Improving Package floatrow Using KOMA-Script Package tocbasic

Markus Kohm

Version 2023-08-16 v1.0

Package floatrowbytocbasic has started as hack module of the KOMA-Script package scrhack years ago to fix an issue when using package floatrow with KOMA-Script 3. This became necessary because package floatrow still depends on an interface once proposed by the KOMA-Script author, but which has long since failed to meet requirements. Unfortunately, this problem could not be solved in dialog with the author of floatrow. Although floatrowbytocbasic still loads the floatrow package, it then changes some internal commands to use and optimally support the KOMA-Script package tocbasic. On the one hand, the user interface of floatrow remains usable unchanged, but at the same time the package benefits from many possibilities of tocbasic.

Contents

1 Why should I use this package instead of floatrow only if I use a KOMA-Script class or KOMA-Script package tocbasic?

The main symptom of using a deprecated interface to KOMA-Script by package floatrow, namely the use of `\float@listhead` and `\float@addtolists`, are corresponding warnings when using one of the KOMA-Script classes at the same time, for example:

```
Class scrbook Warning: \float@addtolists detected!
(scrbook)              Implementation of \float@addtolist became
(scrbook)              deprecated in KOMA-Script v3.01 2008/11/14 and
(scrbook)              has been replaced by several more flexible
(scrbook)              features of package ‘tocbasic‘.
(scrbook)              Since Version 3.12 support for deprecated
(scrbook)              \float@addtolist interface has been
(scrbook)              restricted to only some of the KOMA-Script
(scrbook)              features and been removed from others.
(scrbook)              Loading of package ‘scrhack’ may help to
(scrbook)              avoid this warning, if you are using a
(scrbook)              a package that still implements the
(scrbook)              deprecated \float@addtolist interface.
```

or

```
Class scrbook Warning: \float@listhead detected!
(scrbook)              Implementation of \float@listhead became
(scrbook)              deprecated in KOMA-Script v3.01 2008/11/14 and
(scrbook)              has been replaced by several more flexible
(scrbook)              features of package ‘tocbasic‘.
(scrbook)              Maybe implementation of \float@listhead will
(scrbook)              removed from KOMA-Script soon.
(scrbook)              Loading of package ‘scrhack’ may help to
(scrbook)              avoid this warning, if you are using a
(scrbook)              a package that still implements the
(scrbook)              deprecated \float@listhead interface.
```

If you know, that the issue is because of loading package floatrow you can avoid it using floatrowbytocbasic as explained in section 3.

2 Why should I use this package instead of floatrow only if I don’t use a KOMA-Script class or package?

Package floatrow does not check if a file extension or environment already has been defined when using `\DeclareNewFloatType`. So several definitions of float environments either with the same name or the same file extension could happen by accident. Package floatrowbytocbasic adds test to avoid such issues. But it also allows to redefine an environment or reuse a file extension, if you want. Alternatively you can force the definition or do the definition only if needed. See description of `\DeclareFloatType`, `\RedeclareFloatType` and `\ProvideFloatType` in section 3 for more information.

Package tocbasic provides several additional features, e.g.:

- automatic handling of language switching using babel,
- optional numbered heading for a list/table of contents,
- optional adding an entry to the table of contents for a list/table of contents even if the heading is not numbered,
- easy configuration of entries to a list/table of contents.

Several more features can be added by the class or other packages. See section about tocbasic in the KOMA-Script manual, either [Koh23a] or [Koh23b].

3 How to use floatrowbytocbasic

In the document preamble of your document you just can replace

```
\usepackage{floatrow}
```

by

```
\usepackage{floatrowbytocbasic}
```

to load package floatrowbytocbasic. This does still also load package floatrow but additionally patches several commands of floatrow to avoid the issues shown in section 1.

