The everyhook package

Stephen Checkoway
s@cs.jhu.edu

November 26, 2014

Abstract

The everyhook package takes control of the six \TeX{} token parameters \texttt{\everypar}, \texttt{\everymath}, \texttt{\everydisplay}, \texttt{\everyhbox}, \texttt{\everyvbox}, and \texttt{\everycr}. Real hooks for each of these can be installed using a stack like interface. For compatibility with \TeX{} standard classes and packages, each of the \texttt{\everyX} token lists can be set without interfering with the hooks.

Contents

1 Introduction 1

2 Usage 2

2.1 Options ............................................. 3

2.2 Manipulating hooks ............................... 3

3 Example 3

4 Potential pitfalls 4

5 Implementation 6

Change History 11

Index 11

1 Introduction

\TeX{} contains nine token parameters, seven of which are inserted into the current list at various times. Quoting from \textit{The \TeX{}book}, the seven token parameters of interest are\footnote{This document corresponds to everyhook v1.2, dated 2014/11/26.}

\texttt{\everypar} tokens to insert when a paragraph begins,
\texttt{\everymath} tokens to insert when math in text begins,
\everydisplay tokens to insert when display math begins,  
\everyhbox tokens to insert when an hbox begins,  
\everyvbox tokens to insert when a vbox begins,  
\everyjob tokens to insert when the job begins, and  
\everycr tokens to insert after every \cr or nonredundant \crcr.

Of these, \everyjob is not very useful outside of INITEX and so it won’t be considered further.

The remaining six token parameters can be used to great effect. For example, the \everypar is used in \paragraph to set the title of the paragraph inline allowing constructions like

\paragraph{Paragraph title.}  \textbf{Paragraph title.} A blank line followed by the rest of the paragraph.

which work properly rather than starting a new paragraph due to the blank line.

Similarly, \everymath and \everydisplay are used by the \LaTeX{} kernel to set up math fonts.

Using the \TeX{} primitives directly has the major downside that they cannot be used by multiple packages at the same time. Setting \everypar overwrites a prior usage. Even if one package is careful and always uses

\everypar={\expandafter{\the \everypar new tokens here}}

so as not to stomp on another’s usage, there’s no guarantee that the other package will not later set \everypar={}.  

To get around this, the everyhook package takes control of the six \everyX primitives listed above and for each one provides a stack like interface for two additional token lists, one to be expanded before the \everyX and one to be expanded after. For example,

\PushPreHook{hbox}{{1}}  
\PushPreHook{hbox}{{2}}  
\everyhbox={3}  
\PushPostHook{hbox}{{4}}  
\PushPostHook{hbox}{{5}}

will cause the insertion of the tokens 21345 at the start of an \hbox. Note that \PushPreHook adds tokens to the left of the list of tokens to appear before those in \everyhbox whereas \PushPostHook adds tokens to the right of the list of tokens to appear after those in \everyhbox.

2 Usage

The everyhook package has one (rather experimental) option, \excludeor and is loaded using

\usepackage[excludeor]{everyhook}

or
as required where the option is, of course, optional.

2.1 Options

excludeor Some of the hooks described below can cause unwanted behavior when active during the execution of \LaTeX{}’s output routine. The experimental excludeor option saves and clears the hooks at the beginning of the output routine and restores them at the end.

2.2 Manipulating hooks

There are 12 hooks, a pre and post hook for each of the six token parameters par, math, display, hbox, vbox, and cr. The first argument to all of the macros described in this section must be one of these six. All hook manipulation is global.

Pre hooks. Additional tokens (balanced text) are prepended to the pre hook \( \langle \text{hook} \rangle \) using \PushPreHook{\text{hook}}{\langle \text{balanced text} \rangle}. The most recently pushed tokens can be popped off using \PopPreHook{\langle \text{hook} \rangle}.

