

The everyhook package*

Stephen Checkoway
s@cs.ucsd.edu

February 4, 2011

Abstract

The everyhook package takes control of the six TeX token parameters `\everypar`, `\everymath`, `\everydisplay`, `\everyhbox`, `\everyvbox`, and `\everycr`. Real hooks for each of these can be installed using a stack like interface. For compatibility with L^AT_EX standard classes and packages, each of the `\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 *The TeXbook*, the seven token parameters of interest are¹

`\everypar` tokens to insert when a paragraph begins,
`\everymath` tokens to insert when math in text begins,

*This document corresponds to everyhook v1.1, dated 2011/02/04.

¹The remaining two token parameters are `\output` and `\errhelp`.

`\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 `\crr`.

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

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 L^AT_EX kernel to set up math fonts.

Using the T_EX 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

`\RequirePackage[excludeor]{everyhook}`

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 L^AT_EX'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*.**

`\PushPreHook` **Pre hooks.** Additional tokens *<balanced text>* are prepended to the pre hook *<hook>* using
`\PopPreHook` `\PushPreHook{<hook>}{<balanced text>}`. The most recently pushed tokens can be popped
off using `\PopPreHook{<hook>}`.

`\PushPostHook` **Post hooks.** Additional tokens *<balanced text>* are appended to the post hook *<hook>* using
`\PopPostHook` `\PushPostHook{<hook>}{<balanced text>}`. The most recently pushed tokens can be popped
off using `\PopPostHook{<hook>}`.

`\SavePreHook` **Saving, restoring, and clearing hooks.** Each of the 12 pre and post hooks can be
`\SavePostHook` saved to a macro, restored from a macro, or cleared independently. To save the pre
`\RestorePreHook` hook *<hook>* to the macro `\cs`, use `\SavePreHook{<hook>}{\cs}`. Restoring is accom-
`\RestorePostHook` plished by `\RestorePreHook{<hook>}{\cs}`. To clear all of the tokens in a pre hook use
`\ClearPreHook` `\ClearPreHook{<hook>}`. The `\SavePostHook`, `\RestorePostHook`, and `\ClearPostHook` are
`\ClearPostHook` 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}{\setbox0=\lastbox
\llap{\textbullet\enskip}\box0}}
```

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 propagate 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. L^AT_EX'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 `postmath` 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.

- **Lorem ipsum.** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
- Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.
- Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.
- Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

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.

```

1 \NeedsTeXFormat{LaTeX2e}[1999/12/01]
2 \RequirePackage{svn-prov}
3 \ProvidesPackageSVN
4     {$Id: everyhook.dtx 10 2011-02-04 06:43:36Z steve $}
5     [v1.1 \reinfo\ Hooks for low level TeX everyX primitives.]
6 \DeclareOption{excludeor}{%
7     \AtBeginDocument{%
8         \output\expandafter{%
9             \expandafter\eh@saveclearallhooks
10            \the\output
11            \eh@restoreallhooks
12        }%
13    }%
14 }
15 \ProcessOptions\relax
16 \RequirePackage{etoolbox}

```

`\eh@definehook` 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 shows 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.

```

17 \def\eh@definehook#1#2{%
18     \cslet{eh@every#1}#2%
19     \newtoks#2%
20     #2\csuse{eh@every#1}%
21     \csdef{eh@pre#1}{}%
22     \csdef{eh@post#1}{}%

```

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

```
\eh@everyfoo={\eh@prefoo\the\expandafter\everyfoo\eh@postfoo}.
```

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

```

23     \csuse{eh@every#1}\expandafter{\csname eh@pre#1\expandafter\endcsname
24         \expandafter\the\expandafter\expandafter\expandafter#2%
25         \csname eh@post#1\endcsname}%
26 }

```

`\everypar` Define the hooks for the par hook.

```
27 \eh@definehook{par}\everypar
```

`\frozen@everymath` 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.

