The newfloat package*

Axel Sommerfeldt
https://gitlab.com/axelsommerfeldt/newfloat

2013/04/27

Abstract

This package offers the command `\DeclareFloatingEnvironment` for defining new floating environments which behave like `figure` and `table`.

Contents

*This package has version number v1.1.

1 Loading the package

First of all you need to include this package into your document with

`\usepackage[`⟨*options*⟩`]{newfloat}`

where ⟨*options*⟩ are one or more of

`within=`⟨*"within" counter*⟩ *or* `none`
`chapterlistsgap=`⟨*value*⟩ [1]

The "within" counter specifies the counter which will be used to reset the counter of the floating environments `figure` and `table` . (Furthermore this setting will be used as default setting for `\DeclareFloatingEnvironment`.)

So for example `within=chapter` will give you a numbering scheme ⟨*chapter*⟩.⟨*x*⟩ for floating environments, while `within=section` will give you a numbering scheme ⟨*chapter*⟩.⟨*section*⟩.⟨*x*⟩, or ⟨*section*⟩.⟨*x*⟩ if the document class does not offer `\chapter`. `within=none` will result in a continuous numbering throughout the document, i.e. the numbering scheme will be simply ⟨*x*⟩.

The option `chapterlistsgap=`⟨*value*⟩ sets the amount of the vertical gap inserted into the "List of Figure", "List of Tables", and all lists created with `\Declare-FloatingEnvironment` when a new chapter will be started. The default value is `10pt`. (This option will only be available if the document class used offer the usage of chapters, e.g. the `book` or `report` document class.)

Both options can be changed later on, too, by using the command

`\newfloatsetup`⟨*options*⟩ ,

2 `\DeclareFloatingEnvironment`

After loading the newfloat package you can define your own floating environments with

`\DeclareFloatingEnvironment[`⟨*options*⟩`]{`⟨*type*⟩`}`

where ⟨*options*⟩ are one or more of

`fileext=`⟨*file extension*⟩
`listname=`⟨*list name*⟩
`name=`⟨*prosa name*⟩
`placement=`⟨*combination of* `htbp`⟩
`within=`⟨*"within" counter*⟩ *or* `none`
`chapterlistsgaps=on` *or* `off` [1]

If no ⟨*options*⟩ are given, "lo⟨*type*⟩" will be used as ⟨*file extension*⟩ for the list, "List of ⟨*name*⟩s" as ⟨*list name*⟩. "⟨*name*⟩" as ⟨*name*⟩ (but with the first letter capitalized), "`tbp`"

[1] Please note that although the newfloat package tries hard to offer this option it may not be available or is working incorrectly in cooperation with your document class since there is no standard interface which we could use for that purpose. Therefore we have to try to patch the `\chapter` command, and if this fails, we try to patch `\addtocontents` (in an unusual way) instead.

as ⟨*placement*⟩ specifier, and "chapter" resp. "none" as ⟨*"within" counter*⟩, i.e., the counter which resets the numbering.

The default value of the chapterlistsgaps= option depends on the "within" setting, it is set to on if chapter or section is selected, otherwise it is set to off. (This option will only be available if the document class used offer the usage of chapters, e.g. the book or report document class.)

The list will be typeset using the command \listof⟨*type*⟩s resp. \listof⟨*type*⟩es, analogous to \listoffigures and \listoftables.

If the fltpage package is loaded, an environment called FP⟨*type*⟩ will be defined additionally, same for sideways⟨*type*⟩ (rotating package), SC⟨*type*⟩ (sidecap package), and wrap⟨*type*⟩ (wrapfig package).

So for example

```
\DeclareFloatingEnvironment{diagram}
```

will define a new floating environment called diagram, the list will be stored in a file with the extension lodiagram, the name (used for the caption) will be "Diagram" and the list name "List of Diagrams". The list could be typeset with \listofdiagrams. Dependent on which packages are loaded, the environments FPdiagram, sidewaysdiagram, SCdiagram, and wrapdiagram will be defined additionally.

Another example:

```
\DeclareFloatingEnvironment[
  fileext=lox,
  listname={List of Matrixes},
  name=Matrix,
  placement=p,
  within=section,
  chapterlistsgaps=off,
  ]{matrix}
```

will define a new floating environment called matrix with the given settings. Please note that names which contain spaces needs to be enclosed in curly braces.