Post hooks. Additional tokens (balanced text) are appended to the post hook \( \langle \text{hook} \rangle \) using \PushPostHook{\text{hook}}{\langle \text{balanced text} \rangle}. The most recently pushed tokens can be popped off using \PopPostHook{\langle \text{hook} \rangle}.

Saving, restoring, and clearing hooks. Each of the 12 pre and post hooks can be saved to a macro, restored from a macro, or cleared independently. To save the pre hook \( \langle \text{hook} \rangle \) to the macro \texttt{cs}, use \SavePreHook{\text{hook}}{\texttt{cs}}. Restoring is accomplished by \RestorePreHook{\text{hook}}{\texttt{cs}}. To clear all of the tokens in a pre hook use \ClearPreHook{\text{hook}}. The \SavePostHook, \RestorePostHook, and \ClearPostHook are analogous.

3 Example

As a nontrivial example of where this package can be used, consider the following example.

\documentclass{article}
\usepackage{everyhook}
\usepackage{lipsum}

\begin{document}
\setlength{\parindent}{0pt}
\PushPreHook{par}\llap{\textbullet}\enskip\null
\paragraph{lorem ipsum}
\lipsum[1-4]
\PopPreHook{par}
\end{document}
This code will cause each paragraph of the *lorem ipsum* text to have no indentation and instead to place a bullet in the margin. See Figure 1. If \everypar were used instead, the \paragraph would replace the command to create the bullet with those needed to typeset the paragraph title.

Note that this package is not a panacea. We had to add a \null to the par hook because \paragraph uses \lastbox to remove the indentation box. Without the \null it ends up removing the box constructed by \llap instead.

Using the post par hook solves the \lastbox problem, but then the bullet is placed to the right of the \paragraph title.

Perhaps a better way to solve this problem is to remove the indentation box first, insert the bullet, and then place the box after. In this way, the bullet is always to the left of the paragraph indentation.

\PushPreHook{par}{\setbox{\lastbox}
  \llap{\textbullet \enskip \box{\null}}}

4 Potential pitfalls

As noted in the previous section, it can be tricky to use the par hook correctly. This section contains an (almost certainly) incomplete list of pitfalls to watch out for when using everyhook.

1. When using the par hooks, be aware that TeX will insert a box with the width of \parindent before the tokens in the pre hook. One way to handle this is to propogate the box to the right.

2. It is probably not a good idea to use the hbox, vbox, and par hooks at any place where TeX’s output routine is likely to run. The excludeor option should help with this, but it might cause problems with other packages that also modify the output routine.

3. \LaTeX’s kernel takes control of the \everymath and \everydisplay token parameters to make its own adjustments in much the same way this package does. The trace package uses the kernel’s private macros to insert its own hooks. It is probably best to only use the post math and display hooks to ensure that the kernel has done what it needs to do before you start typesetting stuff in math mode.

4. When using the hbox and vbox hooks, any hbox or vbox that appears in a \setbox will have the \afterassignment token inserted before the hooks. This is no different from TeX’s normal behavior with \afterassignment and \everyhbox/\everyvbox, but can be surprising.

5. I’m sure there are others.


Figure 1: Example output.
5 Implementation

The package begins with the usual package identification and then it declares the one option `excludeor` and finally it loads the `etoolbox` package. This package is not strictly necessary, but it does simplify some stuff and provides handy macros for dealing with control sequence names.

\begin{verbatim}
\NeedsTeXFormat{LaTeX2e}[1999/12/01]
\RequirePackage{svn-prov}
\ProvidesPackage{svn}{DidZ everyhookNdtx 1R RP1T-11-RV 15ZSTZ5Vz steve D}
\DeclareOption{excludeor}{E
\AtBeginDocument{E
\output\expandafter{E
\eh@saveclearallhooks
\the\output
\eh@restoreallhooks
}%
}%
\ProcessOptions}\relax
\RequirePackage{etoolbox}
\end{verbatim}

\begin{verbatim}
\def\eh@definehook{E
\cslet{eh@every#1}{E
\newtoks#2{E
\cslet{eh@private#1}{E
#2\csuse{eh@every#1}{E
\csdef{eh@pre#1}{}{E
\csdef{eh@post#1}{}{E
}%
}%
}%
\expandafter{\csname eh@pre#1\expandafter\endcsname
\expandafter{\csname eh@private#1\expandafter\endcsname
\csname eh@post#1\endcsname}E
\everypar
\end{verbatim}

This performs all of the setup work for each hook. First, it takes control of \TeX's token parameter given in the second argument. Then it shadows the name of the primitive with a normal token register (and copies the current definition). The pre and post hooks are defined to be initially empty.