```

28 \eh@definehook{math}\frozen@everymath
29 \eh@definehook{display}\frozen@everydisplay

```

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

```

\everyvbox 30 \eh@definehook{hbox}\everyhbox
\everycr   31 \eh@definehook{vbox}\everyvbox
           32 \eh@definehook{cr}\everycr
           33 \undef\eh@definehook

```

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

```

34 \def\eh@hookseparator{}

```

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

```

35 \def\eh@checkhook#1#2{%
36     \ifcsdef{eh@every#1}{ }\PackageError{everyhook}{Argument #1 to
37     \protect#2\space is invalid}{There is no hook for
38     \protect\every#1.}%
39 }

```

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

```

40 \def\eh@checkhooknotempty#1#2#3{%
41     \eh@checkhook{#2}#3%
42     \ifcempty{eh@#1#2}{\PackageError{everyhook}{The #1 hook for
43     \protect\every#2\space is empty}{I have seen too many
44     \protect#3{#2}s.}}}%
45 }

```

`\PushPreHook` Prepend tokens to the pre hook, separated via the separator.

```

46 \newrobustcmd\PushPreHook[2]{%
47     \eh@checkhook{#1}\PushPreHook
48     \def\eh@temp{i}{#2}%
49     \letcs\eh@temp{i}{eh@pre#1}%
50     \expandafter\gdef\csname eh@pre#1\expandafter\expandafter
51     \expandafter\endcsname\expandafter\expandafter
52     \expandafter{\expandafter\eh@temp{i}\expandafter
53     \eh@hookseparator\eh@temp{i}}%
54     \undef\eh@temp{i}
55     \undef\eh@temp{i}
56 }

```

`\PopPreHook` Check that the hook is not empty, and then pop off the left tokens and separator. We can
`\eh@popprehook` use delimited parameters to strip off the first set of tokens.

```

57 \newrobustcmd\PopPreHook[1]{%
58     \eh@checkhooknotempty{pre}{#1}\PopPreHook
59     \expandafter\eh@popprehook\csname eh@pre#1\expandafter
60     \expandafter\expandafter\endcsname
61     \csname eh@pre#1\endcsname\eh@hookend
62 }
63 \def\eh@popprehook#1#2\eh@hookseparator#3\eh@hookend{\gdef#1{#3}}

```

```

\PushPostHook Append a separator and tokens to the post hook.
64 \newrobustcmd\PushPostHook[2]{%
65     \eh@checkhook{#1}\PushPostHook
66     \letcs\eh@temp{eh@post#1}%
67     \expandafter\gdef\csname eh@post#1\expandafter\endcsname
68         \expandafter{\eh@temp\eh@hookseparator#2}%
69     \undef\eh@temp
70 }

\PopPostHook Check that the post hook is not empty. Then, iterate over the tokens in the list until we
\eh@popposthook reach the end and strip that off.
\eh@sentinel
71 \newrobustcmd\PopPostHook[1]{%
72     \eh@checkhooknotempty{post}{#1}\PopPostHook
73     \letcs\eh@temp{eh@post#1}%
74     \expandafter\eh@popposthook\csname eh@post#1\expandafter
75         \endcsname\expandafter{\expandafter}\eh@temp
76         \eh@hookend\eh@hookseparator\eh@sentinel\eh@hookend
77     \undef\eh@temp
78 }
79 \def\eh@popposthook#1#2\eh@hookseparator#3\eh@hookseparator#4\eh@hookend{%
80     \def\eh@temp{#4}%
81     \ifdefequal\eh@sentinel\eh@temp%
82         {\gdef#1{#2}\undef\eh@temp}%
83         {\eh@popposthook#1{#2\eh@hookseparator#3}\eh@hookseparator#4\eh@hookend}%
84 }
85 \def\eh@sentinel{\eh@sentinel}

\eh@clearhook Internal hook reset.
86 \def\eh@clearhook#1{%
87     \global\csdef{eh@#1}{}%
88 }

