

Contents

1	Hook Management	2
1.1	add/remove	2
1.2	hook label	2
1.3	One-off hook	2
1.4	hook in pairs	3
2	Generic Hooks	3
2.1	environment hooks	3
2.2	command hooks	3
3	File Hooks	3
4	Shipout Hooks	4
4.1	Background and Foreground	4
4.2	tikz code	4
4.3	Other Shipout Hooks	5
5	Declaring Hooks	6
6	Appendix	6

1 Hook Management

1.1 add/remove

```

1 % Original: Content
2
3 \AddToHook{cmd/cmda/before}{Before\ }
4 % Before Content
5
6 \AddToHook{cmd/cmda/after}{\ After}
7 % Before Content After
8
9 \RemoveFromHook{cmd/cmda/before}
10 % Content After
11
12 \RemoveFromHook{cmd/cmda/after}
13 % Content

```

1.2 hook label

Original Content: Before-1 Before-2 Content, the below contents listing the result of each hook label

```

1 \RemoveFromHook{cmd/cmda/before}
2 % Before-1 Before-2 Content
3
4 \RemoveFromHook{cmd/cmda/before}[bf-1]
5 % Before-2 Content
6
7 \RemoveFromHook{cmd/cmda/before}[bf-2]
8 % Before-1 Content

```

1.3 One-off hook

Execute code only once, i.e., just the next time a hook is called. Because this is one-off code, it is not labeled.

```

1 \AddToHookNext{hook}{code}
2 \ClearHookNext{hook}

```

A simple example:

```

1 \newcommand\cmda{Content}
2 \AddToHook{cmd/cmda/before}[bf-1]{Before-1\ }
3 \AddToHook{cmd/cmda/before}[bf-2]{Before-2\ }
4 \AddToHookNext{cmd/cmda/before}{Before-Once\ }
5 \cmda\par
6 \cmda\par
7 % Before-1 Before-2 Before-Once Content
8 % Before-1 Before-2 Content

```

Typical use cases for `\AddToHookNext` are the hooks related to shipping out pages; e.g., you may want to use a special background on the next page.

1.4 hook in pairs

Some hooks (whether normal or one-time) come in pairs, for example, the hooks `file/before` and `file/after`, which are executed before and after the loading of every file. The second of such hooks is called **Reverse Hook**, means: the execution order in the reversed hook is exactly the opposite

2 Generic Hooks

There is one further category: some hooks are so called “**Generic Hooks**”. Normally a hook has to be explicitly declared before it can be used in code.

2.1 environment hooks

Every environment offers a set of four generic hooks.

- Outer hooks: `before` and `after`
- Inner hooks: `begin` and `end`

Only the `/after` hook is implemented as a reversed hook; Generic environment hooks are never one-time hooks even with environments that are supposed to appear only once in a document.

The hooks are executed only if `\begin{env}` and `\end{env}` are used. If the environment code is executed via low-level calls to `\<env>` and `\end<env>` (e.g., to avoid the environment grouping), they are not available. If you want them available in code using this method, you would need to add them yourself, i.e., write something like

```
1 \UseHook{env/quote/before}\quote
2 ...
3 \endquote\UseHook{env/quote/after}
```

to add the outer hooks, etc.

2.2 command hooks

Similar to environments there are two generic hooks available for any \LaTeX (document-level) command — in theory at least. In practice there are restrictions, and especially the `/after` hooks work only with a subset of commands.

3 File Hooks

An example:

```
1 \AddToHook{package/amsmath/after}{%
2   \AddToHook{cmd/colon/after}{\!}%
3 }
```

This is essential to make our this example involving `package/amsmath/after` work: if the package was already loaded, then the code in `immediately` applied. If we had used a file hook instead (which is a normal hook), the code would have been stored away, waiting forever for a second invocation that would never happen.

Some usefule hooks:

- `file/before`, `file/<file>/before`, `file/<file>/after`, `file/after`
- `package/before`, `package/<name>/before`, `package/<name>/after`, `package/after`
- `class/before`, `class/<name>/before`, `class/<name>/after`, `class/after`

As you may have guessed, the last two are reversed hooks. The `<file>` name has to be given with its extension to be recognized, even if it is `.tex`, so this is different from the behavior of the `\input` command.

The final group of file-related hooks are those specific to files loaded with an `\include` command.

4 Shipout Hooks

4.1 Background and Foreground

During the shipout the background picture is printed first, then the content of the page, and finally the foreground picture, each overwriting the other.