If you want you can alternatively also load both packages explicitly, either floatrow before floatrowbytocbasic or—if you want—floatrowbytocbasic before floatrow. This is also useful, if you use a package, that uses floatrow itself. In this case, you always should load floatrowbytocbasic *before* the package, that uses floatrow. Otherwise it is very likely that for some new float environments the enhancements of floatrowbytocbasic will not be used. Only if floatrowbytocbasic is loaded before the first use of `\DeclareNewFloatType` can it be ensured that the definition of the new sliding environments is really done with the help of tocbasic and thus its extended user interface is used.

When using a class that uses floatrow, the correct operation can be ensured with

```
\AddToHook{package/floatrow/after}{\RequirePackage{floatrowbytocbasic}}
```

even before `\documentclass`. This requires at least LaTeX 2020/10/01. For older versions of LaTeX you can use

```
\RequirePackage{scrlfile}
\AfterPackage{floatrow}{\RequirePackage{floatrowbytocbasic}}
```

also before `\documentclass`. This would require the KOMA-Script package scrlfile.

The user interface of floatrowbytocbasic is the same as of floatrow, see [Lap09]. Following we document only the differences.

`\DeclareNewFloatType`
`\ProvideFloatType`
`\RedeclareFloatType`
`\DeclareFloatType`

See [Lap09, section 2] for the basic usage of `\DeclareNewFloatType`. The command takes two mandatory arguments. It is of the form:

`\DeclareNewFloatType{`⟨*type*⟩`}{`⟨*options*⟩`}`

floatrowbytocbasic adds the registration of the file extension of the new ⟨*type*⟩ given either by option `fileext` in the ⟨*options*⟩ or implicitly (see [Lap09, section 4]). The main problem here is to decide what to do, if ⟨*ext*⟩ already has been registered. Package floatrow simply does not care for this case. It does neither report an error nor a warning, but just redefine the float ⟨*type*⟩ and reuses the auxiliary file extension. Because of the "`New`" in the name of the command, we have decided to report an error instead, but only of the owner/category differs and therefore is not `float`. But we also decided to define three new commands:

```
\DeclareFloatType{⟨type⟩}{⟨options⟩}
\ProvideFloatType{⟨type⟩}{⟨options⟩}
\RedeclareFloatType{⟨type⟩}{⟨options⟩}
```

`\DeclareFloatTpye` also does not care for already existing ⟨*type*⟩ environment or already registered file extension. `\DeclareNewFloatType` reports and error for an already existing ⟨*type*⟩ environment. `\ProvideFloatType` does not overwrite existing ⟨*type*⟩ environment or already registered file extensions for other owners. `\RedeclareFloatType` does overwrite only already existing ⟨*type*⟩ environment with registered file extension. There are also cases like defined `\⟨type⟩` but not `\end⟨type⟩` or not defined environment ⟨*type*⟩ but registered file extension. But there are no special commands to handle these. If users need to take care for these cases, they can test for definition using the LaTeX kernel macro `\@ifundefined` and the tocbasic command `\Ifattoclist`, described in the KOMA-Script user manuals [Koh23a] resp. [Koh23b].

4 Implementation

floatrowbytocbasic depend on KOMA-Script package tocbasic. So we load it already before everything else. We use this package also to require at least KOMA-Script 3.41, which is one version before the intended spin-off.

```
\RequirePackage{tocbasic}[2023/07/07]
```

We also load package floatrow, because we reuse most of the code:

```
\RequirePackage{floatrow}
```

As often as possible, we do not redefine macors of floatrow, but patch them using xpatch:

```
\RequirePackage{xpatch}
```

We also use some LaTeX3 functions. So if the LaTeX kernel is too old, we require expl3:

```
\@ifundefined{ExplSyntaxOn}{%
  \RequirePackage{expl3}%
}{}
```

`\DeclareFloatType` Basicly `\DeclareFloatType` is `\DeclareNewFloatType` of floatrow:

```
\@ifundefined{NewCommandCopy}{\let\DeclareFloatType\DeclareNewFloatType}{%
  \NewCommandCopy\DeclareFloatType\DeclareNewFloatType
}
```

But we also have to take care, that a not yet registered file extension has to be registered.