\ClearPreHook Reset the pre/post hook to empty.
\ClearPostHook
89 \newrobustcmd\ClearPreHook[1]{%
90     \eh@checkhook{#1}\ClearPreHook
91     \eh@clearhook{pre#1}%
92 }
93 \newrobustcmd\ClearPostHook[1]{%
94     \eh@checkhook{#1}\ClearPostHook
95     \eh@clearhook{post#1}%
96 }

\eh@savehook Internal macros to \let the hook to the supplied control sequence to save. Perform the
\eh@restorehook \let in the other direction to restore.
97 \def\eh@savehook#1#2{%
98     \letcs#2{eh@#1}%
99 }
100 \def\eh@restorehook#1#2{%
101     \global\cslet{eh@#1}#2%
102 }

```



```

\SavePreHook      User macros to save and restore hooks.
\SavePostHook    103 \newrobustcmd\SavePreHook[2]{%
\RestorePreHook  104     \eh@checkhook{#1}\SavePreHook
\RestorePostHook 105     \eh@savehook{pre#1}#2%
                  106 }
                  107 \newrobustcmd\SavePostHook[2]{%
                  108     \eh@checkhook{#1}\SavePostHook
                  109     \eh@savehook{post#1}#2%
                  110 }
                  111 \newrobustcmd\RestorePreHook[2]{%
                  112     \eh@checkhook{#1}\RestorePreHook
                  113     \eh@restorehook{pre#1}#2%
                  114 }
                  115 \newrobustcmd\RestorePostHook[2]{%
                  116     \eh@checkhook{#1}\RestorePostHook
                  117     \eh@restorehook{post#1}#2%
                  118 }

```

`\eh@saveclearallhooks` Internal macros to save and clear (resp. restore) all hooks at the start (resp. end) of the
`\eh@restoreallhooks` output routine.

```

119 \def\eh@saveclearallhooks{%
120     \global\eh@savehook{prepar}\eh@or@prepar
121     \global\eh@savehook{postpar}\eh@or@postpar
122     \global\eh@savehook{premath}\eh@or@premath
123     \global\eh@savehook{postmath}\eh@or@postmath
124     \global\eh@savehook{predisplay}\eh@or@predisplay
125     \global\eh@savehook{postdisplay}\eh@or@postdisplay
126     \global\eh@savehook{prehbox}\eh@or@prehbox
127     \global\eh@savehook{posthbox}\eh@or@posthbox
128     \global\eh@savehook{prevbox}\eh@or@prevbox
129     \global\eh@savehook{postvbox}\eh@or@postvbox
130     \global\eh@savehook{precr}\eh@or@precr
131     \global\eh@savehook{postcr}\eh@or@postcr
132     \eh@clearhook{prepar}%
133     \eh@clearhook{postpar}%
134     \eh@clearhook{premath}%
135     \eh@clearhook{postmath}%
136     \eh@clearhook{predisplay}%
137     \eh@clearhook{postdisplay}%
138     \eh@clearhook{prehbox}%
139     \eh@clearhook{posthbox}%
140     \eh@clearhook{prevbox}%
141     \eh@clearhook{postvbox}%
142     \eh@clearhook{precr}%
143     \eh@clearhook{postcr}%
144 }
145 \def\eh@restoreallhooks{%
146     \eh@restorehook{prepar}\eh@or@prepar
147     \eh@restorehook{postpar}\eh@or@postpar
148     \eh@restorehook{premath}\eh@or@premath
149     \eh@restorehook{postmath}\eh@or@postmath
150     \eh@restorehook{predisplay}\eh@or@predisplay

```

```
151     \eh@restorehook{postdisplay}\eh@or@postdisplay
152     \eh@restorehook{prehbox}\eh@or@prehbox
153     \eh@restorehook{posthbox}\eh@or@posthbox
154     \eh@restorehook{prevbox}\eh@or@prevbox
155     \eh@restorehook{postvbox}\eh@or@postvbox
156     \eh@restorehook{prec}\eh@or@prec
157     \eh@restorehook{postcr}\eh@or@postcr
158 }
159 \endinput
```

Change History

v1.0		v1.1	
General: Initial version 6	General: Add excludeor 6