Documented Source Code for flowfram.sty v1.17

Nicola L. C. Talbot

2014-09-30

This is the documented source code for the flowfram package. For a user manual, see `ffuserguide.pdf` (or do `texdoc ffuserguide`).

Contents

Glossary 1

1 The Code 3

1.1 Package Initialisation . . . 3

1.2 Flow Frames . . . 11

1.3 Static Frames . . . 31

1.4 Dynamic Frames . . . 46

1.5 Determining Dimensions and Locations . . . 61

1.6 Determining the relative location of one frame from another . . . 69

1.7 Initialise Flow Frames . . . 87

1.8 Output Routine . . . 89

1.9 Static versions of floats . . . 158

1.10 Standard Layouts . . . 158

1.10.1 Column Styles . . . 158

1.10.2 Backdrop Effects . . . 172

1.10.3 Lines Between Frames . . . 180

1.11 Putting Chapter Headings in Dynamic Frames . . . 189

1.12 Thumbtabs . . . 190

1.13 Minitocs . . . 201

Index 210

Index 210

Glossary

bounding box

The smallest possible rectangle that completely encompasses the object.

dynamic frame

Frames in which text is fixed in place, but the contents are re-typeset after each page.

flow frame

The frames in a document such that the contents of the `document` environment flow from one frame to the next in the order that they were defined. There must be at least one flow frame on every page.

frame

A rectangular area of the page in which text can be placed (not to be confused with a frame making command). There are three types: flow, static and dynamic.

frame making command

A LaTeX command which places some kind of border around its argument. For example: `\fbox`.

identification label (IDL)

A unique label which can be assigned to a frame, enabling you to refer to the frame by label instead of by its IDN.

identification number (IDN)

A unique number assigned to each frame, which you can use to identify the frame when modifying its appearance. Example: if you have defined 3 flow frames, 2 static frames and 1 dynamic frame, the flow frames will have IDNs 1, 2 and 3, the static frames will have IDNs 1 and 2, and the dynamic frame will have IDN 1.

page list

A list of pages. This can either be a single keyword: `all`, `odd`, `even` or `none`, or it can be a comma-separated list of individual page numbers or page ranges. For example: `<3,5,7-11,>15` indicates pages 1,2,5,7,8,9,10,11 and all pages after page 15. Note that these numbers refer to the actual value of the page counter, not the absolute physical page number.

page range

Page ranges can be closed, e.g. 5-10, or open, e.g. <7 or >9.

static frame

Frames in which text is fixed in place. The contents are fixed until explicitly changed.

typeblock

The area of the page where the main body of the text goes. The width and height of this area are given by `\textwidth` and `\textheight`.

1 The Code

1.1 Package Initialisation

Declare package, and identify it as a LaTeX 2$_\varepsilon$ package.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{flowfram}[2014/09/30 v1.17 (NLCT)]
```

Load packages needed by this package

```
\RequirePackage{ifthen}

\RequirePackage{xkeyval}
\RequirePackage{graphics}
\RequirePackage{afterpage}


\RequirePackage{xfor}
\RequirePackage{etoolbox}

\@ifundefined{@ldc@l@r}{\RequirePackage{color}}{}
```

The colour of the bounding box borders when the draft option is specified is given by the commands:

```
\newcommand{\setffdraftcolor}{\color[gray]{0.8}}
\newcommand{\setffdrafttypeblockcolor}{\color[gray]{0.9}}
```

`\fflabelsep` In draft mode, each bounding box (apart from the one indicating the typeblock), has a label positioned to the right of the box, at a distance of `\fflabelsep` from the right hand border.

`\fflabelsep`

```
\newlength\fflabelsep
\fflabelsep=1pt
```

`\fflabelfont` The appearance of the label is set by the declaration:

```
\newcommand*{\fflabelfont}{\small\sffamily}
```

The command `\@ffdraft` is used to switch to draft mode. Allow user the option to show particular types of bounding boxes.

```
\newif\ifshowtypeblock
\newif\ifshowmargins
\newif\ifshowframebbox
```

`\@ffdraft` Set all draft settings.

```
\newcommand*{\@ffdraft}{%
  \showtypeblocktrue
  \showmarginstrue
  \showframebboxtrue
}
```

`\@ffnodraft` Unset all draft settings.

```
\newcommand*{\@ffnodraft}{%
  \showtypeblockfalse
  \showmarginsfalse
  \showframebboxfalse
}
```

`\@fr@meifdraft` Draw bounding box.

```
\newcommand*{\@fr@meifdraft}[3][\setffdraftcolor]{%
  \def\ff@backcol{{none}}%
  \@ifundefined{color}{\frame{#2}}{#1\frame{#2}}%
  \ifthenelse{\equal{#3}{}}{}%
  {%
    \makebox[0pt][l]{\hskip\fflabelsep\fflabelfont{[#3]}}%
  }%
}%
```

Colour setting commands, do nothing by default:

```
\newcommand*{\@s@tffcol}{}
\newcommand*{\@s@tfftextcol}{}
```

`\@ffbackground` Deal with frame background colour. Note that the background colour only extends to the limit of the frame's bounding box. If you want the background colour to be flush with the frames border, you will have to create your own customised border.

```
\newcommand*{\@ffbackground}[1]{#1}
```

Now declare the options.

`draft` If draft, switch to draft definitions.

```
\DeclareOptionX{draft}{\@ffdraft}
```

`final` If not draft, reset commands so that no bounding boxes are drawn.

```
\DeclareOptionX{final}{\@ffnodraft}
```

Set the default to final:

```
\@ffnodraft
```

`verbose` Verbose mode is primarily for debug messages.

```
\define@choicekey{flowfram.sty}%
  {verbose}[\val\nr]%
  {true,false}[true]%
  {%
    \ifcase\nr\relax
      \renewcommand*{\flf@doifverbose}[1]{##1}%
      \renewcommand*{\flf@message}[1]{\PackageInfo{flowfram}{##1}}%
    \or
      \renewcommand*{\flf@doifverbose}[1]{}%
      \renewcommand*{\flf@message}[1]{}%
    \fi
  }
```

`\flf@message` Messaging system (to help debugging):

```
\newcommand*{\flf@message}[1]{%
  \flf@doifverbose
  {%
    \PackageInfo{flowfram}{##1}%
  }%
}
```

`\flf@doifverbose` Initialise:

```
\newcommand*{\flf@doifverbose}[1]{}
```

`rotate` Allow provision to prevent rotation in the thumbtabs. If no rotation, thumbtab text will be stacked vertically. This will also affect whether or not to rotate frames.

```
\define@boolkey{flowfram.sty}[@ttb@]{rotate}[true]{}
\@ttb@rotatetrue
```

`norotate` Provide norotate option for backward compatibility

```
\DeclareOptionX{norotate}{\@ttb@rotatefalse}
```

`\rotateframe` Define command that will only rotate box if rotate option set.

```
\newcommand{\rotateframe}[2]{%
  \if@ttb@rotate
    \rotatebox{#1}{#2}%
  \else
    #2%
  \fi
}
```

Should the thumbtabs include number, title, both or neither?

\if@ttb@num

```
\newif\if@ttb@num
\@ttb@numfalse
```

\if@ttb@title

```
\newif\if@ttb@title
\@ttb@titletrue
```

thumbtabs The thumbtabs option replaces the ttbtitle, ttbnotitle, ttbnum and ttbnonum options.

```
\define@choicekey{flowfram.sty}%
  {thumbtabs}[\val\nr]%
  {title,number,both,none}[title]%
  {%
    \ifcase\nr\relax
```

Thumbtabs to only include title

```
        \@ttb@numfalse
        \@ttb@titletrue
    \or
```

Thumbtabs to only include number

```
        \@ttb@numtrue
        \@ttb@titlefalse
    \or
```

Thumbtabs to include title and number

```
        \@ttb@numtrue
        \@ttb@titletrue
    \or
```

Thumbtabs don't have title or number

```
        \@ttb@numfalse
        \@ttb@titlefalse
    \fi
  }
```

Provide old options for backward compatibility:

ttbtitle

```
\DeclareOptionX{ttbtitle}{\@ttb@titletrue}
```

ttbnotitle

```
\DeclareOptionX{ttbnotitle}{\@ttb@titlefalse}
```

ttbnum

```
\DeclareOptionX{ttbnum}{\@ttb@numtrue}
```

ttbnonum

```
\DeclareOptionX{ttbnonum}{\@ttb@numfalse}
```

`pages` Determine whether the pages key when defining frames refers to the page number as given by `\c@page` or the absolute page number as given by `\c@absolutepage`.

```
\define@choicekey{flowfram.sty}{pages}[\val\nr]%
  {relative,absolute}%
  {%
    \ifcase\nr\relax
```

Relative (use `\c@page`):

```
      \renewcommand*{\@ff@pages@countreg}{\c@page}%
    \or
```

Absolute (use `\c@absolutepage`):

```
      \renewcommand*{\@ff@pages@countreg}{\c@absolutepage}%
    \fi
  }
```

`\@ff@pages@countreg` The default is relative (for backwards compatibility).

```
\newcommand*{\@ff@pages@countreg}{\c@page}
```

`absolutepage`

```
\newcounter{absolutepage}
```

`color` If color=true option specified, set up the default colours for the borders and text for all frame types. Note that the colour name has to be grouped within the definition of `\flowframecol` and `\flowframetextcol`. This was done so that you could do, for example, `\renewcommand{\flowframecol}{[rgb]{1,1,0}}` so that you can specify the colour model as well. The commands `\@s@tffcol` and `\@s@tfftextcol` switch to the border and text colour, respectively. They both assume that `\ff@col` has been set to the relevant colour before use.

```
\define@choicekey{flowfram.sty}{color}[\val\nr]{true,false}[true]{%
  \ifcase\nr\relax
```

Option set to true:

```
    \@ff@enablecolor
  \or
```

Option set to false, ensure that the colour changing commands do nothing:

```
    \@ff@disablecolor
  \fi
}
```

Provide nocolor option for backward compatibility:

```
\DeclareOptionX{nocolor}{%
  \@ff@disablecolor
}
```

`\@ff@enablecolor` Enable colour commands.

```
\newcommand*{\@ff@enablecolor}{%
  \def\flowframecol{{black}}%
  \def\flowframetextcol{{black}}%
  \renewcommand*\@s@tffcol{%
    \ifthenelse{\equal{\ff@col}{}}%
    {}%
    {%
      \expandafter\color\ff@col}%
    }%
    \renewcommand*\@s@tfftextcol{%
      \ifthenelse{\equal{\ff@txtcol}{}}%
      {}%
      {%
        \expandafter\color\ff@txtcol
      }%
    }%
    \renewcommand*{\@ffbackground}[1]{%
      \ifthenelse{\equal{\ff@backcol}{{none}}}%
      {%
        ##1%
      }%
      {%
        {\fboxsep=0pt\expandafter\colorbox\ff@backcol{##1}}%
      }%
    }%
}
```

`\@ff@disablecolor` Disable colour commands.

```
\newcommand*{\@ff@disablecolor}{%
  \def\flowframetextcol{}%
  \def\flowframecol{}%
  \renewcommand{\@s@tffcol}{}\renewcommand{\@s@tfftextcol}{}%
  \renewcommand{\@ffbackground}[1]{##1}%
}
```

`\iflefttorightcolumns` Determine whether to define the Ncolumn style frames from left to right or from right to left.

```
\newif\iflefttorightcolumns
\lefttorightcolumnstrue
```

Define options that set the direction:

`LR`

```
\DeclareOptionX{LR}{\lefttorightcolumnstrue}
```

`RL`

```
\DeclareOptionX{RL}{\lefttorightcolumnsfalse}
```

If the `\normalcolor` command is something other than `\relax`, then implement the color=true option as the default, otherwise implement the color=false option as the default.

```
\ifx\normalcolor\relax
  \@ff@disablecolor
\else
  \@ff@enablecolor
\fi
```

Now the defaults have all been set, the package options specified by the user can be processed:

```
\ProcessOptionsX
```

If color=true option has been specified, but no color package has been loaded yet, load color.sty

```
\ifx\normalcolor\relax
  \ifthenelse{\equal{\flowframetextcol}{}}%
  {}%
  {%
    \RequirePackage{color}%
  }
\fi
```

```
\@ifundefined{chapter}{}%
{%
```

`\chapterfirstpagestyle` User may want a non standard style for the first page of each chapter, so modify chapter commands to take this into account.

```
  \newcommand*{\chapterfirstpagestyle}{plain}%

  \let\@ff@OLD@chapter\@chapter
  \let\@ff@OLD@schapter\@schapter
  \renewcommand{\@chapter}{%
    \thispagestyle{\chapterfirstpagestyle}%
    \@ff@OLD@chapter
  }%
  \renewcommand{\@schapter}{%
    \thispagestyle{\chapterfirstpagestyle}%
    \@ff@OLD@schapter
  }%
```

`\ffprechapterhook` Hook at start of chapter (before page break issued)

```
  \newcommand*{\ffprechapterhook}{}
```

`\chapter` Modify `\chapter` so the hook is called at the start:

```
  \let\@ff@OLD@ch@pter\chapter
  \renewcommand{\chapter}{%
    \ffprechapterhook
    \@ff@OLD@ch@pter
  }
```

End of test if `\chapter` defined:

```
}
```

`maxflow` Now get on with the package. First we need to set up a register to store the number of flow frames that have been defined:

```
\newcounter{maxflow}
\c@maxflow=0\relax
```

`thisframe` Next define a counter to keep track of the identification number (IDN) of the current flow frame.

```
\newcounter{thisframe}
\c@thisframe=0\relax
\@ifpackageloaded{hyperref}
{%
  \def\theHthisframe{\thepage.\arabic{thisframe}}%
}%
{}
```

`\labelflowidn` Define a command to label the current flow frame so that its IDN can be referenced:

```
\newcommand*{\labelflowidn}[1]{%
  {%
    \def\@currentlabel{\thethisframe}%
    \label{#1}%
  }%
}
```

`displayedframe` Define a counter to store the current frame index for the current page. This will be the same as the IDN if all flow frames are displayed on the current page, but may be different to the IDN if some flow frames are not displayed.

```
\newcounter{displayedframe}
\c@displayedframe=0
\@ifpackageloaded{hyperref}%
{%
  \def\theHdisplayedframe{\thepage.\arabic{displayedframe}}%
}%
{}
```

`\labelflow` Define a command to label the current flow frame so that its displayed index can be referenced:

```
\newcommand*{\labelflow}[1]{%
  {%
    \def\@currentlabel{\thedisplayedframe}%
    \label{#1}%
  }%
}
```

`maxstatic` Define a counter to store the total number of static frames:

```
\newcounter{maxstatic}
\c@maxstatic=0\relax
```

`maxdynamic` Define a counter to store the total number of dynamic frames:

```
\newcounter{maxdynamic}
\c@maxdynamic=0\relax
```

Define some temporary variables

```
\newcount\@colN
\newcount\@ff@tmpN
\newcount\ff@id
\newlength\@ff@offset
\newlength\@ff@tmp@x
\newlength\@ff@tmp@x@even
\newlength\@ff@tmp@y
```

`\sdfparindent` Define a length to govern paragraph indentation within static and dynamic frames. This is 0pt by default.

```
\newlength\sdfparindent
```

1.2 Flow Frames

`\flowframesep` Set up default lengths. The gap between the text and the border is given by:

```
\newlength\flowframesep
\flowframesep=\fboxsep
```

`\flowframerule` The width of the frame is given by:

```
\newlength\flowframerule
\flowframerule=\fboxrule
```

`\flowframeshowlayout` Define command to show page layout. This finishes the current page, temporarily sets draft mode, and prints an empty page. Only the frames for that page will be shown.

`\flowframeshowlayout`

```
\newcommand*{\flowframeshowlayout}{%
  \finishthispage
  {%
    \@ffdraft\mbox{}\finishthispage\clearpage
  }%
}
```

`\framebreak` If the flow frames are not all of the same width, the change in `\hsize` will not come into effect until the end of the paragraph. Provide a command to simulate a paragraph break, without making it look as though there is a paragraph. Provides an optional argument that is passed to `\pagebreak`. Make sure it is grouped to localise the change in `\parfillskip` and `\parskip`.

```
\newif\ifusedframebreak
\newcommand{\framebreak}[1][4]{%
  \global\usedframebreaktrue
  {%
    \parfillskip=0pt\pagebreak[#1]\parskip=0pt\par\noindent
    }%
  }
```

\finishthispage

The commands \newpage and \pagebreak can be used to move on to the next flow frame, but to finish the entire page, use \finishthispage. This is (loosely) based on the code for \clearpage. (\@dbltopnum not required as we can't have column-spanning floats.)

```
  \newcommand{\finishthispage}{%
    \ifvmode
      \@colN=\c@thisframe\relax
      \count@=\c@absolutepage\relax
      \ifdim \pagetotal<\topskip
        \hbox{}%
      \fi
      \newpage \write \m@ne {}\vbox {}\penalty -\@Mi
```

If that was the last flow frame on the page, then we're done, otherwise iterate through the remaining flow frames.

```
      \ifnum\count@=\c@absolutepage\relax
        \whiledo{\@colN<\c@maxflow \OR \@colN=\c@maxflow}%
        {%
          \@ff@chckifthispg{\@ff@pages@countreg}{\@colN}%
          \if@notthiscol
          \else
            \c@thisframe=\@colN\relax
            \hbox{}\newpage
          \fi
          \advance\@colN by 1\relax
        }%
      \fi
    \fi
  }
```

\cleardoublepage

Modify the definition of \cleardoublepage. This may or may not be defined so use \def.

```
  \def\cleardoublepage{%
    \clearpage
    \if@twoside
      \ifodd\c@page
      \else
        \hbox{}%
        \clearpage
      \fi
    \fi
  }
```

\newpage

Modify the definition of \newpage so that it sets the usedframebreak flag.

```
  \preto\newpage{\global\usedframebreaktrue}
```

Disable `@twocolumn` flag, as it makes no sense.

```
\@twocolumnfalse
```

Disable `@mparswitch` flag, as each flow frame has its own predefined margin setting.

```
\@mparswitchfalse
```

`\globalreversemargin` The margins get switched during the output routine, so need the effect to be global.

```
\newcommand{\globalreversemargin}{%
  \global\@mparbottom\z@
  \global\@reversemargintrue
}
```

`\globalnormalmargin`

```
\newcommand{\globalnormalmargin}{%
  \global\@mparbottom\z@\global
  \@reversemarginfalse
}
```

`\@getmarginpos` Determine whether the margin should be on the right or left. This depends on the setting, which can either be `right` or `left` (self explanatory) or `inner` (on the spine side, so left for odd pages and right for even pages) or `outer` (on the outside of the page, so right for odd pages and left for even pages.) When `\@getmarginpos` is finished, the setting is stored in `\ff@margin`.

```
\newcommand{\@getmarginpos}[1]{%
 \ifthenelse{\equal{#1}{inner}}%
 {%
  \if@twoside
    \ifodd\c@page\def\ff@margin{left}\else\def\ff@margin{right}\fi
  \else
    \def\ff@margin{left}%
  \fi
 }%
 {%
  \ifthenelse{\equal{#1}{outer}}%
  {%
   \if@twoside
    \ifodd\c@page\def\ff@margin{right}\else\def\ff@margin{left}\fi
   \else
    \def\ff@margin{right}%
   \fi
  }%
  {%
   \def\ff@margin{#1}%
  }%
 }%
}
```

`\setmargin` Set the margin for current flow frame.

```
\newcommand{\setmargin}{%
 \@getmarginpos
 {%
   \csname @ff@margin@\romannumeral\c@thisframe\endcsname
 }%
 \ifthenelse{\equal{\ff@margin}{left}}%
 {\globalreversemargin}%
 {\globalnormalmargin}%
}
```

`\newflowframe` Create a new flow frame. Syntax:
`\newflowframe[`⟨*pages*⟩`]{`⟨*width*⟩`}{`⟨*height*⟩`}{`⟨*x*⟩`}{`⟨*y*⟩`}[`⟨*label*⟩`]`

First increment `\c@maxflow`, and define boolean to indicate whether or not the flow frame has a border, Then check to see whether or not the starred version is begin used. All the settings must be global: the output routine will create a new flow frame, if there are no more defined, and since changes made in the output routine are localised, the new frame will be lost unless it is globally defined. Flow frames should only be set up in the preamble, but if there are not enough frames to fit all the document text, the output routine will create a new flow frame. So, define `\newflowframe` so that it calls `\@n@wflowframe`

```
\newcommand{\newflowframe}{\@n@wflowframe}
```

Set the external command for use only in the preamble, an make the output routine use the internal command

```
\@onlypreamble{\newflowframe}
```

`\@n@wflowframe`

```
\newcommand{\@n@wflowframe}{%
  \global\advance\c@maxflow by 1\relax
  \expandafter\global\expandafter
  \newif\csname ifcolumnframe\romannumeral\c@maxflow\endcsname
  \@ifstar\@snewflowframe\@newflowframe
}
```

`\@snewflowframe` Starred version sets boolean flag to indicate a border

```
\newcommand{\@snewflowframe}{%
  \expandafter\global\expandafter
  \let\csname ifcolumnframe\romannumeral\c@maxflow\endcsname\iftrue
  \@@newflowframe
}
```

`\@newflowframe` The unstarred version unsets boolean flag to indicate no border.

```
\newcommand{\@newflowframe}{%
  \expandafter\global\expandafter
  \let\csname ifcolumnframe\romannumeral\c@maxflow\endcsname\iffalse
  \@@newflowframe
}
```

`\@@newflowframe` Now get on with initialising the flow frame. By default, it will apply the flow frame to all pages, the optional argument can override this.

```
\newcommand{\@@newflowframe}[5][all]{%
  \expandafter\global\expandafter
    \newbox\csname column\romannumeral\c@maxflow\endcsname
  \expandafter\global\expandafter
    \newlength\csname colwidth\romannumeral\c@maxflow\endcsname
  \expandafter\global\expandafter
    \newlength\csname colheight\romannumeral\c@maxflow\endcsname
  \expandafter\global\expandafter
    \newlength\csname col@\romannumeral\c@maxflow @posx\endcsname
  \expandafter\global\expandafter
    \newlength\csname col@\romannumeral\c@maxflow @posy\endcsname
  \expandafter\global\expandafter
    \setlength\csname colwidth\romannumeral\c@maxflow\endcsname{#2}
  \expandafter\global\expandafter
    \setlength\csname colheight\romannumeral\c@maxflow\endcsname{#3}
  \expandafter\global\expandafter
    \setlength\csname col@\romannumeral\c@maxflow @posx\endcsname{#4}
  \expandafter\global\expandafter
    \setlength\csname col@\romannumeral\c@maxflow @posy\endcsname{#5}
  \expandafter\global\expandafter
    \newlength\csname col@\romannumeral\c@maxflow @evenx\endcsname
  \expandafter\global\expandafter
    \newlength\csname col@\romannumeral\c@maxflow @eveny\endcsname
  \expandafter\global\expandafter
    \setlength\csname col@\romannumeral\c@maxflow @evenx\endcsname{#4}
  \expandafter\global\expandafter
    \setlength\csname col@\romannumeral\c@maxflow @eveny\endcsname{#5}
  \expandafter
    \gdef\csname @ff@frametype@\romannumeral\c@maxflow\endcsname{fbox}%
  \expandafter
    \gdef\csname @ff@col@\romannumeral\c@maxflow\endcsname{\flowframecol}
  \expandafter
    \gdef\csname @ff@txtcol@\romannumeral\c@maxflow\endcsname{%
      \flowframetextcol
    }
  \expandafter
    \gdef\csname @ff@backcol@\romannumeral\c@maxflow\endcsname{{none}}
  \expandafter
    \gdef\csname @ff@pages@\romannumeral\c@maxflow\endcsname{#1}%
```

Page exclusion list:

```
  \expandafter
    \gdef\csname @ff@xpages@\romannumeral\c@maxflow\endcsname{}%
  \expandafter
    \gdef\csname @ff@offset@\romannumeral\c@maxflow\endcsname{compute}
  \expandafter
    \gdef\csname @ff@angle@\romannumeral\c@maxflow\endcsname{0}%
  \expandafter
    \gdef\csname @ff@margin@\romannumeral\c@maxflow\endcsname{right}
  \ifnum\c@thisframe=0\relax
    \ifthenelse{\equal{#1}{all}\TE@or\equal{#1}{odd}}%
    {%
      \c@thisframe=\c@maxflow
      \global\setlength{\hsize}{#2}%
      \global\usedframebreaktrue
    }%
    {%
      \ifthenelse{\equal{#1}{even}\TE@or\equal{#1}{none}}%
      {}%
      {%
        \def\ff@pages{#1}%
        \@for\@ff@pp:=\ff@pages\do
        {%
          \def\@ff@numstart{0}\def\@ff@numend{0}%
          \@ff@getrange{\@ff@pp}%
          \ifnum\@ff@numstart=0\relax
            \def\@ff@numstart{1}%
          \fi
          \ifnum\@ff@numstart=1\relax
            \c@thisframe=\c@maxflow
            \global\setlength{\hsize}{#2}%
            \global\usedframebreaktrue
          \fi
        }%
      }%
    }%
  \fi
  \@ifnextchar[%
  {\@s@tflowframeid{\c@maxflow}}%
  {%
    \@s@tflowframeid{\c@maxflow}[\number\c@maxflow]%
  }%
}
```

`\@s@tflowframeid` If square brackets occur after `\newflowframe`, take the contents to be the label, otherwise the label will be the flow frame number.

```
\def\@s@tflowframeid#1[#2]{%
  \edef\ff@label{#2}%
  \@ff@checkuniqueidl{#1}{\ff@label}%
  \expandafter
    \xdef\csname @col@id@\romannumeral#1\endcsname{\ff@label}%
}
```

`\@ff@checkuniqueidl` Check identification label (IDL) #2 for flow frame #1 is unique

```
\newcommand*{\@ff@checkuniqueidl}[2]{%
  {%
```

```
    \@colN=0\relax
    \whiledo{\@colN<\c@maxflow}%
    {%
      \advance\@colN by 1\relax
      \ifnum\@colN=#1\relax
      \else
        \ifthenelse
        {%
           \equal{#2}%
           {%
             \csname @col@id@\romannumeral\@colN\endcsname
           }%
        }%
        {%
          \PackageError{flowfram}%
          {Flow frame IDL '#2' already defined}%
          {%
            You can't assign this label, as it is already defined
            for flow frame \number\@colN
          }%
        }%
        {}%
      \fi
    }%
  }%
}
```

`\getflowlabel` `\getflowlabel{⟨idn⟩}` Gets the IDL for the flow frame identified by its IDN.

```
\newcommand*{\getflowlabel}[1]{%
  \csname @col@id@\romannumeral#1\endcsname
}
```

`\getflowid` `\getflowid{⟨cmd⟩}{⟨idl⟩}` Gets the IDN for the flow frame identified by its IDL and stores in ⟨*cmd*⟩ which must be a control sequence.

```
\newcommand*{\getflowid}[2]{%
  \@flowframeid{#2}%
  \edef#1{\number\ff@id}%
}
```

`\@flowframeid` Work out the flow frame IDN from the label. This iterates through the flow frames, so if you have a lot of them it is quicker to identifiy them by their IDN rather than their IDL. The IDN stored in `\ff@id`.

```
\newcommand*{\@flowframeid}[1]{%
  \@colN=0\relax
  \ff@id=0\relax
  \whiledo{\@colN<\c@maxflow}%
  {%
    \advance\@colN by 1\relax
    \ifthenelse
```

```
      {%
        \equal{#1}{\csname @col@id@\romannumeral\@colN\endcsname}%
      }%
      {%
        \ff@id=\@colN\relax
```

Break out of loop

```
        \@colN=\c@maxflow
      }%
      {}%
    }%
    \ifnum\ff@id=0\relax
      \PackageError{flowfram}{Can't find flow frame id '#1'}{}%
    \fi
  }
```

Set up the keys for use with `\setflowframe`, `\setstaticframe` and `\setdynamicframe`.

Frame width is stored in `\ff@width`.

```
  \define@key{flowframe}{width}%
  {%
    \ifthenelse{\equal{#1}{}}%
    {%
      \PackageError{flowfram}{Missing value for 'width' key}{}%
    }%
    {}%
    \def\ff@width{#1}%
  }
```

Frame height is stored in `\ff@height`.

```
  \define@key{flowframe}{height}%
  {%
    \ifthenelse{\equal{#1}{}}%
    {%
      \PackageError{flowfram}{Missing value for 'height' key}{}%
    }%
    {}%
    \def\ff@height{#1}%
  }
```

Frame x co-ordinate (odd and even pages) is stored in `\ff@x`.

```
  \define@key{flowframe}{x}%
  {%
    \ifthenelse{\equal{#1}{}}%
    {%
      \PackageError{flowfram}{Missing value for 'x' key}{}%
    }%
    {}%
    \def\ff@x{#1}%
  }
```

Frame y co-ordinate (odd and even pages) is stored in `\ff@y`.

```
\define@key{flowframe}{y}%
{%
  \ifthenelse{\equal{#1}{}}%
  {%
    \PackageError{flowfram}{Missing value for 'y' key}{}%
  }%
  {}%
  \def\ff@y{#1}%
}
```

Frame *x* co-ordinate (even pages only) is stored in `\ff@evenx`.

```
\define@key{flowframe}{evenx}%
{%
  \ifthenelse{\equal{#1}{}}%
  {%
    \PackageError{flowfram}{Missing value for 'evenx' key}{}%
  }%
  {}%
  \def\ff@evenx{#1}%
}
```

Frame *y* co-ordinate (even pages only) is stored in `\ff@eveny`.

```
\define@key{flowframe}{eveny}%
{%
  \ifthenelse{\equal{#1}{}}%
  {%
    \PackageError{flowfram}{Missing value for 'eveny' key}{}%
  }%
  {}%
  \def\ff@eveny{#1}%
}
```

Frame *x* co-ordinate (odd pages only if twoside implemented) is stored in `\ff@oddx`.

```
\define@key{flowframe}{oddx}%
{%
  \ifthenelse{\equal{#1}{}}%
  {%
    \PackageError{flowfram}{Missing value for 'oddx' key}{}%
  }%
  {}%
  \def\ff@oddx{#1}%
}
```

Frame *y* co-ordinate (odd pages only if twoside implemented) is stored in `\ff@oddy`.

```
\define@key{flowframe}{oddy}%
{%
  \ifthenelse{\equal{#1}{}}%
  {%
    \PackageError{flowfram}{Missing value for 'oddy' key}{}%
  }%
  {}%
  \def\ff@oddy{#1}%
}
```

New IDL for frame is stored in `\ff@label`.

```
  \define@key{flowframe}{label}%
  {%
    \ifthenelse{\equal{#1}{}}%
    {%
      \PackageError{flowfram}{Missing value for 'label' key}{}%
    }%
    {}%
    \def\ff@label{#1}%
  }
```

Frame border. If `none`, define `\ff@frame` as `false`, otherwise define `\ff@frame` as `true`. If `plain`, define `\ff@frametype` as `fbox`, otherwise define it to be the specified type, which should be the name of a frame making command without the preceding backslash.

```
  \define@key{flowframe}{border}[plain]%
  {%
    \ifthenelse{\equal{#1}{}}%
    {%
      \PackageError{flowfram}%
      {%
         Missing value for 'border' key - use
         'none' for no border%
      }%
      {}%
    }%
    {}%
    \ifthenelse{\equal{#1}{none}}%
    {%
      \def\ff@frame{false}%
    }%
    {%
      \def\ff@frame{true}%
      \ifthenelse{\equal{#1}{plain}}%
      {%
        \def\ff@frametype{fbox}%
      }%
      {%
        \def\ff@frametype{#1}%
      }%
    }%
  }
```

Frame's border colour. (This may not work for non-standard frame making commands.)

```
\define@key{flowframe}{bordercolor}%
{%
  \ifthenelse{\equal{#1}{}}%
  {%
    \PackageError{flowfram}{Missing value for 'bordercolor' key}{}%
  }%
  {}%
  \def\ff@col{#1}%
}
```

Frame's text colour.

```
\define@key{flowframe}{textcolor}%
{%
  \ifthenelse{\equal{#1}{}}%
  {%
    \PackageError{flowfram}{Missing value for 'textcolor' key}{}%
  }%
  {}%
  \def\ff@txtcol{#1}%
}
```

The background colour of the frame. Note this only covers the region of the bounding box, not any extra space between the bounding box and the border. If you want the background colour to go right up to the border, you will need to define your own customised border.

```
\define@key{flowframe}{backcolor}%
{%
  \ifthenelse{\equal{#1}{}}%
  {%
    \PackageError{flowfram}{Missing value for 'backcolor' key}{}%
  }%
  {}%
  \def\ff@backcol{#1}%
}
```

Page list for which the frame should appear.

```
\define@key{flowframe}{pages}%
{%
  \ifthenelse{\equal{#1}{}}%
  {%
    \PackageError{flowfram}{Missing value for 'pages' key}{}%
  }%
  {}%
  \def\ff@pages{#1}%
}
```

Exclusion list:

```
\define@key{flowframe}{excludepages}%
{%
  \def\ff@xpages{#1}%
}
```

The border takes up extra space, which needs to be adjusted. This can be done for standard border types, but non-standard borders may require some help.

```
\define@key{flowframe}{offset}%
{%
  \def\ff@offset{#1}%
  \ifthenelse{\equal{#1}{}}%
  {%
    \PackageError{flowframe}%
    {%
      Invalid value for key 'offset'%
    }%
    {%
      'offset' can either be 'compute' (to compute it according
      to certain pre-defined rules) or a length%
    }%
  }%
  {}%
}
```

Angle to rotate flow frame:

```
\define@key{flowframe}{angle}{\def\ff@angle{#1}%
}
```

This key is only for flow frames:

```
\define@choicekey{flowframe}{margin}{left,right,inner,outer}%
{%
  \def\ff@margin{#1}%
}
```

This key is only for static frames:

```
\define@choicekey{flowframe}{clear}{true,false}[true]{%
  \def\ff@clear{#1}%
}
```

This key is only for dynamic frames:

```
\define@key{flowframe}{style}%
{%
  \ifthenelse{\equal{#1}{}}%
  {%
    \PackageError{flowfram}{Missing value for 'style' key}{}%
  }%
  {}%
  \ifthenelse{\equal{#1}{none}}%
  {%
    \def\ff@style{relax}%
  }%
  {%
    \def\ff@style{#1}%
  }%
}
```

This key is only for static frames and dynamic frames.