This is slightly tricky to get right. Basically, we want to set the `everyfoo` primitive which we have saved as `eh@everyfoo` like 

\begin{verbatim}
\eh@everyfoo={\eh@prefoo\the\expandafter\everyfoo\eh@postfoo}.
\end{verbatim}

The reason for the \texttt{\expandafter} is to make sure it is expanded before the the token register \texttt{\everyfoo} is expanded. Thus if the post hook is empty, then code in \texttt{\everyfoo} sees no additional tokens, in case that is important.

Unfortunately, some code wants to redefine \texttt{\everyfoo} itself in order to prevent other code that uses \texttt{\everyfoo} from actually setting anything. To deal with that, we use the private token list

\begin{verbatim}
\eh@everyfoo={\eh@prefoo\the\expandafter\everyfoo\eh@postfoo}
\csuse{eh@every#1}\expandafter{\csname eh@pre#1\expandafter\endcsname
\expandafter{\csname eh@private#1\expandafter\endcsname
\csname eh@post#1\endcsname}%
\end{verbatim}

\everypar Define the hooks for the \par hook.
\@definehook{par}\everypar

\frozen@everymath
\frozen@everydisplay

Define the math and display hooks. Since the \LaTeX kernel has already saved \everymath
and \everydisplay into the frozen macros, we take control by redefining the frozen ones
instead.

\@definehook{math}\frozen@everymath
\@definehook{display}\frozen@everydisplay

\everyhbox
\everyvbox
\everycr

Define the hbox, vbox, and cr hooks and free up some used memory.

\@definehook{hbox}\everyhbox
\@definehook{vbox}\everyvbox
\@definehook{cr}\everycr
\@undef\@definehook

\@hookseparator
An separator used to separate tokens in each hook.

\@checkhook
Check that the hook is one of the six.

\@checkhooknotempty
Check that the hook is both defined and not empty so that we can pop.

\newrobustcmd\pushprehook
Prepend tokens to the pre hook, separated via the separator.

\newrobustcmd\popprehook
Pop the hook is not empty, and then pop off the left tokens and separator. We can
use delimited parameters to strip off the first set of tokens.
Append a separator and tokens to the post hook.

Check that the post hook is not empty. Then, iterate over the tokens in the list until we reach the end and strip that off.

Reset the pre/post hook to empty.

Internal hook reset.

Internal macros to \let the hook to the supplied control sequence to save. Perform the \let in the other direction to restore.
User macros to save and restore hooks.

```
\newrobustcmd\SavePreHook[2]{%}
  \eh@checkhook(1)\SavePreHook
  \eh@savehook(pre1)#2%
  \eh@checkhook(1)\SavePreHook}
\newrobustcmd\SavePostHook[2]{%}
  \eh@checkhook(1)\SavePostHook
  \eh@savehook(post1)#2%
  \eh@checkhook(1)\SavePostHook}
\newrobustcmd\RestorePreHook[2]{%}
  \eh@checkhook(1)\RestorePreHook
  \eh@restorehook(pre1)#2%
  \eh@checkhook(1)\RestorePreHook}
\newrobustcmd\RestorePostHook[2]{%}
  \eh@checkhook(1)\RestorePostHook
  \eh@restorehook(post1)#2%
  \eh@checkhook(1)\RestorePostHook}
```