Note: We do not need to setup feature `chapteratlist`, because classes like the KOMA-Script classes should use

```
\AtAddToTocList[float]{\setuptoc{\@currext}{chapteratlist}}
```

to implement such features.

```
\newcommand*{\floatrowbytocbasic@incompatibility@error}[1]{%
  \PackageError{floatrowbytocbasic}{incompatible definition of
    \expandafter\string\csname #1\endcsname}{%
    Package `floatrowbytocbasic' depends on the original definition of
    package\MessageBreal
    `floatrow'.\MessageBreak
    Some changes to that definition are tolerated.\MessageBreak
    \@ifundefined{#1}%
    {But it seems the definition is completely missing!}%
    {But the current definition is incompatible!}%
    \MessageBreak
    Make sure, you have installed the original package `floatrow' as
    referred\MessageBreak
    by section ``References'' of the manual.%
  }%
}
\xpatchcmd{\DeclareFloatType}{%
  \xdef\@tempa{\noexpand\flrow@types{\the\flrow@types \FR@tmp{#1}}}%
  \@tempa
}{%
  \Ifattoclist{\@nameuse{ext@\FB@captype}}{%
    \let\reserved@a\relax
    \owneroftoc[\def\reserved@a]{\@nameuse{ext@\FB@captype}}\relax
    \@ifundefined{reserved@a}{}{%
      \Ifstr{\reserved@a}{float}{%
        \PackageInfo{floarowtbytocbasic}{%
          reusing file extension `\@nameuse{ext@\FB@captype}' for\MessageBreak
          float type `#1'%
        }%
      }{%
        \PackageWarning{floatrowbytocbasic}{%
          reusing file extension `\@nameuse{ext@\FB@captype}' of owner\MessageBreak
          `\reserved@a' for float `#1'\MessageBreak
          not recommended%
        }%
      }%
    }%
  }{%
    \addtotoclist[float]{\@nameuse{ext@\FB@captype}}%
  }%
  \xdef\@tempa{\noexpand\flrow@types{\the\flrow@types \FR@tmp{#1}}}%
  \@tempa
}{}{%
  \floatrowbytocbasic@incompatibility@error{DeclareNewFloatType}%
}
```

Note: In difference to floatrow we think, reusing a counter is not a big problem, so we don't report an error.

```
\xpatchcmd{\DeclareFloatType}{%
  \newcounter{#1}%
}{%
  \@ifundefined{c@#1}{\newcounter{#1}}{%
    \PackageInfo{floatrowbytocbasic}{%
      reusing existing counter for float type ‘#1’%
    }%
  }%
}{}{%
  \floatrowbytocbasic@incompatibility@error{DeclareNewFloatType}%
}
\xpatchcmd{\DeclareFloatType}{%
  \@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}%
}{%
  \@ifundefined{l@#1}{\expandafter\let\csname l@#1\endcsname\l@figure
    \@ifundefined{l@#1}{%
      \DeclareTOCStyleEntry[level=1,numwith=2.3em,indent=1.5em]{default}{#1}%
    }{}%
  }{}%
}{}{%
  \floatrowbytocbasic@incompatibility@error{DeclareNewFloatType}%
}
```

`\DeclareNewFloatType`
`\ProvideFloatType`
`\RedeclareFloatType`

`\DeclareNewFloatType` and `\RedeclareFloatType` similar to `\newcommand` and `\renewcommand` do some tests and maybe report an error, before (nevertheless) defining the float.