```
\define@key{flowframe}{shape}%
{%
  \def\ff@shape{#1}%
}
```

This key is only for static frames and dynamic frames.

```
\define@choicekey{flowframe}{valign}{c,t,b}%
{%
  \def\ff@valign{#1}%
}
```

This key is only for static frames and dynamic frames:

```
\define@choicekey{flowframe}{hide}{true,false}[true]{%
  \def\ff@hide{#1}%
}
```

This key is only for static frames and dynamic frames:

```
\define@choicekey{flowframe}{hidethis}{true,false}[true]{%
  \def\ff@hidethis{#1}%
}
```

`\setallflowframes` Provide a command to change the settings for all flow frames. This just iterates through all the flow frames, and sets each one in turn.

```
\newcommand*{\setallflowframes}[1]{%
  \@colN=0\relax
  \whiledo{\@colN<\c@maxflow}%
  {%
    \advance\@colN by 1\relax
    \@@setflowframe{\@colN}{#1}%
  }%
}
```

`\setflowframe` Define `\setflowframe` command. Check to see whether or not the starred version is being used.

```
\newcommand*{\setflowframe}{\@ifstar\@ssetflowframe\@setflowframe}
```

`\@ssetflowframe` This is the starred version. It finds the IDN for each label in the comma-separated list (first argument), and applies the setting for that numbered flow frame.

```
\newcommand{\@ssetflowframe}[2]{%
  \@for\@ff@id:=#1\do{%
    \@flowframeid{\@ff@id}%
    \@@setflowframe{\ff@id}{#2}%
  }%
}
```

`\@setflowframe` This is the unstarred version. It iterates through each IDN in the comma-separated list passed as the first argument, but it also checks for number ranges, and sets the values for that flow frame. Ensures that number ranges do not lie out of bounds.

```
\newcommand*{\@setflowframe}[2]{%
  \ifthenelse{\equal{#1}{all}}%
  {%
    \setallflowframes{#2}%
  }%
  {%
    \ifthenelse{\equal{#1}{odd} \TE@or \equal{#1}{even}}%
    {%
      \ifthenelse{\equal{#1}{odd}}%
      {%
        \@colN=1\relax
      }%
      {%
        \@colN=2\relax
      }%
      \whiledo{\@colN<\c@maxflow\TE@or\@colN=\c@maxflow}%
      {%
        \@@setflowframe{\@colN}{#2}%
        \advance\@colN by 2\relax
      }%
    }%
    {%
      \@for\@ff@id:=#1\do
      {%
        \def\@ff@numstart{0}%
        \def\@ff@numend{10000}%
        \@ff@getrange{\@ff@id}%
        \ifnum\@ff@numstart=0\relax
          \def\@ff@numstart{1}%
        \fi
        \ifnum\@ff@numend>\c@maxflow\relax
          \def\@ff@numend{\c@maxflow}%
        \fi
        \@colN=\@ff@numstart\relax
        \whiledo{\@colN<\@ff@numend \TE@or \@colN=\@ff@numend}%
        {%
          \@@setflowframe{\@colN}{#2}%
          \advance\@colN by 1\relax
        }%
      }%
    }%
  }%
}
```

`\@@setflowframe` This is the command that actually sets the values for the flow frame whose IDN is specified by the first parameter.

```
\newcommand*{\@@setflowframe}[2]{%
```

```
\def\ff@frame{}\def\ff@width{}\def\ff@height{}\def\ff@margin{}%
\def\ff@x{}\def\ff@y{}\def\ff@frametype{}\def\ff@col{}%
\def\ff@valign{}\def\ff@style{}%
\def\ff@hide{}\def\ff@hidethis{}%
\def\ff@txtcol{}\def\ff@clear{}\def\ff@offset{}\def\ff@pages{}%
\def\ff@label{}\def\ff@backcol{}\def\ff@evenx{}\def\ff@eveny{}%
\def\ff@oddx{}\def\ff@oddy{}\def\ff@angle{}%
\let\ff@xpages\undefined
\let\ff@shape\undefined
\setkeys{flowframe}{#2}%
\ifdefempty{\ff@frame}{}%
{%
  \setboolean{columnframe\romannumeral#1}{\ff@frame}%
}%
\ifdefempty{\ff@width}{}%
{%
  \expandafter
    \setlength\csname colwidth\romannumeral#1\endcsname
    {\ff@width}%
}%
\ifdefempty{\ff@height}{}%
{%
  \expandafter
    \setlength\csname colheight\romannumeral#1\endcsname
    {\ff@height}%
}%
\ifdefempty{\ff@x}{}%
{%
  \expandafter
    \setlength\csname col@\romannumeral#1@posx\endcsname
    {\ff@x}%
  \expandafter
    \setlength\csname col@\romannumeral#1@evenx\endcsname
    {\ff@x}%
}
\ifdefempty{\ff@y}{}%
{%
  \expandafter
    \setlength\csname col@\romannumeral#1@posy\endcsname
    {\ff@y}%
  \expandafter
    \setlength\csname col@\romannumeral#1@eveny\endcsname
    {\ff@y}%
}%
\ifdefempty{\ff@evenx}{}%
{%
  \expandafter
    \setlength\csname col@\romannumeral#1@evenx\endcsname
    {\ff@evenx}%
```

```
}%
\ifdefempty{\ff@eveny}{}%
{%
  \expandafter
    \setlength\csname col@\romannumeral#1@eveny\endcsname
    {\ff@eveny}%
}%
\ifdefempty{\ff@oddx}{}%
{%
  \expandafter
    \setlength\csname col@\romannumeral#1@posx\endcsname
    {\ff@oddx}%
}%
\ifdefempty{\ff@oddy}{}%
{%
  \expandafter
    \setlength\csname col@\romannumeral#1@posy\endcsname
    {\ff@oddy}%
}%
\ifdefempty{\ff@label}{}%
{%
  \@s@tflowframeid{#1}[\ff@label]%
}%
\ifdefempty{\ff@frametype}{}%
{%
  \expandafter
    \edef\csname @ff@frametype@\romannumeral#1\endcsname{%
      \ff@frametype}%
}%
\ifdefempty{\ff@col}{}%
{%
  \expandafter\@setframecol\ff@col\end{#1}{col}{ff}%
}%
\ifdefempty{\ff@txtcol}{}%
{%
  \expandafter\@setframecol\ff@txtcol\end{#1}{txtcol}{ff}%
}%
\ifdefempty{\ff@backcol}{}%
{%
  \expandafter\@setframecol\ff@backcol\end{#1}{backcol}{ff}%
}%
\ifdefempty{\ff@margin}{}%
{%
  \expandafter
    \xdef\csname @ff@margin@\romannumeral#1\endcsname{%
      \ff@margin}%
}%
\ifdefempty{\ff@pages}{}%
{%
```

```
    \flowsetpagelist{#1}{\ff@pages}%
  }%
  \ifundef{\ff@xpages}{}%
  {%
    \flowsetexclusion{#1}{\ff@xpages}%
  }%
  \ifdefempty{\ff@offset}{}%
  {%
    \expandafter
      \xdef\csname @ff@offset@\romannumeral#1\endcsname{%
        \ff@offset}%
  }%
  \ifdefempty{\ff@angle}{}%
  {%
    \expandafter
      \xdef\csname @ff@angle@\romannumeral#1\endcsname{%
        \ff@angle}%
  }%
  \ifdefempty{\ff@clear}{}%
  {%
    \PackageError{flowfram}%
      {Key 'clear' not available for flow frames}{}%
  }%
  \ifdefempty{\ff@style}{}%
  {%
    \PackageError{flowfram}%
      {Key 'style' not available for flow frames}{}%
  }%
  \ifundef{\ff@shape}{}%
  {%
    \PackageError{flowfram}%
    {Key 'shape' not available for flow frames}{}%
  }%
  \ifdefempty{\ff@valign}{}%
  {%
    \PackageError{flowfram}%
    {Key 'valign' not available for flow frames}{}%
  }%
  \ifdefempty{\ff@hide}{}%
  {%
    \PackageError{flowfram}%
      {Key 'hide' not available for flow frames}{}%
  }%
  \ifdefempty{\ff@hidethis}{}%
  {%
    \PackageError{flowfram}%
      {Key 'hidethis' not available for flow frames}{}%
  }%
}
```

`\flowsetpagelist` Sets the page list for the flow frame given by `#1` (the IDN).

```
\newcommand*{\flowsetpagelist}[2]{%
  \expandafter
    \xdef\csname @ff@pages@\romannumeral#1\endcsname{#2}%
  \flf@message{Setting page range for flow frame
    \number#1\space\space to "#2"}%
}
```

`\flowsetexclusion` Sets the exclusion list for the flow frame given by `#1` (the IDN).

```
\newcommand*{\flowsetexclusion}[2]{%
  \expandafter
    \xdef\csname @ff@xpages@\romannumeral#1\endcsname{#2}%
  \flf@message{Setting exclusion for flow frame
    \number#1\space\space to "#2"}%
}
```

`\flowaddexclusion` Adds to the exclusion list for the flow frame given by `#1` (the IDN).

```
\newcommand*{\flowaddexclusion}[2]{%
  \ifcsempty{@ff@xpages@\romannumeral#1}
  {%
    \expandafter
      \xdef\csname @ff@xpages@\romannumeral#1\endcsname{#2}%
  }%
  {%
    \expandafter
      \xdef\csname @ff@xpages@\romannumeral#1\endcsname{%
       \csname @ff@xpages@\romannumeral#1\endcsname,#2}%
  }%
  \flf@message{Setting exclusion for flow frame
    \number#1\space\space to
    "\csname @ff@xpages@\romannumeral#1\endcsname"}%
}
```

`\ffswapoddeven` Swap odd and even offsets for a given flow frame. Do the main stuff for a given flow frame IDN.

```
\newcommand*{\@@flowframeswapcoords}[1]{%
  \setlength{\@ff@tmp@x}%
    {\csname col@\romannumeral#1@evenx\endcsname}
  \expandafter\setlength\csname col@\romannumeral#1@evenx\endcsname
    {\csname col@\romannumeral#1@posx\endcsname}%
  \expandafter\setlength\csname col@\romannumeral#1@posx\endcsname
    {\@ff@tmp@x}%
  \setlength{\@ff@tmp@y}%
    {\csname col@\romannumeral#1@eveny\endcsname}
  \expandafter\setlength\csname col@\romannumeral#1@eveny\endcsname
    {\csname col@\romannumeral#1@posy\endcsname}%
  \expandafter\setlength\csname col@\romannumeral#1@posy\endcsname
    {\@ff@tmp@y}%
}
```

`\ffswapoddeven` Allow user to specify flow frame either by IDN or IDL:

```
\newcommand*{\ffswapoddeven}{%
  \@ifstar\@sflowframeswapcoords\@flowframeswapcoords
}
```

`\@sflowframeswapcoords` Starred form

```
\newcommand*{\@sflowframeswapcoords}[1]{%
  \@for\@ff@id:=#1\do
  {%
    \@flowframeid{\@ff@id}%
    \@@flowframeswapcoords{\ff@id}%
  }%
}
```

`\@flowframeswapcoords` Unstarred form:

```
\newcommand*{\@flowframeswapcoords}[1]{%
  \ifthenelse{\equal{#1}{all}}%
  {%
    \ff@id=0\relax
    \whiledo{\ff@id<\c@maxflow}%
    {%
      \advance\ff@id by 1\relax
      \@@flowframeswapcoords{\ff@id}%
    }%
  }%
  {%
    \ifthenelse{\equal{#1}{odd} \TE@or \equal{#1}{even}}%
    {%
      \ifthenelse{\equal{#1}{odd}}{\@colN=1}{\@colN=2}%
      \whiledo{\@colN<\c@maxflow\TE@or\@colN=\c@maxflow}%
      {%
        \@@flowframeswapcoords{\@colN}%
        \advance\@colN by 2\relax
      }%
    }%
    {%
      \@for\@ff@id:=#1\do
      {%
        \def\@ff@numstart{0}%
        \def\@ff@numend{100000}%
        \@ff@getrange{\@ff@id}%
        \ifnum\@ff@numstart=0\relax
          \def\@ff@numstart{1}%
        \fi
        \ifnum\@ff@numend>\c@maxflow
          \def\@ff@numend{\c@maxflow}%
        \fi
        \@colN=\@ff@numstart
        \whiledo{\@colN<\@ff@numend \TE@or \@colN=\@ff@numend}%
        {%
          \@@flowframeswapcoords{\@colN}%
          \advance\@colN by 1\relax
        }%
      }%
    }%
  }%
}
```

Allow user to get the dimensions of flow frame (useful for flow frames created using `\Ncolumns` etc.) Only the IDN can be used for these commands.

`\flowframex`

```
\newcommand*{\flowframex}[1]{%
  \csname col@\romannumeral#1@posx\endcsname
}
```

`\flowframey`

```
\newcommand*{\flowframey}[1]{%
  \csname col@\romannumeral#1@posy\endcsname
}
```

`\flowframeevenx`

```
\newcommand*{\flowframeevenx}[1]{%
  \csname col@\romannumeral#1@evenx\endcsname
}
```

`\flowframeeveny`

```
\newcommand*{\flowframeeveny}[1]{%
  \csname col@\romannumeral#1@eveny\endcsname
}
```

`\flowframewidth`

```
\newcommand{\flowframewidth}[1]{%
  \csname colwidth\romannumeral#1\endcsname
}
```

`\flowframeheight`

```
\newcommand*{\flowframeheight}[1]{%
  \csname colheight\romannumeral#1\endcsname
}
```

`\@setframecol` Set the colour of the frame, this is a little tricky because the model may need to be specified in square brackets. First check to see if a colour model has been specified

```
\def\@setframecol{\@ifnextchar[\@@setframecol\@@setfr@mecol}
```

`\@@setframecol` A colour model has been specified.

```
\def\@@setframecol[#1]#2\end#3#4#5{%
  \expandafter\edef\csname @#5@#4@\romannumeral#3\endcsname{%
    [#1]{#2}}%
}
```

`\@@setfr@mecol` A colour model has not been specified.

```
\def\@@setfr@mecol#1\end#2#3#4{%
  \expandafter\edef\csname @#4@#3@\romannumeral#2\endcsname{{#1}}%
}
```

1.3 Static Frames

`\newstaticframe` Now deal with setting up the static frames. This is similar to the flow frames, except it has an associated LaTeX savebox rather than a TeX box. Syntax:
\newstaticframe[⟨*pages*⟩]{⟨*width*⟩}{⟨*height*⟩}{⟨*x*⟩}{⟨*y*⟩}[⟨*label*⟩]

As with \newflowframe, the final optional argument is dealt with at the end.

```
\newcommand*{\newstaticframe}{\@n@wstaticframe}
```

`\@n@wstaticframe`

```
\newcommand*{\@n@wstaticframe}{%
  \global\advance\c@maxstatic by 1\relax
  \newboolean{staticframe\romannumeral\c@maxstatic}%
  \@ifstar\@snewstaticframe\@newstaticframe
}
```

`\@snewstaticframe` Starred version (has a border):

```
\newcommand{\@snewstaticframe}{%
  \setboolean{staticframe\romannumeral\c@maxstatic}{true}%
  \@@newstaticframe
}
```

`\@newstaticframe` Unstarred version (no border):

```
\newcommand{\@newstaticframe}{%
  \setboolean{staticframe\romannumeral\c@maxstatic}{false}%
  \@@newstaticframe
}
```

`\@@newstaticframe` Now set up the static frame:

```
\newcommand*{\@@newstaticframe}[5][all]{%
  \expandafter
    \newbox\csname @staticframe@\romannumeral\c@maxstatic\endcsname
  \expandafter
    \newlength\csname @sf@\romannumeral\c@maxstatic @posx\endcsname
  \expandafter
    \newlength\csname @sf@\romannumeral\c@maxstatic @posy\endcsname
  \expandafter\setlength
    \csname @sf@\romannumeral\c@maxstatic @posx\endcsname{#4}%
  \expandafter\setlength
    \csname @sf@\romannumeral\c@maxstatic @posy\endcsname{#5}%
  \expandafter\newlength
    \csname @sf@\romannumeral\c@maxstatic @evenx\endcsname
  \expandafter\newlength
    \csname @sf@\romannumeral\c@maxstatic @eveny\endcsname
  \expandafter\setlength
    \csname @sf@\romannumeral\c@maxstatic @evenx\endcsname{#4}%
  \expandafter\setlength
    \csname @sf@\romannumeral\c@maxstatic @eveny\endcsname{#5}%
    {\@ff@tmp@x=#2\relax
  \@ff@tmp@y=#3\relax
  \expandafter
    \xdef\csname @sf@dim@\romannumeral\c@maxstatic\endcsname{%
      [c][\the\@ff@tmp@y][c]{\the\@ff@tmp@x}}}%
  \expandafter
    \def\csname @sf@col@\romannumeral\c@maxstatic\endcsname{%
      \flowframecol}%
  \expandafter
    \def\csname @sf@txtcol@\romannumeral\c@maxstatic\endcsname{%
      \flowframetextcol}%
  \expandafter
    \def\csname @sf@backcol@\romannumeral\c@maxstatic\endcsname{%
      {none}}%
  \expandafter
    \xdef\csname @sf@pages@\romannumeral\c@maxstatic\endcsname{#1}%
```

Page exclusion list:

```
\expandafter
  \gdef\csname @sf@xpages@\romannumeral\c@maxflow\endcsname{}%
\expandafter
  \gdef\csname @sf@offset@\romannumeral\c@maxstatic\endcsname{%
    compute}%
\expandafter
  \gdef\csname @sf@angle@\romannumeral\c@maxstatic\endcsname{0}%
\expandafter
  \gdef\csname @sf@shape@\romannumeral\c@maxstatic\endcsname{\relax}%
\expandafter
  \def\csname @sf@frametype@\romannumeral\c@maxstatic\endcsname{%
    fbox}%
\newboolean{@sf@clear@\romannumeral\c@maxstatic}%
\setboolean{@sf@clear@\romannumeral\c@maxstatic}{false}

\newboolean{@sf@hide@\romannumeral\c@maxstatic}%
\setboolean{@sf@hide@\romannumeral\c@maxstatic}{false}%
\newboolean{@sf@hidethis@\romannumeral\c@maxstatic}%
\setboolean{@sf@hidethis@\romannumeral\c@maxstatic}{false}%
\@ifnextchar[{\@s@tstaticframeid{\c@maxstatic}}%
{\@s@tstaticframeid{\c@maxstatic}[\number\c@maxstatic]}%
}
```

`\@s@tstaticframeid` Set the label for the static frame:

```
\def\@s@tstaticframeid#1[#2]{%
  \edef\ff@label{#2}%
  \@sf@checkuniqueidl{#1}{\ff@label}%
  \expandafter
    \xdef\csname @sf@id@\romannumeral#1\endcsname{\ff@label}%
}
```

`\@sf@checkuniqueidl` Check IDL #2 for static frame #1 is unique

```
\newcommand*{\@sf@checkuniqueidl}[2]{%
  \@colN=0\relax
  \whiledo{\@colN<\c@maxstatic}%
  {%
    \advance\@colN by 1\relax
    \ifnum\@colN=#1\relax
    \else
      \ifthenelse
      {%
        \equal{#2}{\csname @sf@id@\romannumeral\@colN\endcsname}%
      }%
      {%
        \PackageError{flowfram}%
        {Static frame IDL '#2' already defined}%
        {%
          You can't assign this label, as it is already defined
          for static frame \number\@colN
        }%
      }%
      {}%
    \fi
  }%
}
```

`\getstaticlabel` `\getstaticlabel{`⟨*idn*⟩`}` Gets the IDL for the static frame identified by its IDN.

```
\newcommand*{\getstaticlabel}[1]{%
  \csname @sf@id@\romannumeral#1\endcsname
}
```

`\getstaticid` `\getstaticid{`⟨*cmd*⟩`}{`⟨*idl*⟩`}` Gets the IDN for the static frame identified by its IDL and stores in ⟨*cmd*⟩ which must be a control sequence.

```
\newcommand*{\getstaticid}[2]{%
  \@staticframeid{#2}\edef#1{\number\ff@id}%
}
```

`\@staticframeid` Work out the IDN of the static frame with the given label. This iterates through each static frame, so if there are a lot of static frames, it may take a while. The IDN stored in `\ff@id`.

```
\newcommand*{\@staticframeid}[1]{%
  \@colN=0\relax
  \ff@id=0\relax
  \whiledo{\@colN<\c@maxstatic}%
  {%
    \advance\@colN by 1\relax
    \ifthenelse
    {%
      \equal{#1}{\csname @sf@id@\romannumeral\@colN\endcsname}%
    }%
    {%
      \ff@id=\@colN\relax
```

Break out of loop

```
      \@colN=\c@maxstatic
    }%
    {}%
  }%
  \ifnum\ff@id=0\relax
    \PackageError{flowfram}%
    {Can't find static frame id '#1'}{}%
  \fi
}
```

Make it easier to get the x and y values for static frames. (Width and height stored differently.)

`\staticframex`

```
\newcommand*{\staticframex}[1]{%
  \csname @sf@\romannumeral#1@posx\endcsname
}
```

`\staticframey`

```
\newcommand*{\staticframey}[1]{%
  \csname @sf@\romannumeral#1@posy\endcsname
}
```

`\staticframeevenx`

```
\newcommand*{\staticframeevenx}[1]{%
  \csname @sf@\romannumeral#1@evenx\endcsname
}
```

`\staticframeeveny`

```
\newcommand*{\staticframeeveny}[1]{%
  \csname @sf@\romannumeral#1@eveny\endcsname
}
```

`\setallstaticframes` Modify the settings for all the static frames:

```
\newcommand*{\setallstaticframes}[1]{%
  \@colN=0\relax
  \whiledo{\@colN<\c@maxstatic}%
  {%
    \advance\@colN by 1\relax
    \@@setstaticframe{\@colN}{#1}%
  }%
}
```

`\setstaticframe` Modify the settings for the specified static frames:

```
\newcommand*{\setstaticframe}{%
  \@ifstar\@ssetstaticframe\@setstaticframe
}
```

`\@ssetstaticframe` Starred version: Iterate through the comma-separated list of labels.

```
\newcommand*{\@ssetstaticframe}[2]{%
  \@for\@ff@id:=#1\do
  {%
    \@staticframeid{\@ff@id}%
    \@@setstaticframe{\ff@id}{#2}%
  }%
}
```

`\@setstaticframe` Unstarred version. Iterate through the comma-separated list of IDNs, and check for number ranges. Ensures that number ranges do not lie out of bounds.

```
\newcommand*{\@setstaticframe}[2]{%
  \ifthenelse{\equal{#1}{all}}%
  {%
    \setallstaticframes{#2}%
  }%
  {%
    \ifthenelse{\equal{#1}{odd} \TE@or \equal{#1}{even}}%
    {%
      \ifthenelse{\equal{#1}{odd}}{\@colN=1}{\@colN=2}%
      \whiledo{\@colN<\c@maxstatic\TE@or\@colN=\c@maxstatic}%
      {%
        \@@setstaticframe{\@colN}{#2}%
        \advance\@colN by 2\relax
      }%
    }%
    {%
      \@for\@ff@id:=#1\do
      {%
        \def\@ff@numstart{0}%
        \def\@ff@numend{10000}%
        \@ff@getrange{\@ff@id}%
        \ifnum\@ff@numstart=0\relax
          \def\@ff@numstart{1}%
        \fi
        \ifnum\@ff@numend>\c@maxstatic\relax
          \def\@ff@numend{\c@maxstatic}%
        \fi
        \@colN=\@ff@numstart\relax
        \whiledo{\@colN<\@ff@numend \TE@or \@colN=\@ff@numend}%
        {%
          \@@setstaticframe{\@colN}{#2}%
          \advance\@colN by 1\relax
        }%
      }%
    }%
  }%
}
```

`\@@setstaticframe` Modify the settings for the static frame whose IDN is given by the first argument.

```
\newcommand*{\@@setstaticframe}[2]{%
  \expandafter\expandafter\expandafter
    \@ff@getstaticpos\csname @sf@dim@\romannumeral#1\endcsname
  \def\ff@frame{}\edef\ff@width{\the\@ff@tmp@x}\def\ff@angle{}%
  \edef\ff@height{\the\@ff@tmp@y}\def\ff@style{}\def\ff@frametype{}%
  \def\ff@x{}\def\ff@y{}\def\ff@col{}\def\ff@txtcol{}%
  \def\ff@backcol{}%
  \def\ff@clear{}\def\ff@margin{}\def\ff@offset{}\def\ff@pages{}%
  \def\ff@label{}\def\ff@evenx{}\def\ff@eveny{}%
  \def\ff@oddx{}\def\ff@oddy{}%
  \def\ff@hide{}\def\ff@hidethis{}%
  \let\ff@shape\undefined
  \let\ff@xpages\undefined
  \setkeys{flowframe}{#2}%
  \ifdefempty{\ff@frame}{}%
  {%
    \setboolean{staticframe\romannumeral#1}{\ff@frame}%
  }%
  \ifdefempty{\ff@x}{}%
  {%
    \expandafter\global\expandafter
      \setlength\csname @sf@\romannumeral#1@posx\endcsname
        {\ff@x}%
    \expandafter\global\expandafter
      \setlength\csname @sf@\romannumeral#1@evenx\endcsname
        {\ff@x}%
  }%
  \ifdefempty{\ff@y}{}%
  {%
    \expandafter\global\expandafter
      \setlength\csname @sf@\romannumeral#1@posy\endcsname
        {\ff@y}%
    \expandafter\global\expandafter
      \setlength\csname @sf@\romannumeral#1@eveny\endcsname
        {\ff@y}%
  }%
  \ifdefempty{\ff@evenx}{}%
  {%
    \expandafter\global\expandafter
      \setlength\csname @sf@\romannumeral#1@evenx\endcsname
        {\ff@evenx}%
  }%
  \ifdefempty{\ff@eveny}{}%
  {%
    \expandafter\global\expandafter
      \setlength\csname @sf@\romannumeral#1@eveny\endcsname
      {\ff@eveny}%
  }%
  \ifdefempty{\ff@oddx}{}%
  {%
    \expandafter\global\expandafter
      \setlength\csname @sf@\romannumeral#1@posx\endcsname
      {\ff@oddx}%
  }%
  \ifdefempty{\ff@oddy}{}%
  {%
    \expandafter\global\expandafter
      \setlength\csname @sf@\romannumeral#1@posy\endcsname
      {\ff@oddy}%
  }%
  \expandafter
    \xdef\csname @sf@dim@\romannumeral#1\endcsname{%
      [c][\ff@height][\ff@valign]{\ff@width}}%
  \ifdefempty{\ff@frametype}{}%
  {%
    \expandafter
      \xdef\csname @sf@frametype@\romannumeral#1\endcsname{%
        \ff@frametype}%
  }%
  \ifdefempty{\ff@label}{}%
  {%
    \@s@tstaticframeid{#1}[\ff@label]%
  }
  \ifdefempty{\ff@col}{}%
  {%
    \expandafter\@setframecol\ff@col\end{#1}{col}{sf}%
  }%
  \ifdefempty{\ff@txtcol}{}%
  {%
    \expandafter\@setframecol\ff@txtcol\end{#1}{txtcol}{sf}%
  }%
```

```
\ifdefempty{\ff@backcol}{}%
{%
  \expandafter\@setframecol\ff@backcol\end{#1}{backcol}{sf}%
}%
\ifdefempty{\ff@offset}{}%
{%
  \expandafter
    \xdef\csname @sf@offset@\romannumeral#1\endcsname{\ff@offset}%
}%
\ifdefempty{\ff@angle}{}%
{%
  \expandafter
    \xdef\csname @sf@angle@\romannumeral#1\endcsname{\ff@angle}%
}%
\ifundef{\ff@shape}{}%
{%
  \expandafter\global\expandafter
    \let\csname @sf@shape@\romannumeral#1\endcsname\ff@shape
}%
\ifdefempty{\ff@pages}{}%
{%
   \staticsetpagelist{#1}{\ff@pages}%
}%
\ifundef{\ff@xpages}{}%
{%
   \staticsetexclusion{#1}{\ff@xpages}%
}%
\ifdefempty{\ff@hide}{}%
{%
  \setboolean{@sf@hide@\romannumeral#1}{\ff@hide}%
}%
\ifdefempty{\ff@hidethis}{}%
{%
  \global\csletcs{if@sf@hidethis@\romannumeral#1}{if\ff@hidethis}%
}%
\ifdefempty{\ff@clear}{}%
{%
  \setboolean{@sf@clear@\romannumeral#1}{\ff@clear}%
}%
\ifdefempty{\ff@margin}{}%
{%
  \PackageError{flowfram}%
  {Key 'margin' not available for static frames}%
  {Static frames don't have marginal notes}%
}%
\ifdefempty{\ff@style}{}%
{%
  \PackageError{flowfram}%
  {Key 'style' not available for static frames}{}%
  }%
}
```

\simpar Simulate paragraph break inside \shapepar

```
 %\newcommand*{\simpar}{\hfil\vadjust{\vskip\parskip}\break\indent}
\newcommand*{\simpar}{\hfill\\\indent\mbox{}}
```

\ffpshpar Provide means to allow parshape to be carried over a paragraph break.

```
\let\FLForgpar\par
\newcommand{\ffpshpar}{%
  \edef\flf@next{\hangafter=\the\hangafter
    \hangindent=\the\hangindent}%
  \FLForgpar\flf@next
  \edef\flf@next{\prevgraf=\the\prevgraf}%
  \@ff@parshape\indent\mbox{}\flf@next
}
```

Provide a means to have section headings within \parshape.

\@ff@parshape

```
\def\@ff@parshape{\parshape=0}
```

\@ff@sectionhead

```
\newcommand*{\@ff@sectionhead}[1]{%
  \def\ff@sechead{#1}%
  \ffpshpar
  \@ifstar{\@s@ff@heading}{\@dblarg\@ff@heading}%
}
```

\@s@ff@heading

```
\def\@s@ff@heading#1{%
  \@ifundefined{@ff@old\ff@sechead}%
  {%
    \PackageError{flowfram}%
    {Unknown heading command '\ff@sechead'}{}%
  }%
  {%
    \begingroup
    \edef\flf@next{\hangafter=\the\hangafter
      \hangindent=\the\hangindent}%
    \FLForgpar\flf@next
    \let\par=\FLForgpar
    \edef\flf@next{\prevgraf=\the\prevgraf}%
    \csname @ff@old\ff@sechead\endcsname*{%
      \@ff@parshape\flf@next #1}%
    \xdef\flf@next{%
      \@ff@parshape
      \prevgraf=\the\prevgraf}%
    \endgroup
  }%
  \mbox{}\flf@next
  \let\flf@next\undefined
}
```

`\@ff@heading`

```
\def\@ff@heading[#1]#2{%
  \@ifundefined{@ff@old\ff@sechead}%
  {%
    \PackageError{flowfram}%
    {Unknown heading command '\ff@sechead'}{}%
  }%
  {%
    \begingroup
    \edef\flf@next{%
      \hangafter=\the\hangafter
      \hangindent=\the\hangindent}%
    \FLForgpar\flf@next
    \let\par=\FLForgpar
    \edef\flf@next{\prevgraf=\the\prevgraf}%
    \csname @ff@old\ff@sechead\endcsname[#1]{%
      \@ff@parshape\flf@next #2}%
    \xdef\flf@next{\@ff@parshape
      \prevgraf=\the\prevgraf}%
    \endgroup
  }%
  \mbox{}\flf@next
  \let\flf@next\undefined
}
```

`\@ff@setsecthead` Define command to switch to adjusted section headings:

```
\newcommand*{\@ff@setsecthead}{%
  \let\@ff@oldsection=\section
  \let\@ff@oldsubsection=\subsection
  \let\@ff@oldsubsubsection=\subsubsection
  \let\@ff@oldparagraph=\paragraph
  \let\@ff@oldsubparagraph=\subparagraph
  \def\section{\@ff@sectionhead{section}}%
  \def\subsection{\@ff@sectionhead{subsection}}%
  \def\subsubsection{\@ff@sectionhead{subsubsection}}%
  \def\paragraph{\@ff@sectionhead{paragraph}}%
  \def\subparagraph{\@ff@sectionhead{subparagraph}}%
}
```

`\@ff@getshape` Determine what shape command is being used:

```
\def\@ff@getshape#1#2\relax{%
  \ifdefequal{#1}{\parshape}%
  {%
    \def\ff@shape{1}%
  }%
  {%
    \ifdefequal{#1}{\shapepar}%
    {%
      \def\ff@shape{2}%
    }%
    {%

       \ifdefequal{#1}{\Shapepar}%
       {%
         \def\ff@shape{2}%
       }%
       {%
         \ifx#1\relax
           \def\ff@shape{0}%
         \else
           \PackageError{flowfram}{Unknown shape \string#1}{}%
           \def\ff@shape{2}%
         \fi
      }%
    }%
  }%
}
```

`\@ff@disablesec` Disable sectioning commands

```
\newcommand*{\@ff@disablesec}{%
  \def\section{%
    \PackageError{flowfram}%
    {You can't have sectioning commands within a \string\shapepar}{}%
  }%
  \def\subsection{%
    \PackageError{flowfram}%
    {You can't have sectioning commands within a \string\shapepar}{}%
  }%
  \def\subsubsection{%
    \PackageError{flowfram}%
    {You can't have sectioning commands within a \string\shapepar}{}%
  }%
  \def\paragraph{%
    \PackageError{flowfram}%
    {You can't have sectioning commands within a \string\shapepar}{}%
  }%
  \def\subparagraph{%
    \PackageError{flowfram}%
    {You can't have sectioning commands within a \string\shapepar}{}%
  }%
}
```

`staticcontents` Set the contents of the static frame given by its IDN. Syntax: `\begin{staticcontents}{`⟨*idn*⟩`}`.

```
\newbox\staticframe
\newenvironment{staticcontents}[1]{%
  \let\continueonframe=\@staticcontinueonframe
  \@beginstaticcontents{#1}%
}%
{%
  \@endstaticcontents
  \ignorespaces
}
```

staticcontents* Set the contents of the static frame given by its IDL. Syntax: `\begin{staticcontents*}{`⟨*label*⟩`}`.

```
\newenvironment{staticcontents*}[1]{%
  \@staticframeid{#1}%
  \let\continueonframe=\@staticscontinueonframe
  \@beginstaticcontents{\ff@id}%
}%
{%
  \@endstaticcontents
  \ignorespaces
}
```

Begin staticcontents stuff.

