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 1
2 The Implementation 2

1 User Interface

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 \textit{(prototype)} together with the \texttt{hfill} command in \textit{(prefix)} and/or \textit{(suffix)} parameters, you can align left, align right, or center narrow list labels.

The following example shows the benefits of the \texttt{desclist} environment:

\begin{description}
\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.] Here we skip the \texttt{desclist} suffix.
\end{description}

It was prepared as follows:

\begin{verbatim}
\begin{desclist}{\rm\hfill\textemdash}\[Mark\]
\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 \texttt{desclist} suffix.
\end{desclist}
\end{verbatim}

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

\begin{verbatim}
\begin{desclist}{\normalfont\bfseries}{}\[(\textit{prototype})]\n\end{desclist}
\end{verbatim}

\section{The Implementation}

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

\begin{verbatim}
\begin{Verbatim}
\begin{desclist}{\rm\hfill\textemdash}\[Mark\]
\end{Verbatim}
\end{verbatim}

\texttt{\textbackslash NCC\textbackslash dlist} The internal definition of the \texttt{desclist} is based on the \texttt{\list}:

\begin{verbatim}
\begin{Verbatim}
\def\NCC\textbackslash dlist#1[#2][#3]{{\NCC\textbackslash dlist#1[#2]}}\end{Verbatim}
\end{verbatim}

7 \texttt{\def\NCC\textbackslash dlist#1[#2][#3]}\%
8 \texttt{\list}\%
Specify just the same list parameters as in the \texttt{description} if the optional parameter is empty:

\begin{verbatim}
  \def\@tempa{#3}\ifx\@tempa\@empty
    \labelwidth\z@ \itemindent-\leftmargin
  \else
    \settowidth\leftmargin{#1#3{#2}}\%
    \advance\leftmargin\labelsep
    \labelwidth=\leftmargin
    \itemindent\z@ \fi
  \fi
\end{verbatim}

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

\begin{verbatim}
  \labelwidth\z@ \itemindent-\leftmargin
  \settowidth\leftmargin{#1#3{#2}}\%
  \advance\leftmargin\labelsep
  \labelwidth=\leftmargin
  \itemindent\z@ \fi
\end{verbatim}

Finally, define the \texttt{makelabel} command adding the desclist prefix and suffix and finishing it with \texttt{\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):

\begin{verbatim}
  \def\makelabel##1{\hspace\labelsep #1##1{#2}\hfil}\
\end{verbatim}

\texttt{description} Now we redefine the \texttt{description} environment. We do not test is it defined or not.

\begin{verbatim}
  \def\description{\desclist{\normalfont\bfseries}{} \}
  \let\enddescription=\enddesclist
\end{verbatim}