3 \SetupFloatingEnvironment

While \DeclareFloatingEnvironment will create new floating environments,

\SetupFloatingEnvironment{⟨*floating environment*⟩}{⟨*options*⟩}

will change the settings of existing ones, i.e. either figure or table, or a one created with \DeclareFloatingEnvironment, or a one created with \newfloat offered by the float package, or a one created with \newfloat offered by the memoir document class, or a one created with \DeclareNewFloatType offered by the floatrow package, or . . .

The ⟨*options*⟩ are the same as the options for \DeclareFloatingEnvironment, but one should avoid changing the file extension of existing floating environments, i.e. using the fileext= option within \SetupFloatingEnvironment is usually a very bad idea.

An example:

```
\SetupFloatingEnvironment{lstlisting}{chapterlistsgaps=off}
```

will switch off the chapter lists gaps for `lstlisting` environments offered by the listings package.

4 \ForEachFloatingEnvironment

`\ForEachFloatingEnvironment{`⟨*code with #1*⟩`}`

will execute the given ⟨*code*⟩ for all known floating environments, and for ones defined with `\DeclareFloatingEnvironment` later on.
So for example the subcaption packages uses

```
\ForEachFloatingEnvironment{\DeclareCaptionSubType{#1}}
```

for initializing itself for all floating environments which are known to the newfloat package.

There is also a starred variant `\ForEachFloatingEnvironment*` which will execute the given code for already existing floating environments only, i.e. no hook will be placed inside `\DeclareFloatingEnvironment`.
An example:

```
\ForEachFloatingEnvironment*{\typeout{#1}}
```

will typeout the names of all already known floating environments to the terminal and log file.

5 \PrepareListOf

Unfortunately there is no standardized way of typesetting and customizing lists of floating environments, i.e. every document class offers it's own stuff here, and there are plenty of LaTeX 2ε packages for this purpose.
For that reason the the newfloat package simply typesets the lists of newly defined floating environments as "List of Figure" internally (using `\listoffigure`) but using a different list (the right one, of course) and using a different list name, the one you have specified with either `\DeclareFloatingEnvironment` or `\SetupFloatingEnvironment`.[1]
While this usually results in the same layout as the "List of Figure" and is sufficient for most cases, there are some cases where this might lead to an unwanted side effect.
Therefore the the newfloat package offers

`\PrepareListOf{`⟨*floating environment*⟩`}{`⟨*code*⟩`}`

which will execute the given ⟨*code*⟩ right before actually typesetting the list with `\listoffigures` internally.

[1] Please note that this is not the fact when using a KOMA-Script document class, not `\listoffigure` but `\listoftoc` will be used here, and therefore no problems should occur.

One example:

```
\PrepareListOf{diagram}{%
  \renewcommand{\cftfigpresnum}{Diagram~}}
```

will change the part typeset in front of the diagram number within the "List of Diagrams" when using the tocloft package [8]. Without the code above you will get the part defined for figures instead.

However, when dealing with the tocloft package it's a better idea not to use `\PrepareListOf` but the customization facilities of that package instead, e.g.:

```
\documentclass{book}
...
\usepackage{newfloat,tocloft}
\newlistof{diagram}{lod}{List of Diagrams}
\DeclareFloatingEnvironment[name=Diagram,fileext=lod]{diagram}
      % Note: Due of a bug in tocloft the file extension
      % must be repeated at \DeclareFloatingEnvironment,
      % otherwise you will get an empty list.

% Customize 'List of Figures'
\renewcommand{\cftfigpresnum}{Figure~}
\setlength{\cftfignumwidth}{2cm}

% Customize 'List of Diagrams'
\renewcommand{\cftdiagrampresnum}{Diagram~}
\setlength{\cftdiagramnumwidth}{2cm}

...

\begin{document}
...
\listoffigures
\listofdiagrams
...
\end{document}
```

References

[1] Peter Wilson:
The Memoir Class for Configurable Typesetting,
2011/03/06

[2] Victor Eijkhout:
An introduction to the Dutch LaTeX document classes,
3 September 1989

[3] Markus Kohm & Jens-Uwe-Morawski:
KOMA-Script – a versatile LaTeX 2ε bundle,
2012-07-22

[4] Anselm Lingnau:
An Improved Environment for Floats,
2001/11/08

[5] Sebastian Gross:
Welcome to the beta test of fltpage package!,
1998/11/13

[6] Sebastian Rahtz and Leonor Barroca:
A style option for rotated objects in LaTeX,
1997/09/26

[7] Rolf Niepraschk & Hubert Gäßlein:
The sidecap package,
2003/06/06

[8] Peter Wilson, Herries Press, Will Robertson:
The tocloft package,
2010/10/13

[9] Donald Arseneau:
WRAPFIG.STY ver 3.6,
2003/01/31