```
\newcommand{\@beginstaticcontents}[1]{%
  \@ifundefined{@staticframe@\romannumeral#1}%
  {%
    \PackageError{flowfram}{Static frame '#1' not defined}{}%
  }%
  {}%
  \expandafter\let\expandafter\@ff@parshape\csname @sf@shape@\romannumeral#1\endcsname
  \expandafter\@ff@getshape\@ff@parshape\relax
  \ifcase\ff@shape
```

no shape:

```
    \edef\@sf@mpg{%
      \noexpand
      \begin{minipage}\csname @sf@dim@\romannumeral#1\endcsname
        \noexpand\begingroup
        \noexpand\let\noexpand\FLForgpar=\noexpand\par
    }%
  \or
```

`\parshape`:

```
    \edef\@sf@mpg{%
      \noexpand
      \begin{minipage}\csname @sf@dim@\romannumeral#1\endcsname
        \@ff@parshape
        \noexpand\begingroup
        \noexpand\let\noexpand\FLForgpar=\noexpand\par
        \noexpand\let\noexpand\par=\noexpand\ffpshpar
        \noexpand\@ff@setsecthead
    }%
  \or
```

\shapepar or \Shapepar:

```
    \edef\@sf@mpg{%
      \noexpand
      \begin{minipage}\csname @sf@dim@\romannumeral#1\endcsname
        \noexpand\begingroup
        \noexpand\@ff@disablesec
        \noexpand\@ff@parshape
    }%
  \fi
  \edef\@sf@thisframe{\csname @staticframe@\romannumeral#1\endcsname}%
  \begin{lrbox}{\staticframe}%
    \edef\ff@txtcol{\csname @sf@txtcol@\romannumeral#1\endcsname}%
    \@s@tfftextcol\noindent
    \@sf@mpg
    \setlength\parindent\sdfparindent
}
```

End staticcontents stuff

```
\newcommand*{\@endstaticcontents}{%
  \ifnum\ff@shape=2\relax
    \par
  \else
    \FLForgpar
  \fi
  \endgroup
  \end{minipage}%
  \end{lrbox}%
  \expandafter\global\expandafter
    \sbox\@sf@thisframe{\usebox\staticframe}%
}
```

`\setstaticcontents`

Provide a command version. Syntax: `\setstaticcontents{`⟨*idn*⟩`}{`⟨*text*⟩`}`.

```
\newcommand{\setstaticcontents}{%
  \@ifstar\@sstaticconts\@staticconts
}
```

`\@sstaticconts`

Starred version: static frame identified by label.

```
\newcommand{\@sstaticconts}[2]{%
  \begin{staticcontents*}{#1}%
    #2%
  \end{staticcontents*}%
}
```

`\@staticconts`

Unstarred version: static frame identified by IDN.

```
\newcommand{\@staticconts}[2]{%
  \begin{staticcontents}{#1}%
  #2%
  \end{staticcontents}%
}
```

`\staticsetpagelist` Sets the page list for the static frame given by #1 (the IDN).

```
\newcommand*{\staticsetpagelist}[2]{%
  \expandafter
    \xdef\csname @sf@pages@\romannumeral#1\endcsname{#2}%
  \flf@message{Setting page range for static frame
    \number#1\space\space to "#2"}%
}
```

`\staticsetexclusion` Sets the exclusion list for the static frame given by #1 (the IDN).

```
\newcommand*{\staticsetexclusion}[2]{%
  \expandafter
    \xdef\csname @sf@xpages@\romannumeral#1\endcsname{#2}%
  \flf@message{Setting exclusion for static frame
    \number#1\space\space to "#2"}%
}
```

`\staticaddexclusion` Adds to the exclusion list for the static frame given by #1 (the IDN).

```
\newcommand*{\staticaddexclusion}[2]{%
  \ifcsempty{@sf@xpages@\romannumeral#1}
  {%
    \expandafter
      \xdef\csname @sf@xpages@\romannumeral#1\endcsname{#2}%
  }%
  {%
    \expandafter
      \xdef\csname @sf@xpages@\romannumeral#1\endcsname{%
       \csname @sf@xpages@\romannumeral#1\endcsname,#2}%
  }%
  \flf@message{Setting exclusion for static frame
    \number#1\space\space to
    "\csname @sf@xpages@\romannumeral#1\endcsname"}%
}
```

`@@staticframeswapcoords` Swap odd and even offsets for a given static frame. Do the main stuff for a given static frame IDN.

```
\newcommand*{\@@staticframeswapcoords}[1]{%
  \setlength{\@ff@tmp@x}%
    {\csname @sf@\romannumeral#1@evenx\endcsname}
  \expandafter\setlength\csname @sf@\romannumeral#1@evenx\endcsname
    {\csname @sf@\romannumeral#1@posx\endcsname}%
  \expandafter\setlength\csname @sf@\romannumeral#1@posx\endcsname
    {\@ff@tmp@x}%
  \setlength{\@ff@tmp@y}%
    {\csname @sf@\romannumeral#1@eveny\endcsname}
  \expandafter\setlength\csname @sf@\romannumeral#1@eveny\endcsname
    {\csname @sf@\romannumeral#1@posy\endcsname}%
  \expandafter\setlength\csname @sf@\romannumeral#1@posy\endcsname
    {\@ff@tmp@y}%
}
```

`\sfswapoddeven` Allow user to specify flow frame either by IDN or IDL:

```
\newcommand*{\sfswapoddeven}{%
  \@ifstar\@sstaticframeswapcoords\@staticframeswapcoords
}
```

`@sstaticframeswapcoords` Starred form

```
\newcommand*{\@sstaticframeswapcoords}[1]{%
  \@for\@ff@id:=#1\do
  {%
    \@staticframeid{\@ff@id}%
    \@@staticframeswapcoords{\ff@id}%
  }%
}
```

`\@staticframeswapcoords` Unstarred form:

```
\newcommand*{\@staticframeswapcoords}[1]{%
  \ifthenelse{\equal{#1}{all}}%
  {%
    \ff@id=0\relax
    \whiledo{\ff@id<\c@maxflow}%
    {%
      \advance\ff@id by 1\relax
      \@@staticframeswapcoords{\ff@id}%
    }%
  }%
  {%
    \ifthenelse{\equal{#1}{odd} \TE@or \equal{#1}{even}}%
    {%
      \ifthenelse{\equal{#1}{odd}}{\@colN=1}{\@colN=2}%
      \whiledo{\@colN<\c@maxflow\TE@or\@colN=\c@maxflow}%
      {%
        \@@staticframeswapcoords{\@colN}%
        \advance\@colN by 2\relax
      }%
    }%
    {%
      \@for\@ff@id:=#1\do
      {%
        \def\@ff@numstart{0}\def\@ff@numend{100000}%
        \@ff@getrange{\@ff@id}%
        \ifnum\@ff@numstart=0\relax
          \def\@ff@numstart{1}%
        \fi
        \ifnum\@ff@numend>\c@maxflow
          \def\@ff@numend{\c@maxflow}%
        \fi
        \@colN=\@ff@numstart
        \whiledo{\@colN<\@ff@numend \TE@or \@colN=\@ff@numend}%
        {%
          \@@staticframeswapcoords{\@colN}%
          \advance\@colN by 1\relax
        }%
      }%
    }%
  }%
}
```

1.4 Dynamic Frames

Now deal with the dynamic frames. These are very similar to the static frames, but instead of having a savebox, the contents of the dynamic frame are stored in a macro.

`\newdynamicframe` Syntax:

`\newdynamicframe[`⟨*pages*⟩`]{`⟨*width*⟩`}{`⟨*height*⟩`}{`⟨*x*⟩`}{`⟨*y*⟩`}[`⟨*label*⟩`]`

```
\newcommand*{\newdynamicframe}{%
  \@n@wdynamicframe
}
\newcommand*{\@n@wdynamicframe}{%
  \global\advance\c@maxdynamic by 1\relax
  \newboolean{dynamicframe\romannumeral\c@maxdynamic}
  \@ifstar\@snewdynamicframe\@newdynamicframe
}
```

`\@snewdynamicframe` Starred version: has a border.

```
\newcommand*{\@snewdynamicframe}{%
  \setboolean{dynamicframe\romannumeral\c@maxdynamic}{true}%
  \@@newdynamicframe
}
```

`\@newdynamicframe` Unstarred version: no border.

```
\newcommand*{\@newdynamicframe}{%
  \setboolean{dynamicframe\romannumeral\c@maxdynamic}{false}%
  \@@newdynamicframe
}
```

`\@@newdynamicframe` Create new dynamic frame:

```
\newcommand*{\@@newdynamicframe}[5][all]{%
  \expandafter
    \gdef\csname @dynamicframe@\romannumeral\c@maxdynamic\endcsname{}%
  \expandafter
    \newlength\csname @df@\romannumeral\c@maxdynamic @posx\endcsname
  \expandafter
    \newlength\csname @df@\romannumeral\c@maxdynamic @posy\endcsname
  \expandafter\setlength
    \csname @df@\romannumeral\c@maxdynamic @posx\endcsname{#4}%
  \expandafter\setlength
    \csname @df@\romannumeral\c@maxdynamic @posy\endcsname{#5}%
  \expandafter\newlength
    \csname @df@\romannumeral\c@maxdynamic @evenx\endcsname
  \expandafter\newlength
    \csname @df@\romannumeral\c@maxdynamic @eveny\endcsname
  \expandafter\setlength
    \csname @df@\romannumeral\c@maxdynamic @evenx\endcsname{#4}%
  \expandafter\setlength
    \csname @df@\romannumeral\c@maxdynamic @eveny\endcsname{#5}%
  {%
    \@ff@tmp@x=#2\relax
    \@ff@tmp@y=#3\relax
    \expandafter
      \xdef\csname @df@dim@\romannumeral\c@maxdynamic\endcsname{%
        [c][\the\@ff@tmp@y][t]{\the\@ff@tmp@x}%
      }%
  }%
  \expandafter
    \gdef\csname @df@col@\romannumeral\c@maxdynamic\endcsname{%
      \flowframecol
    }%
  \expandafter
    \gdef\csname @df@txtcol@\romannumeral\c@maxdynamic\endcsname{%
      \flowframetextcol
    }%
  \expandafter
    \gdef\csname @df@backcol@\romannumeral\c@maxdynamic\endcsname{%
      {none}}%
  \expandafter
    \gdef\csname @df@pages@\romannumeral\c@maxdynamic\endcsname{#1}%
```

Page exclusion list:

```
\expandafter
  \gdef\csname @df@xpages@\romannumeral\c@maxflow\endcsname{}%
\expandafter
  \gdef\csname @df@frametype@\romannumeral\c@maxdynamic\endcsname{%
    fbox}%
\expandafter
  \gdef\csname @df@style@\romannumeral\c@maxdynamic\endcsname{relax}%
\expandafter
  \gdef\csname @df@offset@\romannumeral\c@maxdynamic\endcsname{compute}%
\expandafter
  \gdef\csname @df@angle@\romannumeral\c@maxdynamic\endcsname{0}%
\expandafter
  \gdef\csname @df@shape@\romannumeral\c@maxdynamic\endcsname{\relax}%
\newboolean{@df@clear@\romannumeral\c@maxdynamic}%
```

```
    \setboolean{@df@clear@\romannumeral\c@maxdynamic}{false}%

    \newboolean{@df@hide@\romannumeral\c@maxdynamic}%
    \setboolean{@df@hide@\romannumeral\c@maxdynamic}{false}%
    \newboolean{@df@hidethis@\romannumeral\c@maxdynamic}%
    \setboolean{@df@hidethis@\romannumeral\c@maxdynamic}{false}%
    \@ifnextchar[{\@s@tdynamicframeid{\c@maxdynamic}}%
      {\@s@tdynamicframeid{\c@maxdynamic}[\number\c@maxdynamic]}%
  }
```

`\@s@tdynamicframeid` Set the label for the given dynamic frame:

```
  \def\@s@tdynamicframeid#1[#2]{%
    \edef\ff@label{#2}%
    \@df@checkuniqueidl{#1}{\ff@label}%
    \expandafter
      \xdef\csname @df@id@\romannumeral#1\endcsname{\ff@label}%
  }
```

`\@df@checkuniqueidl` Check IDL #2 for static frame #1 is unique

```
  \newcommand*{\@df@checkuniqueidl}[2]{%
    \@colN=0\relax
    \whiledo{\@colN<\c@maxdynamic}%
    {%
      \advance\@colN by 1\relax
      \ifnum\@colN=#1\relax
      \else
        \ifthenelse
        {%
           \equal{#2}%
             {\csname @df@id@\romannumeral\@colN\endcsname}%
        }%
        {%
          \PackageError{flowfram}%
          {Dynamic frame IDL '#2' already defined}%
          {%
            You can't assign this label, as it is already defined
            for dynamic frame \number\@colN
          }%
        }%
        {}%
      \fi
    }%
  }
```

`\getdynamiclabel` `\getdynamiclabel{`⟨*idn*⟩`}` Gets the IDL for the dynamic frame identified by its IDN.

```
  \newcommand*{\getdynamiclabel}[1]{%
    \csname @df@id@\romannumeral#1\endcsname
  }
```

`\getdynamicid` `\getdynamicid{`⟨*cmd*⟩`}{`⟨*idl*⟩`}` Gets the IDN for the dynamic frame identified by its IDL and stores in ⟨*cmd*⟩ which must be a control sequence.

```
\newcommand*{\getdynamicid}[2]{%
  \@dynamicframeid{#2}\edef#1{\number\ff@id}%
}
```

`\@dynamicframeid` Determine the IDN of the dynamic frame from its label. The IDN is stored in `\ff@id`.

```
\newcommand*{\@dynamicframeid}[1]{%
  \@colN=0\relax
  \ff@id=0\relax
  \whiledo{\@colN<\c@maxdynamic}%
  {%
    \advance\@colN by 1\relax
    \ifthenelse
    {%
      \equal{#1}{\csname @df@id@\romannumeral\@colN\endcsname}%
    }%
    {%
      \ff@id=\@colN\relax
```

Break out of loop

```
      \@colN=\c@maxdynamic
    }%
    {}%
  }%
  \ifnum\ff@id=0\relax
    \PackageError{flowfram}%
    {Can't find dynamic frame id '#1'}{}%
  \fi
}
```

`\@getframeid` `\@getframeid{`⟨*type*⟩`}{`⟨*idl*⟩`}`

Gets the IDL for the frame of type ⟨*type*⟩ whose IDL is given by ⟨*idl*⟩. The IDN is stored in `\ff@id`.

```
\newcommand*{\@getframeid}[2]{%
  \@ifdefined{@#1frameid}%
  {\csname @#1frameid\endcsname{#2}}%
  {%
    \PackageError{flowfram}%
    {Unknown frame type `#1'}%
    {Frame types can be one of: flow, static or dynamic}%
  }%
}
```

Make it easier to get the x and y values for dynamic frames. (Width and height stored differently.)

`\dynamicframex`

```
\newcommand*{\dynamicframex}[1]{%
  \csname @df@\romannumeral#1@posx\endcsname
}
```

`\dynamicframey`

```
\newcommand*{\dynamicframey}[1]{%
  \csname @df@\romannumeral#1@posy\endcsname
}
```

`\dynamicframeevenx`

```
\newcommand*{\dynamicframeevenx}[1]{%
  \csname @df@\romannumeral#1@evenx\endcsname
}
```

`\dynamicframeeveny`

```
\newcommand*{\dynamicframeeveny}[1]{%
  \csname @df@\romannumeral#1@eveny\endcsname
}
```

`\setalldynamicframes` Change the settings for all the dynamic frames:

```
\newcommand*{\setalldynamicframes}[1]{%
  \@colN=0\relax
  \whiledo{\@colN<\c@maxdynamic}%
  {%
    \advance\@colN by 1\relax
    \@@setdynamicframe{\@colN}{#1}%
  }%
}
```

`\setdynamicframe` Change the settings for specified dynamic frames:

```
\newcommand*{\setdynamicframe}{%
  \@ifstar\@ssetdynamicframe\@setdynamicframe
}
```

`\@ssetdynamicframe` Starred version: iterate through comma-separated list of labels.

```
\newcommand*{\@ssetdynamicframe}[2]{%
  \@for\@ff@id:=#1\do{%
    \@dynamicframeid{\@ff@id}%
    \@@setdynamicframe{\ff@id}{#2}%
  }%
}
```

`\@setdynamicframe` Unstarred version: iterate through comma-separated list of ID numbers. Include provision for number ranges. If necessary, modify number ranges to ensure they are valid.

```
\newcommand*{\@setdynamicframe}[2]{%
  \ifthenelse{\equal{#1}{all}}%
  {%
```

```
    \setalldynamicframes{#2}%
  }%
  {%
    \ifthenelse{\equal{#1}{odd} \TE@or \equal{#1}{even}}%
    {%
      \ifthenelse{\equal{#1}{odd}}%
      {\@colN=1}%
      {\@colN=2}%
      \whiledo{\@colN<\c@maxdynamic\TE@or\@colN=\c@maxdynamic}%
      {%
        \@@setdynamicframe{\@colN}{#2}%
        \advance\@colN by 2\relax
      }%
    }%
    {%
      \@for\@ff@id:=#1\do{%
        \def\@ff@numstart{0}%
        \def\@ff@numend{10000}%
        \@ff@getrange{\@ff@id}%
        \ifnum\@ff@numstart=0\relax
          \def\@ff@numstart{1}%
        \fi
        \ifnum\@ff@numend>\c@maxdynamic\relax
          \def\@ff@numend{\c@maxdynamic}%
        \fi
        \@colN=\@ff@numstart\relax
        \whiledo{\@colN<\@ff@numend \TE@or \@colN=\@ff@numend}%
        {%
          \@@setdynamicframe{\@colN}{#2}%
          \advance\@colN by 1\relax
        }%
      }%
    }%
  }%
}
```

`\@@setdynamicframe` Change the setting for the dynamic frame given by its IDN.

```
\newcommand*{\@@setdynamicframe}[2]{%
  \expandafter\expandafter\expandafter
    \@ff@getstaticpos\csname @df@dim@\romannumeral#1\endcsname
  \def\ff@frame{}\edef\ff@width{\the\@ff@tmp@x}%
  \edef\ff@height{\the\@ff@tmp@y}\def\ff@style{}\def\ff@frametype{}%
  \def\ff@x{}\def\ff@y{}\def\ff@col{}\def\ff@txtcol{}\def\ff@backcol{}%
  \def\ff@clear{}\def\ff@margin{}\def\ff@offset{}\def\ff@pages{}%
  \def\ff@label{}\def\ff@evenx{}\def\ff@eveny{}%
  \def\ff@oddx{}\def\ff@oddy{}\def\ff@angle{}%
  \def\ff@hide{}\def\ff@hidethis{}%
  \let\ff@shape\undefined
  \let\ff@xpages\undefined
```

```
\setkeys{flowframe}{#2}%
\ifdefempty{\ff@frame}%
{}%
{%
  \setboolean{dynamicframe\romannumeral#1}{\ff@frame}%
}%
\ifdefempty{\ff@x}%
{}%
{%
  \expandafter\global\expandafter\setlength
    \csname @df@\romannumeral#1@posx\endcsname{\ff@x}%
  \expandafter\global\expandafter\setlength
    \csname @df@\romannumeral#1@evenx\endcsname{\ff@x}%
}%
\ifdefempty{\ff@y}%
{}%
{%
  \expandafter\global\expandafter\setlength
    \csname @df@\romannumeral#1@posy\endcsname{\ff@y}%
  \expandafter\global\expandafter\setlength
    \csname @df@\romannumeral#1@eveny\endcsname{\ff@y}%
}%
\ifdefempty{\ff@evenx}%
{}%
{%
  \expandafter\global\expandafter\setlength
    \csname @df@\romannumeral#1@evenx\endcsname{\ff@evenx}%
}%
\ifdefempty{\ff@eveny}%
{}%
{%
  \expandafter\global\expandafter\setlength
    \csname @df@\romannumeral#1@eveny\endcsname{\ff@eveny}%
}%
\ifdefempty{\ff@oddx}%
{}%
{%
  \expandafter\global\expandafter\setlength
    \csname @df@\romannumeral#1@posx\endcsname{\ff@oddx}%
}%
\ifdefempty{\ff@oddy}%
{}%
{%
  \expandafter\global\expandafter\setlength
    \csname @df@\romannumeral#1@posy\endcsname{\ff@oddy}%
}%
\expandafter\xdef\csname @df@dim@\romannumeral#1\endcsname{%
  [c][\ff@height][\ff@valign]{\ff@width}%
}%
```

```
\ifdefempty{\ff@label}%
{}%
{%
  \@s@tdynamicframeid{#1}[\ff@label]%
}%
\ifdefempty{\ff@frametype}%
{}%
{%
  \expandafter
    \xdef\csname @df@frametype@\romannumeral#1\endcsname{%
      \ff@frametype
    }%
}%
\ifdefempty{\ff@col}%
{}%
{%
  \expandafter\@setframecol\ff@col\end{#1}{col}{df}%
}%
\ifdefempty{\ff@txtcol}%
{}%
{%
  \expandafter\@setframecol\ff@txtcol\end{#1}{txtcol}{df}%
}%
\ifdefempty{\ff@backcol}%
{}%
{%
  \expandafter\@setframecol\ff@backcol\end{#1}{backcol}{df}%
}%
\ifdefempty{\ff@offset}%
{}%
{%
  \expandafter
    \xdef\csname @df@offset@\romannumeral#1\endcsname{\ff@offset}%
}%
\ifdefempty{\ff@angle}%
{}%
{%
  \expandafter
    \xdef\csname @df@angle@\romannumeral#1\endcsname{\ff@angle}%
}%
\ifundef{\ff@shape}{}%
{%
  \expandafter\global\expandafter
    \let\csname @df@shape@\romannumeral#1\endcsname\ff@shape
}%
\ifdefempty{\ff@pages}%
{}%
{%
  \dynamicsetpagelist{#1}{\ff@pages}%
  }%
  \ifundef{\ff@xpages}{}%
  {%
    \dynamicsetexclusion{#1}{\ff@xpages}%
  }%
  \ifdefempty{\ff@style}%
  {}%
  {%
    \ifcsundef{\ff@style}%
    {%
      \PackageError{flowfram}%
      {Unknown style '\ff@style'}%
      {%
        The command \expandafter\@gobble\string\\\ff@style
        \space has not been defined%
      }%
    }%
    {%
      \expandafter
        \xdef\csname @df@style@\romannumeral#1\endcsname{\ff@style}%
    }%
  }%
  \ifdefempty{\ff@clear}%
  {}%
  {%
    \setboolean{@df@clear@\romannumeral#1}{\ff@clear}%
  }%
  \ifdefempty{\ff@margin}%
  {}%
  {%
    \PackageError{flowfram}%
    {%
      Key 'margin' not available for dynamic frames%
    }%
    {dynamic frames don't have marginal notes}%
  }%
  \ifdefempty{\ff@hide}{}%
  {%
    \setboolean{@df@hide@\romannumeral#1}{\ff@hide}%
  }%
  \ifdefempty{\ff@hidethis}{}%
  {%
    \global\csletcs{if@df@hidethis@\romannumeral#1}{if\ff@hidethis}%
  }%
}
```

`\dynamicsetpagelist` Sets the page list for the dynamic frame given by #1 (the IDN).

```
\newcommand*{\dynamicsetpagelist}[2]{%
  \expandafter
```

```
    \xdef\csname @df@pages@\romannumeral#1\endcsname{#2}%
  \flf@message{Setting page range for dynamic frame
    \number#1\space\space to "#2"}%
}
```

`\dynamicsetexclusion` Sets the exclusion list for the dynamic frame given by `#1` (the IDN).

```
\newcommand*{\dynamicsetexclusion}[2]{%
  \expandafter
    \xdef\csname @df@xpages@\romannumeral#1\endcsname{#2}%
  \flf@message{Setting exclusion for dynamic frame
    \number#1\space\space to "#2"}%
}
```

`\dynamicaddexclusion` Adds to the exclusion list for the dynamic frame given by `#1` (the IDN).

```
\newcommand*{\dynamicaddexclusion}[2]{%
  \ifcsempty{@df@xpages@\romannumeral#1}
  {%
    \expandafter
      \xdef\csname @df@xpages@\romannumeral#1\endcsname{#2}%
  }%
  {%
    \expandafter
      \xdef\csname @df@xpages@\romannumeral#1\endcsname{%
       \csname @df@xpages@\romannumeral#1\endcsname,#2}%
  }%
  \flf@message{Setting exclusion for dynamic frame
    \number#1\space\space to
    "\csname @df@xpages@\romannumeral#1\endcsname"}%
}
```

`@dynamicframeswapcoords` Swap odd and even offsets for a given dynamic frame. Do the main stuff for a given dynamic frame IDN.

```
\newcommand*{\@@dynamicframeswapcoords}[1]{%
  \setlength{\@ff@tmp@x}%
    {\csname @df@\romannumeral#1@evenx\endcsname}%
  \expandafter\setlength
    \csname @df@\romannumeral#1@evenx\endcsname
    {\csname @df@\romannumeral#1@posx\endcsname}%
  \expandafter\setlength
    \csname @df@\romannumeral#1@posx\endcsname{\@ff@tmp@x}%
  \setlength{\@ff@tmp@y}%
    {\csname @df@\romannumeral#1@eveny\endcsname}%
  \expandafter\setlength
    \csname @df@\romannumeral#1@eveny\endcsname
    {\csname @df@\romannumeral#1@posy\endcsname}%
  \expandafter\setlength\csname @df@\romannumeral#1@posy\endcsname
    {\@ff@tmp@y}%
}
```

`\dfswapoddeven` Allow user to specify flow frame either by IDN or IDL:

```
\newcommand*{\dfswapoddeven}{%
\@ifstar\@sdynamicframeswapcoords\@dynamicframeswapcoords}
```

`sdynamicframeswapcoords` Starred form

```
\newcommand*{\@sdynamicframeswapcoords}[1]{%
  \@for\@ff@id:=#1\do{%
  \@dynamicframeid{\@ff@id}%
  \@@dynamicframeswapcoords{\ff@id}}%
}
```

`@dynamicframeswapcoords` Unstarred form:

```
\newcommand*{\@dynamicframeswapcoords}[1]{%
  \ifthenelse{\equal{#1}{all}}%
  {%
    \ff@id=0\relax
    \whiledo{\ff@id<\c@maxflow}%
    {%
      \advance\ff@id by 1\relax
      \@@dynamicframeswapcoords{\ff@id}%
    }%
  }%
  {%
    \ifthenelse{\equal{#1}{odd} \TE@or \equal{#1}{even}}%
    {%
      \ifthenelse{\equal{#1}{odd}}%
      {\@colN=1}%
      {\@colN=2}%
      \whiledo{\@colN<\c@maxflow\TE@or\@colN=\c@maxflow}%
      {%
        \@@dynamicframeswapcoords{\@colN}%
        \advance\@colN by 2\relax
      }%
    }%
    {%
      \@for\@ff@id:=#1\do{%
        \def\@ff@numstart{0}%
        \def\@ff@numend{10000}%
        \@ff@getrange{\@ff@id}%
        \ifnum\@ff@numstart=0\relax
          \def\@ff@numstart{1}%
        \fi
        \ifnum\@ff@numend>\c@maxflow
          \def\@ff@numend{\c@maxflow}%
        \fi
        \@colN=\@ff@numstart
        \whiledo{\@colN<\@ff@numend \TE@or \@colN=\@ff@numend}%
        {%
          \@@dynamicframeswapcoords{\@colN}%
          \advance\@colN by 1\relax
        }%
      }%
    }%
  }%
}
```

Set the contents of a dynamic frame.

dynamiccontents Syntax: `\begin{dynamiccontents}{⟨idn⟩}`

The contents of the dynamiccontents environment needs to be stored in the control sequence `\@dynamicframe@`⟨*rn*⟩ (where ⟨*rn*⟩ is the ⟨*idn*⟩ as a roman numeral.)

```
\newenvironment{dynamiccontents}[1]{%
  \def\@flf@{dynamiccontents}%
  \xdynamiccontents{#1}}{%
  \endxdynamiccontents
}
```

Token to store contents of environment:

```
\newtoks\@dynamictok
```

Start of the environment (unstarred):

```
\def\xdynamiccontents#1{%
  \def\@flf@idn{#1}%
  \@dynamictok{}\@flf@get@body
}
```

Get the body of the environment:

```
\long\def\@flf@get@body#1\end{%
  \@flf@checkcontinued#1\continueonframe\@nil
  \ifdfcontinued
     \expandafter\flf@ta\expandafter{\@flf@tmpa}%
     \edef\@flf@tmp{\the\@dynamictok\the\flf@ta}%
     \@dynamictok\expandafter{\@flf@tmp}%
  \else
     \@dynamictok\expandafter{\the\@dynamictok#1}%
  \fi
  \@flf@find@end
}
```

Check if `\continueonframe` has been used.

```
\newif\ifdfcontinued
\long\def\@flf@checkcontinued#1\continueonframe#2\@nil{%
  \long\def\@flf@tmpa{#1}\long\def\@flf@tmpb{#2}%
  \ifx\@flf@tmpb\@lempty
    \dfcontinuedfalse
  \else
    \dfcontinuedtrue
    \flf@getcontargs#2\@ff@text\@ff@nextid\@ff@rest
  \fi
}
```

Long equivalent of `\@empty`:

```
\long\def\@lempty{}
```

Get the first optional argument and store in the forth argument (which should be a control sequence). Get the second argument and store in the fifth argument (which should be a control sequence). Get the third argument and store in the sixth argument (which should be a control sequence).

```
\def\flf@getcontargs{%
  \@ifnextchar[{\@flf@getcontargs}{\@flf@getcontargs[]}%
}

\long\def\@flf@getcontargs[#1]#2#3\continueonframe#4#5#6{%
  \def#4{#1}\def#5{#2}\def#6{#3}%
}
```

Find the end of the environment:

```
\def\@flf@find@end#1{%
  \def\@tempa{#1}%
  \global\let\flf@next=\relax
  \ifdfcontinued
    \@dynamictok\expandafter
       {\the\@dynamictok\ffcontinuedtextlayout}%
    \protected@edef\@tmpa{\the\@dynamictok{\@ff@text}}%
    \@dynamictok\expandafter{\@tmpa}%
    \toks@\expandafter{\@ff@rest}%
    \edef\flf@next{\noexpand\@flf@get@body\noexpand\end{#1}%
       \noexpand\begin{#1}{\@ff@nextid}\noexpand\par
       \noexpand\noindent\noexpand\ignorespaces
       \the\toks@\noexpand\end{#1}}%
  \else
    \ifx\@tempa\@flf@
      \let\flf@next=\@flf@endxdynamiccontents
    \else
      \@dynamictok\expandafter
        {\the\@dynamictok\end{#1}}%
      \let\flf@next=\@flf@get@body
    \fi
  \fi
  \flf@next
}
```

End of the environment:

```
\let\endxdynamiccontents\relax
\def\@flf@endxdynamiccontents{%
  \ifnum\@flf@idn>\c@maxdynamic
    \PackageError{flowfram}%
    {Dynamic frame \number\@flf@idn\ does not exist}%
    {%
```

```
        You have specified dynamic frame number \number\@flf@idn,
        but there are only \number\c@maxdynamic\space dynamic
        frames currently defined%
      }%
    \else
      \expandafter
      \xdef\csname @dynamicframe@\romannumeral\@flf@idn\endcsname{%
         \the\@dynamictok}%
      \expandafter
    \fi
    \expandafter\end\expandafter{\@flf@}%
  }
```

dynamiccontents* Starred version

```
\newenvironment{dynamiccontents*}[1]{%
  \def\@flf@{dynamiccontents*}%
  \@dynamicframeid{#1}%
  \xdynamiccontents{\ff@id}}{%
  \enddynamiccontents
}
```

\setdynamiccontents

```
\newcommand{\setdynamiccontents}{%
  \@ifstar\@ssetdynamiccontents\@setdynamiccontents
}
```

\@ssetdynamiccontents Starred version: identify dynamic frame by its IDL:

```
\newcommand{\@ssetdynamiccontents}[2]{%
  \@dynamicframeid{#1}\@setdynamiccontents{\ff@id}{#2}%
}
```

\@setdynamiccontents Unstarred version: identify dynamic frame by its IDN:

```
\newcommand{\@setdynamiccontents}[2]{%
  \ifnum#1>\c@maxdynamic
    \PackageError{flowfram}%
    {Dynamic frame \number#1\ does not exist}%
    {%
      You have specified dynamic frame number \number#1, but there are
      only \number\c@maxdynamic\space dynamic frames currently defined%
    }%
  \else
    \expandafter
      \gdef\csname @dynamicframe@\romannumeral#1\endcsname{#2}%
  \fi
}
```

\appenddynamiccontents Append information to dynamic frame. First check to see if starred or unstarred version is being used.

```
\newcommand{\appenddynamiccontents}{%
  \@ifstar\@sappenddynamic\@appenddynamic
}
```

`\@sappenddynamic` Starred version: find the IDN and pass it to the unstarred version.

```
\newcommand{\@sappenddynamic}[2]{%
  \@dynamicframeid{#1}\@appenddynamic{\ff@id}{#2}%
}
```

`\@appenddynamic` Unstarred version.

```
\newcommand{\@appenddynamic}[2]{%
  \ifnum#1>\c@maxdynamic
    \PackageError{flowfram}%
    {Dynamic frame \number#1 does not exist}%
    {%
      You have specified dynamic frame number \number#1,
      but there are only
      \number\c@maxdynamic\space dynamic frames currently defined%
    }%
  \else
    \expandafter\@ff@addtolist
      \csname @dynamicframe@\romannumeral#1\endcsname\entry{#2}%
  \fi
}
```

`\@ff@addtolist` Append #2 onto the end of #1.

```
\newtoks\flf@ta \newtoks\flf@tb
\long\def\@ff@addtolist#1\entry#2{%
 \flf@ta={{#2}}%
 \flf@tb=\expandafter{#1}%
 \xdef#1{\the\flf@tb\the\flf@ta}%
}
```

`\continueonframe` `\continueonframe[`⟨*text*⟩`]{`⟨*id*⟩`}` Ends current staticcontents or dynamiccontents environment and starts environment of the same type for frame given by ⟨*id*⟩. Can only be used inside staticcontents or dynamiccontents environments. If the starred version of the environment is used, `{`⟨*id*⟩`}` refers to the IDL, otherwise it refers to the IDN of the new frame.

