28 \@tempdimb\@tempdima \@tempswatruue
29 \fi
30 }%
31 }
```

```
\NCC@parsedash Pattern parser:
```

```

32 \def\NCC@parsedash#1 #2\@nil#3{%
33   \if/#1/\else % Empty arg. ignored
34     \ifx#1!\else % Exclamation mark ends the list
35       \@defaultunits\@tempdima#1pt\relax\@nnil
36       #3\NCC@parsedash#2\@nil#3}%
37   \fi
38 \fi
39 }

```

`\NCC@vdash` `\NCC@vdash{h-pattern}` is applied when a vertical dash is composed. `\@tempdimb` contains the required skip, `\@tempdima` contains the rule height.

```

40 \def\NCC@vdash#1{%
41   \setbox\@tempboxa\vbox{%
42     \unvbox\@tempboxa \vskip\@tempdimb
43     \setbox\@tempboxa\hbox{}}%
44   \@tempskipa\@tempdima % Save rule height in \@tempskipa
45   \NCC@composedash{\NCC@hdash}{#1}%
46   \hb@xt@\@tempdimc{\hss\unhbox\@tempboxa\hss}%
47   }%
48 }

```

`\NCC@hdash` This action is applied when a horizontal dash is composed. `\@tempdimb` contains the required skip, `\@tempdima` contains the rule width, `\@tempskipa` contains the rule height.

```

49 \def\NCC@hdash{%
50   \setbox\@tempboxa\hbox{%
51     \unhbox\@tempboxa \kern\@tempdimb
52     \vrule \@width\@tempdima \@height\@tempskipa
53   }%
54 }

```

`\NCC@fnoterule` `\NCC@fnoterule{width}{rule code}`

This is the footnote rule producing command. As usual, two kerns must be inserted surround the rule: a negative kern before the rule and a nonnegative kern after the rule. The sum of kern values and of the total rule height must vanish. Kerns are calculated in such a way to vertically center the rule relative to 2.8pt distance top to the current position. If the total height of rule is greater than 5.6pt, the rule is moved up on the total rule height (the kern after the rule is zero in this case).

```

55 \def\NCC@fnoterule#1#2{%
56   \setlength\@tempdima{#1}%
57   \setbox\@tempboxa\hb@xt@\@tempdima{#2}%
58   \@tempdima\ht\@tempboxa \advance\@tempdima\dp\@tempboxa
59   \@tempdimb 2.8\p@ \@tempdimc .5\@tempdima
60   \ifdim\@tempdimc>\@tempdimb \@tempdimb\@tempdimc \fi
61   \advance\@tempdimb\@tempdimc
62   \kern-\@tempdimb
63   \box\@tempboxa
64   \advance\@tempdimb -\@tempdima

```

```

65 \kern\@tempdimb
66 }

\NCC@fnotedashrule \NCC@fnotedashrule{<width>}{<h-pattern>}{<v-pattern>}
Produces the rule based on \dashrulefill. A special case of empty <h-pattern>
means the solid (maybe multi-line) rule.
67 \def\NCC@fnotedashrule#1#2#3{%
68 \NCC@fnoterule{#1}{\def\@tempa{#2}%
69 \ifx\@tempa\@empty
70 \@tempskipb\@tempdima \dashrule{\@tempskipb}{#3}%
71 \else
72 \dashrulefill[\z@][s]{#2}{#3}%
73 \fi
74 }%
75 }

\NCC@fnotecreate \NCC@fnotecreate{<prefix>}{<body>}
Creates a footnote rule whose name is composed from the given <prefix> and
footnoterule. For example if the prefix is dashed, the \dashedfootnoterule
command will be created.
76 \def\NCC@fnotecreate#1{%
77 \edef\@tempa{\noexpand\newcommand*%
78 \expandafter\noexpand\csname #1footnoterule\endcsname}%
79 \@tempa
80 }
81 \@onlypreamble\NCC@fnotecreate

\NCC@fnoteprepare \NCC@fnoteprepare{<driver>}{<prefix>}[<width>]
Prepares a footnote rule command. The <driver> is a command that will be called
for collecting rule code (it gets more arguments from the input). When a driver
prepares the code, it calls the \NCC@fnotecreate command. If <width> is omitted,
the standard width of 0.4\columnwidth is used.
82 \def\NCC@fnoteprepare#1#2{%
83 \@ifnextchar[#{#1}{#2}{#1}{#2}[.4\columnwidth]}%
84 }
85 \@onlypreamble\NCC@fnoteprepare

Footnote rule creation driver has the following syntax:

\<driver>{<prefix>}[<width>]<more arguments>

\newfootnoterule \newfootnoterule{<prefix>}[<width>]{<rule code>}
86 \newcommand\newfootnoterule{\NCC@fnoteprepare\NCC@fnotedriver}
87 \def\NCC@fnotedriver#1[#2]#3{%
88 \NCC@fnotecreate{#1}{\NCC@fnoterule{#2}{#3}}%
89 }
90 \@onlypreamble\newfootnoterule
91 \@onlypreamble\NCC@fnotedriver

```

```
\newfootnotedashrule \newfootnotedashrule{<prefix>}[<width>]{<h-pattern>}{<v-pattern>}
92 \newcommand\newfootnotedashrule{\NCC@fnoteprepare\NCC@fnotedashdriver}
93 \def\NCC@fnotedashdriver#1[#2]#3#4{%
94   \NCC@fnotecreate{#1}{\NCC@fnotedashrule{#2}{#3}{#4}}%
95 }
96 \@onlypreamble\newfootnotedashrule
97 \@onlypreamble\NCC@fnotedashdriver
98 </package>
```