With the starting point being the top-left corner of the page, your vertical co- ordinate values should be negative.

The hooks should therefore contain only `\put` commands or other commands suitable within a `picture` environment;

```
1 \AddToHookNext{shipout/background}
2   {\put(.5\paperwidth, -.5\paperheight)
3     {\makebox(0, 0){\includegraphics
4       [width=.5\paperwidth]{vs_logo}}}}
5 % remove background picture
6 \RemoveFromHook{shipout/background}
```

4.2 tikz code

Using tikz code for watermark:

```
1 \AddToHook{shipout/background}{%
2   \put(1cm,-\paperheight)
3   {\begin{tikzpicture}[remember picture]%
4     % tikz code
5   \end{tikzpicture}}}%
6 }
```

For example:

```

1 \AddToHook{shipout/background}{%
2 \put(.5\paperwidth, -.5\paperheight)
3 {\makebox(0, 0){\begin{tikzpicture}[remember picture]%
4   \draw[pink, fill=pink] (0,0) rectangle (5, 5);
5 \end{tikzpicture}}}%
6 }

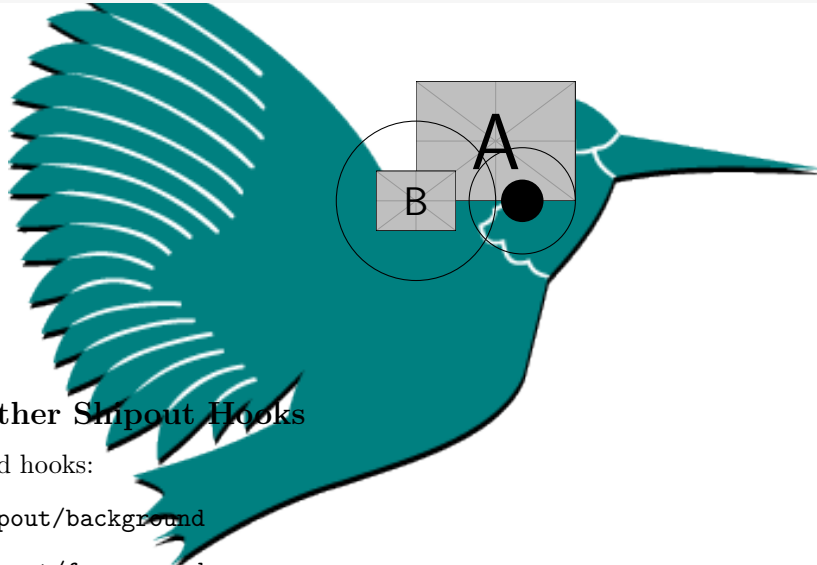
```

Review of the `pict2e` which is extension of the `picture` environment:

```

1 \begin{picture}(0,0)
2   \put(0,0) {\includegraphics[width=6em]{example-image-a}}
3   \put(0,0) {\makebox(0, 0){
4     \includegraphics[width=3em]{example-image-b}
5   }}
6   \put(0,0) {\circle{60}}
7   \put(40,0){\circle{40}} \put(40,0){\circle*{16}}
8 \end{picture}

```



4.3 Other Shipout Hooks

All related hooks:

- `shipout/background`
- `shipout/foreground`
- `shipout/firstpage`: can also be set using the command `\AtBeginDvi`
- `shipout/lastpage`
- `shipout/before`: can be used to manipulate the collected page box before it is being shipped out (or even discard it)
- `shipout`: is executed right in front of the page being shipped out, i.e., after any foreground or background material has been added.
- `shipout/after`: is called after the current page has been shipped out.

Caution: Note that it is not possible (or advisable) to try to use these hooks to typeset material with the intention of returning it to the main vertical list. It will go wrong and give unexpected results in many cases

5 Declaring Hooks

We now look briefly at what is necessary to define new hooks and use them in your own package code.

```
1 \NewHook{hook}          \NewReversedHook{hook}
2 \NewMirroredHookPair{hook1}{hook2}
```

How to call (normal) hooks ?

- `\UseHook{hook}`: execute a normal hook code
- `\UseOneTimeHook{hook}`: If the hook is intended to be a one-time hook, you call it with this.

6 Appendix

Example image in `graphicx` package:

```
1 \includegraphics[width=3cm]{example-image-a}
2 \includegraphics[width=3cm]{example-grid-100x100pt}
3 \includegraphics[width=3cm]{example-image-duck}
```