```
\newcommand{\continueonframe}{%
  \PackageError{flowfram}%
  {%
    Can't continue to new frame: not in static or dynamic frame%
  }%
  {%
    \string\continueonframe\space may only
    be used inside `staticcontents' or `dynamiccontents'
    environments (or their starred versions)%
  }%
}
```

`\@scontinueonframe` and `\@continueonframe` are set by staticcontents and dynamiccontents environments (and their starred forms).

Static starred version uses IDL

```
\newcommand*{\@staticscontinueonframe}[2][]{%
  \ffcontinuedtextlayout{#1}%
  \end{staticcontents*}%
  \begin{staticcontents*}{#2}\par\noindent\ignorespaces
}
```

Static unstarred version uses IDN

```
\newcommand*{\@staticcontinueonframe}[2][]{%
  \ffcontinuedtextlayout{#1}%
  \end{staticcontents}%
  \begin{staticcontents}{#2}\par\noindent\ignorespaces
}
```

`\ffcontinuedtextlayout` Displays the continued text used by `\continueonframe`.

```
\newcommand{\ffcontinuedtextlayout}[1]{%
  \parfillskip=0pt\par\hfill
  \ffcontinuedtextfont{#1}%
}
```

`\ffcontinuedtextfont` Sets the font to display the continuation text used by `\continueonframe`

```
\newcommand*{\ffcontinuedtextfont}[1]{\emph{\small #1}}
```

1.5 Determining Dimensions and Locations

`\computeleftedgeodd` Compute the position of the left most edge of the page, relative to the left side of the typeblock. Since odd and even pages may have a different offset if `\oddsidemargin` and `\evensidemargin` have different values, it is necessary to have two separate commands for odd and even pages. First the odd pages.

```
\newcommand*{\computeleftedgeodd}[1]{%
  \setlength{#1}{-1in}%
  \addtolength{#1}{-\hoffset}%
  \addtolength{#1}{-\oddsidemargin}%
}
```

`\computeleftedgeeven` Now for the even pages

```
\newcommand*{\computeleftedgeeven}[1]{%
  \setlength{#1}{-1in}%
  \addtolength{#1}{-\hoffset}%
  \addtolength{#1}{-\evensidemargin}%
}
```

`\computetopedge` Compute the top edge of the page, relative to the bottom of the typeblock.

```
\newcommand*{\computetopedge}[1]{%
  \setlength{#1}{\textheight}%
  \addtolength{#1}{\headheight}%
  \addtolength{#1}{\headsep}%
  \addtolength{#1}{1in}%
  \addtolength{#1}{\voffset}%
  \addtolength{#1}{\topmargin}%
}
```

`\computebottomedge` Compute the bottom edge of the page, relative to the bottom of the typeblock.

```
\newcommand*{\computebottomedge}[1]{%
  \computetopedge{#1}%
  \addtolength{#1}{-\paperheight}%
}
```

`\computerightedgeodd` Compute the right edge of the page, relative to the left edge of the typeblock. Again, two commands are needed for odd and even pages. First the odd pages.

```
\newcommand*{\computerightedgeodd}[1]{%
  \computeleftedgeodd{#1}%
  \addtolength{#1}{\paperwidth}%
}
```

`\computerightedgeeven` Now for the even pages.

```
\newcommand*{\computerightedgeeven}[1]{%
  \computeleftedgeeven{#1}%
  \addtolength{#1}{\paperwidth}%
}
```

Compute the minimum area surrounding the listed flow frames. Values stored in `\ffareawidth`, `\ffareaheight`, `\ffareax` and `\ffareay`

```
\newlength\ffareawidth
\newlength\ffareaheight
\newlength\ffareax
\newlength\ffareay
\newlength\ffareaevenx
\newlength\ffareaeveny
```

`\computeflowframearea` Starred version identifies frame by IDL, unstarred version identifies frame by IDN.

```
\newcommand*{\computeflowframearea}{%
  \@ifstar\@scomputeffarea\@computeffarea
}
```

`\@scomputeffarea` Starred version.

```
\newcommand*{\@scomputeffarea}[1]{%
  \setlength{\ffareax}{\paperwidth}%
  \setlength{\ffareay}{\paperheight}%
  \setlength{\@ff@tmp@x}{0pt}%
  \setlength{\@ff@tmp@y}{0pt}%
  \@for\@ff@id:=#1\do{%
    \@flowframeid{\@ff@id}%
```

`\ff@id` is the IDN

```
  \ifnum\ffareax>\flowframex{\ff@id}%
   \setlength{\ffareax}{\flowframex{\ff@id}}%
  \fi
  \ifnum\ffareay>\flowframey{\ff@id}%
   \setlength{\ffareay}{\flowframey{\ff@id}}%
  \fi
  \setlength{\@ff@offset}{\flowframex{\ff@id}}%
  \addtolength{@ff@offset}{\flowframewidth{\ff@id}}%
  \ifnum\@ff@tmp@x<\@ff@offset
   \setlength{\@ff@tmp@x}{\@ff@offset}%
  \fi
  \setlength{\@ff@offset}{\flowframey{\ff@id}}%
  \addtolength{@ff@offset}{\flowframeheight{\ff@id}}%
  \ifnum\@ff@tmp@y<\@ff@offset
   \setlength{\@ff@tmp@y}{\@ff@offset}%
  \fi
 }%
 \setlength{\ffareawidth}{\@ff@tmp@x}%
 \addtolength{\ffareawidth}{-\ffareax}%
 \setlength{\ffareaheight}{\@ff@tmp@y}%
 \addtolength{\ffareaheight}{-\ffareay}%
}
```

`\@computeffarea` Unstarred version.

```
\newcommand*{\@computeffarea}[1]{%
 \setlength{\ffareax}{\paperwidth}%
 \setlength{\ffareay}{\paperheight}%
 \setlength{\@ff@tmp@x}{0pt}%
 \setlength{\@ff@tmp@y}{0pt}%
 \@for\@ff@id:=#1\do{%
  \ff@id=\@ff@id\relax
  \setlength{\@ff@offset}{\flowframex{\ff@id}}%
  \ifdim\ffareax>\@ff@offset
   \setlength{\ffareax}{\@ff@offset}%
  \fi
  \setlength{\@ff@offset}{\flowframey{\ff@id}}%
  \ifdim\ffareay>\@ff@offset
   \setlength{\ffareay}{\@ff@offset}%
  \fi
  \setlength{\@ff@offset}{\flowframex{\ff@id}}%
  \addtolength{\@ff@offset}{\flowframewidth{\ff@id}}%
  \ifdim\@ff@tmp@x<\@ff@offset
   \setlength{\@ff@tmp@x}{\@ff@offset}%
  \fi
  \setlength{\@ff@offset}{\flowframey{\ff@id}}%
  \addtolength{\@ff@offset}{\flowframeheight{\ff@id}}%
  \ifdim\@ff@tmp@y<\@ff@offset
   \setlength{\@ff@tmp@y}{\@ff@offset}%
```

```
      \fi
    }%
    \setlength{\ffareawidth}{\@ff@tmp@x}%
    \addtolength{\ffareawidth}{-\ffareax}%
    \setlength{\ffareaheight}{\@ff@tmp@y}%
    \addtolength{\ffareaheight}{-\ffareay}%
  }
```

`\@ff@swaplen` Swap the values of two lengths

```
  \newcommand*{\@ff@swaplen}[2]{%
    \setlength{\@ff@tmp@x}{#1}%
    \setlength{#1}{#2}%
    \setlength{#2}{\@ff@tmp@x}%
  }
```

`\@ff@getdim` Get the dimensions for the given type of frame. The first parameter should be a number indictating type of frame : 1 (flow), 2 (static), 3 (dynamic). The second number is its IDN. Values are stored in `\ffareax`, `\ffareay`, `\ffareawidth` and `\ffareaheight`.

```
  \newcommand*{\@ff@getdim}[2]{%
    \ifnum#2<1\relax
      \PackageError{flowfram}%
      {Frame IDNs start from 1}%
      {%
        You have specified a frame IDN of '\number#2'%
      }%
    \fi
    \ifcase#1\relax
      \PackageError{flowfram}%
      {Unknown frame ID type '#1'}%
      {%
        Frame ID types are: 1 (flow), 2 (static) and 3 (dynamic)%
      }%
    \or
```

Flow frame

```
      \ifnum#2>\c@maxflow\relax
        \PackageError{flowfram}{Invalid flow frame IDN '\number#2'}{%
        Flow frame IDNs go from 1 to \number\c@maxflow}%
      \else
        \setlength{\ffareax}{\flowframex{#2}}%
        \setlength{\ffareay}{\flowframey{#2}}%
        \setlength{\ffareaevenx}{\flowframeevenx{#2}}%
        \setlength{\ffareaeveny}{\flowframeeveny{#2}}%
        \setlength{\ffareawidth}{\flowframewidth{#2}}%
        \setlength{\ffareaheight}{\flowframeheight{#2}}%
      \fi
    \or
```

Static frame

```
    \ifnum#2>\c@maxstatic\relax
      \PackageError{flowfram}%
      {Invalid static frame IDN '\number#2'}%
      {%
        Static frame IDNs go from 1 to \number\c@maxstatic
      }%
    \else
      \setlength{\ffareax}{\staticframex{#2}}%
      \setlength{\ffareay}{\staticframey{#2}}%
      \setlength{\ffareaevenx}{\staticframeevenx{#2}}%
      \setlength{\ffareaeveny}{\staticframeeveny{#2}}%
      \expandafter\expandafter\expandafter
        \@ff@getstaticpos
        \csname @sf@dim@\romannumeral#2\endcsname
      \setlength{\ffareawidth}{\@ff@tmp@x}%
      \setlength{\ffareaheight}{\@ff@tmp@y}%
    \fi
  \or
```

Dynamic frame

```
    \ifnum#2>\c@maxdynamic\relax
      \PackageError{flowfram}%
      {Invalid dynamic frame IDN '\number#2'}%
      {%
        Dynamic frame IDNs go from 1 to \number\c@maxdynamic
      }%
    \else
      \setlength{\ffareax}{\dynamicframex{#2}}%
      \setlength{\ffareay}{\dynamicframey{#2}}%
      \setlength{\ffareaevenx}{\dynamicframeevenx{#2}}%
      \setlength{\ffareaeveny}{\dynamicframeeveny{#2}}%
      \expandafter\expandafter\expandafter
        \@ff@getstaticpos
        \csname @df@dim@\romannumeral#2\endcsname
      \setlength{\ffareawidth}{\@ff@tmp@x}%
      \setlength{\ffareaheight}{\@ff@tmp@y}%
     \fi
  \else
    \PackageError{flowfram}%
    {Unknown frame ID type '#1'}%
    {%
      Frame ID types are: 1 (flow), 2 (static) and 3 (dynamic)%
    }%
  \fi
}
```

`\@ff@getevendim` Get the dimensions for the given type of frame on even pages. The first parameter should be a number indictating type of frame : 1 (flow), 2 (static), 3 (dynamic). The second number is its IDN. Values are stored in `\ffareax`,

`\ffareay`, `\ffareawidth` and `\ffareaheight`.

```
\newcommand*{\@ff@getevendim}[2]{%
  \ifnum#2<1\relax
    \PackageError{flowfram}%
    {Frame IDNs start from 1}%
    {%
      You have specified a frame IDN of '\number#2'%
    }%
  \fi
  \ifcase#1\relax
    \PackageError{flowfram}%
    {Unknown frame ID type '#1'}%
    {%
      Frame ID types are: 1 (flow), 2 (static) and 3 (dynamic)%
    }
  \or
```

Flow frame

```
    \ifnum#2>\c@maxflow
      \PackageError{flowfram}%
      {Invalid flow frame IDN '\number#2'}%
      {%
        Flow frame IDNs go from 1 to \number\c@maxflow
      }%
    \else
      \setlength{\ffareax}{\flowframeevenx{#2}}%
      \setlength{\ffareay}{\flowframeeveny{#2}}%
      \setlength{\ffareawidth}{\flowframewidth{#2}}%
      \setlength{\ffareaheight}{\flowframeheight{#2}}%
    \fi
  \or
```

Static frame

```
    \ifnum#2>\c@maxstatic\relax
      \PackageError{flowfram}%
      {Invalid static frame IDN '\number#2'}%
      {%
        Static frame IDNs go from 1 to \number\c@maxstatic
      }%
    \else
      \setlength{\ffareax}{\staticframeevenx{#2}}%
      \setlength{\ffareay}{\staticframeeveny{#2}}%
      \expandafter\expandafter\expandafter
        \@ff@getstaticpos
        \csname @sf@dim@\romannumeral#2\endcsname
      \setlength{\ffareawidth}{\@ff@tmp@x}%
      \setlength{\ffareaheight}{\@ff@tmp@y}%
    \fi
  \or
```

Dynamic frame

```
      \ifnum#2>\c@maxdynamic\relax
        \PackageError{flowfram}%
        {Invalid dynamic frame IDN '\number#2'}%
        {%
          Dynamic frame IDNs go from 1 to \number\c@maxdynamic
        }%
      \else
        \setlength{\ffareax}{\dynamicframeevenx{#2}}%
        \setlength{\ffareay}{\dynamicframeeveny{#2}}%
        \expandafter\expandafter\expandafter
          \@ff@getstaticpos
          \csname @df@dim@\romannumeral#2\endcsname
        \setlength{\ffareawidth}{\@ff@tmp@x}%
        \setlength{\ffareaheight}{\@ff@tmp@y}%
      \fi
    \else
      \PackageError{flowfram}%
      {Unknown frame ID type '#1'}%
      {%
        Frame ID types are: 1 (flow), 2 (static) and 3 (dynamic)%
      }%
    \fi
  }
```

\getstaticbounds Convenience method for calling the above. Firstly for static frames:

```
\newcommand*{\getstaticbounds}{%
  \@ifstar\@sgetstaticbounds\@getstaticbounds
}
```

\@sgetstaticbounds Starred version (specify by IDL):

```
\newcommand*{\@sgetstaticbounds}[1]{%
  \@staticframeid{#1}\@getstaticbounds{\ff@id}%
}
```

\@getstaticbounds Unstarred version (specify by IDN):

```
\newcommand*{\@getstaticbounds}[1]{\@ff@getdim{2}{#1}}
```

\getstaticevenbounds Even pages

```
\newcommand*{\getstaticevenbounds}{%
  \@ifstar\@sgetstaticevenbounds\@getstaticevenbounds
}
```

\@sgetstaticevenbounds Starred version (specify by IDL):

```
\newcommand*{\@sgetstaticevenbounds}[1]{%
  \@staticframeid{#1}\@getstaticevenbounds{\ff@id}%
}
```

`\@getstaticevenbounds` Unstarred version (specify by IDN):

```
\newcommand*{\@getstaticevenbounds}[1]{\@ff@getevendim{2}{#1}}
```

`\getflowbounds` Next flow frames:

```
\newcommand*{\getflowbounds}{%
  \@ifstar\@sgetflowbounds\@getflowbounds
}
```

`\@sgetflowbounds` Starred version (specify by IDL):

```
\newcommand*{\@sgetflowbounds}[1]{%
  \@flowframeid{#1}\@getflowbounds{\ff@id}%
}
```

`\@getflowbounds` Unstarred version (specify by IDN):

```
\newcommand*{\@getflowbounds}[1]{\@ff@getdim{1}{#1}}
```

`\getflowevenbounds` Even pages:

```
\newcommand*{\getflowevenbounds}{%
  \@ifstar\@sgetflowevenbounds\@getflowevenbounds
}
```

`\@sgetflowevenbounds` Starred version (specify by IDL):

```
\newcommand*{\@sgetflowevenbounds}[1]{%
  \@flowframeid{#1}\@getflowevenbounds{\ff@id}%
}
```

`\@getflowevenbounds` Unstarred version (specify by IDN):

```
\newcommand*{\@getflowevenbounds}[1]{\@ff@getevendim{1}{#1}}
```

`\getdynamicbounds` Next dynamic frames:

```
\newcommand*{\getdynamicbounds}{%
  \@ifstar\@sgetdynamicbounds\@getdynamicbounds
}
```

`\@sgetdynamicbounds` Starred version (specify by IDL):

```
\newcommand*{\@sgetdynamicbounds}[1]{%
  \@dynamicframeid{#1}\@getdynamicbounds{\ff@id}%
}
```

`\@getdynamicbounds` Unstarred version (specify by IDN):

```
\newcommand*{\@getdynamicbounds}[1]{\@ff@getdim{3}{#1}}
```

`\getdynamicevenbounds` Even pages:

```
\newcommand*{\getdynamicevenbounds}{%
  \@ifstar\@sgetdynamicevenbounds\@getdynamicevenbounds
}
```

\@sgetdynamicevenbounds

Starred version (specify by IDL):

```
\newcommand*{\@sgetdynamicevenbounds}[1]{%
  \@dynamicframeid{#1}\@getdynamicevenbounds{\ff@id}%
}
```

\@getdynamicevenbounds

Unstarred version (specify by IDN):

```
\newcommand*{\@getdynamicevenbounds}[1]{\@ff@getevendim{3}{#1}}
```

1.6 Determining the relative location of one frame from another

The commands in this section set the following boolean variables:

```
\newif\ifFLFabove
\newif\ifFLFbelow
\newif\ifFLFleft
\newif\ifFLFright
```

These can then be used after one of the `\checkifframe`⟨*loc*⟩ commands defined below. For example:

```
\checkifframeabove{static}{1}{flow}{1}
\ifFLFabove
  Static frame is above flow frame.
\else
  Static frame isn't above flow frame.
\fi
```

\checkifframeabove

`\checkifframeabove{`⟨*type1*⟩`}{`⟨*id1*⟩`}{`⟨*type2*⟩`}{`⟨*id2*⟩`}`

Checks if the first frame is above the second frame where the first frame is of type ⟨*type1*⟩ with IDN given by ⟨*id1*⟩ and the second frame is of type ⟨*type2*⟩ with IDN given by ⟨*id2*⟩. The starred version uses the IDL instead of the IDN. The first frame is not considered to be above the second frame if they overlap. This code checks the page number to determine whether to use `\oddcheckifframeabove` or `\evencheckifframeabove` so it should not be used in the first paragraph of the first flow frame on the page if the paragraph spans the page break.

```
\newcommand*{\checkifframeabove}{%
  \@ifstar\@scheckifframeabove\@checkifframeabove
}
```

Starred version:

```
\newcommand*{\@scheckifframeabove}[4]{%
  \ifodd\c@page
    \@soddcheckifframeabove{#1}{#2}{#3}{#4}%
  \else
    \@sevencheckifframeabove{#1}{#2}{#3}{#4}%
  \fi
}
```

Unstarred version:

```
\newcommand*{\@checkifframeabove}[4]{%
  \ifodd\c@page
    \@oddcheckifframeabove{#1}{#2}{#3}{#4}%
  \else
    \@evencheckifframeabove{#1}{#2}{#3}{#4}%
  \fi
}
```

`\oddcheckifframeabove`

`\oddcheckifframeabove{`⟨*type1*⟩`}{`⟨*id1*⟩`}{`⟨*type2*⟩`}{`⟨*id2*⟩`}` Checks if the first frame is above the second frame where the first frame is of type ⟨*type1*⟩ with IDN given by ⟨*id1*⟩ and the second frame is of type ⟨*type2*⟩ with IDN given by ⟨*id2*⟩ for odd pages. The starred version uses the IDL instead of the IDN. The first frame is not considered to be above the second frame if they overlap.

```
\newcommand*{\oddcheckifframeabove}{%
  \@ifstar\@soddcheckifframeabove\@oddcheckifframeabove
}
```

The starred version

```
\newcommand*{\@soddcheckifframeabove}[4]{%
  \@ifundefined{@sget#1bounds}%
  {%
    \PackageError{flowfram}%
    {Unknown frame type ‘#1’}%
    {%
      Frame types may only be one of: static, dynamic or flow%
    }%
  }%
  {}%
  \csname @sget#1bounds\endcsname{#2}%
  \edef\@ff@check{\the\ffareay}%
  \@ifundefined{@sget#3bounds}%
  {%
    \PackageError{flowfram}%
    {Unknown frame type ‘#3’}%
    {%
      Frame types may only be one of: static, dynamic or flow%
    }%
  }%
  {}%
  \csname @sget#3bounds\endcsname{#4}%
  \advance\ffareay by \ffareaheight\relax
  \expandafter\ifdim\@ff@check>\ffareay
    \FLFabovetrue
  \else
    \FLFabovefalse
  \fi
}
```

The unstarred version

```
\newcommand*{\@oddcheckifframeabove}[4]{%
  \@ifundefined{@get#1bounds}%
  {%
    \PackageError{flowfram}%
    {Unknown frame type ‘#1’}%
    {%
      Frame types may only be one of: static, dynamic or
      flow%
    }%
  }%
  {}%
  \csname @get#1bounds\endcsname{#2}%
  \edef\@ff@check{\the\ffareay}%
  \@ifundefined{@get#3bounds}%
  {%
    \PackageError{flowfram}%
    {Unknown frame type ‘#3’}%
    {%
      Frame types may only be one of: static, dynamic or
      flow%
    }%
  }%
  {}%
  \csname @get#3bounds\endcsname{#4}%
  \advance\ffareay by \ffareaheight\relax
  \expandafter\ifdim\@ff@check>\ffareay
    \FLFabovetrue
  \else
    \FLFabovefalse
  \fi
}
```

`\checkifframebelow` `\checkifframebelow{`⟨*type1*⟩`}{`⟨*id1*⟩`}{`⟨*type2*⟩`}{`⟨*id2*⟩`}` Checks if the first frame is below the second frame where the first frame is of type ⟨*type1*⟩ with IDN given by ⟨*id1*⟩ and the second frame is of type ⟨*type2*⟩ with IDN given by ⟨*id2*⟩. The starred version uses the IDL instead of the IDN. The first frame is not considered to be below the second frame if they overlap. This code checks the page number to determine whether to use `\oddcheckifframebelow` or `\evencheckifframebelow` so it should not be used in the first paragraph of the first flow frame on the page if the paragraph spans the page break.

```
\newcommand*{\checkifframebelow}{%
  \@ifstar\@scheckifframebelow\@checkifframebelow
}
```

Starred version:

```
\newcommand*{\@scheckifframebelow}[4]{%
  \ifodd\c@page
    \@soddcheckifframebelow{#1}{#2}{#3}{#4}%
  \else
```

```
      \@sevencheckifframebelow{#1}{#2}{#3}{#4}%
    \fi
  }
```

Unstarred version:

```
  \newcommand*{\@checkifframebelow}[4]{%
    \ifodd\c@page
      \@oddcheckifframebelow{#1}{#2}{#3}{#4}%
    \else
      \@evencheckifframebelow{#1}{#2}{#3}{#4}%
    \fi
  }
```

`\oddcheckifframebelow` `\oddcheckifframebelow{`⟨*type1*⟩`}{`⟨*id1*⟩`}{`⟨*type2*⟩`}{`⟨*id2*⟩`}`

Checks if the first frame is below the second frame where the first frame is of type ⟨*type1*⟩ with IDN given by ⟨*id1*⟩ and the second frame is of type ⟨*type2*⟩ with IDN given by ⟨*id2*⟩ on odd pages. The starred version uses the IDL instead of the IDN. The first frame is not considered to be below the second frame if they overlap.

```
  \newcommand*{\oddcheckifframebelow}{%
    \@ifstar\@soddcheckifframebelow\@oddcheckifframebelow
  }
```

The starred version

```
  \newcommand*{\@soddcheckifframebelow}[4]{%
    \@ifundefined{@sget#1bounds}%
    {%
      \PackageError{flowfram}%
      {Unknown frame type ‘#1’}%
      {%
        Frame types may only be one of: static, dynamic or
        flow%
      }%
    }%
    {}%
    \csname @sget#1bounds\endcsname{#2}%
    \advance\ffareay by \ffareaheight\relax
    \edef\@ff@check{\the\ffareay}%
    \@ifundefined{@sget#3bounds}%
    {%
      \PackageError{flowfram}%
      {Unknown frame type ‘#3’}%
      {%
        Frame types may only be one of: static, dynamic or
        flow%
      }%
    }%
    {}%
    \csname @sget#3bounds\endcsname{#4}%
    \expandafter\ifdim\@ff@check<\ffareay
      \FLFbelowtrue
    \else
      \FLFbelowfalse
    \fi
  }
```

The unstarred version

```
\newcommand*{\@oddcheckifframebelow}[4]{%
  \@ifundefined{@get#1bounds}%
  {%
    \PackageError{flowfram}%
    {Unknown frame type ‘#1’}%
    {%
      Frame types may only be one of: static, dynamic or
      flow%
    }%
  }%
  {}%
  \csname @get#1bounds\endcsname{#2}%
  \advance\ffareay by \ffareaheight\relax
  \edef\@ff@check{\the\ffareay}%
  \@ifundefined{@get#3bounds}%
  {%
    \PackageError{flowfram}%
    {Unknown frame type ‘#3’}%
    {%
      Frame types may only be one of: static, dynamic or
      flow%
    }%
  }%
  {}%
  \csname @get#3bounds\endcsname{#4}%
  \expandafter\ifdim\@ff@check<\ffareay
    \FLFbelowtrue
  \else
    \FLFbelowfalse
  \fi
}
```

`\checkifframeleft`

`\checkifframeleft{`⟨*type1*⟩`}{`⟨*id1*⟩`}{`⟨*type2*⟩`}{`⟨*id2*⟩`}` Checks if the first frame is to the left of the second frame where the first frame is of type ⟨*type1*⟩ with IDN given by ⟨*id1*⟩ and the second frame is of type ⟨*type2*⟩ with IDN given by ⟨*id2*⟩. The starred version uses the IDL instead of the IDN. The first frame is not considered to be to the left of the second frame if they overlap. This code checks the page number to determine whether to use `\oddcheckifframeleft` or `\evencheckifframeleft` so it should not be used in the first paragraph of the first flow frame on the page if the paragraph spans the page break.

```
\newcommand*{\checkifframeleft}{%
```

```
  \@ifstar\@scheckifframeleft\@checkifframeleft
}
```

Starred version:

```
\newcommand*{\@scheckifframeleft}[4]{%
  \ifodd\c@page
    \@soddcheckifframeleft{#1}{#2}{#3}{#4}%
  \else
    \@sevencheckifframeleft{#1}{#2}{#3}{#4}%
  \fi
}
```

Unstarred version:

```
\newcommand*{\@checkifframeleft}[4]{%
  \ifodd\c@page
    \@oddcheckifframeleft{#1}{#2}{#3}{#4}%
  \else
    \@evencheckifframeleft{#1}{#2}{#3}{#4}%
  \fi
}
```

`\oddcheckifframeleft` `\oddcheckifframeleft{`⟨*type1*⟩`}{`⟨*id1*⟩`}{`⟨*type2*⟩`}{`⟨*id2*⟩`}`

Checks if the first frame is to the left of the second frame where the first frame is of type ⟨*type1*⟩ with IDN given by ⟨*id1*⟩ and the second frame is of type ⟨*type2*⟩ with IDN given by ⟨*id2*⟩ on odd pages. The starred version uses the IDL instead of the IDN. The first frame is not considered to be to the left of the second frame if they overlap.

```
\newcommand*{\oddcheckifframeleft}{%
  \@ifstar\@soddcheckifframeleft\@oddcheckifframeleft
}
```

The starred version

```
\newcommand*{\@soddcheckifframeleft}[4]{%
  \@ifundefined{@sget#1bounds}%
  {%
    \PackageError{flowfram}%
    {Unknown frame type ‘#1’}%
    {%
      Frame types may only be one of: static, dynamic or
      flow%
    }%
  }%
  {}%
  \csname @sget#1bounds\endcsname{#2}%
  \advance\ffareax by \ffareawidth\relax
  \edef\@ff@check{\the\ffareax}%
  \@ifundefined{@sget#3bounds}%
  {%
    \PackageError{flowfram}%
    {Unknown frame type ‘#3’}%
    {%
      Frame types may only be one of: static, dynamic or
      flow%
    }%
  }%
  {}%
  \csname @sget#3bounds\endcsname{#4}%
  \expandafter\ifdim\@ff@check<\ffareax
    \FLFlefttrue
  \else
    \FLFleftfalse
  \fi
}
```

The unstarred version

```
\newcommand*{\@oddcheckifframeleft}[4]{%
  \@ifundefined{@get#1bounds}%
  {%
    \PackageError{flowfram}%
    {Unknown frame type '#1'}%
    {%
      Frame types may only be one of: static, dynamic or
      flow%
    }%
  }%
  {}%
  \csname @get#1bounds\endcsname{#2}%
  \advance\ffareax by \ffareawidth\relax
  \edef\@ff@check{\the\ffareax}%
  \@ifundefined{@get#3bounds}%
  {%
    \PackageError{flowfram}%
    {Unknown frame type '#3'}%
    {%
      Frame types may only be one of: static, dynamic or
      flow%
    }%
  }%
  {}%
  \csname @get#3bounds\endcsname{#4}%
  \expandafter\ifdim\@ff@check<\ffareax
    \FLFlefttrue
  \else
    \FLFleftfalse
  \fi
}
```

`\checkifframeright` `\checkifframeright{⟨type1⟩}{⟨id1⟩}{⟨type2⟩}{⟨id2⟩}` Checks if the first frame is to the right of the second frame where the first frame is of type ⟨*type1*⟩ with IDN given by ⟨*id1*⟩ and the second frame is of type ⟨*type2*⟩ with IDN given by ⟨*id2*⟩. The starred version uses the IDL instead of the IDN. The first frame is not considered to be to the right of the second frame if they overlap. This code checks the page number to determine whether to use `\oddcheckifframeright` or `\evencheckifframeright` so it should not be used in the first paragraph of the first flow frame on the page if the paragraph spans the page break.

```
\newcommand*{\checkifframeright}{%
  \@ifstar\@scheckifframeright\@checkifframeright
}
```

Starred version:

```
\newcommand*{\@scheckifframeright}[4]{%
  \ifodd\c@page
    \@soddcheckifframeright{#1}{#2}{#3}{#4}%
  \else
    \@sevencheckifframeright{#1}{#2}{#3}{#4}%
  \fi
}
```

Unstarred version:

```
\newcommand*{\@checkifframeright}[4]{%
  \ifodd\c@page
    \@oddcheckifframeright{#1}{#2}{#3}{#4}%
  \else
    \@evencheckifframeright{#1}{#2}{#3}{#4}%
  \fi
}
```

`\oddcheckifframeright`

`\oddcheckifframeright{`⟨*type1*⟩`}{`⟨*id1*⟩`}{`⟨*type2*⟩`}{`⟨*id2*⟩`}`

Checks if the first frame is to the right of the second frame where the first frame is of type ⟨*type1*⟩ with IDN given by ⟨*id1*⟩ and the second frame is of type ⟨*type2*⟩ with IDN given by ⟨*id2*⟩ on odd pages. The starred version uses the IDL instead of the IDN. The first frame is not considered to be to the right of the second frame if they overlap.

```
\newcommand*{\oddcheckifframeright}{%
  \@ifstar\@soddcheckifframeright\@oddcheckifframeright
}
```

The starred version

```
\newcommand*{\@soddcheckifframeright}[4]{%
  \@ifundefined{@sget#1bounds}%
  {%
    \PackageError{flowfram}%
    {Unknown frame type ‘#1’}%
    {%
      Frame types may only be one of: static, dynamic or
      flow%
    }%
  }%
  {}%
  \csname @sget#1bounds\endcsname{#2}%
  \edef\@ff@check{\the\ffareax}%
  \@ifundefined{@sget#3bounds}%
  {%
    \PackageError{flowfram}%
    {Unknown frame type ‘#3’}%
    {%
      Frame types may only be one of: static, dynamic or
     flow%
    }%
  }%
  {}%
  \csname @sget#3bounds\endcsname{#4}%
  \advance\ffareax by \ffareawidth\relax
  \expandafter\ifdim\@ff@check>\ffareax
    \FLFrighttrue
  \else
    \FLFrightfalse
  \fi
}
```

The unstarred version

```
\newcommand*{\@oddcheckifframeright}[4]{%
  \@ifundefined{@get#1bounds}%
  {%
    \PackageError{flowfram}%
    {Unknown frame type ‘#1’}%
    {%
      Frame types may only be one of: static, dynamic or
      flow%
    }%
  }%
  {}%
  \csname @get#1bounds\endcsname{#2}%
  \edef\@ff@check{\the\ffareax}%
  \@ifundefined{@get#3bounds}%
  {%
    \PackageError{flowfram}%
    {Unknown frame type ‘#3’}%
    {%
      Frame types may only be one of: static, dynamic or
      flow%
    }%
  }%
  {}%
  \csname @get#3bounds\endcsname{#4}%
  \advance\ffareax by \ffareawidth\relax
  \expandafter\ifdim\@ff@check>\ffareax
    \FLFrighttrue
  \else
    \FLFrightfalse
  \fi
}
```

`\evencheckifframeabove` `\evencheckifframeabove{⟨type1⟩}{⟨id1⟩}{⟨type2⟩}{⟨id2⟩}` Checks if the first frame is above the second frame where the first frame is of type ⟨*type1*⟩ with IDN given by ⟨*id1*⟩ and the second frame is of type ⟨*type2*⟩ with IDN given by ⟨*id2*⟩ for even pages. The starred version uses the IDL instead of the IDN. The first frame is not considered to be above the second frame if they overlap.

```
\newcommand*{\evencheckifframeabove}{%
  \@ifstar\@sevencheckifframeabove\@evencheckifframeabove
}
```

The starred version

```
\newcommand*{\@sevencheckifframeabove}[4]{%
  \@ifundefined{@sget#1evenbounds}%
  {%
    \PackageError{flowfram}%
    {Unknown frame type ‘#1’}%
    {%
      Frame types may only be one of: static, dynamic or
      flow%
    }%
  }%
  {}%
  \csname @sget#1evenbounds\endcsname{#2}%
  \edef\@ff@check{\the\ffareay}%
  \@ifundefined{@sget#3evenbounds}%
  {%
    \PackageError{flowfram}%
    {Unknown frame type ‘#3’}%
    {%
      Frame types may only be one of: static, dynamic or
      flow%
    }%
  }%
  {}%
  \csname @sget#3evenbounds\endcsname{#4}%
  \advance\ffareay by \ffareaheight\relax
  \expandafter\ifdim\@ff@check>\ffareay
    \FLFabovetrue
  \else
    \FLFabovefalse
  \fi
}
```

The unstarred version

```
\newcommand*{\@evencheckifframeabove}[4]{%
  \@ifundefined{@get#1evenbounds}%
```

```
    {%
      \PackageError{flowfram}%
      {Unknown frame type `#1'}%
      {%
        Frame types may only be one of: static, dynamic or
        flow%
      }%
    }%
    {}%
    \csname @get#1evenbounds\endcsname{#2}%
    \edef\@ff@check{\the\ffareay}%
    \@ifundefined{@get#3evenbounds}%
    {%
      \PackageError{flowfram}%
      {Unknown frame type `#3'}%
      {%
        Frame types may only be one of: static, dynamic or
        flow%
      }%
    }%
    {}%
    \csname @get#3evenbounds\endcsname{#4}%
    \advance\ffareay by \ffareaheight\relax
    \expandafter\ifdim\@ff@check>\ffareay
      \FLFabovetrue
    \else
      \FLFabovefalse
    \fi
  }
```

`\evencheckifframebelow`

`\checkifframebelow{`⟨*type1*⟩`}{`⟨*id1*⟩`}{`⟨*type2*⟩`}{`⟨*id2*⟩`}` Checks if the first frame is below the second frame where the first frame is of type ⟨*type1*⟩ with IDN given by ⟨*id1*⟩ and the second frame is of type ⟨*type2*⟩ with IDN given by ⟨*id2*⟩. The starred version uses the IDL instead of the IDN. The first frame is not considered to be below the second frame if they overlap.

