The lstautogobble Package

Martin Scharrer
martin.scharrer@web.de

Version v0.1 – 2012/05/03

CTAN: https://www.ctan.org/pkg/lstautogobble

VC: https://github.com/MartinScharrer/lstaddons/

Abstract

This add-on package to `listings` provides a boolean `autogobble` setting which will automatically set the `gobble` setting to indention of the first line.

This package was created as response to the question "How to automatically skip leading white spaces in listings?" on TEX Stack Exchange.

1 Introduction

The `listings` package has a setting `gobble`=⟨*number*⟩ which allows to remove a certain number of characters from the beginning of every line in the listing. This can be used to indent the listing in the source code without affecting the printed result. However, this forces the user to set a suitable value manually. An incorrect value will lead either to an indented listing or to missing leading characters.

A solution for this is to automatically detect the used indention of the listing and that the `gobble` setting to this value. This functionality is provided by this package. For this it reads and scans the first listing line and reinserts it again afterwards.

2 Usage

After loading `lstautogobble` the following new `listings` setting is available:

```
autogobble=true|false
```

This boolean setting switches the autogobble feature on or off. If no value is used the default is '`true`'. The initial setting is '`false`'. One enabled the first line of any `lstlisting` is scanned and the amount of spaces or tabulators is used to set the `gobble` setting. If `gobble` is set manually it will not be overwritten and `autogobble`=`true` is ignored.

3 Examples / Tests

The following code is intended as examples and also for testing the package. Here the `autogobble` feature is globally enabled.

```
test
it
```

```
\begin{lstlisting}
  test
  it
\end{lstlisting}
```

Example 1: Only environment (with `autogobble` enabled globally).

```
test
it
```

```
\begin{lstlisting}[basicstyle=\ttfamily\scriptsize]
    test
    it
\end{lstlisting}
```

Example 2: With options (must be skipped and reinserted).

```
est
t
```

```
\begin{lstlisting}[gobble=7]
      test
      it
\end{lstlisting}
```

Example 3: Manual gobble option (override). Intentionally set to an incorrect value.

```
    test
    it
```

```
\begin{lstlisting}[autogobble=false]
    test
    it
\end{lstlisting}
```

Example 4: Locally turned-off `autogobble`.

```
test
it
```

```
\begin{lstlisting} some text at the first line
    test
    it
\end{lstlisting}
```

Example 5: With some material on the same line as `\begin` (dropped by listings. The warning message got preserved).

```
test
it
```

```
\begin{lstlisting}[] some text at the first line
    test
    it
\end{lstlisting}
```

Example 6: As before, but with optional argument.

```
test
it
```

```
\begin{lstlisting}
    test
    it
\end{lstlisting}
```

```
test
it
```

```
    \begin{lstlisting}
        test
        it
    \end{lstlisting}
```

```
test
it
```

```
    \begin{lstlisting}
test
it
    \end{lstlisting}
```

```
test
it
```

```
       \begin{lstlisting}
                     test
                     it
       \end{lstlisting}
```

```
test
it
```

```
\begin{lstlisting}
              test
              it
\end{lstlisting}
```

Example 7: Different indention levels.

```
#include <stdio.h>
int main(){
  printf("tex.stackexchange.com: the coolest community ever!\n");
}
```

```
\begin{lstlisting}[autogobble]
  #include <stdio.h>
  int main(){
    printf("tex.stackexchange.com: the coolest community ever!\n⁄
        ");
  }
\end{lstlisting}
```

Example 8: Some real C Code.

4 Implementation

```
%<!COPYRIGHT>
\ProvidesPackage{lstautogobble}[%
%<!DATE>
%<!VERSION>
%<*DRIVER>
    2099/01/01 develop
%</DRIVER>
    Implements 'autogobble' option for 'listings']


% This is an add-on to the `listings` package
\RequirePackage{listings}

% Counter for leading spaces
\newcount\lstag@spacecount

% Some macros for comparison:
\def\lstag@activespace{\lst@ProcessSpace}%  Definition of an active space
\def\lstag@tabulator{\lst@ProcessTabulator}%  Definition of an tabulator

\begingroup
\catcode`\^^M=\active%
\gdef\lstag@activenl{^^M}%  Active CR (ASCII 13) character which is used as line break
\endgroup


% Define `autogobble` option as boolean (by default off)
\lst@Key{autogobble}{false}[t]{\lstKV@SetIf{#1}\lst@ifautogobble}

% `ungobble` option
\lst@Key{ungobble}{0}{\def\lst@ungobble{#1}}

% Insert required code at environment init
\lst@AddToHook{Init}{\lst@autogobble}

% Autogobble init macro.
% If the option is active and `gobble` is not set, init vars and overwrite the process macro with own definition.
\def\lst@autogobble{%
    \lst@ifautogobble
        \ifnum\lst@gobble>0\else
            \def\lst@gobble{\lstag@gobble}%
            \def\lstag@gobble{0}%
            \lstag@spacecount\z@
            \def\lstag@spaceaccu{}%
            \let\lstag@restofline\empty
            \let\lstag@origlstenv@Process\lstenv@Process
            \let\lstenv@Process\lstag@countleadingspaces
        \fi
    \fi
}

% Checks if the next following character (read as argument) is a line break (as it is supposed to be)
% Otherwise there is some text direct after the ‘\begin{<env>}[<options>]‘ which is dropped by ‘listings‘.
\def\lstag@countleadingspaces#1{%
    \expandafter\ifx\lstag@activenl#1\relax
        \expandafter\lstag@countleadingspaces@
    \else
        \def\lstag@restofline{Dummy replacement of text after begin of listing to trigger original warning message}%
        \expandafter\lstag@countleadingspaces
    \fi
}

% After the new line is found this macro counts the spaces and tabulators
\def\lstag@countleadingspaces@#1{%
    \ifx\lstag@activespace#1\relax
        \advance\lstag@spacecount by \@ne
        % Accumulate spaces (i.e. their definitions) for later re-insertion:
        \expandafter\def\expandafter\lstag@spaceaccu\expandafter{\lstag@spaceaccu\lst@ProcessSpace}%
        \let\next\lstag@countleadingspaces@
    \else% Character wasn't a space
    \ifx\lstag@tabulator#1\relax
        \advance\lstag@spacecount by \lst@tabsize\relax
        % Accumulate spaces (i.e. their definitions) for later re-insertion:
        \@tempcnta=\lst@tabsize\relax
        \loop
        \ifnum\@tempcnta>\z@
            \expandafter\def\expandafter\
               lstag@spaceaccu\expandafter{\↙
               lstag@spaceaccu\lst@ProcessSpace}%
            \advance\@tempcnta\m@ne
        \repeat
        \let\next\lstag@countleadingspaces@
    \else% Character wasn't a tabulator either
        % Set gobble option (indirect):
        \xdef\lstag@gobble{\the\numexpr\↙
           lstag@spacecount-\lst@ungobble\relax}%
        % Restore original definition of process ↙
           macro:
        \global\let\lstenv@Process\↙
           lstag@origlstenv@Process
        % Re-insert all collected material or ↙
           appropriate replacement material:
        \edef\next{\noexpand\lstenv@Process\↙
           lstag@restofline\expandafter\noexpand\↙
           lstag@activenl\expandafter\unexpanded\↙
           expandafter{\lstag@spaceaccu}\noexpand#1}%
    \fi\fi
    \next
}
```