Internal macros to save and clear (resp. restore) all hooks at the start (resp. end) of the output routine.

```
\def\eh@saveclearallhooks{%
  \global\eh@savehook{prepar}\eh@or@prepar
  \global\eh@savehook{postpar}\eh@or@postpar
  \global\eh@savehook{premath}\eh@or@premath
  \global\eh@savehook{postmath}\eh@or@postmath
  \global\eh@savehook{predisplay}\eh@or@predisplay
  \global\eh@savehook{postdisplay}\eh@or@postdisplay
  \global\eh@savehook{prehbox}\eh@or@prehbox
  \global\eh@savehook{posthbox}\eh@or@posthbox
  \global\eh@savehook{prevbox}\eh@or@prevbox
  \global\eh@savehook{postvbox}\eh@or@postvbox
  \global\eh@savehook{precr}\eh@or@precr
  \global\eh@savehook{postcr}\eh@or@postcr
  \eh@clearhook(prepar)%
  \eh@clearhook(postpar)%
  \eh@clearhook(premath)%
  \eh@clearhook(postmath)%
  \eh@clearhook(predisplay)%
  \eh@clearhook(postdisplay)%
  \eh@clearhook(prehbox)%
  \eh@clearhook(posthbox)%
  \eh@clearhook(prevbox)%
  \eh@clearhook(postvbox)%
  \eh@clearhook(precr)%
  \eh@clearhook(postcr)%
}
\def\eh@restoreallhooks{%
  \eh@restorehook(prepar)\eh@or@prepar
  \eh@restorehook(postpar)\eh@or@postpar
```
Change History

v1.0
General: Initial version .......................... 6

v1.1
General: Add excludeor .......................... 6

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

\_ \_ \_ .......................... 5
A \AtBeginDocument .................. 7
C \ClearPostHook ................. 90
\ClearPreHook ................. 90
D \DeclareOption .................. 6
E \eh@checkhook .............. 36, 42, 48, 66,
91, 95, 105, 109, 113, 117
\eh@checkhooknotempty ...... 41, 59, 73
\eh@clearhook ............... 87, 92, 96, 133–144
\eh@definehook ...... 17, 28–34
\eh@hookend 62, 64, 77, 80, 84
\eh@hookseparator ... 35,
54, 64, 69, 77, 80, 84
\eh@or@postcr ....... 132, 158
\eh@or@postdisplay 126, 152
\leh@or@postbox ... 128, 154
\leh@or@postmath ... 124, 150
\leh@or@postpar ... 122, 148
\leh@or@postvbox ... 130, 156
\leh@or@precrr ...... 131, 157
\leh@or@predisplay ... 125, 151
\leh@or@prehbox ... 127, 153
\leh@or@premath ... 123, 149
\leh@or@prepar ... 121, 147
\leh@or@prevbox ... 129, 155
\leh@popposthook ....... 72
\leh@popprehook ........ 58
\leh@restoreallhooks 11, 120
\leh@restorehook ....... 98, 114, 118, 147–158
\leh@saveclearallhooks 9, 120
\leh@savehook ........... 98, 106, 110, 121–132
\leh@sentinel ............ 72
\leh@tempi ............. 49, 53, 55, 67,
69, 70, 74, 76, 78, 81–83
\leh@tempii ...... 50, 54, 56
\ everycr ............ 31
\everyhbox .............. 31
\everypar ............ 28
\everyvbox .............. 31
\frozen@everydisplay ... 20
\frozen@everymath ..... 20
\NeedsTeXFormat ........... 1
\output .................. 8, 10
\PopPostHook .......... 72
\PopPreHook .......... 58
\ProcessOptions ....... 15
\ProvidesPackageSVN .... 3
\PushPostHook .......... 65
\PushPreHook .......... 47
\relax .................. 15
\restorePostHook ...... 104
\restorePreHook ...... 104
\revinfo ............... 5
\SavePostHook ...... 104
\SavePreHook ...... 104

S
\SavePostHook ...... 104
\SavePreHook ...... 104

11