```
  \newcommand*{\evencheckifframebelow}{%
    \@ifstar\@sevencheckifframebelow\@evencheckifframebelow
  }
```

The starred version

```
  \newcommand*{\@sevencheckifframebelow}[4]{%
    \@ifundefined{@sget#1evenbounds}%
    {%
      \PackageError{flowfram}%
      {Unknown frame type `#1'}%
      {%
        Frame types may only be one of: static, dynamic or
        flow%
      }%
    }%
    {}%
    \csname @sget#1evenbounds\endcsname{#2}%
    \advance\ffareay by \ffareaheight\relax
    \edef\@ff@check{\the\ffareay}%
    \@ifundefined{@sget#3evenbounds}%
    {%
      \PackageError{flowfram}%
      {Unknown frame type ‘#3’}%
      {%
        Frame types may only be one of: static, dynamic or
        flow%
      }%
    }%
    {}%
    \csname @sget#3evenbounds\endcsname{#4}%
    \expandafter\ifdim\@ff@check<\ffareay
      \FLFbelowtrue
    \else
      \FLFbelowfalse
    \fi
  }
```

The unstarred version

```
  \newcommand*{\@evencheckifframebelow}[4]{%
    \@ifundefined{@get#1evenbounds}%
    {%
      \PackageError{flowfram}%
      {Unknown frame type ‘#1’}%
      {%
        Frame types may only be one of: static, dynamic or
        flow%
      }%
    }{}%
    \csname @get#1evenbounds\endcsname{#2}%
    \advance\ffareay by \ffareaheight\relax
    \edef\@ff@check{\the\ffareay}%
    \@ifundefined{@get#3evenbounds}%
    {%
      \PackageError{flowfram}%
      {Unknown frame type ‘#3’}%
      {%
        Frame types may only be one of: static, dynamic or
        flow%
      }%
    }%
    {}%
    \csname @get#3evenbounds\endcsname{#4}%
    \expandafter\ifdim\@ff@check<\ffareay
      \FLFbelowtrue
    \else
      \FLFbelowfalse
    \fi
  }
```

`\evencheckifframeleft`

`\evencheckifframeleft{`⟨*type1*⟩`}{`⟨*id1*⟩`}{`⟨*type2*⟩`}{`⟨*id2*⟩`}` Checks if the first frame is to the left of the second frame where the first frame is of type ⟨*type1*⟩ with IDN given by ⟨*id1*⟩ and the second frame is of type ⟨*type2*⟩ with IDN given by ⟨*id2*⟩. The starred version uses the IDL instead of the IDN. The first frame is not considered to be to the left of the second frame if they overlap.

```
  \newcommand*{\evencheckifframeleft}{%
    \@ifstar\@sevencheckifframeleft\@evencheckifframeleft
  }
```

The starred version

```
  \newcommand*{\@sevencheckifframeleft}[4]{%
    \@ifundefined{@sget#1evenbounds}%
    {%
      \PackageError{flowfram}%
      {Unknown frame type ‘#1’}%
      {%
        Frame types may only be one of: static, dynamic or
        flow%
      }%
    }%
    {}%
    \csname @sget#1evenbounds\endcsname{#2}%
    \advance\ffareax by \ffareawidth\relax
    \edef\@ff@check{\the\ffareax}%
    \@ifundefined{@sget#3evenbounds}%
    {%
      \PackageError{flowfram}%
      {Unknown frame type ‘#3’}%
      {%
        Frame types may only be one of: static, dynamic or
        flow%
      }%
    }%
    {}%
    \csname @sget#3evenbounds\endcsname{#4}%
    \expandafter\ifdim\@ff@check<\ffareax
      \FLFlefttrue
    \else
      \FLFleftfalse
    \fi
  }
```

The unstarred version

```
  \newcommand*{\@evencheckifframeleft}[4]{%
    \@ifundefined{@get#1evenbounds}%
    {%
      \PackageError{flowfram}%
      {Unknown frame type ‘#1’}%
      {%
        Frame types may only be one of: static, dynamic or
        flow%
      }%
    }%
    {}%
    \csname @get#1evenbounds\endcsname{#2}%
    \advance\ffareax by \ffareawidth\relax
    \edef\@ff@check{\the\ffareax}%
    \@ifundefined{@get#3evenbounds}%
    {%
      \PackageError{flowfram}%
      {Unknown frame type ‘#3’}%
      {%
        Frame types may only be one of: static, dynamic or
        flow%
      }%
    }%
    {}%
    \csname @get#3evenbounds\endcsname{#4}%
    \expandafter\ifdim\@ff@check<\ffareax
      \FLFlefttrue
    \else
      \FLFleftfalse
    \fi
  }
```

`\evencheckifframeright`

`\evencheckifframeright{`⟨*type1*⟩`}{`⟨*id1*⟩`}{`⟨*type2*⟩`}{`⟨*id2*⟩`}` Checks if the first frame is to the right of the second frame where the first frame is of type ⟨*type1*⟩ with IDN given by ⟨*id1*⟩ and the second frame is of type ⟨*type2*⟩ with IDN given by ⟨*id2*⟩. The starred version uses the IDL instead of the IDN. The first frame is not considered to be to the right of the second frame if they overlap.

```
  \newcommand*{\evencheckifframeright}{%
    \@ifstar\@sevencheckifframeright\@evencheckifframeright
  }
```

The starred version

```
  \newcommand*{\@sevencheckifframeright}[4]{%
    \@ifundefined{@sget#1evenbounds}%
    {%
      \PackageError{flowfram}%
      {Unknown frame type ‘#1’}%
      {%
        Frame types may only be one of: static, dynamic or
        flow%
      }%
    }%
    {}%
    \csname @sget#1evenbounds\endcsname{#2}%
    \edef\@ff@check{\the\ffareax}%
    \@ifundefined{@sget#3evenbounds}%
    {%
      \PackageError{flowfram}%
      {Unknown frame type ‘#3’}%
      {%
        Frame types may only be one of: static, dynamic or
        flow%
      }%
    }%
    {}%
    \csname @sget#3evenbounds\endcsname{#4}%
    \advance\ffareax by \ffareawidth\relax
    \expandafter\ifdim\@ff@check>\ffareax
      \FLFrighttrue
    \else
      \FLFrightfalse
    \fi
  }
```

The unstarred version

```
  \newcommand*{\@evencheckifframeright}[4]{%
    \@ifundefined{@get#1evenbounds}%
    {%
      \PackageError{flowfram}%
      {Unknown frame type ‘#1’}%
      {%
        Frame types may only be one of: static, dynamic or
        flow%
      }%
    }%
    {}%
    \csname @get#1evenbounds\endcsname{#2}%
    \edef\@ff@check{\the\ffareax}%
    \@ifundefined{@get#3evenbounds}%
    {%
      \PackageError{flowfram}%
      {Unknown frame type ‘#3’}%
      {%
        Frame types may only be one of: static, dynamic or
        flow%
      }%
    }%
    {}%
    \csname @get#3evenbounds\endcsname{#4}%
    \advance\ffareax by \ffareawidth\relax
    \expandafter\ifdim\@ff@check>\ffareax
      \FLFrighttrue
    \else
      \FLFrightfalse
    \fi
  }
```

Textual labels used to indicate relative location of one frame to another.

\FFaboveleft

```
\newcommand*{\FFaboveleft}{above left}
```

\FFaboveright

```
\newcommand*{\FFaboveright}{above right}
```

\FFbelowleft

```
\newcommand*{\FFbelowleft}{below left}
```

\FFbelowright

```
\newcommand*{\FFbelowright}{below right}
```

\FFleft

```
\newcommand*{\FFleft}{on the left}
```

\FFbelowright

```
\newcommand*{\FFright}{on the right}
```

\FFabove

```
\newcommand*{\FFabove}{above}
```

\FFbelow

```
\newcommand*{\FFbelow}{below}
```

\FFoverlap

```
\newcommand*{\FFoverlap}{overlap}
```

\relativeframelocation

`\relativeframelocation{`⟨*type1*⟩`}{`⟨*id1*⟩`}{`⟨*type2*⟩`}{`⟨*id2*⟩`}` Displays one of the above commands depending on the relative locations of the first frame to the second frame. The arguments ⟨*id1*⟩ and ⟨*id2*⟩ refer to the IDN for the unstarred version and to the IDL for the starred version.

```
\DeclareRobustCommand*{\relativeframelocation}{%
  \@ifstar\@srelativeframelocation\@relativeframelocation
}
```

Starred version:

```
\newcommand*{\@srelativeframelocation}[4]{%
  \@scheckifframeabove{#1}{#2}{#3}{#4}%
  \@scheckifframebelow{#1}{#2}{#3}{#4}%
  \@scheckifframeleft{#1}{#2}{#3}{#4}%
  \@scheckifframeright{#1}{#2}{#3}{#4}%
  \ifFLFabove
    \ifFLFleft
      \FFaboveleft
    \else
      \ifFLFright
        \FFaboveright
      \else
         \FFabove
      \fi
    \fi
  \else
    \ifFLFbelow
      \ifFLFleft
        \FFbelowleft
      \else
        \ifFLFright
          \FFbelowright
        \else
           \FFbelow
        \fi
      \fi
    \else
      \ifFLFleft
        \FFleft
      \else
        \ifFLFright
          \FFright
        \else
           \FFoverlap
        \fi
      \fi
    \fi
  \fi
}
```

Unstarred version:

```
\newcommand*{\@relativeframelocation}[4]{%
  \@checkifframeabove{#1}{#2}{#3}{#4}%
  \@checkifframebelow{#1}{#2}{#3}{#4}%
  \@checkifframeleft{#1}{#2}{#3}{#4}%
  \@checkifframeright{#1}{#2}{#3}{#4}%
  \ifFLFabove
    \ifFLFleft
      \FFaboveleft
    \else
      \ifFLFright
        \FFaboveright
      \else
         \FFabove
      \fi
    \fi
  \else
    \ifFLFbelow
      \ifFLFleft
        \FFbelowleft
      \else
        \ifFLFright
          \FFbelowright
        \else
           \FFbelow
        \fi
      \fi
    \else
      \ifFLFleft
        \FFleft
      \else
        \ifFLFright
          \FFright
        \else
           \FFoverlap
        \fi
      \fi
    \fi
  \fi
}
```

Short cut commands for frames of the same type.

`\reldynamicloc` `\reldynamicloc{⟨id1⟩}{⟨id2⟩}`

```
\DeclareRobustCommand*{\reldynamicloc}{%
  \@ifstar\@sreldynamicloc\@reldynamicloc
}
```

Starred version:

```
\newcommand*{\@sreldynamicloc}[2]{%
  \@srelativeframelocation{dynamic}{#1}{dynamic}{#2}%
}
```

Unstarred version:

```
\newcommand*{\@reldynamicloc}[2]{%
  \@relativeframelocation{dynamic}{#1}{dynamic}{#2}%
}
```

`\relstaticloc` `\relstaticloc{⟨id1⟩}{⟨id2⟩}`

```
\DeclareRobustCommand*{\relstaticloc}{%
  \@ifstar\@srelstaticloc\@relstaticloc
}
```

Starred version:

```
\newcommand*{\@srelstaticloc}[2]{%
```

```
  \@srelativeframelocation{static}{#1}{static}{#2}%
}
```

Unstarred version:

```
\newcommand*{\@relstaticloc}[2]{%
  \@relativeframelocation{static}{#1}{static}{#2}%
}
```

`\relflowloc` \relflowloc{⟨*id1*⟩}{⟨*id2*⟩}

```
\DeclareRobustCommand*{\relflowloc}{%
  \@ifstar\@srelflowloc\@relflowloc
}
```

Starred version:

```
\newcommand*{\@srelflowloc}[2]{%
  \@srelativeframelocation{flow}{#1}{flow}{#2}%
}
```

Unstarred version:

```
\newcommand*{\@relflowloc}[2]{%
  \@relativeframelocation{flow}{#1}{flow}{#2}%
}
```

1.7 Initialise Flow Frames

`\setinitialframe` Specify initial frame. This should be the first flow frame that is defined on the first page of the document. Having another flow frame as the initial frame is not a good idea, and may have unexpected results.

```
\newcommand*{\setinitialframe}[1]{%
  \c@thisframe=#1%
  \global\usedframebreaktrue
  \global\setlength{\hsize}
  {%
    \csname colwidth\romannumeral\c@thisframe\endcsname
  }%
}
```

`\setframes` Set the initial frame.

```
\newif\if@setfr@mes
\@setfr@mesfalse
\newcommand*{\setframes}{%
  \ifnum\c@thisframe=0\relax
    \PackageWarning{flowfram}%
    {Can't find a flow frame on page 1.
     \MessageBreak
     Attempting to find the first page with a flow frame%
    }%
    \@nxtcol=1\relax
    \c@curpg=1\relax
    \@g@tnextcol{\@nxtcol}%
```

Shipout pages without flow frames.

```
      \advance\c@curpg by -1\relax
        \whiledo{\c@curpg>0}%
        {%
          \advance\c@curpg by -1\relax
          \setbox\@outputbox\vbox{\hbox to \textwidth{\@ff@do@allframes}}%
          \@outputpage
        }%
      \c@thisframe=\@nxtcol
    \fi
    \@setcol{\c@thisframe}\relax
    \@setfr@mestrue
    \edef\ff@txtcol{%
    \csname @ff@txtcol@\romannumeral\c@thisframe\endcsname}%
    \@s@tfftextcol
  }
```

`\emulatetwocolumn` Emulate original `\twocolumn` declaration. This is provided for backward compatibility, and may be removed in later versions.

```
  \newcommand{\emulatetwocolumn}[1][]{%
    \finishthispage
    \setallflowframes{pages=none}%
    \settoheight{\@ff@staticH}{#1}%
    \settodepth{\@ff@tmp@y}{#1}%
    \addtolength{\@ff@staticH}{\@ff@tmp@y}%
    \ifdim\@ff@staticH>0pt\relax
      \twocolumnStop[\@ff@pages@countreg]{\@ff@staticH}%
      \c@thisframe=\c@maxflow
      \advance\c@thisframe by -1\relax
      \@twocolumn[>\@ff@pages@countreg]%
        \setstaticcontents{\c@maxstatic}{#1}%
    \else
        \@twocolumn
        \c@thisframe=\c@maxflow
        \advance\c@thisframe by -1\relax
    \fi
    \@setcol{\c@thisframe}%
    \relax
  }
```

`\emulateonecolumn` Emulate original `\onecolumn` declaration. This is provided for backward compatibility, and may be removed in later versions.

```
  \newcommand{\emulateonecolumn}[1][]{%
    \finishthispage
    \setallflowframes{pages=none}%
    \settoheight{\@ff@staticH}{#1}%
    \settodepth{\@ff@tmp@y}{#1}%
    \addtolength{\@ff@staticH}{\@ff@tmp@y}%
    \ifdim\@ff@staticH>0pt\relax
    \onecolumnStop[\@ff@pages@countreg]{\@ff@staticH}%
    \c@thisframe=\c@maxflow
    \advance\c@thisframe by -1\relax
    \@onecolumn[>\@ff@pages@countreg]%
    \setstaticcontents{\c@maxstatic}{#1}%
  \else
    \@twocolumn
    \c@thisframe=\c@maxflow
    \advance\c@thisframe by -1\relax
  \fi
  \@setcol{\c@thisframe}%
  \relax
}
```

If no flow frames have been defined, create one big one the size of the typeblock, and initialise the frames.

```
\AtBeginDocument{%
  \c@absolutepage=1\relax
  \ifnum\c@maxflow=0\relax
    \PackageWarning{flowfram}{No flow frames, adding one}%
    \@onecolumn
  \fi
  \setframes
  \renewcommand{\onecolumn}[1][]{%
    \PackageWarning{flowfram}%
    {%
       Ignoring \string\onecolumn\space found in document environment.
       Frames must be defined in the preamble%
    }%
    #1%
  }%
  \renewcommand{\twocolumn}[1][]{%
    \PackageWarning{flowfram}%
    {%
      Ignoring \string\twocolumn\space found in document environment.
      Frames must be defined in the preamble}%
      #1%
  }%
}
```

1.8 Output Routine

`\fftolerance`

The flowfram package does a check to see if text has flowed between frames of different widths, which will cause a discrepancy in the line widths of the paragraph spanning the break. Before version 1.14, the output routine just checked if the widths were different, but this means that warning messages will be generated even if there's only a tiny difference that can be caused by rounding errors (for example, if the frames were created using jpgfdraw). So add a tolerance and only complain if the difference exceeds this value.

```
\newlength\fftolerance
\setlength\fftolerance{2pt}
```

`\@setcol` Set up the output box so it has the correct dimensions for specified flow frame. This is used by the output routine.

```
\newcommand{\@setcol}[1]{%
  \ifnum\c@maxflow<#1\relax
    \PackageError{flowfram}%
    {Can't set frame '\number#1', doesn't exist}{}%
  \else
    \flf@message{Switching to flow frame \number#1\space on page
      \number\@ff@pages@countreg}%
    \expandafter\global\expandafter\columnwidth
     \csname colwidth\romannumeral#1\endcsname
    \dimen@\columnwidth
    \advance\dimen@ by -\hsize\relax
    \ifdim\dimen@<0pt\relax
      \dimen@=-\dimen@
    \fi
    \ifdim\dimen@>\fftolerance
      \ifusedframebreak
      \else
        \PackageWarning{flowfram}%
        {Moving to flow frame of unequal
        width,\MessageBreak use of \string\framebreak\space advised,
        or text might not appear correctly (difference =
        \the\dimen@, tolerance = \the\fftolerance)}%
      \fi
    \fi
    \global\usedframebreakfalse
    \global\hsize\columnwidth
    \expandafter\global
    \expandafter\vsize\csname colheight\romannumeral#1\endcsname
    \global\@colht\vsize
    \global\@colroom\@colht
```

We may be inside an environment that has modified the line width, such as one of the list environments so we can't just set `\linewidth` to `\columnwidth`. Test if we're in a list environment by checking if `\@listdepth` is greater than 0. If true, only modify `\linewidth` if it's larger than the new column width.

```
    \ifnum\@listdepth>0\relax
      \ifnum\linewidth>\columnwidth
        \global\linewidth\columnwidth
      \fi
    \else
      \global\linewidth\columnwidth
    \fi
    %\global\textwidth\columnwidth
    \setmargin
  \fi
  \stepcounter{displayedframe}%
}
```

Modify the output routine so that it uses `\vsize` instead of `\textheight`.

```
\output={%
  \let\par\@@par
  \ifnum\outputpenalty <-\@M
    \@specialoutput
  \else
    \@makecol
    \@opcol \@startcolumn
    \@whilesw \if@fcolmade \fi {\@opcol \@startcolumn }%
  \fi
  \ifnum\outputpenalty>-\@Miv
    \ifdim\@colroom<1.5\baselineskip
      \ifdim\@colroom<\vsize
        \@latex@warning@no@line{Text page \thepage \space
        contains only floats}%
        \@emptycol
      \else
        \global\vsize\@colroom
      \fi
    \else
      \global\vsize\@colroom
    \fi
  \else
    \global\vsize\maxdimen
  \fi
}
```

`\@doclearpage` Modify `\@doclearpage`, again replace `\textheight` with `\vsize`, and only use the twocolumn stuff.

```
\def\@doclearpage{%
  \ifvoid\footins
    \setbox\@tempboxa\vsplit\@cclv to\z@
    \unvbox\@tempboxa
    \setbox\@tempboxa\box\@cclv
    \xdef\@deferlist{\@toplist\@botlist\@deferlist}%
    \global\let\@toplist\@empty
    \global\let\@botlist\@empty
    \global\@colroom\@colht
    \ifx\@currlist\@empty
    \else
      \@latexerr{Float(s) lost}\@ehb
      \global\let\@currlist\@empty
    \fi
```

```
      \@makefcolumn\@deferlist
      \@whilesw \if@fcolmade \fi
      {%
        \@opcol
        \@makefcolumn\@deferlist
      }%
      \if@firstcolumn
       \xdef\@dbldeferlist{\@dbltoplist\@dbldeferlist}%
       \global\let\@dbltoplist\@empty
       \global\@colht\vsize
       \begingroup
         \@dblfloatplacement
         \@makefcolumn\@dbldeferlist
         \@whilesw \if@fcolmade \fi
         {%
           \@outputpage
           \@makefcolumn\@dbldeferlist
         }%
       \endgroup
      \else
        \vbox{}%
        \clearpage
      \fi
    \else
      \setbox\@cclv\vbox{\box\@cclv\vfil}%
      \@makecol\@opcol
      \clearpage
    \fi
  }
```

Modify `\@outputpage` slightly. Add provision for turning headers and footers into dynamic frames.

`\@dothehead` First define macro to do the header. This will be modified if it is turned into a dynamic frame.

```
\newcommand{\@dothehead}{%
  \vbox to \headheight
  {%
    \color@hbox\normalcolor\hbox to \textwidth{\@thehead}%
    \color@endbox
  }%
}
```

`\@dothefoot` Same again for the footer.

```
\newcommand{\@dothefoot}{%
  \color@hbox\normalcolor\hbox to \textwidth{\@thefoot}%
  \color@endbox
}
\newcommand{\@dodynamicthehead}{}
```

```
\newcommand{\@dodynamicthefoot}{}
```

`\@outputpage` Now for the modified version of `\@outputpage`. The page style stuff has been moved to `\@outputdblcol` so that the headers and footers can be set in dynamic frames before the dynamic frames are put on the page.

```
\def\@outputpage{%
  \begingroup
    \let\protect\noexpand
    \@resetactivechars
    \global\let\@@if@newlist\if@newlist
    \global\@newlistfalse\@parboxrestore
    \shipout\vbox
    {%
      \set@typeset@protect
      \aftergroup
      \endgroup
      \aftergroup
      \set@typeset@protect
      \reset@font\normalsize\normalsfcodes
      \let\label\@gobble
      \let\index\@gobble
      \let\glossary\@gobble
      \baselineskip\z@skip
      \lineskip\z@skip
      \lineskiplimit\z@
      \vskip\topmargin\moveright\@themargin
      \vbox
      {%
        \vskip\headheight
        \vskip\headsep
        \box\@outputbox
      }%
    }%
    \global\let\if@newlist\@@if@newlist
    \stepcounter{page}%
```

Also increment absolutepage counter.

```
    \stepcounter{absolutepage}%
    \setcounter{displayedframe}{0}%
    \let\firstmark\botmark
}
```

`\makedfheaderfooter` Make the headers and footers be in dynamic frames. There will initially be no difference in appearance until the settings are changed using `\setdynamicframe`. The header frame is given the IDL `header`, and the footer is given the IDL `footer`.

```
\newcommand*{\makedfheaderfooter}{%
```

create dynamic frames at the standard location

```
  \setlength{\@ff@tmp@y}{\textheight}%
  \addtolength{\@ff@tmp@y}{\headsep}%
  \newdynamicframe{\textwidth}{\headheight}{0pt}{\@ff@tmp@y}[header]%
  \newdynamicframe{\textwidth}{\headheight}{0pt}{-\footskip}[footer]%
  \renewcommand{\@dothehead}{}%
  \renewcommand{\@dothefoot}{}%
  \renewcommand{\@dodynamicthehead}{%
    \@dynamicframeid{header}%
    \expandafter
      \def\csname @dynamicframe@\romannumeral\ff@id\endcsname{%
        \vfill\@thehead\vfill
      }%
  }%
  \renewcommand{\@dodynamicthefoot}{%
    \@dynamicframeid{footer}%
    \expandafter
     \def\csname @dynamicframe@\romannumeral\ff@id\endcsname{%
      \vfill\@thefoot\vfill
     }%
  }%
}
```

This should only be done in the preamble.

```
\@onlypreamble{\makedfheaderfooter}
```

`\footnotecolor` Set footnotes in `\footnotecolor` rather than `\normalcolor` This ensures that the footnotes appear in the same colour as the text colour for the flow frame to which they belong.

```
\newcommand{\footnotecolor}{%
  \@ifundefined{@ff@txtcol@\romannumeral\c@thisframe}%
  {%
    \normalcolor
  }%
  {%
    \edef\ff@txtcol{%
      \csname @ff@txtcol@\romannumeral\c@thisframe\endcsname
    }%
    \@s@tfftextcol
  }%
}
```

`\@makecol` Modify `\@makecol` so that the footnotes, and the footnote rule are in the colour for that frame.

```
\renewcommand{\@makecol}{%
  \ifvoid\footins
    \setbox\@outputbox\box\@cclv
  \else
    \setbox\@outputbox\vbox
    {%
```

```
      \boxmaxdepth\@maxdepth\@tempdima\dp\@cclv
      \unvbox\@cclv
      \vskip\skip\footins
      \color@begingroup
        \footnotecolor
        \footnoterule
        \unvbox\footins
      \color@endgroup
    }%
  \fi
  \xdef\@freelist{\@freelist\@midlist}%
  \global\let\@midlist\@empty
  \@combinefloats
  \ifvbox\@kludgeins
    \@makespecialcolbox
  \else
    \setbox\@outputbox\vbox to\@colht{%
      \@texttop\dimen@\dp\@outputbox
      \unvbox \@outputbox
      \vskip -\dimen@\@textbottom
    }%
  \fi
  \global\maxdepth\@maxdepth
}
```

`\@opcol`

Modify `\@opcol`, as `\if@twocolumn` is now irrelevant.

```
\def\@opcol{%
  \@outputdblcol
  \global\@mparbottom\z@
  \global\@textfloatsheight\z@
  \@floatplacement
}
```

`\@ff@checkifmoreframes`

Check to see if there are more flow frames defined, and set `\if@ff@moreframes` as appropriate. This involves iterating through all flow frames, and through each frame's page list.

```
\newif\if@ff@moreframes
\newcommand*{\@ff@checkifmoreframes}{%
  \@ff@moreframesfalse
  \@colN=\c@thisframe
  \whiledo{\@colN<\c@maxflow}%
  {%
    \advance\@colN by 1\relax
```

Skip if this page is in this frame's exclusion list.

```
    \edef\ff@xpages{\csname @ff@xpages@\romannumeral\@colN\endcsname}%
    \@for\@ff@pp:=\ff@xpages\do
    {%
      \ifnum0\@ff@pp=\@ff@pages@countreg\relax
```

```
          \@endfortrue
        \fi
      }%
      \if@endfor
```

If for loop was terminated prematurely, then this page is in this frame's exclusion list.

```
      \else
        \edef\ff@pages{\csname @ff@pages@\romannumeral\@colN\endcsname}%
        \@ff@checkpages{\ff@pages}%
```

If found a frame, break out of loop.

```
        \if@ff@moreframes
          \@colN=\c@maxflow\relax
        \fi
      \fi
    }%
    \if@ff@moreframes
    \else

      \@ff@tmpN=\@ff@pages@countreg
```

Look ahead up to a maximum of 4 pages.

```
      \count@=0\relax
      \loop
        \advance\@ff@tmpN by 1\relax
        \@colN=0\relax
        \whiledo{\@colN<\c@maxflow}%
        {%
          \advance\@colN by 1\relax
```

Skip if page is in this frame's exclusion list.

```
            \edef\ff@xpages{\csname @ff@xpages@\romannumeral\@colN\endcsname}%
            \@for\@ff@pp:=\ff@xpages\do
            {%
              \ifnum0\@ff@pp=\@ff@tmpN\relax
                \@endfortrue
              \fi
            }%
            \if@endfor
```

If for loop was terminated prematurely, then page is in this frame's exclusion list.

```
            \else
              \edef\ff@pages{\csname @ff@pages@\romannumeral\@colN\endcsname}%
              \@ff@checkpages[\@ff@tmpN]{\ff@pages}%
```

If found a frame, break out of loop.

```
              \if@ff@moreframes
                \@colN=\c@maxflow\relax
              \fi
            \fi
```

```
          }%
          \if@ff@moreframes
            \count@=4\relax
          \else
            \advance\count@ by 1\relax
          \fi
        \ifnum\count@<4
        \repeat
      \fi
    }
```

`\@ff@checkpages` Check to see if the current page lies in the page list given by `#1`.

```
\newcommand*{\@ff@checkpages}[2][\@ff@pages@countreg]{%
  \@for\@ff@pp:=#2\do{%
    \@ff@checkthispage{#1}{\@ff@pp}%
  }%
}
```

`\@ff@checkthispage` Check to see if the current page lies in the page range given by `#1`. If the page range is specified by `all`, `odd` or `even` then there are definitely more frames available, otherwise check to see if the current page lies within the number range. If the page range is `none`, ignore it.

```
\newcommand*{\@ff@checkthispage}[2]{%
  \ifthenelse{\equal{#2}{all}\or\equal{#2}{even}\or\equal{#2}{odd}}%
  {%
    \@ff@moreframestrue
  }%
  {%
    \ifthenelse{\equal{#2}{none}}%
    {}%
    {%
      \@ff@checknumrange{#1}{#2}%
    }%
  }%
}
```

`\@ff@checknumrange` The number range could be a single number, a closed range (e.g. `2-6`) or an open range (e,g. `<4` or `>10`). Use `\@ff@getrange` to find the start and end ranges. For open ended ranges assume a maximum value of 10000. If the current page is less than or equal to the maximum, there are still more flow frames available.

```
\newcommand*{\@ff@checknumrange}[2]{%
  \def\@ff@numstart{0}%
  \def\@ff@numend{100000}%
  \@ff@getrange{#2}%
  \ifnum\@ff@numend>#1\relax
    \@ff@moreframestrue
  \else
```

```
      \ifnum\@ff@numend=#1\relax
        \@ff@moreframestrue
      \fi
    \fi
  }
```

Work out the minimum and maximum values of a number range which could either be a single number, a closed number range or an open number range. If the first character is < or > then it is an open range, otherwise it is a closed range or a single number. Define a counter to use whilst determining the range.

```
  \newcount\c@ffrangenum
```

`\@ff@getrange` Now to find out what kind of range it is. If it is a single number, e.g. 24, then it will do, e.g. `\@ff@@getrange24-\relax`. If it is a closed range, e.g. 30-40, it will do , e.g. `\@ff@@getrange30-40-\relax`. If it is an open range, e.g. >25, it will do, e.g. `\@ff@@getrange>25-\relax`.

```
  \newcommand*{\@ff@getrange}[1]{%
    \expandafter\@ff@@getrange#1-\relax\end
  }
```

`\@ff@@getrange` The ranges can now be picked out. If the first character is a < or > it is an open ended range, otherwise it is either a single value, or a close ended range.

```
  \def\@ff@@getrange#1#2\end{%
    \ifx#1<\relax
     \@ff@getrangeless#1#2\end
    \else
      \ifx#1>\relax
        \@ff@getrangegreater#1#2\end
      \else
        \@@ff@getrange#1#2\end
      \fi
    \fi
  }
```

`\@ff@getrangeless` Get the values for an open ended range with an upper bound. A minimum value of 0 is assumed.

```
  \def\@ff@getrangeless<#1-\relax\end{%
    \c@ffrangenum=#1\relax
    \advance\c@ffrangenum by -1\relax
    \def\@ff@numstart{0}%
    \edef\@ff@numend{\number\c@ffrangenum}%
  }
```

`\@ff@getrangegreater` Get the values for an open ended range with a lower bound. A maximum value of 100000 is assumed.

```
  \def\@ff@getrangegreater>#1-\relax\end{%
    \c@ffrangenum=#1\relax
    \advance\c@ffrangenum by 1\relax
    \edef\@ff@numstart{\number\c@ffrangenum}%
    \def\@ff@numend{100000}%
  }
```

`\@@ff@getrange` Determine whether we have a single number or a closed range. If #2 is `\relax`, it is a single value, otherwise it is a range.

```
  \def\@@ff@getrange#1-#2\end{%
    \ifx\relax#2\relax
      \def\@ff@numstart{#1}%
      \def\@ff@numend{#1}%
    \else
      \def\@ff@numstart{#1}%
      \@@@ff@getrange#2\end
    \fi
  }
```

`\@@@ff@getrange` Extract the end value from the closed range.

```
  \def\@@@ff@getrange#1-\relax\end{%
    \def\@ff@numend{#1}%
  }
```

`@ff@output@adjustframes` Provide a hook to adjust frame settings in the output routine.

```
  \newcommand*{\@ff@output@adjustframes}{}
```

`\flowswitchonnext` Switch on the listed flow frames from the next page onwards

```
  \newcommand*{\flowswitchonnext}{%
    \@ifstar\@sflowswitchonnext\@flowswitchonnext
  }
```

`\@sflowswitchonnext` The starred version uses IDLs.

```
  \newcommand{\@sflowswitchonnext}[1]{%
    \@for\@ff@id:=#1\do{%
      \@flowframeid{\@ff@id}%
```

Is this frame already on?

```
      \@ff@chckifthispg{\@ff@pages@countreg}{\ff@id}%
      \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
      \if@notthiscol
        \xdef\@ff@output@adjustframes{%
          \the\toks@
          \noexpand\flowsetpagelist{\number\ff@id}{>\number\@ff@pages@countreg}%
        }%
      \else
        \xdef\@ff@output@adjustframes{%
          \the\toks@
          \noexpand\flowsetpagelist{\number\ff@id}%
            {\number\@ff@pages@countreg,>\number\@ff@pages@countreg}%
        }%
      \fi
    }%
  }
```

\@flowswitchonnext The unstarred version uses IDNs.