Note: The test for already defined floats is somehow complicated, because floatrow delays the definition until `\begin{document}`. Before, there is only a list of `\FR@tmp` commands stored in toks register `\flrow@types`. But we also have to test for reusing another environment instead of a float.

```
\renewcommand{\DeclareNewFloatType}[2]{%
  \floatrowbytocbasic@get@currext{#1}{#2}%
  \@tempswatrue
  \begingroup
    \def\FR@tmp##1{%
      \Ifstr{#1}{##1}{\aftergroup\@tempswafalse}{}%
    }%
    \the\flrow@types
  \endgroup
  \if@tempswa
    \expandafter\@ifdefinable\csname #1\endcsname {%
      \Ifattoclist{\floatrowbytocbasic@currext}{%
        \let\reserved@a\relax
        \owneroftoc[\def\reserved@a]{\floatrowbytocbasic@currext}\relax
        \@ifundefined{reserved@a}{%
          \DeclareFloatType{#1}{#2}%
        }{%
          \Ifstr{\reserved@a}{float}{%
            \PackageInfo{floatrowbytocbasic}{%
              reusing ‘\floatrowbytocbasic@currext’ of owner ‘float’%
            }%
            \DeclareFloatType{#1}{#2}%
          }{%
```

```
            \PackageError{floatrowbytocbasic}{%
              reusing `\floatrowbytocbasic@currext' of owner `\reserved@a' not
              allowed%
            }{%
              Each file extension should be used only once.\MessageBreak
              Reusing is only allowed with the same owner/category.\MessageBreak
              You, the class, or another package already uses
              `\floatrowbytocbasic@currext'.\MessageBreak
              \@eha
            }%
          }%
        }%
      }{%
        \DeclareFloatType{#1}{#2}%
      }%
    }%
  \else
    \PackageError{floatrowbytocbasic}{Float `#1' already defined}{%
      Each float can be defined only once using
      \string\DeclareNewFloatType.\MessageBreak
      If you want to redefine an already existing float, please
      use\MessageBreak
      \string\RedeclareFloatType.\MessageBreak
      \@eha
    }%
  \fi
}
\newcommand{\ProvideFloatType}[2]{%
  \@ifundefined{#1}{%
    \@tempswatrue
    \begingroup
      \def\FR@tmp##1{%
        \Ifstr{#1}{##1}{\aftergroup\@tempswafalse}{}%
      }%
      \the\flrow@types
    \endgroup
    \if@tempswa
      \DeclareNewFloatType{#1}{#2}%
    \fi
  }{}%
}
\newcommand{\RedeclareFloatType}[2]{%
  \@tempswafalse
  \@ifundefined{#1}{%
    \begingroup
      \def\FR@tmp##1{%
        \Ifstr{#1}{##1}{\aftergroup\@tempswatrue}{}%
      }%
      \the\flrow@types
    \endgroup
  }{\@tempswatrue}%
  \floatrowbytocbasic@get@currext{#1}{#2}%
  \if@tempswa
    \Ifattoclist{\floatrowbytocbasic@currext}{%
      \removefromtoclist{\floatrowbytocbasic@currext}%
    }{%
      \PackageError{floatrowbytocbasic}{%
        extension ‘\floatrowbytocbasic@currext’ not yet registered%
      }{%
        I can only renew floats with already registered
        extensions.\MessageBreak
        \@ehc
      }%
    }%
  \else
    \@latex@error{Environment #1 undefined}\@ehc
    \removefromtoclist{\floatrowbytocbasic@currext}%
  \fi
  \DeclareFloatType{#1}{#2}%
}
```

