The **desclist** package[*]

Alexander I. Rozhenko

rozhenko@oapmg.sscc.ru

2004/11/23

The package implements the `desclist` environment and redefines the `description` environment to provide an optional parameter as a prototype for calculation the hang indent of the list.

Contents

1   User Interface

desclist   The `desclist` environment is similar to the `description` environment but adds more control for typesetting list labels. The usage:

> \begin{desclist}{⟨*prefix*⟩}{⟨*suffix*⟩}[⟨*prototype*⟩]
> \item[⟨*label*⟩] ⟨*text*⟩
> . . .
> \end{desclist}

The ⟨*prefix*⟩ is inserted before the label and the suffix is inserted after the label. The optional ⟨*prototype*⟩ parameter specifies a prototype for calculation the minimum width of a list label and for calculation the hang indent of a list text. If this parameter is omitted, the standard hang indent value is used and the minimum width of list labels is equal to zero.

A list label is generated in the \item[⟨*label*⟩] command as follows:

> ⟨*prefix*⟩⟨*label*⟩{⟨*suffix*⟩}\hfil

Therefore, you can finish the ⟨*label*⟩ with a command having a parameter to replace a suffix with something else. The \hfil command at the end of label adds necessary horizontal space if the label width is less than the minimum width of list

---

[*]This file has version number v1.1, last revised 2004/11/23.

label. Using a ⟨*prototype*⟩ together with the `\hfill` command in ⟨*prefix*⟩ and/or ⟨*suffix*⟩ parameters, you can align left, align right, or center narrow list labels.

The following example shows the benefits of the `desclist` environment:

M   — an item with a narrow label;

MM   — an item with a narrow label;

Mark — an item with a normal label;

Long mark — an item with too wide label; and

A mark without dash. Here we skip the desclist suffix.

It was prepared as follows:

```
\begin{desclist}{\sf}{ \rm\hfill---}[Mark]
\newcommand*\Skip[1]{}
\item[M] an item with a narrow label;
\item[MM] an item with a narrow label;
\item[Mark] an item with a normal label;
\item[Long mark] an item with too wide label; and
\item[A mark without dash.\Skip] Here we skip
the desclist suffix.
\end{desclist}
```

description   The package also redefines the `description` environment to allow an optional parameter. In fact, the new implementation of `description` is equivalent to

```
\begin{desclist}{\normalfont\bfseries}{}[⟨prototype⟩]
. . .
\end{desclist}
```

2   The Implementation

desclist   The definition of the `desclist` environment is quite simple. If an optional parameter is omitted, pass an empty one to the internal command.

```
1 ⟨∗package⟩
2 \newenvironment*{desclist}[2]{%
3   \@ifnextchar[{\NCC@dlist{#1}{#2}}{\NCC@dlist{#1}{#2}[]}%
4 }{%
5   \endlist
6 }
```

\NCC@dlist   The internal definition of the `desclist` is based on the `\list`:

```
7 \def\NCC@dlist#1#2[#3]{%
8   \list{}{%
```

Specify just the same list parameters as in the `description` if the optional parameter is empty:

```
9       \def\@tempa{#3}\ifx\@tempa\@empty
10        \labelwidth\z@ \itemindent-\leftmargin
```

In other case, use the prototype to calculate the label width and margins:

```
11      \else
12        \settowidth\leftmargin{#1#3{#2}}%
13        \advance\leftmargin\labelsep
14        \labelwidth=\leftmargin
15        \itemindent\z@
16      \fi
```

Finally, define the `\makelabel` command adding the desclist prefix and suffix and finishing it with `\hfil` to allow left alignment for narrow labels. The suffix is enclosed in braces, to allow a trick with ignoring it in the label mark if necessary (see the example above):

```
17      \def\makelabel##1{\hspace\labelsep #1##1{#2}\hfil}%
18    }%
19 }
```

description   Now we redefine the `description` environment. We do not test is it defined or not.

```
20 \def\description{\desclist{\normalfont\bfseries}{}}
21 \let\enddescription=\enddesclist
22 ⟨/package⟩
```