```
  \newcommand{\@flowswitchonnext}[1]{%
    \@for\@ff@id:=#1\do{%
```

Is this frame already on?

```
      \@ff@chckifthispg{\@ff@pages@countreg}{\@ff@id}%
      \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
      \if@notthiscol
        \xdef\@ff@output@adjustframes{%
          \the\toks@
          \noexpand\flowsetpagelist{\number\@ff@id}{>\number\@ff@pages@countreg}%
        }%
      \else
        \xdef\@ff@output@adjustframes{%
          \the\toks@
          \noexpand\flowsetpagelist{\number\@ff@id}%
             {\number\@ff@pages@countreg,>\number\@ff@pages@countreg}%
        }%
      \fi
    }%
  }
```

\flowswitchonnextodd Switch on the listed flow frames from the next odd page onwards

```
  \newcommand*{\flowswitchonnextodd}{%
    \@ifstar\@sflowswitchonnextodd\@flowswitchonnextodd
  }
```

\@sflowswitchonnextodd The starred version uses IDLs.

```
  \newcommand{\@sflowswitchonnextodd}[1]{%
    \count@=\@ff@pages@countreg\relax
    \ifodd\count@\relax
      \advance\count@ by 1\relax
    \fi
    \@for\@ff@id:=#1\do{%
      \@flowframeid{\@ff@id}%
```

Is this frame already on?

```
      \@ff@chckifthispg{\@ff@pages@countreg}{\ff@id}%
      \def\@ff@prepages{}%
      \if@notthiscol
      \else
         \def\@ff@prepages{\number\@ff@pages@countreg,}%
      \fi
```

Is this frame already switched on for the next page?

```
      \@ff@chckifthispg{\count@}{\ff@id}%
      \ifnum\count@=\@ff@pages@countreg\relax
      \else
        \if@notthiscol
        \else
           \edef\@ff@prepages{\@ff@prepages\number\count@,}%
        \fi
      \fi
      \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
      \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\flowsetpagelist{\number\ff@id}%
          {\@ff@prepages>\number\count@}%
      }%
    }%
  }
```

`\@flowswitchonnextodd` The unstarred version uses IDNs.

```
  \newcommand{\@flowswitchonnextodd}[1]{%
    \count@=\@ff@pages@countreg\relax
    \ifodd\count@\relax
      \advance\count@ by 1\relax
    \fi
    \@for\@ff@id:=#1\do{%
```

Is this frame already on?

```
      \@ff@chckifthispg{\@ff@pages@countreg}{\@ff@id}%
      \def\@ff@prepages{}%
      \if@notthiscol
      \else
         \def\@ff@prepages{\number\@ff@pages@countreg,}%
      \fi
```

Is this frame already switched on for the next page?

```
      \@ff@chckifthispg{\count@}{\@ff@id}%
      \ifnum\count@=\@ff@pages@countreg\relax
      \else
        \if@notthiscol
        \else
           \edef\@ff@prepages{\@ff@prepages\number\count@,}%
        \fi
      \fi
      \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
      \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\flowsetpagelist{\number\@ff@id}%
          {\@ff@prepages>\number\count@}%
      }%
    }%
  }
```

`\flowswitchoffnext` Switch off the listed flow frames from the next page onwards

```
\newcommand*{\flowswitchoffnext}{%
  \@ifstar\@sflowswitchoffnext\@flowswitchoffnext
}
```

`\@sflowswitchoffnext` The starred version uses IDLs.

```
\newcommand{\@sflowswitchoffnext}[1]{%
  \@for\@ff@id:=#1\do{%
    \@flowframeid{\@ff@id}%
```

Is this frame already off on this page?

```
    \@ff@chckifthispg{\@ff@pages@countreg}{\ff@id}%
    \if@notthiscol
       \def\@ff@pages{none}%
    \else
       \def\@ff@pages{\number\@ff@pages@countreg}%
    \fi
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\flowsetpagelist{\number\ff@id}{\@ff@pages}%
    }%
  }%
}
```

`\@flowswitchoffnext` The unstarred version uses IDNs.

```
\newcommand{\@flowswitchoffnext}[1]{%
  \@for\@ff@id:=#1\do{%
```

Is this frame already off on this page?

```
    \@ff@chckifthispg{\@ff@pages@countreg}{\@ff@id}%
    \if@notthiscol
       \def\@ff@pages{none}%
    \else
       \def\@ff@pages{\number\@ff@pages@countreg}%
    \fi
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\flowsetpagelist{\number\@ff@id}{\@ff@pages}%
    }%
  }%
}
```

`\flowswitchoffnextodd` Switch off the listed flow frames from the next odd page onwards

```
\newcommand*{\flowswitchoffnextodd}{%
  \@ifstar\@sflowswitchoffnextodd\@flowswitchoffnextodd
}
```

\@sflowswitchoffnextodd The starred version uses IDLs.

```
\newcommand{\@sflowswitchoffnextodd}[1]{%
  \count@=\@ff@pages@countreg\relax
  \ifodd\@ff@pages@countreg\relax
    \advance\count@ by 1\relax
  \fi
  \@for\@ff@id:=#1\do{%
    \@flowframeid{\@ff@id}%
```

Is this frame already off on this page?

```
    \@ff@chckifthispg{\@ff@pages@countreg}{\ff@id}%
    \if@notthiscol
```

It's off on this page. Is it on or off on the next page, if this page is odd? First, is this page odd?

```
      \ifnum\@ff@pages@countreg=\count@\relax
```

This page is even and the frame is off on this page, so set to none.

```
        \def\@ff@nextpages{none}%
      \else
```

This page is odd. Is the frame on or off on the next page?

```
        \@ff@chckifthispg{\count@}{\ff@id}%
        \if@notthiscol
```

Off on the next page as well, so set to none.

```
          \def\@ff@nextpages{none}%
        \else
```

Not off on the next page, so set to next page only.

```
          \def\@ff@nextpages{\number\count@}%
        \fi
      \fi
    \else
```

It's not off on this page. Is it on or off on the next page, if this page is odd? First, is this page odd?

```
      \ifnum\@ff@pages@countreg=\count@\relax
```

This page is even and the frame is not off on this page, so set to this page.

```
        \def\@ff@nextpages{\number\@ff@pages@countreg}%
      \else
```

This page is odd. Is the frame on or off on the next page?

```
        \@ff@chckifthispg{\count@}{\ff@id}%
        \if@notthiscol
```

Off on the next page but not off on this page. So set to just this page.

```
          \def\@ff@nextpages{\number\@ff@pages@countreg}%
        \else
```

Not off on the next page as well, so set to this page and next page.

```
          \def\@ff@nextpages{\number\@ff@pages@countreg,\number\count@}%
        \fi
      \fi

    \fi
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\flowsetpagelist{\number\ff@id}{\@ff@nextpages}%
    }%
  }%
}
```

\@flowswitchoffnextodd The unstarred version uses IDNs.

```
\newcommand{\@flowswitchoffnextodd}[1]{%
  \count@=\@ff@pages@countreg\relax
  \ifodd\@ff@pages@countreg\relax
    \advance\count@ by 1\relax
  \fi
  \@for\@ff@id:=#1\do{%
```

Is this frame already off on this page?

```
    \@ff@chckifthispg{\@ff@pages@countreg}{\@ff@id}%
    \if@notthiscol
```

It's off on this page. Is it on or off on the next page, if this page is odd? First, is this page odd?

```
      \ifnum\@ff@pages@countreg=\count@\relax
```

This page is even and the frame is off on this page, so set to none.

```
        \def\@ff@nextpages{none}%
      \else
```

This page is odd. Is the frame on or off on the next page?

```
        \@ff@chckifthispg{\count@}{\@ff@id}%
        \if@notthiscol
```

Off on the next page as well, so set to none.

```
          \def\@ff@nextpages{none}%
        \else
```

Not off on the next page, so set to next page only.

```
          \def\@ff@nextpages{\number\count@}%
        \fi
      \fi
    \else
```

It's not off on this page. Is it on or off on the next page, if this page is odd? First, is this page odd?

```
      \ifnum\@ff@pages@countreg=\count@\relax
```

This page is even and the frame is not off on this page, so set to this page.

```
    \def\@ff@nextpages{\number\@ff@pages@countreg}%
   \else
```

This page is odd. Is the frame on or off on the next page?

```
    \@ff@chckifthispg{\count@}{\@ff@id}%
    \if@notthiscol
```

Off on the next page but not off on this page. So set to just this page.

```
     \def\@ff@nextpages{\number\@ff@pages@countreg}%
    \else
```

Not off on the next page as well, so set to this page and next page.

```
     \def\@ff@nextpages{\number\@ff@pages@countreg,\number\count@}%
    \fi
   \fi
  \fi
  \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
  \xdef\@ff@output@adjustframes{%
   \the\toks@
   \noexpand\flowsetpagelist{\number\@ff@id}{\@ff@nextpages}%
  }%
 }%
}
```

\flowswitchonnextonly Switch on the listed flow frames for just the next page

```
\newcommand*{\flowswitchonnextonly}{%
 \@ifstar\@sflowswitchonnextonly\@flowswitchonnextonly
}
```

\@sflowswitchonnextonly The starred version uses IDLs.

```
\newcommand{\@sflowswitchonnextonly}[1]{%
 \count@=\@ff@pages@countreg\relax
 \advance\count@ by 1\relax
 \@for\@ff@id:=#1\do{%
  \@flowframeid{\@ff@id}%
```

Is this frame already on?

```
  \@ff@chckifthispg{\@ff@pages@countreg}{\ff@id}%
  \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
  \if@notthiscol
```

Not, it isn't, so just set to the next page:

```
   \xdef\@ff@output@adjustframes{%
    \the\toks@
    \noexpand\flowsetpagelist{\number\ff@id}{\number\count@}%
   }%
  \else
```

Yes, it is, so set to this page and the next page:

```
      \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\flowsetpagelist{\number\ff@id}%
          {\number\@ff@pages@countreg,\number\count@}%
      }%
    \fi
  }%
}
```

\@flowswitchonnextonly

The unstarred version uses IDNs.

```
\newcommand{\@flowswitchonnextonly}[1]{%
  \count@=\@ff@pages@countreg\relax
  \advance\count@ by 1\relax
  \@for\@ff@id:=#1\do{%
```

Is this frame already on?

```
    \@ff@chckifthispg{\@ff@pages@countreg}{\@ff@id}%
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \if@notthiscol
```

Not, it isn't, so just set to the next page:

```
      \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\flowsetpagelist{\number\@ff@id}{\number\count@}%
      }%
    \else
```

Yes, it is, so set to this page and the next page:

```
      \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\flowsetpagelist{\number\@ff@id}%
          {\number\@ff@pages@countreg,\number\count@}%
      }%
    \fi
  }%
}
```

flowswitchonnextoddonly

Switch on the listed flow frames for just the next odd page

```
\newcommand*{\flowswitchonnextoddonly}{%
  \@ifstar\@sflowswitchonnextoddonly\@flowswitchonnextoddonly
}
```

flowswitchonnextoddonly

The starred version uses IDLs.

```
\newcommand{\@sflowswitchonnextoddonly}[1]{%
  \@for\@ff@id:=#1\do{%
    \@flowframeid{\@ff@id}%
```

Is this frame already on?

```
    \@ff@chckifthispg{\@ff@pages@countreg}{\ff@id}%
    \if@notthiscol
```

No, it isn't. If this is an odd page, is it on or off on the next page? First, is this an odd page?

```
    \ifodd\@ff@pages@countreg
```

Yes, it's odd. So this frame isn't on this page, but is it on or off on the next page?

```
      \count@=\@ff@pages@countreg\relax
      \advance\count@ by 1\relax
      \@ff@chckifthispg{\count@}{\ff@id}%
      \if@notthiscol
```

It's not switched on either on this (odd) page or the next (even) page. So the page list should be just the next odd page after this one.

```
       \advance\count@ by 1\relax
       \edef\@ff@pages{\number\count@}%
      \else
```

It's not switched on for this (odd) page but it is for the next (even) page. So the page list should be the next even and odd pages after this page.

```
       \edef\@ff@pages{\number\count@}%
       \advance\count@ by 1\relax
       \edef\@ff@pages{\@ff@pages,\number\count@}%
      \fi
    \else
```

No, it's even. So it's not on this (even) page, but needs to be on for the following (odd) page.

```
     \count@=\@ff@pages@countreg\relax
     \advance\count@ by 1\relax
     \edef\@ff@pages{\number\count@}%
    \fi
  \else
```

Frame is on this page. If this is an odd page, is it on or off on the next page? First, is this an odd page?

```
    \ifodd\@ff@pages@countreg
```

Yes, it's odd. Is the frame on or off for the next (even) page?

```
      \count@=\@ff@pages@countreg\relax
      \advance\count@ by 1\relax
      \@ff@chckifthispg{\count@}{\ff@id}%
      \if@notthiscol
```

Frame is off. So the frame is switched on for this (odd) page but is off for the next (even) page. So the page list needs to be this (odd) page and the following odd page, skipping the even page in between.

```
        \advance\count@ by 1\relax
        \edef\@ff@pages{\number\@ff@pages@countreg,\number\count@}%
      \else
```

Frame is on. So the frame is switched on for this (odd) page and the next (even) page. So the page list needs to be this (odd) page, the next even page and the following odd page.

```
          \advance\count@ by 1\relax
          \edef\@ff@pages{\number\@ff@pages@countreg-\number\count@}%
        \fi
      \else
```

Frame is switched on for this page and this page is even. So the page list needs to be this (even) page and the next (odd) page.

```
        \count@=\@ff@pages@countreg\relax
        \advance\count@ by 1\relax
        \edef\@ff@pages{\number\@ff@pages@countreg,\number\count@}%
      \fi
```

```
    \fi
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\flowsetpagelist{\number\ff@id}{\@ff@pages}%
    }%
  }%
}
```

flowswitchonnextoddonly The unstarred version uses IDNs.

```
\newcommand{\@flowswitchonnextoddonly}[1]{%
  \@for\@ff@id:=#1\do{%
```

Is this frame already on?

```
    \@ff@chckifthispg{\@ff@pages@countreg}{\@ff@id}%
    \if@notthiscol
```

No, it isn't. If this is an odd page, is it on or off on the next page? First, is this an odd page?

```
      \ifodd\@ff@pages@countreg
```

Yes, it's odd. So this frame isn't on this page, but is it on or off on the next page?

```
        \count@=\@ff@pages@countreg\relax
        \advance\count@ by 1\relax
        \@ff@chckifthispg{\count@}{\@ff@id}%
        \if@notthiscol
```

It's not switched on either on this (odd) page or the next (even) page. So the page list should be just the next odd page after this one.

```
         \advance\count@ by 1\relax
         \edef\@ff@pages{\number\count@}%
        \else
```

It's not switched on for this (odd) page but it is for the next (even) page. So the page list should be the next even and odd pages after this page.

```
         \edef\@ff@pages{\number\count@}%
         \advance\count@ by 1\relax
         \edef\@ff@pages{\@ff@pages,\number\count@}%
        \fi
      \else
```

No, it's even. So it's not on this (even) page, but needs to be on for the following (odd) page.

```
       \count@=\@ff@pages@countreg\relax
       \advance\count@ by 1\relax
       \edef\@ff@pages{\number\count@}%
      \fi
    \else
```

Frame is on this page. If this is an odd page, is it on or off on the next page? First, is this an odd page?

```
      \ifodd\@ff@pages@countreg
```

Yes, it's odd. Is the frame on or off for the next (even) page?

```
        \count@=\@ff@pages@countreg\relax
        \advance\count@ by 1\relax
        \@ff@chckifthispg{\count@}{\@ff@id}%
        \if@notthiscol
```

Frame is off. So the frame is switched on for this (odd) page but is off for the next (even) page. So the page list needs to be this (odd) page and the following odd page, skipping the even page in between.

```
          \advance\count@ by 1\relax
          \edef\@ff@pages{\number\@ff@pages@countreg,\number\count@}%
        \else
```

Frame is on. So the frame is switched on for this (odd) page and the next (even) page. So the page list needs to be this (odd) page, the next even page and the following odd page.

```
          \advance\count@ by 1\relax
          \edef\@ff@pages{\number\@ff@pages@countreg-\number\count@}%
        \fi
      \else
```

Frame is switched on for this page and this page is even. So the page list needs to be this (even) page and the next (odd) page.

```
        \count@=\@ff@pages@countreg\relax
        \advance\count@ by 1\relax
        \edef\@ff@pages{\number\@ff@pages@countreg,\number\count@}%
      \fi
```

```
    \fi
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\flowsetpagelist{\number\@ff@id}{\@ff@pages}%
    }%
  }%
}
```

`\flowswitchoffnextonly` Switch off the listed flow frames for just the next page

```
\newcommand*{\flowswitchoffnextonly}{%
  \@ifstar\@sflowswitchoffnextonly\@flowswitchoffnextonly
}
```

@sflowswitchoffnextonly The starred version uses IDLs.

```
\newcommand{\@sflowswitchoffnextonly}[1]{%
  \count@=\@ff@pages@countreg\relax
  \advance\count@ by 1\relax
  \@for\@ff@id:=#1\do{%
    \@flowframeid{\@ff@id}%
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\flowaddexclusion{\number\ff@id}{\number\count@}%
    }%
  }%
}
```

\@flowswitchoffnextonly The unstarred version uses IDNs.

```
\newcommand{\@flowswitchoffnextonly}[1]{%
  \count@=\@ff@pages@countreg\relax
  \advance\count@ by 1\relax
  \@for\@ff@id:=#1\do{%
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\flowaddexclusion{\number\@ff@id}{\number\count@}%
    }%
  }%
}
```

lowswitchoffnextoddonly Switch off the listed flow frames for just the next odd page

```
\newcommand*{\flowswitchoffnextoddonly}{%
  \@ifstar\@sflowswitchoffnextoddonly\@flowswitchoffnextoddonly
}
```

lowswitchoffnextoddonly The starred version uses IDLs.

```
\newcommand{\@sflowswitchoffnextoddonly}[1]{%
  \count@=\@ff@pages@countreg\relax
  \advance\count@ by 1\relax
  \ifodd\count@\relax
  \else
    \advance\count@ by 1\relax
  \fi
  \@for\@ff@id:=#1\do{%
    \@flowframeid{\@ff@id}%
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
```

```
   \noexpand\flowaddexclusion{\number\ff@id}{\number\count@}%
  }%
 }%
}
```

`lowswitchoffnextoddonly` The unstarred version uses IDNs.

```
\newcommand{\@flowswitchoffnextoddonly}[1]{%
 \count@=\@ff@pages@countreg\relax
 \advance\count@ by 1\relax
 \ifodd\count@\relax
 \else
  \advance\count@ by 1\relax
 \fi
 \@for\@ff@id:=#1\do{%
  \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
  \xdef\@ff@output@adjustframes{%
   \the\toks@
   \noexpand\flowaddexclusion{\number\@ff@id}{\number\count@}%
  }%
 }%
}
```

`\dynamicswitchonnext` Switch on the listed dynamic frames from the next page onwards

```
\newcommand*{\dynamicswitchonnext}{%
 \@ifstar\@sdynamicswitchonnext\@dynamicswitchonnext
}
```

`\@sdynamicswitchonnext` The starred version uses IDLs.

```
\newcommand{\@sdynamicswitchonnext}[1]{%
 \@for\@ff@id:=#1\do{%
  \@dynamicframeid{\@ff@id}%
```

Is this frame already on?

```
  \@df@chckifthispg[\@ff@pages@countreg]{\ff@id}%
  \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
  \if@notthiscol
   \xdef\@ff@output@adjustframes{%
    \the\toks@
    \noexpand\dynamicsetpagelist{\number\ff@id}{>\number\@ff@pages@countreg}%
   }%
  \else
   \xdef\@ff@output@adjustframes{%
    \the\toks@
    \noexpand\dynamicsetpagelist{\number\ff@id}%
     {\number\@ff@pages@countreg,>\number\@ff@pages@countreg}%
   }%
  \fi
 }%
}
```

`\@dynamicswitchonnext` The unstarred version uses IDNs.

```
\newcommand{\@dynamicswitchonnext}[1]{%
  \@for\@ff@id:=#1\do{%
```

Is this frame already on?

```
    \@df@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \if@notthiscol
      \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\dynamicsetpagelist{\number\@ff@id}{>\number\@ff@pages@countreg}%
      }%
    \else
      \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\dynamicsetpagelist{\number\@ff@id}%
           {\number\@ff@pages@countreg,>\number\@ff@pages@countreg}%
      }%
    \fi
  }%
}
```

`\dynamicswitchonnextodd` Switch on the listed dynamic frames from the next odd page onwards

```
\newcommand*{\dynamicswitchonnextodd}{%
  \@ifstar\@sdynamicswitchonnextodd\@dynamicswitchonnextodd
}
```

`sdynamicswitchonnextodd` The starred version uses IDLs.

```
\newcommand{\@sdynamicswitchonnextodd}[1]{%
  \count@=\@ff@pages@countreg\relax
  \ifodd\count@\relax
    \advance\count@ by 1\relax
  \fi
  \@for\@ff@id:=#1\do{%
    \@dynamicframeid{\@ff@id}%
```

Is this frame already on?

```
    \@df@chckifthispg[\@ff@pages@countreg]{\ff@id}%
    \def\@ff@prepages{}%
    \if@notthiscol
    \else
       \def\@ff@prepages{\number\@ff@pages@countreg,}%
    \fi
```

Is this frame already switched on for the next page?

```
    \@df@chckifthispg[\count@]{\ff@id}%
    \ifnum\count@=\@ff@pages@countreg\relax
    \else
      \if@notthiscol
      \else
```

```
         \edef\@ff@prepages{\@ff@prepages\number\count@,}%
      \fi
    \fi
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\dynamicsetpagelist{\number\ff@id}%
        {\@ff@prepages>\number\count@}%
    }%
  }%
}
```

@dynamicswitchonnextodd

The unstarred version uses IDNs.

```
\newcommand{\@dynamicswitchonnextodd}[1]{%
  \count@=\@ff@pages@countreg\relax
  \ifodd\count@\relax
    \advance\count@ by 1\relax
  \fi
  \@for\@ff@id:=#1\do{%
```

Is this frame already on?

```
    \@df@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
    \def\@ff@prepages{}%
    \if@notthiscol
    \else
       \def\@ff@prepages{\number\@ff@pages@countreg,}%
    \fi
```

Is this frame already switched on for the next page?

```
    \@df@chckifthispg[\count@]{\@ff@id}%
    \ifnum\count@=\@ff@pages@countreg\relax
    \else
      \if@notthiscol
      \else
         \edef\@ff@prepages{\@ff@prepages\number\count@,}%
      \fi
    \fi
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\dynamicsetpagelist{\number\@ff@id}%
        {\@ff@prepages>\number\count@}%
    }%
  }%
}
```

\dynamicswitchoffnext

Switch off the listed dynamic frames from the next page onwards

```
\newcommand*{\dynamicswitchoffnext}{%
  \@ifstar\@sdynamicswitchoffnext\@dynamicswitchoffnext
}
```

`\@sdynamicswitchoffnext` The starred version uses IDLs.

```
\newcommand{\@sdynamicswitchoffnext}[1]{%
  \@for\@ff@id:=#1\do{%
    \@dynamicframeid{\@ff@id}%
```

Is this frame already off on this page?

```
    \@df@chckifthispg[\@ff@pages@countreg]{\ff@id}%
    \if@notthiscol
       \def\@ff@pages{none}%
    \else
       \def\@ff@pages{\number\@ff@pages@countreg}%
    \fi
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\dynamicsetpagelist{\number\ff@id}{\@ff@pages}%
    }%
  }%
}
```

`\@dynamicswitchoffnext` The unstarred version uses IDNs.

```
\newcommand{\@dynamicswitchoffnext}[1]{%
  \@for\@ff@id:=#1\do{%
```

Is this frame already off on this page?

```
    \@df@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
    \if@notthiscol
       \def\@ff@pages{none}%
    \else
       \def\@ff@pages{\number\@ff@pages@countreg}%
    \fi
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\dynamicsetpagelist{\number\@ff@id}{\@ff@pages}%
    }%
  }%
}
```

`dynamicswitchoffnextodd` Switch off the listed dynamic frames from the next odd page onwards

```
\newcommand*{\dynamicswitchoffnextodd}{%
  \@ifstar\@sdynamicswitchoffnextodd\@dynamicswitchoffnextodd
}
```

`dynamicswitchoffnextodd` The starred version uses IDLs.

```
\newcommand{\@sdynamicswitchoffnextodd}[1]{%
  \count@=\@ff@pages@countreg\relax
  \ifodd\@ff@pages@countreg\relax
    \advance\count@ by 1\relax
  \fi
  \@for\@ff@id:=#1\do{%
    \@dynamicframeid{\@ff@id}%
```

Is this frame already off on this page?

```
    \@df@chckifthispg[\@ff@pages@countreg]{\ff@id}%
    \if@notthiscol
```

It's off on this page. Is it on or off on the next page, if this page is odd? First, is this page odd?

```
      \ifnum\@ff@pages@countreg=\count@\relax
```

This page is even and the frame is off on this page, so set to none.

```
        \def\@ff@nextpages{none}%
      \else
```

This page is odd. Is the frame on or off on the next page?

```
        \@df@chckifthispg[\count@]{\ff@id}%
        \if@notthiscol
```

Off on the next page as well, so set to none.

```
          \def\@ff@nextpages{none}%
        \else
```

Not off on the next page, so set to next page only.

```
          \def\@ff@nextpages{\number\count@}%
        \fi
      \fi
    \else
```

It's not off on this page. Is it on or off on the next page, if this page is odd? First, is this page odd?

```
      \ifnum\@ff@pages@countreg=\count@\relax
```

This page is even and the frame is not off on this page, so set to this page.

```
        \def\@ff@nextpages{\number\@ff@pages@countreg}%
      \else
```

This page is odd. Is the frame on or off on the next page?

```
        \@df@chckifthispg[\count@]{\ff@id}%
        \if@notthiscol
```

Off on the next page but not off on this page. So set to just this page.

```
          \def\@ff@nextpages{\number\@ff@pages@countreg}%
        \else
```

Not off on the next page as well, so set to this page and next page.

```
          \def\@ff@nextpages{\number\@ff@pages@countreg,\number\count@}%
        \fi
      \fi

    \fi
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
```

```
      \the\toks@
      \noexpand\dynamicsetpagelist{\number\ff@id}{\@ff@nextpages}%
    }%
  }%
}
```

dynamicswitchoffnextodd The unstarred version uses IDNs.

```
\newcommand{\@dynamicswitchoffnextodd}[1]{%
  \count@=\@ff@pages@countreg\relax
  \ifodd\@ff@pages@countreg\relax
    \advance\count@ by 1\relax
  \fi
  \@for\@ff@id:=#1\do{%
```

Is this frame already off on this page?

```
    \@df@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
    \if@notthiscol
```

It's off on this page. Is it on or off on the next page, if this page is odd? First, is this page odd?

```
      \ifnum\@ff@pages@countreg=\count@\relax
```

This page is even and the frame is off on this page, so set to none.

```
        \def\@ff@nextpages{none}%
      \else
```

This page is odd. Is the frame on or off on the next page?

```
        \@df@chckifthispg[\count@]{\@ff@id}%
        \if@notthiscol
```

Off on the next page as well, so set to none.

```
          \def\@ff@nextpages{none}%
        \else
```

Not off on the next page, so set to next page only.

```
          \def\@ff@nextpages{\number\count@}%
        \fi
      \fi
    \else
```

It's not off on this page. Is it on or off on the next page, if this page is odd? First, is this page odd?

```
      \ifnum\@ff@pages@countreg=\count@\relax
```

This page is even and the frame is not off on this page, so set to this page.

```
        \def\@ff@nextpages{\number\@ff@pages@countreg}%
      \else
```

This page is odd. Is the frame on or off on the next page?

```
        \@df@chckifthispg[\count@]{\@ff@id}%
        \if@notthiscol
```

Off on the next page but not off on this page. So set to just this page.

```
     \def\@ff@nextpages{\number\@ff@pages@countreg}%
    \else
```

Not off on the next page as well, so set to this page and next page.

```
     \def\@ff@nextpages{\number\@ff@pages@countreg,\number\count@}%
    \fi
   \fi

  \fi
  \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
  \xdef\@ff@output@adjustframes{%
   \the\toks@
   \noexpand\dynamicsetpagelist{\number\@ff@id}{\@ff@nextpages}%
  }%
 }%
}
```

dynamicswitchonnextonly

Switch on the listed dynamic frames for just the next page

```
\newcommand*{\dynamicswitchonnextonly}{%
 \@ifstar\@sdynamicswitchonnextonly\@dynamicswitchonnextonly
}
```

dynamicswitchonnextonly

The starred version uses IDLs.

```
\newcommand{\@sdynamicswitchonnextonly}[1]{%
 \count@=\@ff@pages@countreg\relax
 \advance\count@ by 1\relax
 \@for\@ff@id:=#1\do{%
  \@dynamicframeid{\@ff@id}%
```

Is this frame already on?

```
  \@df@chckifthispg[\@ff@pages@countreg]{\ff@id}%
  \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
  \if@notthiscol
```

Not, it isn't, so just set to the next page:

```
   \xdef\@ff@output@adjustframes{%
    \the\toks@
    \noexpand\dynamicsetpagelist{\number\ff@id}{\number\count@}%
   }%
  \else
```

Yes, it is, so set to this page and the next page:

```
   \xdef\@ff@output@adjustframes{%
    \the\toks@
    \noexpand\dynamicsetpagelist{\number\ff@id}%
     {\number\@ff@pages@countreg,\number\count@}%
   }%
  \fi
 }%
}
```

dynamicswitchonnextonly

The unstarred version uses IDNs.

```
\newcommand{\@dynamicswitchonnextonly}[1]{%
  \count@=\@ff@pages@countreg\relax
  \advance\count@ by 1\relax
  \@for\@ff@id:=#1\do{%
```

Is this frame already on?

```
    \@df@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \if@notthiscol
```

Not, it isn't, so just set to the next page:

```
      \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\dynamicsetpagelist{\number\@ff@id}{\number\count@}%
      }%
    \else
```

Yes, it is, so set to this page and the next page:

```
      \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\dynamicsetpagelist{\number\@ff@id}%
          {\number\@ff@pages@countreg,\number\count@}%
      }%
    \fi
  }%
}
```

amicswitchonnextoddonly

Switch on the listed dynamic frames for just the next odd page

```
\newcommand*{\dynamicswitchonnextoddonly}{%
  \@ifstar\@sdynamicswitchonnextoddonly\@dynamicswitchonnextoddonly
}
```

amicswitchonnextoddonly

The starred version uses IDLs.

```
\newcommand{\@sdynamicswitchonnextoddonly}[1]{%
  \@for\@ff@id:=#1\do{%
    \@dynamicframeid{\@ff@id}%
```

Is this frame already on?

```
    \@df@chckifthispg[\@ff@pages@countreg]{\ff@id}%
    \if@notthiscol
```

No, it isn't. If this is an odd page, is it on or off on the next page? First, is this an odd page?

```
      \ifodd\@ff@pages@countreg
```

Yes, it's odd. So this frame isn't on this page, but is it on or off on the next page?

```
        \count@=\@ff@pages@countreg\relax
        \advance\count@ by 1\relax
        \@df@chckifthispg[\count@]{\ff@id}%
        \if@notthiscol
```

It’s not switched on either on this (odd) page or the next (even) page. So the page list should be just the next odd page after this one.

```
           \advance\count@ by 1\relax
           \edef\@ff@pages{\number\count@}%
          \else
```

It’s not switched on for this (odd) page but it is for the next (even) page. So the page list should be the next even and odd pages after this page.

```
           \edef\@ff@pages{\number\count@}%
           \advance\count@ by 1\relax
           \edef\@ff@pages{\@ff@pages,\number\count@}%
          \fi
        \else
```

No, it’s even. So it’s not on this (even) page, but needs to be on for the following (odd) page.

```
         \count@=\@ff@pages@countreg\relax
         \advance\count@ by 1\relax
         \edef\@ff@pages{\number\count@}%
        \fi
      \else
```

Frame is on this page. If this is an odd page, is it on or off on the next page? First, is this an odd page?

```
        \ifodd\@ff@pages@countreg
```

Yes, it’s odd. Is the frame on or off for the next (even) page?

```
          \count@=\@ff@pages@countreg\relax
          \advance\count@ by 1\relax
          \@df@chckifthispg[\count@]{\ff@id}%
          \if@notthiscol
```

Frame is off. So the frame is switched on for this (odd) page but is off for the next (even) page. So the page list needs to be this (odd) page and the following odd page, skipping the even page in between.

```
            \advance\count@ by 1\relax
            \edef\@ff@pages{\number\@ff@pages@countreg,\number\count@}%
          \else
```

Frame is on. So the frame is switched on for this (odd) page and the next (even) page. So the page list needs to be this (odd) page, the next even page and the following odd page.

```
            \advance\count@ by 1\relax
            \edef\@ff@pages{\number\@ff@pages@countreg-\number\count@}%
          \fi
        \else
```

Frame is switched on for this page and this page is even. So the page list needs to be this (even) page and the next (odd) page.

```
          \count@=\@ff@pages@countreg\relax
          \advance\count@ by 1\relax
```

```
        \edef\@ff@pages{\number\@ff@pages@countreg,\number\count@}%
      \fi
%    \begin{macrocode}
    \fi
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\dynamicsetpagelist{\number\ff@id}{\@ff@pages}%
    }%
  }%
}
```

amicswitchonnextoddonly The unstarred version uses IDNs.

```
\newcommand{\@dynamicswitchonnextoddonly}[1]{%
  \@for\@ff@id:=#1\do{%
```

Is this frame already on?

```
    \@df@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
    \if@notthiscol
```

No, it isn't. If this is an odd page, is it on or off on the next page? First, is this an odd page?

```
      \ifodd\@ff@pages@countreg
```

Yes, it's odd. So this frame isn't on this page, but is it on or off on the next page?

```
        \count@=\@ff@pages@countreg\relax
        \advance\count@ by 1\relax
        \@df@chckifthispg[\count@]{\@ff@id}%
        \if@notthiscol
```

It's not switched on either on this (odd) page or the next (even) page. So the page list should be just the next odd page after this one.

```
         \advance\count@ by 1\relax
         \edef\@ff@pages{\number\count@}%
        \else
```

It's not switched on for this (odd) page but it is for the next (even) page. So the page list should be the next even and odd pages after this page.