`ytocbasic_get_ext_from_prop:nn`
`floatrowbytocbasic@get@currext`

Helper macros to set `\floatrowbytocbasic@currext` to the extension set by the options (=#2). #1 is the name of the new float type. We are not really interested in handling of all the options. So we just use the options to setup a property list and then get the one interesting property.

```
\ExplSyntaxOn
\prop_new:N \@@_DeclareNewFloatType_prop
\tl_new:N \@@_ext_tl
\cs_new:Nn \@@_get_ext_from_prop:nn
  {
    \prop_set_from_keyval:Nn \@@_DeclareNewFloatType_prop { #2 }
    \prop_get:NnN \@@_DeclareNewFloatType_prop { fileext } \@@_ext_tl
    \quark_if_no_value:NT \@@_ext_tl
      {
        \tl_set:Nn \@@_ext_tl { lo#1 }
      }
  }
\newcommand*{\floatrowbytocbasic@get@currext}[2]{%
  \@@_get_ext_from_prop:nn { #1 } { #2 }
  \let\floatrowbytocbasic@currext\@@_ext_tl
}
\ExplSyntaxOff
```

`\listof` This command is redefined using a complete new definition using tocbasic's `\listoftoc`.

```
\renewcommand*{\listof}[2]{%
  \@ifundefined{ext@#1}{\flrow@error{Unknown float style ‘#1’}}{%
    \@ifundefined{l@#1}{\expandafter\let\csname l@#1\endcsname\l@figure
      \@ifundefined{l@#1}{%
        \DeclareTOCStyleEntry[level=1,numwidth=2.3em,indent=1.5em]
        {default}{#1}%
      }{}%
    }{}%
    \listoftoc[{#2}]{\csname ext@#1\endcsname}%
  }%
}%
```

References

[Gre20] Enrico Gregorio. *xpatch – Extending etoolbox patching commands.* Version 0.3. The package generalises the macro patching commands provided by Philipp Lehmann's etoolbox. Mar. 25, 2020. URL: https://ctan.org/pkg/xpatch (visited on 07/19/2023).

[Koh23a] Markus Kohm. *KOMA-Script. The Guide.* June 16, 2023. URL: http://mirrors.ctan.org/macros/latex/contrib/koma-script/scrguide-en.pdf (visited on 07/14/2023).

[Koh23b] Markus Kohm. *KOMA-Script. Die Anleitung.* June 16, 2023. URL: http://mirrors.ctan.org/macros/latex/contrib/koma-script/scrguide-de.pdf (visited on 07/04/2023).

[Koh23c] Markus Kohm. *KOMA-Script — A bundle of versatile classes and packages.* Version 3.41. The KOMA-Script bundle provides replacements for the article, report, and book classes with emphasis on typography and versatility. There is also a letter class. July 7, 2023. URL: https://ctan.org/pkg/koma-script (visited on 07/14/2023).

[Koh23d] Markus Kohm. *scrlfile – Installation control (not only) for KOMA-Script packages.* Version 3.41. The package provides hooks for the execution of commands before or after loading files, classes or packages independent from the LaTeX kernel version. July 7, 2023. URL: https://ctan.org/pkg/scrlfile (visited on 07/19/2023).

[Koh23e] Markus Kohm. *tocbasic — Management of tables/lists of contents (and the like).* Version 3.41. The package provides means to create specialised "table of contents"-like lists of features of a document. July 7, 2023. URL: https://ctan.org/pkg/tocbasic (visited on 07/14/2023).

[Lap08] Olga Lapko. *floatrow — Modifying the layout of floats.* Version 0.3b. The floatrow package provides many ways to customize layouts of floating environments and has code to cooperate with the caption 3.x package. The package offers mechanisms to put floats side by side, and to put the caption beside its float. The floatrow settings could be expanded to the floats created by packages rotating, wrapfig, subfig (in the case of rows of subfloats), and longtable. Aug. 2, 2008. URL: https://ctan.org/pkg/floatrow (visited on 07/18/2023).

[Lap09] Olga Lapko. *The floatrow package.* Aug. 2, 2009. URL: http://mirrors.ctan.org/macros/latex/contrib/floatrow/floatrow.pdf (visited on 07/18/2023).

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.

Change History

v0.1 – 2023/06/01
General: start of KOMA-Script spin-off 1

v1.0 – 2023/08/19
General: first package release 1