```
         \edef\@ff@pages{\number\count@}%
         \advance\count@ by 1\relax
         \edef\@ff@pages{\@ff@pages,\number\count@}%
        \fi
      \else
```

No, it's even. So it's not on this (even) page, but needs to be on for the following (odd) page.

```
       \count@=\@ff@pages@countreg\relax
       \advance\count@ by 1\relax
       \edef\@ff@pages{\number\count@}%
      \fi
    \else
```

Frame is on this page. If this is an odd page, is it on or off on the next page? First, is this an odd page?

```
   \ifodd\@ff@pages@countreg
```

Yes, it's odd. Is the frame on or off for the next (even) page?

```
    \count@=\@ff@pages@countreg\relax
    \advance\count@ by 1\relax
    \@df@chckifthispg[\count@]{\@ff@id}%
    \if@notthiscol
```

Frame is off. So the frame is switched on for this (odd) page but is off for the next (even) page. So the page list needs to be this (odd) page and the following odd page, skipping the even page in between.

```
     \advance\count@ by 1\relax
     \edef\@ff@pages{\number\@ff@pages@countreg,\number\count@}%
    \else
```

Frame is on. So the frame is switched on for this (odd) page and the next (even) page. So the page list needs to be this (odd) page, the next even page and the following odd page.

```
     \advance\count@ by 1\relax
     \edef\@ff@pages{\number\@ff@pages@countreg-\number\count@}%
    \fi
   \else
```

Frame is switched on for this page and this page is even. So the page list needs to be this (even) page and the next (odd) page.

```
    \count@=\@ff@pages@countreg\relax
    \advance\count@ by 1\relax
    \edef\@ff@pages{\number\@ff@pages@countreg,\number\count@}%
   \fi
```

```
  \fi
  \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
  \xdef\@ff@output@adjustframes{%
   \the\toks@
   \noexpand\dynamicsetpagelist{\number\@ff@id}{\@ff@pages}%
  }%
 }%
}
```

`ynamicswitchoffnextonly` Switch off the listed dynamic frames for just the next page

```
\newcommand*{\dynamicswitchoffnextonly}{%
 \@ifstar\@sdynamicswitchoffnextonly\@dynamicswitchoffnextonly
}
```

`ynamicswitchoffnextonly` The starred version uses IDLs.

```
\newcommand{\@sdynamicswitchoffnextonly}[1]{%
 \count@=\@ff@pages@countreg\relax
 \advance\count@ by 1\relax
```

```
  \@for\@ff@id:=#1\do{%
    \@dynamicframeid{\@ff@id}%
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\dynamicaddexclusion{\number\ff@id}{\number\count@}%
    }%
  }%
}
```

`ynamicswitchoffnextonly` The unstarred version uses IDNs.

```
\newcommand{\@dynamicswitchoffnextonly}[1]{%
  \count@=\@ff@pages@countreg\relax
  \advance\count@ by 1\relax
  \@for\@ff@id:=#1\do{%
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\dynamicaddexclusion{\number\@ff@id}{\number\count@}%
    }%
  }%
}
```

`micswitchoffnextoddonly` Switch off the listed dynamic frames for just the next odd page

```
\newcommand*{\dynamicswitchoffnextoddonly}{%
  \@ifstar\@sdynamicswitchoffnextoddonly\@dynamicswitchoffnextoddonly
}
```

`micswitchoffnextoddonly` The starred version uses IDLs.

```
\newcommand{\@sdynamicswitchoffnextoddonly}[1]{%
  \count@=\@ff@pages@countreg\relax
  \advance\count@ by 1\relax
  \ifodd\count@\relax
  \else
    \advance\count@ by 1\relax
  \fi
  \@for\@ff@id:=#1\do{%
    \@dynamicframeid{\@ff@id}%
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\dynamicaddexclusion{\number\ff@id}{\number\count@}%
    }%
  }%
}
```

`micswitchoffnextoddonly` The unstarred version uses IDNs.

```
\newcommand{\@dynamicswitchoffnextoddonly}[1]{%
  \count@=\@ff@pages@countreg\relax
```

```
    \advance\count@ by 1\relax
    \ifodd\count@\relax
    \else
      \advance\count@ by 1\relax
    \fi
    \@for\@ff@id:=#1\do{%
      \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
      \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\dynamicaddexclusion{\number\@ff@id}{\number\count@}%
      }%
    }%
  }
```

`\staticswitchonnext` Switch on the listed static frames from the next page onwards

```
  \newcommand*{\staticswitchonnext}{%
    \@ifstar\@sstaticswitchonnext\@staticswitchonnext
  }
```

`\@sstaticswitchonnext` The starred version uses IDLs.

```
  \newcommand{\@sstaticswitchonnext}[1]{%
    \@for\@ff@id:=#1\do{%
      \@staticframeid{\@ff@id}%
```

Is this frame already on?

```
      \@sf@chckifthispg[\@ff@pages@countreg]{\ff@id}%
      \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
      \if@notthiscol
        \xdef\@ff@output@adjustframes{%
          \the\toks@
          \noexpand\staticsetpagelist{\number\ff@id}{>\number\@ff@pages@countreg}%
        }%
      \else
        \xdef\@ff@output@adjustframes{%
          \the\toks@
          \noexpand\staticsetpagelist{\number\ff@id}%
            {\number\@ff@pages@countreg,>\number\@ff@pages@countreg}%
        }%
      \fi
    }%
  }
```

`\@staticswitchonnext` The unstarred version uses IDNs.

```
  \newcommand{\@staticswitchonnext}[1]{%
    \@for\@ff@id:=#1\do{%
```

Is this frame already on?

```
      \@sf@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
      \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
      \if@notthiscol
      \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\staticsetpagelist{\number\@ff@id}{>\number\@ff@pages@countreg}%
      }%
    \else
      \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\staticsetpagelist{\number\@ff@id}%
           {\number\@ff@pages@countreg,>\number\@ff@pages@countreg}%
      }%
    \fi
  }%
}
```

`\staticswitchonnextodd` Switch on the listed static frames from the next odd page onwards

```
\newcommand*{\staticswitchonnextodd}{%
  \@ifstar\@sstaticswitchonnextodd\@staticswitchonnextodd
}
```

`@sstaticswitchonnextodd` The starred version uses IDLs.

```
\newcommand{\@sstaticswitchonnextodd}[1]{%
  \count@=\@ff@pages@countreg\relax
  \ifodd\count@\relax
    \advance\count@ by 1\relax
  \fi
  \@for\@ff@id:=#1\do{%
    \@staticframeid{\@ff@id}%
```

Is this frame already on?

```
    \@sf@chckifthispg[\@ff@pages@countreg]{\ff@id}%
    \def\@ff@prepages{}%
    \if@notthiscol
    \else
       \def\@ff@prepages{\number\@ff@pages@countreg,}%
    \fi
```

Is this frame already switched on for the next page?

```
    \@sf@chckifthispg[\count@]{\ff@id}%
    \ifnum\count@=\@ff@pages@countreg\relax
    \else
      \if@notthiscol
      \else
         \edef\@ff@prepages{\@ff@prepages\number\count@,}%
      \fi
    \fi
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\staticsetpagelist{\number\ff@id}%
        {\@ff@prepages>\number\count@}%
    }%
  }%
}
```

`\@staticswitchonnextodd`

The unstarred version uses IDNs.

```
\newcommand{\@staticswitchonnextodd}[1]{%
  \count@=\@ff@pages@countreg\relax
  \ifodd\count@\relax
    \advance\count@ by 1\relax
  \fi
  \@for\@ff@id:=#1\do{%
```

Is this frame already on?

```
    \@sf@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
    \def\@ff@prepages{}%
    \if@notthiscol
    \else
       \def\@ff@prepages{\number\@ff@pages@countreg,}%
    \fi
```

Is this frame already switched on for the next page?

```
    \@sf@chckifthispg[\count@]{\@ff@id}%
    \ifnum\count@=\@ff@pages@countreg\relax
    \else
      \if@notthiscol
      \else
         \edef\@ff@prepages{\@ff@prepages\number\count@,}%
      \fi
    \fi
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\staticsetpagelist{\number\@ff@id}%
        {\@ff@prepages>\number\count@}%
    }%
  }%
}
```

`\staticswitchoffnext`

Switch off the listed static frames from the next page onwards

```
\newcommand*{\staticswitchoffnext}{%
  \@ifstar\@sstaticswitchoffnext\@staticswitchoffnext
}
```

`\@sstaticswitchoffnext`

The starred version uses IDLs.

```
\newcommand{\@sstaticswitchoffnext}[1]{%
  \@for\@ff@id:=#1\do{%
    \@staticframeid{\@ff@id}%
```

Is this frame already off on this page?

```
    \@sf@chckifthispg[\@ff@pages@countreg]{\ff@id}%
```

```
  \if@notthiscol
    \def\@ff@pages{none}%
  \else
    \def\@ff@pages{\number\@ff@pages@countreg}%
  \fi
  \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
  \xdef\@ff@output@adjustframes{%
   \the\toks@
   \noexpand\staticsetpagelist{\number\ff@id}{\@ff@pages}%
  }%
 }%
}
```

`\@staticswitchoffnext` The unstarred version uses IDNs.

```
\newcommand{\@staticswitchoffnext}[1]{%
 \@for\@ff@id:=#1\do{%
```

Is this frame already off on this page?

```
  \@sf@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
  \if@notthiscol
    \def\@ff@pages{none}%
  \else
    \def\@ff@pages{\number\@ff@pages@countreg}%
  \fi
  \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
  \xdef\@ff@output@adjustframes{%
   \the\toks@
   \noexpand\staticsetpagelist{\number\@ff@id}{\@ff@pages}%
  }%
 }%
}
```

`\staticswitchoffnextodd` Switch off the listed static frames from the next odd page onwards

```
\newcommand*{\staticswitchoffnextodd}{%
 \@ifstar\@sstaticswitchoffnextodd\@staticswitchoffnextodd
}
```

`sstaticswitchoffnextodd` The starred version uses IDLs.

```
\newcommand{\@sstaticswitchoffnextodd}[1]{%
 \count@=\@ff@pages@countreg\relax
 \ifodd\@ff@pages@countreg\relax
  \advance\count@ by 1\relax
 \fi
 \@for\@ff@id:=#1\do{%
  \@staticframeid{\@ff@id}%
```

Is this frame already off on this page?

```
  \@sf@chckifthispg[\@ff@pages@countreg]{\ff@id}%
  \if@notthiscol
```

It's off on this page. Is it on or off on the next page, if this page is odd? First, is this page odd?

```
      \ifnum\@ff@pages@countreg=\count@\relax
```

This page is even and the frame is off on this page, so set to none.

```
        \def\@ff@nextpages{none}%
      \else
```

This page is odd. Is the frame on or off on the next page?

```
        \@sf@chckifthispg[\count@]{\ff@id}%
        \if@notthiscol
```

Off on the next page as well, so set to none.

```
          \def\@ff@nextpages{none}%
        \else
```

Not off on the next page, so set to next page only.

```
          \def\@ff@nextpages{\number\count@}%
        \fi
      \fi
    \else
```

It's not off on this page. Is it on or off on the next page, if this page is odd? First, is this page odd?

```
      \ifnum\@ff@pages@countreg=\count@\relax
```

This page is even and the frame is not off on this page, so set to this page.

```
        \def\@ff@nextpages{\number\@ff@pages@countreg}%
      \else
```

This page is odd. Is the frame on or off on the next page?

```
        \@sf@chckifthispg[\count@]{\ff@id}%
        \if@notthiscol
```

Off on the next page but not off on this page. So set to just this page.

```
          \def\@ff@nextpages{\number\@ff@pages@countreg}%
        \else
```

Not off on the next page as well, so set to this page and next page.

```
          \def\@ff@nextpages{\number\@ff@pages@countreg,\number\count@}%
        \fi
      \fi

    \fi
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\staticsetpagelist{\number\ff@id}{\@ff@nextpages}%
    }%
  }%
}
```

@staticswitchoffnextodd The unstarred version uses IDNs.

```
\newcommand{\@staticswitchoffnextodd}[1]{%
  \count@=\@ff@pages@countreg\relax
  \ifodd\@ff@pages@countreg\relax
    \advance\count@ by 1\relax
  \fi
  \@for\@ff@id:=#1\do{%
```

Is this frame already off on this page?

```
    \@sf@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
    \if@notthiscol
```

It's off on this page. Is it on or off on the next page, if this page is odd? First, is this page odd?

```
      \ifnum\@ff@pages@countreg=\count@\relax
```

This page is even and the frame is off on this page, so set to none.

```
        \def\@ff@nextpages{none}%
      \else
```

This page is odd. Is the frame on or off on the next page?

```
        \@sf@chckifthispg[\count@]{\@ff@id}%
        \if@notthiscol
```

Off on the next page as well, so set to none.

```
          \def\@ff@nextpages{none}%
        \else
```

Not off on the next page, so set to next page only.

```
          \def\@ff@nextpages{\number\count@}%
        \fi
      \fi
    \else
```

It's not off on this page. Is it on or off on the next page, if this page is odd? First, is this page odd?

```
      \ifnum\@ff@pages@countreg=\count@\relax
```

This page is even and the frame is not off on this page, so set to this page.

```
        \def\@ff@nextpages{\number\@ff@pages@countreg}%
      \else
```

This page is odd. Is the frame on or off on the next page?

```
        \@sf@chckifthispg[\count@]{\@ff@id}%
        \if@notthiscol
```

Off on the next page but not off on this page. So set to just this page.

```
          \def\@ff@nextpages{\number\@ff@pages@countreg}%
        \else
```

Not off on the next page as well, so set to this page and next page.

```
          \def\@ff@nextpages{\number\@ff@pages@countreg,\number\count@}%
        \fi
      \fi
```

```
    \fi
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\staticsetpagelist{\number\@ff@id}{\@ff@nextpages}%
    }%
  }%
}
```

\staticswitchonnextonly

Switch on the listed static frames for just the next page

```
\newcommand*{\staticswitchonnextonly}{%
  \@ifstar\@sstaticswitchonnextonly\@staticswitchonnextonly
}
```

sstaticswitchonnextonly

The starred version uses IDLs.

```
\newcommand{\@sstaticswitchonnextonly}[1]{%
  \count@=\@ff@pages@countreg\relax
  \advance\count@ by 1\relax
  \@for\@ff@id:=#1\do{%
    \@staticframeid{\@ff@id}%
```

Is this frame already on?

```
    \@sf@chckifthispg[\@ff@pages@countreg]{\ff@id}%
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \if@notthiscol
```

Not, it isn't, so just set to the next page:

```
      \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\staticsetpagelist{\number\ff@id}{\number\count@}%
      }%
    \else
```

Yes, it is, so set to this page and the next page:

```
      \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\staticsetpagelist{\number\ff@id}%
          {\number\@ff@pages@countreg,\number\count@}%
      }%
    \fi
  }%
}
```

@staticswitchonnextonly

The unstarred version uses IDNs.

```
\newcommand{\@staticswitchonnextonly}[1]{%
  \count@=\@ff@pages@countreg\relax
  \advance\count@ by 1\relax
  \@for\@ff@id:=#1\do{%
```

Is this frame already on?

```
    \@sf@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \if@notthiscol
```

Not, it isn't, so just set to the next page:

```
      \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\staticsetpagelist{\number\@ff@id}{\number\count@}%
      }%
    \else
```

Yes, it is, so set to this page and the next page:

```
      \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\staticsetpagelist{\number\@ff@id}%
          {\number\@ff@pages@countreg,\number\count@}%
      }%
    \fi
  }%
}
```

aticswitchonnextoddonly

Switch on the listed static frames for just the next odd page

```
\newcommand*{\staticswitchonnextoddonly}{%
  \@ifstar\@sstaticswitchonnextoddonly\@staticswitchonnextoddonly
}
```

aticswitchonnextoddonly

The starred version uses IDLs.

```
\newcommand{\@sstaticswitchonnextoddonly}[1]{%
  \@for\@ff@id:=#1\do{%
    \@staticframeid{\@ff@id}%
```

Is this frame already on?

```
    \@sf@chckifthispg[\@ff@pages@countreg]{\ff@id}%
    \if@notthiscol
```

No, it isn't. If this is an odd page, is it on or off on the next page? First, is this an odd page?

```
      \ifodd\@ff@pages@countreg
```

Yes, it's odd. So this frame isn't on this page, but is it on or off on the next page?

```
        \count@=\@ff@pages@countreg\relax
        \advance\count@ by 1\relax
        \@sf@chckifthispg[\count@]{\ff@id}%
        \if@notthiscol
```

It's not switched on either on this (odd) page or the next (even) page. So the page list should be just the next odd page after this one.

```
         \advance\count@ by 1\relax
         \edef\@ff@pages{\number\count@}%
        \else
```

It's not switched on for this (odd) page but it is for the next (even) page. So the page list should be the next even and odd pages after this page.

```
     \edef\@ff@pages{\number\count@}%
     \advance\count@ by 1\relax
     \edef\@ff@pages{\@ff@pages,\number\count@}%
    \fi
  \else
```

No, it's even. So it's not on this (even) page, but needs to be on for the following (odd) page.

```
   \count@=\@ff@pages@countreg\relax
   \advance\count@ by 1\relax
   \edef\@ff@pages{\number\count@}%
  \fi
\else
```

Frame is on this page. If this is an odd page, is it on or off on the next page? First, is this an odd page?

```
  \ifodd\@ff@pages@countreg
```

Yes, it's odd. Is the frame on or off for the next (even) page?

```
    \count@=\@ff@pages@countreg\relax
    \advance\count@ by 1\relax
    \@sf@chckifthispg[\count@]{\ff@id}%
    \if@notthiscol
```

Frame is off. So the frame is switched on for this (odd) page but is off for the next (even) page. So the page list needs to be this (odd) page and the following odd page, skipping the even page in between.

```
      \advance\count@ by 1\relax
      \edef\@ff@pages{\number\@ff@pages@countreg,\number\count@}%
    \else
```

Frame is on. So the frame is switched on for this (odd) page and the next (even) page. So the page list needs to be this (odd) page, the next even page and the following odd page.

```
      \advance\count@ by 1\relax
      \edef\@ff@pages{\number\@ff@pages@countreg-\number\count@}%
    \fi
  \else
```

Frame is switched on for this page and this page is even. So the page list needs to be this (even) page and the next (odd) page.

```
    \count@=\@ff@pages@countreg\relax
    \advance\count@ by 1\relax
    \edef\@ff@pages{\number\@ff@pages@countreg,\number\count@}%
  \fi
```

```
\fi
\expandafter\toks@\expandafter{\@ff@output@adjustframes}%
\xdef\@ff@output@adjustframes{%
```

```
      \the\toks@
      \noexpand\staticsetpagelist{\number\ff@id}{\@ff@pages}%
    }%
  }%
}
```

aticswitchonnextoddonly

The unstarred version uses IDNs.

```
\newcommand{\@staticswitchonnextoddonly}[1]{%
  \@for\@ff@id:=#1\do{%
```

Is this frame already on?

```
    \@sf@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
    \if@notthiscol
```

No, it isn't. If this is an odd page, is it on or off on the next page? First, is this an odd page?

```
      \ifodd\@ff@pages@countreg
```

Yes, it's odd. So this frame isn't on this page, but is it on or off on the next page?

```
        \count@=\@ff@pages@countreg\relax
        \advance\count@ by 1\relax
        \@sf@chckifthispg[\count@]{\@ff@id}%
        \if@notthiscol
```

It's not switched on either on this (odd) page or the next (even) page. So the page list should be just the next odd page after this one.

```
         \advance\count@ by 1\relax
         \edef\@ff@pages{\number\count@}%
        \else
```

It's not switched on for this (odd) page but it is for the next (even) page. So the page list should be the next even and odd pages after this page.

```
         \edef\@ff@pages{\number\count@}%
         \advance\count@ by 1\relax
         \edef\@ff@pages{\@ff@pages,\number\count@}%
        \fi
      \else
```

No, it's even. So it's not on this (even) page, but needs to be on for the following (odd) page.

```
       \count@=\@ff@pages@countreg\relax
       \advance\count@ by 1\relax
       \edef\@ff@pages{\number\count@}%
      \fi
    \else
```

Frame is on this page. If this is an odd page, is it on or off on the next page? First, is this an odd page?

```
      \ifodd\@ff@pages@countreg
```

Yes, it's odd. Is the frame on or off for the next (even) page?

```
        \count@=\@ff@pages@countreg\relax
        \advance\count@ by 1\relax
        \@sf@chckifthispg[\count@]{\@ff@id}%
        \if@notthiscol
```

Frame is off. So the frame is switched on for this (odd) page but is off for the next (even) page. So the page list needs to be this (odd) page and the following odd page, skipping the even page in between.

```
          \advance\count@ by 1\relax
          \edef\@ff@pages{\number\@ff@pages@countreg,\number\count@}%
        \else
```

Frame is on. So the frame is switched on for this (odd) page and the next (even) page. So the page list needs to be this (odd) page, the next even page and the following odd page.

```
          \advance\count@ by 1\relax
          \edef\@ff@pages{\number\@ff@pages@countreg-\number\count@}%
        \fi
      \else
```

Frame is switched on for this page and this page is even. So the page list needs to be this (even) page and the next (odd) page.

```
        \count@=\@ff@pages@countreg\relax
        \advance\count@ by 1\relax
        \edef\@ff@pages{\number\@ff@pages@countreg,\number\count@}%
      \fi
```

```
    \fi
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\staticsetpagelist{\number\@ff@id}{\@ff@pages}%
    }%
  }%
}
```

staticswitchoffnextonly

Switch off the listed static frames for just the next page

```
\newcommand*{\staticswitchoffnextonly}{%
  \@ifstar\@sstaticswitchoffnextonly\@staticswitchoffnextonly
}
```

staticswitchoffnextonly

The starred version uses IDLs.

```
\newcommand{\@sstaticswitchoffnextonly}[1]{%
  \count@=\@ff@pages@countreg\relax
  \advance\count@ by 1\relax
  \@for\@ff@id:=#1\do{%
    \@staticframeid{\@ff@id}%
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
```

```
      \the\toks@
      \noexpand\staticaddexclusion{\number\ff@id}{\number\count@}%
    }%
  }%
}
```

staticswitchoffnextonly The unstarred version uses IDNs.

```
\newcommand{\@staticswitchoffnextonly}[1]{%
  \count@=\@ff@pages@countreg\relax
  \advance\count@ by 1\relax
  \@for\@ff@id:=#1\do{%
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\staticaddexclusion{\number\@ff@id}{\number\count@}%
    }%
  }%
}
```

ticswitchoffnextoddonly Switch off the listed static frames for just the next odd page

```
\newcommand*{\staticswitchoffnextoddonly}{%
  \@ifstar\@sstaticswitchoffnextoddonly\@staticswitchoffnextoddonly
}
```

ticswitchoffnextoddonly The starred version uses IDLs.

```
\newcommand{\@sstaticswitchoffnextoddonly}[1]{%
  \count@=\@ff@pages@countreg\relax
  \advance\count@ by 1\relax
  \ifodd\count@\relax
  \else
    \advance\count@ by 1\relax
  \fi
  \@for\@ff@id:=#1\do{%
    \@staticframeid{\@ff@id}%
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\staticaddexclusion{\number\ff@id}{\number\count@}%
    }%
  }%
}
```

ticswitchoffnextoddonly The unstarred version uses IDNs.

```
\newcommand{\@staticswitchoffnextoddonly}[1]{%
  \count@=\@ff@pages@countreg\relax
  \advance\count@ by 1\relax
  \ifodd\count@\relax
  \else
    \advance\count@ by 1\relax
  \fi
  \@for\@ff@id:=#1\do{%
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\staticaddexclusion{\number\@ff@id}{\number\count@}%
    }%
  }%
}
```

`ffaddtoadjustframeshook` Add stuff to the output hook.

```
  \newcommand*{\ffaddtoadjustframeshook}[1]{%
    \@ff@addtolist\@ff@output@adjustframes\entry{#1}%
  }
```

`\@g@tnextcol` Find the next flow frame. If there are no more flow frames, define a new one the size of the typeblock. (Otherwise the remaining document text will be lost.)

```
  \newif\if@notthiscol
  \newif\if@ff@nwpg
  \newcount\c@curpg
  \newcommand*{\@g@tnextcol}[1]{%
```

Do any frame adjustments

```
    \@ff@output@adjustframes
```

Now clear the hook

```
    \global\let\@ff@output@adjustframes\@empty
```

Now check for any more frames.

```
    \@ff@checkifmoreframes
    \if@ff@moreframes
    \else
```

No more frames, add new frame

```
      \PackageWarning{flowfram}%
      {Run out of flows frames on page \number\@ff@pages@countreg, adding new one}%
      \flf@doifverbose
      {%
        \def\flf@messinfo{Here's the list of flow frames:}%
        \count@=0\relax
        \loop
          \advance\count@ by 1\relax
          \expandafter\toks@\expandafter{\flf@messinfo\MessageBreak}%
          \edef\flf@messinfo{\the\toks@
            \number\count@.
             Pages: \csname @ff@pages@\romannumeral\count@\endcsname.
             Exclusions: \csname @ff@xpages@\romannumeral\count@\endcsname.
          }%
        \ifnum\count@<\c@maxflow
        \repeat
        \PackageInfo{flowfram}{\flf@messinfo\@gobbletwo}%
```

```
    }%
    \@onecolumn
    #1=\c@maxflow
  \fi
  \@notthiscoltrue
  \@ff@nwpgfalse
  \@colN=#1\relax

  \c@curpg=\@ff@pages@countreg
  \loop
    \ifnum\@colN=\c@maxflow
```

Reached the end of the page. Try the next one.

```
      \@colN=1\relax
      \@ff@nwpgtrue
      \advance\c@curpg by 1\relax
    \else
```

Move on to the next flow frame on this page.

```
      \advance\@colN by 1\relax
    \fi
    \@ff@chckifthispg{\c@curpg}{\@colN}%
  \if@notthiscol
  \repeat
  #1=\@colN\relax
}
```

`\@ff@chckifthispg` This is used to determine the next flow frame, since not all flow frames may be defined on every page. Checks to see if flow frame #2 is defined on page #1. First set up some variables.

```
\newcommand*{\@ff@chckifthispg}[2]{%
  \@notthiscolfalse
  \edef\ff@xpages{\csname @ff@xpages@\romannumeral#2\endcsname}%
  \@for\@ff@pp:=\ff@xpages\do
  {%
    \ifnum0\@ff@pp=#1\relax
      \@notthiscoltrue
      \@endfortrue
    \fi
  }%
  \if@notthiscol
  \else
    \@notthiscoltrue
    \edef\ff@pages{\csname @ff@pages@\romannumeral#2\endcsname}%
    \@@ff@chckifthispg{#1}%
  \fi
}
```

`\@@ff@chckifthispg` Now go ahead and check.

```
\newcommand*{\@@ff@chckifthispg}[1]{%
```

```
  \ifthenelse{\equal{\ff@pages}{none}}%
  {}%
  {%
    \ifthenelse{\equal{\ff@pages}{all}}%
    {%
      \@notthiscolfalse
    }%
    {%
      \ifthenelse{\equal{\ff@pages}{odd}}%
      {%
        \ifodd#1\@notthiscolfalse\fi
      }%
      {%
        \ifthenelse{\equal{\ff@pages}{even}}%
        {%
          \ifodd#1\else\@notthiscolfalse\fi
        }%
        {%
```

check through list of page numbers

```
          \@for\@ff@pp:=\ff@pages\do{%
            \def\@ff@numstart{0}%
            \def\@ff@numend{0}%
            \@ff@getrange{\@ff@pp}%
            \ifthenelse{#1<\@ff@numstart \or #1>\@ff@numend}%
            {}%
            {%
              \@notthiscolfalse
            }%
          }%
        }%
      }%
    }%
  }%
}
```

`\@sf@chckifthispg`

Checks to see if static frame #1 is defined on the current page (or the page given by the optional argument).

```
\newcommand*{\@sf@chckifthispg}[2][\@ff@pages@countreg]{%
  \@notthiscoltrue
  \edef\ff@pages{\csname @sf@pages@\romannumeral#2\endcsname}%

  \@@ff@chckifthispg{#1}%
}
```

`\@df@chckifthispg`

Checks to see if dynamic frame #1 is defined on the current page (or the page given by the optional argument).

```
\newcommand*{\@df@chckifthispg}[2][\@ff@pages@countreg]{%
  \@notthiscoltrue
  \edef\ff@pages{\csname @df@pages@\romannumeral#2\endcsname}%
```

```
    \@@ff@chckifthispg{#1}%
  }
```

\@setcolbox Sets the TEX box defining the flow frame to the output box. This saves the output until the page is shipped out after all the flow frames have been filled for that page.

```
  \newcommand*{\@setcolbox}[1]{%
    \flf@message{Setting contents of box for flow frame \number#1}%
    \expandafter\global\expandafter\setbox
      \csname column\romannumeral#1\endcsname\box\@outputbox
  }
```

\@docolbox Put flow frame on the page with the correct border, if it has one.

```
  \newcommand*{\@docolbox}[1]{%
    \flf@message{Doing flow frame \number#1\space
      (page \number\@ff@pages@countreg)}%
    \edef\ff@frametype{%
      \csname @ff@frametype@\romannumeral#1\endcsname}%
```

Frame colour

```
    \edef\ff@col{\csname @ff@col@\romannumeral#1\endcsname}%
```

Text colour

```
    \edef\ff@txtcol{\csname @ff@txtcol@\romannumeral#1\endcsname}%
```

Background colour

```
    \edef\ff@backcol{\csname @ff@backcol@\romannumeral#1\endcsname}%
```

Compute offset for this frame

```
    \@ff@setoffset{#1}%
```

Rotate frame if required

```
    \rotateframe{\csname @ff@angle@\romannumeral#1\endcsname}%
    {%
```

Check if frame has a border

```
      \ifthenelse{\boolean{columnframe\romannumeral#1}}%
      {%
  %   \end{macrocode}
  % Put the required border around the frame
  %   \begin{macrocode}
         \@ff@fbox
           {\csname colwidth\romannumeral#1\endcsname}%
           {\csname colheight\romannumeral#1\endcsname}%
           {%
             \expandafter\box\csname column\romannumeral#1\endcsname
           }%
           {%
             \csname\ff@frametype\endcsname
           }%
      }%
      {%
```

Do the frame without a border

```
      \@ff@box
        {\csname colwidth\romannumeral#1\endcsname}%
        {\csname colheight\romannumeral#1\endcsname}%
        {%
           \expandafter\box\csname column\romannumeral#1\endcsname
        }%
    }%
  }%
}
```

`\@docolbbox` Do the bounding box for given flow frame.

```
\newcommand*{\@docolbbox}[1]{%
  \@ff@setoffset{#1}%
  \def\ff@col{}\def\ff@txtcol{}%
  \@fr@meifdraft
  {%
    \@ff@box
      {\csname colwidth\romannumeral#1\endcsname}%
      {\csname colheight\romannumeral#1\endcsname}%
      {%
        \expandafter\box\csname column\romannumeral#1\endcsname
      }%
  }%
  {F:\number#1;\csname @col@id@\romannumeral#1\endcsname}%
}
```

`\@ff@fbox` Put the TeX box #3 of width #1 and height #2, and frame making command specified by #4.

```
\newcommand{\@ff@fbox}[4]{%
  {%
    \fboxsep=\flowframesep
    \fboxrule=\flowframerule
    \@s@tffcol
    \kern\@ff@offset
    #4{\@ff@box{#1}{#2}{#3}}%
  }%
}
```

`\@ff@box` Put the TeX box #3 of width #1 and height #2 on the page.

```
\newcommand{\@ff@box}[3]{%
  {%
    \@ffbackground
    {%
      \vbox to#2 {\hb@xt@ #1{\hss{\@s@tfftextcol #3}\hss}\vss\kern\z@}%
    }%
  }%
}
```

`\@putcolbox` Display the flow frame on the page, at its given position. If the document is two-sided, need to check whether the current page is odd or even to determine the correct location.

```
\newcommand*{\@putcolbox}[1]{%

  \@ff@chckifthispg{\@ff@pages@countreg}{#1}%

  \if@notthiscol
    \expandafter\ifvoid\csname column\romannumeral#1\endcsname
    \else
      \PackageWarning{flowfram}{Box \number#1\space is not void.
       Dumping. This page: \number\@ff@pages@countreg.
       Page list: "\csname @ff@pages@\romannumeral#1\endcsname".
       Exclusion list: "\csname @ff@xpages@\romannumeral#1\endcsname".
       (Maybe the page list was changed after this frame was
       selected or maybe you should use package option pages=absolute)}%
      \@notthiscolfalse
    \fi
  \fi
  \if@notthiscol
    \flf@message{Flow frame \number#1\space is not required on page
      \number\@ff@pages@countreg}%
  \else
    \@killglue
    \if@twoside
      \ifodd\c@page
        \expandafter\raise\csname col@\romannumeral#1@posy\endcsname
        \hb@xt@\z@
        {%
          \expandafter\kern \csname col@\romannumeral#1@posx\endcsname
          \@docolbox{#1}\hss
        }%
      \else
        \expandafter\raise\csname col@\romannumeral#1@eveny\endcsname
        \hb@xt@\z@
        {%
          \expandafter\kern \csname col@\romannumeral#1@evenx\endcsname
          \@docolbox{#1}\hss
        }%
      \fi
    \else
      \expandafter\raise\csname col@\romannumeral#1@posy\endcsname
      \hb@xt@\z@
      {%
        \expandafter\kern \csname col@\romannumeral#1@posx\endcsname
        \@docolbox{#1}\hss
      }%
    \fi
  \fi
}
```

\@putcolbbox

Same for flow frame bounding box:

```
\newcommand*{\@putcolbbox}[1]{%

  \@ff@chckifthispg{\@ff@pages@countreg}{#1}%
  \if@notthiscol
  \else
    \@killglue
    \if@twoside
      \ifodd\c@page
        \expandafter\raise\csname col@\romannumeral#1@posy\endcsname
          \hb@xt@\z@
          {%
            \expandafter\kern \csname col@\romannumeral#1@posx\endcsname
            \@docolbbox{#1}\hss
          }%
      \else
        \expandafter\raise\csname col@\romannumeral#1@eveny\endcsname
        \hb@xt@\z@
        {%
          \expandafter\kern \csname col@\romannumeral#1@evenx\endcsname
          \@docolbbox{#1}\hss
        }%
      \fi
    \else
     \expandafter\raise\csname col@\romannumeral#1@posy\endcsname
     \hb@xt@\z@
     {%
       \expandafter\kern \csname col@\romannumeral#1@posx\endcsname
       \@docolbbox{#1}\hss
     }%
    \fi
  \fi
}
```

If an offset hasn't been specified, compute it. If the frame making command is known (e.g. doublebox), compute the offset according to known specifications, otherwise set the negative offset to `\flowframesep` plus `\flowframerule`, which may or may not be correct.

@ff@s@t@doubleboxoffset

Compute offset for `\doublebox`:

```
\newcommand*{\@ff@s@t@doubleboxoffset}{%
  \setlength{\@ff@offset}{-\flowframesep}%
  \addtolength{\@ff@offset}{-3.75\flowframerule}%
  \addtolength{\@ff@offset}{-.5pt}%
}
```

\@ff@s@t@ovalboxoffset

Compute offset for `\ovalbox`:

```
\newcommand*{\@ff@s@t@ovalboxoffset}{%
  \@ff@offset=-\fontdimen 8\tenln\relax
  \advance\@ff@offset by -\flowframesep\relax
}
```

\@ff@s@t@Ovalboxoffset Compute offset for `\ovalbox`:

```
\newcommand*{\@ff@s@t@Ovalboxoffset}{%
  \@ff@offset=-\fontdimen 8\tenlnw\relax
  \advance\@ff@offset by -\flowframesep\relax
}
```

\@ff@s@t@defaultoffset Compute default offset:

```
\newcommand*{\@ff@s@t@defaultoffset}{%
  \@ff@offset=-\flowframesep\relax
  \addtolength{\@ff@offset}{-\flowframerule}%
}
```

\@ff@setoffset Compute offset for flow frame #1. Stores offset value in `\ff@offset`.

```
\newcommand*{\@ff@setoffset}[1]{%
  \ifthenelse
    {\equal{\csname @ff@offset@\romannumeral#1\endcsname}{compute}}%
  {%
    \ifthenelse{\boolean{columnframe\romannumeral#1}}%
    {%
      \ifthenelse
      {%
        \equal{\csname @ff@frametype@\romannumeral#1\endcsname}%
              {doublebox}%
      }%
      {%
        \@ff@s@t@doubleboxoffset
      }%
      {%
        \ifthenelse
        {%
          \equal{\csname @ff@frametype@\romannumeral#1\endcsname}%
                {ovalbox}%
        }%
        {%
          \@ff@s@t@ovalboxoffset
        }%
        {%
          \ifthenelse
          {%
            \equal{\csname @ff@frametype@\romannumeral#1\endcsname}%
                  {Ovalbox}%
          }%
          {%
            \@ff@s@t@Ovalboxoffset
          }%
          {%
```

```
              \@ff@s@t@defaultoffset
            }%
          }%
        }%
      }%
      {}%
    }%
    {%
      \setlength{\@ff@offset}%
      {\csname @ff@offset@\romannumeral#1\endcsname}%
    }%
  }
```

`\@sf@setoffset` Compute offset for static frame #1. Stores offset value in `\ff@offset`.

```
  \newcommand*{\@sf@setoffset}[1]{%
    \ifthenelse
    {%
      \equal{\csname @sf@offset@\romannumeral#1\endcsname}%
            {compute}%
    }%
    {%
      \ifthenelse{\boolean{staticframe\romannumeral#1}}%
      {%
        \ifthenelse
        {%
          \equal{\csname @sf@frametype@\romannumeral#1\endcsname}%
                {doublebox}%
        }%
        {%
          \@ff@s@t@doubleboxoffset
        }%
        {%
          \ifthenelse
          {%
            \equal{\csname @sf@frametype@\romannumeral#1\endcsname}%
                  {ovalbox}%
          }%
          {%
            \@ff@s@t@ovalboxoffset
          }%
          {%
            \ifthenelse
            {%
              \equal{\csname @sf@frametype@\romannumeral#1\endcsname}%
                    {Ovalbox}%
            }%
            {%
              \@ff@s@t@Ovalboxoffset
            }%
            {%
              \@ff@s@t@defaultoffset
            }%
          }%
        }%
      }%
      {}%
    }%
    {%
      \setlength{\@ff@offset}%
      {\csname @sf@offset@\romannumeral#1\endcsname}%
    }%
  }
```

`\@df@setoffset` Compute offset for dynamic frame #1. Stores offset value in `\ff@offset`.

```
  \newcommand*{\@df@setoffset}[1]{%
    \ifthenelse
    {%
      \equal{\csname @df@offset@\romannumeral#1\endcsname}%
            {compute}%
    }%
    {%
      \setlength{\@ff@offset}{0pt}%
      \ifthenelse{\boolean{dynamicframe\romannumeral#1}}%
      {%
        \ifthenelse
        {%
          \equal{\csname @df@frametype@\romannumeral#1\endcsname}%
                {doublebox}%
        }%
        {%
          \@ff@s@t@doubleboxoffset
        }%
        {%
          \ifthenelse
          {%
            \equal{\csname @df@frametype@\romannumeral#1\endcsname}%
                  {ovalbox}%
          }%
          {%
            \@ff@s@t@ovalboxoffset
          }%
          {%
            \ifthenelse
            {%
              \equal{\csname @df@frametype@\romannumeral#1\endcsname}%
                    {Ovalbox}%
            }%
            {%
              \@ff@s@t@Ovalboxoffset
            }%
            {%
              \@ff@s@t@defaultoffset
            }%
          }%
        }%
      }%
      {}%
    }%
    {%
      \setlength{\@ff@offset}%
      {\csname @df@offset@\romannumeral#1\endcsname}%
    }%
  }
```

`\@putmarginbox` Draw box representing the margin for flow frame #1.

```
\newcommand*{\@putmarginbox}[1]{%

  \@ff@chckifthispg{\@ff@pages@countreg}{#1}%
  \if@notthiscol
  \else
    \@killglue
    \if@twoside
      \ifodd\c@page
        \edef\ff@x{\csname col@\romannumeral#1@posx\endcsname}%
        \edef\ff@y{\csname col@\romannumeral#1@posy\endcsname}%
      \else
        \edef\ff@x{\csname col@\romannumeral#1@evenx\endcsname}%
        \edef\ff@y{\csname col@\romannumeral#1@eveny\endcsname}%
      \fi
    \else
      \edef\ff@x{\csname col@\romannumeral#1@posx\endcsname}%
      \edef\ff@y{\csname col@\romannumeral#1@posy\endcsname}%
    \fi
    \setlength{\@ff@tmp@x}{\ff@x}%
    \setlength{\@ff@tmp@y}{\ff@y}%
    \@getmarginpos{\csname @ff@margin@\romannumeral#1\endcsname}%
    \ifthenelse{\equal{\ff@margin}{left}}%
    {%
      \addtolength{\@ff@tmp@x}{-\marginparwidth}%
      \addtolength{\@ff@tmp@x}{-\marginparsep}%
      \ifthenelse{\boolean{columnframe\romannumeral#1}}%
      {}%
      {}%
    }%
    {%
      \addtolength{\@ff@tmp@x}%
      {\csname colwidth\romannumeral#1\endcsname}%
      \addtolength{\@ff@tmp@x}{\marginparsep}%
```

```
        \ifthenelse{\boolean{columnframe\romannumeral#1}}%
        {}%
        {}%
      }%
      \raise\@ff@tmp@y
      \hb@xt@\z@
      {%
        \expandafter\kern\@ff@tmp@x
        \@fr@meifdraft{\@ff@box{\marginparwidth}%
        {\csname colheight\romannumeral#1\endcsname}{}}%
        {M:\number#1}\hss
      }%
    \fi
    \ignorespaces
  }
```

`\@ff@drawmargins` Draw all the margins associated with the flow frames defined on the current page.

```
  \newcommand*{\@ff@drawmargins}{%
    \@colN=0\relax
    \whiledo{\@colN<\c@maxflow}%
    {%
      \advance\@colN by 1\relax
      \makebox[0pt][l]{\@putmarginbox{\@colN}}%
    }%
  }
```

`\@ff@getstaticpos` Extract the width and height for static or dynamic frame specified in the form `[`⟨*c*⟩`][`⟨*height*⟩`][`⟨*valign*⟩`]{`⟨*width*⟩`}`

```
  \def\@ff@getstaticpos[#1][#2][#3]#4{%
    \@ff@tmp@x=#4\relax
    \@ff@tmp@y=#2\relax
    \def\ff@valign{#3}%
  }
```

`\@dostaticbox` Display the savebox associated with static frame `#1`

```
  \newcommand*{\@dostaticbox}[1]{%
    \edef\ff@frametype{%
      \csname @sf@frametype@\romannumeral#1\endcsname
    }%
    \edef\ff@col{\csname @sf@col@\romannumeral#1\endcsname}%
    \edef\ff@backcol{\csname @sf@backcol@\romannumeral#1\endcsname}%
    \@sf@setoffset{#1}%
    \expandafter\expandafter\expandafter
      \@ff@getstaticpos\csname @sf@dim@\romannumeral#1\endcsname
    \rotateframe
      {\csname @sf@angle@\romannumeral#1\endcsname}%
      {%
        \ifthenelse{\boolean{staticframe\romannumeral#1}}%
        {%
          \@ff@fbox{\@ff@tmp@x}{\@ff@tmp@y}%
          {%
            \expandafter\usebox\csname @staticframe@\romannumeral#1\endcsname
          }
          {\csname\ff@frametype\endcsname}%
        }%
        {%
          \@ff@box{\@ff@tmp@x}{\@ff@tmp@y}%
          {%
            \expandafter\usebox\csname @staticframe@\romannumeral#1\endcsname
          }%
        }%
     }%
  }
```

`\@dostaticbbox` Now for the bounding box:

```
\newcommand*{\@dostaticbbox}[1]{%
  \edef\ff@col{}%
  \@sf@setoffset{#1}%
  \expandafter\expandafter\expandafter
    \@ff@getstaticpos\csname @sf@dim@\romannumeral#1\endcsname
  \@fr@meifdraft
  {%
    \@ff@box{\@ff@tmp@x}{\@ff@tmp@y}%
    {%
      \expandafter\usebox\csname @staticframe@\romannumeral#1\endcsname
    }%
  }%
  {S:\number#1;\csname @sf@id@\romannumeral#1\endcsname}%
}
```

`\@putstaticbox` Put the static box #1 at its given position, with its associated border.

```
\newcommand*{\@putstaticbox}[1]{%
%\changes{2014-06-04}{1.16}{added check for ‘hide’ and ‘hidethis’
%attributes}
% Check the ‘hide’ and ‘hidethis’ attributes
%    \begin{macrocode}
  \ifthenelse{\boolean{@sf@hidethis@\romannumeral#1}}%
  {%
    \@notthiscoltrue
    \global\csletcs{if@sf@hidethis@\romannumeral#1}{iffalse}%
  }%
  {%
    \ifthenelse{\boolean{@sf@hide@\romannumeral#1}}%
    {%
      \@notthiscoltrue
    }%
    {%
```

Neither ‘hide’ nor ‘hidethis’ have been set so check the page list.

```
      \@sf@chckifthispg{#1}%
    }%
  }%
  \if@notthiscol
  \else
    \@killglue
    \if@twoside
      \ifodd\c@page
        \expandafter\raise\csname @sf@\romannumeral#1@posy\endcsname
        \hb@xt@\z@
        {%
          \expandafter\kern \csname @sf@\romannumeral#1@posx\endcsname
          \@dostaticbox{#1}\hss
        }%
      \else
        \expandafter\raise\csname @sf@\romannumeral#1@eveny\endcsname
        \hb@xt@\z@
        {%
          \expandafter\kern \csname @sf@\romannumeral#1@evenx\endcsname
          \@dostaticbox{#1}\hss
        }%
      \fi
    \else
      \expandafter\raise\csname @sf@\romannumeral#1@posy\endcsname
      \hb@xt@\z@
      {%
        \expandafter\kern \csname @sf@\romannumeral#1@posx\endcsname
        \@dostaticbox{#1}\hss
      }%
  \fi
\fi
}
```

`\@putstaticbbox`

Now for the bounding box:

```
\newcommand*{\@putstaticbbox}[1]{%
  \@sf@chckifthispg{#1}%
  \if@notthiscol
  \else
    \@killglue
    \if@twoside
      \ifodd\c@page
        \expandafter\raise\csname @sf@\romannumeral#1@posy\endcsname
        \hb@xt@\z@
        {%
          \expandafter\kern \csname @sf@\romannumeral#1@posx\endcsname
          \@dostaticbbox{#1}\hss
        }%
        \ignorespaces
      \else
        \expandafter\raise\csname @sf@\romannumeral#1@eveny\endcsname
        \hb@xt@\z@
        {%
          \expandafter\kern \csname @sf@\romannumeral#1@evenx\endcsname
          \@dostaticbbox{#1}\hss
        }%
        \ignorespaces
      \fi
    \else
      \expandafter\raise\csname @sf@\romannumeral#1@posy\endcsname
      \hb@xt@\z@
      {%
        \expandafter\kern \csname @sf@\romannumeral#1@posx\endcsname
        \@dostaticbbox{#1}\hss
      }%
      \ignorespaces
    \fi
  \fi
}
```

`\@resetst@tics` Clear the contents of all the static frames that have the `clear` option set.

```
\newcommand*{\@resetst@tics}{%
  \@colN=0\relax
  \whiledo{\@colN<\c@maxstatic}%
  {%
    \advance\@colN by 1\relax
```

Has the clear flag been set?

```
    \ifthenelse{\boolean{@sf@clear@\romannumeral\@colN}}%
    {%
```

Set the contents of the box to empty

```
      \global\sbox
      {%
        \csname @staticframe@\romannumeral\@colN\endcsname
      }%
      {}%
    }%
    {}%
  }%
}
```

`\@resetdyn@mics` Clear the contents of the dynamic frames that have the `clear` option set.

```
\newcommand*{\@resetdyn@mics}{%
  \@colN=0\relax
  \whiledo{\@colN<\c@maxdynamic}%
  {%
    \advance\@colN by 1\relax
    \ifthenelse{\boolean{@df@clear@\romannumeral\@colN}}%
```

```
      {%
        \expandafter\global\expandafter
        \gdef\csname @dynamicframe@\romannumeral\@colN\endcsname{}%
      }%
      {}%
    }%
  }
```

`\@dodfparbox` Display contents of dynamic box (contents stored in `\ff@contents`, style given by `\ff@style`):

```
  \newcommand*{\@dodfparbox}[1]{%
    \expandafter\let\expandafter
      \@ff@parshape\csname @df@shape@\romannumeral#1\endcsname
    \expandafter\@ff@getshape\@ff@parshape\relax
    \ifcase\ff@shape
```

no shape

```
      \expandafter\expandafter\expandafter
      \parbox\csname @df@dim@\romannumeral#1\endcsname
      {%
        \setlength\parindent\sdfparindent
        \csname\ff@style\endcsname{\ff@contents}%
      }%
    \or
```

`\parshape`

```
      \expandafter\expandafter\expandafter
      \parbox\csname @df@dim@\romannumeral#1\endcsname
      {%
        \setlength\parindent\sdfparindent
        \csname\ff@style\endcsname
        {{%
          \let\oldpar=\par
          \let\par=\ffpshpar
          \@ff@setsecthead
          \@ff@parshape
          \ff@contents\oldpar
        }}%
      }%
    \or
```

`\shapepar`

```
      \expandafter\expandafter\expandafter
      \parbox\csname @df@dim@\romannumeral#1\endcsname
      {%
        \setlength\parindent\sdfparindent
        \csname\ff@style\endcsname
        {{%
          \@ff@disablesec\@ff@parshape
          \ff@contents\par
      }}%
    }%
  \fi
}
```

`\@dodynamicbox`

Typeset the dynamic box with its associated border.

```
\newcommand*{\@dodynamicbox}[1]{%
  \edef\ff@frametype{%
    \csname @df@frametype@\romannumeral#1\endcsname
  }%
  \edef\ff@col{\csname @df@col@\romannumeral#1\endcsname}%
  \edef\ff@txtcol{\csname @df@txtcol@\romannumeral#1\endcsname}%
  \edef\ff@backcol{\csname @df@backcol@\romannumeral#1\endcsname}%
  \edef\ff@style{\csname @df@style@\romannumeral#1\endcsname}%
  \def\ff@contents{\csname @dynamicframe@\romannumeral#1\endcsname}%
  \@df@setoffset{#1}%
  \expandafter\expandafter\expandafter
    \@ff@getstaticpos\csname @df@dim@\romannumeral#1\endcsname
  \rotateframe{\csname @df@angle@\romannumeral#1\endcsname}%
  {%
    \ifthenelse{\boolean{dynamicframe\romannumeral#1}}%
    {%
      \@ff@fbox{\@ff@tmp@x}{\@ff@tmp@y}%
      {\@dodfparbox{#1}}%
      {\csname\ff@frametype\endcsname}%
    }%
    {%
      \@ff@box{\@ff@tmp@x}{\@ff@tmp@y}%
      {%
        \@dodfparbox{#1}%
      }%
    }%
  }%
}
```

`\@dodynamicbbox`

Now for the bounding box:

```
\newcommand*{\@dodynamicbbox}[1]{%
  \edef\ff@col{}%
  \@df@setoffset{#1}%
  \expandafter\expandafter\expandafter
  \@ff@getstaticpos\csname @df@dim@\romannumeral#1\endcsname
  \@fr@meifdraft
  {%
    \@ff@box{\@ff@tmp@x}{\@ff@tmp@y}%
    {%
      \expandafter\expandafter\expandafter
      \parbox\csname @df@dim@\romannumeral#1\endcsname
      {}%
    }%
  }%
  {D:\number#1;\csname @df@id@\romannumeral#1\endcsname}%
}
```

`\@putdynamicbox` Put the dynamic frame #1 at its given position

```
\newcommand*{\@putdynamicbox}[1]{%
```

Check the 'hide' and 'hidethis' attributes

```
  \ifthenelse{\boolean{@df@hidethis@\romannumeral#1}}%
  {%
    \@notthiscoltrue
    \global\csletcs{if@df@hidethis@\romannumeral#1}{iffalse}%
  }%
  {%
    \ifthenelse{\boolean{@df@hide@\romannumeral#1}}%
    {%
      \@notthiscoltrue
    }%
    {%
```

Neither 'hide' nor 'hidethis' have been set so check the page list.

```
      \@df@chckifthispg{#1}%
    }%
  }%
  \if@notthiscol
  \else
    \@killglue
    \if@twoside
      \ifodd\c@page
        \expandafter\raise\csname @df@\romannumeral#1@posy\endcsname
        \hb@xt@\z@
        {%
          \expandafter\kern \csname @df@\romannumeral#1@posx\endcsname
          \@dodynamicbox{#1}\hss
        }%
        \ignorespaces
      \else
        \expandafter\raise\csname @df@\romannumeral#1@eveny\endcsname
        \hb@xt@\z@
        {%
          \expandafter\kern \csname @df@\romannumeral#1@evenx\endcsname
          \@dodynamicbox{#1}\hss
        }%
        \ignorespaces
      \fi
    \else
      \expandafter\raise\csname @df@\romannumeral#1@posy\endcsname
      \hb@xt@\z@
      {%
        \expandafter\kern \csname @df@\romannumeral#1@posx\endcsname
        \@dodynamicbox{#1}\hss
      }%
      \ignorespaces
    \fi
  \fi
 }
```

`\@putdynamicbbox` Bounding box:

```
\newcommand*{\@putdynamicbbox}[1]{%
 \@df@chckifthispg{#1}%
 \if@notthiscol
 \else
  \@killglue
  \if@twoside
   \ifodd\c@page
    \expandafter\raise\csname @df@\romannumeral#1@posy\endcsname
    \hb@xt@\z@
    {%
     \expandafter\kern \csname @df@\romannumeral#1@posx\endcsname
     \@dodynamicbbox{#1}\hss
    }%
    \ignorespaces
   \else
    \expandafter\raise\csname @df@\romannumeral#1@eveny\endcsname
    \hb@xt@\z@
    {%
     \expandafter\kern \csname @df@\romannumeral#1@evenx\endcsname
     \@dodynamicbbox{#1}\hss
    }%
    \ignorespaces
   \fi
  \else
   \expandafter\raise\csname @df@\romannumeral#1@posy\endcsname
   \hb@xt@\z@
   {%
    \expandafter\kern \csname @df@\romannumeral#1@posx\endcsname
    \@dodynamicbbox{#1}\hss
   }%
   \ignorespaces
  \fi
 \fi
}
```

`\@@doheader` Do standard header in the standard place.

```
\newcommand*{\@@doheader}{%
 \setlength\@ff@tmp@y{\textheight}%
 \addtolength{\@ff@tmp@y}{\headsep}%
 \def\ff@col{}%
 \def\ff@txtcol{}%
```

```
    \def\ff@backcol{{none}}%
    \@ff@box{0pt}{\@ff@tmp@y}{\makebox[0pt][l]{\@dothehead}}%
  }
```

`\@@dofooter` Do standard footer in the standard place.

```
  \newcommand*{\@@dofooter}{%
    \setlength\@ff@tmp@y{-\footskip}%
    \def\ff@col{}%
    \def\ff@txtcol{}%
    \def\ff@backcol{{none}}%
    \@ff@box{0pt}{\@ff@tmp@y}{\makebox[0pt][l]{\@dothefoot}}%
  }
```

`\@s@tfr@mes` This is a modified version of the way the picture environment works:

```
  \newcommand{\@s@tfr@mes}[1]{%
    {%
      \@picht\textheight
      \setbox\@picbox\hb@xt@ \textwidth
      \bgroup
        \hbox
          \bgroup
            #1\relax
          \egroup
        \hss
      \egroup
      \ht\@picbox\@picht
      \dp\@picbox\z@
      \mbox{\box\@picbox}%
    }%
  }
```

`\@ff@doallflowframes` Puts all the flow frames defined on the current page

```
  \newcommand*{\@ff@doallflowframes}{%
    \@colN=0\relax
    \whiledo{\@colN<\c@maxflow}%
    {%
      \advance\@colN by 1\relax
      \@putcolbox{\@colN}%
    }%
  }
```

`@ff@doallflowframesbbox` Flow frame bounding boxes:

```
  \newcommand*{\@ff@doallflowframesbbox}{%
    \@colN=0\relax
    \whiledo{\@colN<\c@maxflow}%
    {%
      \advance\@colN by 1\relax
      \@putcolbbox{\@colN}%
    }%
```

```
}
```

`\@ff@doallstatics` Puts all static frames defined on the current page

```
\newcommand*{\@ff@doallstatics}{%
  \@colN=0\relax
  \whiledo{\@colN<\c@maxstatic}%
  {%
    \advance\@colN by 1\relax
    \@putstaticbox{\@colN}%
  }%
}
```

`\@ff@doallstaticsbbox` Static frame bounding boxes:

```
\newcommand*{\@ff@doallstaticsbbox}{%
  \@colN=0\relax
  \whiledo{\@colN<\c@maxstatic}%
  {%
    \advance\@colN by 1\relax
    \@putstaticbbox{\@colN}%
  }%
}
```

`\@ff@doalldynamics` Puts all the dynamic frames defined on the current page

```
\newcommand*{\@ff@doalldynamics}{%
  \@colN=0\relax
  \whiledo{\@colN<\c@maxdynamic}%
  {%
    \advance\@colN by 1\relax
    \@putdynamicbox{\@colN}%
  }%
}
```

`\@ff@doalldynamicsbbox` Dynamic frame bounding boxes:

```
\newcommand*{\@ff@doalldynamicsbbox}{%
  \@colN=0\relax
  \whiledo{\@colN<\c@maxdynamic}%
  {%
    \advance\@colN by 1\relax
    \@putdynamicbbox{\@colN}%
  }%
}
```

`\@ff@dotypeblock` Draw typeblock frame if draft.

```
\newcommand*{\@ff@dotypeblock}{%
  \makebox[0pt][l]%
  {%
    \@fr@meifdraft[\setffdrafttypeblockcolor]%
    {%
      \vbox to \textheight{\hbox to \textwidth{}}%
    }%
    {}%
  }%
}
```

`\@ff@do@allframes` Put all frames defined on the current page.

```
\newlength\ffevenoffset
\newcommand*{\@ff@do@allframes}{%
  \ffevenoffset=0pt\relax
  \if@twoside
    \ifodd\c@page
    \else
      \ffevenoffset=-\oddsidemargin\relax
      \advance\ffevenoffset by \evensidemargin\relax
      \kern\ffevenoffset\relax
    \fi
  \fi
  \setlength{\@ff@tmp@x}{\textwidth}%
  \advance\@ff@tmp@x by -\ffevenoffset\relax
  \makebox[\@ff@tmp@x][l]%
  {%
    \@s@tfr@mes
    {%
      \@ff@doallstatics
      \@@doheader
      \@@dofooter
      \@ff@doallflowframes
      \@ff@doalldynamics
      \ifshowtypeblock
        \@ff@dotypeblock
      \fi
      \ifshowframebbox
        \@ff@doallstaticsbbox
        \@ff@doallflowframesbbox
        \@ff@doalldynamicsbbox
      \fi
      \ifshowmargins
        \@ff@drawmargins
      \fi
    }%
  }%
}
```

`\@outputdblcol` This was modified from the output routine for standard two column format. After `\@g@tnextcol`, the register `\c@curpg` contains the page that the next flow frame is on. If `\c@curpg` minus `\c@page` is greater than 1, then there is at least one page without a flow frame. These pages will have to be shipped before TEX can continue with the rest of the document.

```
\newcount\@nxtcol
```

```
\def\@outputdblcol{%
  \@nxtcol=\c@thisframe

  \c@curpg=\@ff@pages@countreg
  \@g@tnextcol{\@nxtcol}%
  \if@ff@nwpg
```

Next flow frame starts on new page.

```
    \global\@firstcolumntrue
    \@setcolbox\c@thisframe
    \if@specialpage
      \global\@specialpagefalse
      \@nameuse{ps@\@specialstyle}\relax
    \fi
    \if@twoside
      \ifodd\count\z@
        \let\@thehead\@oddhead
        \let\@thefoot\@oddfoot
      \else
        \let\@thehead\@evenhead
        \let\@thefoot\@evenfoot
      \fi
    \else
      \let\@thehead\@oddhead
      \let\@thefoot\@oddfoot
    \fi
    \@begindvi
    \@dodynamicthehead\@dodynamicthefoot
    \vbadness=\@M
    \setbox\@outputbox\vbox{\hbox to \textwidth{\@ff@do@allframes}}%
    \@combinedblfloats
    \@outputpage
```

Shipout pages without flow frames.

```
    \advance\c@curpg by -\@ff@pages@countreg\relax
    \whiledo{\c@curpg>0}%
    {%
      \advance\c@curpg by -1\relax
      \setbox\@outputbox\vbox{\hbox to \textwidth{\@ff@do@allframes}}%
      \@outputpage
    }
    \begingroup
      \@dblfloatplacement
      \@startdblcolumn
      \@whilesw \if@fcolmade \fi
         {\@outputpage \@startdblcolumn }%
    \endgroup
    \@resetst@tics
    \@resetdyn@mics
  \else
```

Still on same page, save contents of box255

```
      \global\@firstcolumnfalse
      \@setcolbox\c@thisframe
    \fi
    \global\c@thisframe=\@nxtcol
    \@setcol{\c@thisframe}\relax
    \global\@colht\vsize
  }
```

`\@dblfloatplacement` Modify `\@dblfloatplacement` replacing `\textheight` with `\vsize`.

```
  \def\@dblfloatplacement{%
    \global\@dbltopnum\c@dbltopnumber
    \global\@dbltoproom\dbltopfraction\@colht\@textmin
    \@colht\advance\@textmin -\@dbltoproom
    \@fpmin\dblfloatpagefraction\vsize
    \@fptop \@dblfptop \@fpsep \@dblfpsep \@fpbot \@dblfpbot
  }
```

1.9 Static versions of floats

Floats can not go in saveboxes or minipages, so define static versions to go in static and dynamic frames. These just set `\@captype` so that the `\caption` command may be used.

`statictable`

```
  \newenvironment{statictable}{\def\@captype{table}}{}
```

`staticfigure`

```
  \newenvironment{staticfigure}{\def\@captype{figure}}{}
```

1.10 Standard Layouts

1.10.1 Column Styles

Redefine `\twocolumn` and `\onecolumn` to set up flow frames from the dimensions of the typeblock. Ignore the optional argument. The flow frame height will be adjusted to make sure that it is an integer multiple of `\baselineskip`, unless `\ffvajdustfalse` is used.

```
  \newif\ifffvadjust
  \ffvadjusttrue
```

`\onecolumn` `\onecolumn` will make a single flow frame that takes up the entire area of the typeblock (adjusted according to `\ifffvadjust`.) Frames should only be created in the preamble, otherwise the next flow frame may not be detected by the output routine. The exception to this is when the output routine can't find any more flow frames to use, in which case it creates a single flow frame using `\@onecolumn`. Therefore, make `\onecolumn` use `\@onecolumn`, and then set \onecolumn as a preamble command, so it can't be used in the document, but the output routine can use \@onecolumn. Syntax: \onecolumn[⟨*pages*⟩], where ⟨*pages*⟩ is the page list for which the new flow frame is defined.

```
\renewcommand*{\onecolumn}{\@onecolumn}
```

\@onecolumn

```
\newcommand*{\@onecolumn}[1][all]{%
  \@onecolumninarea[#1]{\textwidth}{\textheight}{0pt}{0pt}%
}
```

Need a length to store the height of the flow frame so that it can be adjusted.

```
\newlength\columnheight
```

\onecolumninarea \onecolumn is in fact a special case of \onecolumninarea which sets up one flow frame in the specified area, given by bottom left corner (⟨x⟩, ⟨y⟩), relative to the typeblock, with width ⟨w⟩ and height ⟨h⟩. The only difference between \onecolumninarea and explicitly creating the flow frame using \newflowframe is the \onecolumninarea will adjust the vertical height the ensure it is a multiple of \baselineskip. There is also no starred version, so if you want a border, you will need to set it explicitly using \setflowframe. Syntax:
\onecolumninarea[⟨*pages*⟩]{⟨w⟩}{⟨h⟩}{⟨x⟩}{⟨y⟩}.

```
\newcommand*{\onecolumninarea}{\@onecolumninarea}
\@onlypreamble{\onecolumninarea}
```

\@onecolumninarea

```
\newcommand*{\@onecolumninarea}[5][all]{%
  \setlength{\columnheight}{#3}%
  \ifffvadjust
    \adjustheight{\columnheight}%
  \fi
  \@n@wflowframe[#1]{#2}{\columnheight}{#4}{#5}%
}
```

\twocolumn Set up two flow frames parallel to each other with a distance of \columnsep between them, to fill the entire typeblock (although the frames may end up marginally shorter than \textheight after they have been adjusted.) Again, these commands may only be used in the preamble. Note that unlike the standard \twocolumn command, this one has an optional argument that indicates which pages the two flow frames should appear on. Syntax: \twocolumn[⟨*pages*⟩].

```
\renewcommand*{\twocolumn}{\@twocolumn}
```

\@twocolumn

```
\newcommand*{\@twocolumn}[1][all]{%
  \@twocolumninarea[#1]{\textwidth}{\textheight}{0pt}{0pt}%
}
```

`\twocolumninarea` Again, `\twocolumn` is actually a special case of `\twocolumninarea`. Syntax: `\twocolumninarea[`⟨*pages*⟩`]{`⟨*w*⟩`}{`⟨*h*⟩`}{`⟨*x*⟩`}{`⟨*y*⟩`}`.

```
\newcommand*{\twocolumninarea}{\@twocolumninarea}
\@onlypreamble{\twocolumninarea}
```

`\@twocolumninarea`

```
\newcommand*{\@twocolumninarea}[5][all]{%
  \setlength{\columnheight}{#3}%
  \ifffvadjust
    \adjustheight{\columnheight}%
  \fi
  \setlength{\columnwidth}{#2}%
  \addtolength{\columnwidth}{-\columnsep}%
  \divide\columnwidth by 2\relax
  \setlength{\@ff@tmp@x}{#4}%
  \addtolength{\@ff@tmp@x}{\columnwidth}%
  \addtolength{\@ff@tmp@x}{\columnsep}%
  \iflefttorightcolumns
    \@n@wflowframe[#1]{\columnwidth}{\columnheight}{#4}{#5}%
    \setflowframe{\c@maxflow}{margin=left}%
  \else
    \@n@wflowframe[#1]{\columnwidth}{\columnheight}{\@ff@tmp@x}{#5}%
    \setflowframe{\c@maxflow}{margin=right}%
  \fi
  \iflefttorightcolumns
    \@n@wflowframe[#1]{\columnwidth}{\columnheight}{\@ff@tmp@x}{#5}%
    \setflowframe{\c@maxflow}{margin=right}%
  \else
    \@n@wflowframe[#1]{\columnwidth}{\columnheight}{#4}{#5}%
    \setflowframe{\c@maxflow}{margin=left}%
  \fi
}
```

`\Ncolumn` Again for an aribtrary number of columns (⟨*n*⟩). Syntax: `\Ncolumn[`⟨*pages*⟩`]{`⟨*n*⟩`}`.

```
\newcommand*{\Ncolumn}[2][all]{%
  \Ncolumninarea[#1]{#2}{\textwidth}{\textheight}{0pt}{0pt}%
}
\@onlypreamble{\Ncolumn}
```

`\Ncolumninarea` Check the number of flow frames requested, and do one of the special cases if available. Syntax:
`\Ncolumninarea[`⟨*pages*⟩`]{`⟨*n*⟩`}{`⟨*w*⟩`}{`⟨*h*⟩`}{`⟨*x*⟩`}{`⟨*y*⟩`}`.

```
\newcommand*{\Ncolumninarea}[6][all]{%
  \ifnum#2>2\relax
    \@Ncolumninarea[#1]{#2}{#3}{#4}{#5}{#6}%
  \else
    \ifcase#2\relax
      \PackageError{flowfram}%
      {%
        You have requested 0 flowframes!%
      }%
      {%
        It does not make much sense to ask to create 0 flow frames%
      }%
    \or
      \onecolumninarea[#1]{#3}{#4}{#5}{#6}%
    \or
      \twocolumninarea[#1]{#3}{#4}{#5}{#6}%
    \else
      \PackageError{flowfram}%
      {%
        Can't create a negative number of flow frames!%
      }%
      {%
        You have asked for \number#2 \space flow frames
        which really doesn't make sense%
      }%
    \fi
  \fi
}

\@onlypreamble{\Ncolumninarea}
```

`\@Ncolumninarea` Set up ⟨*n*⟩ columns in the area specified. There is a horizontal distance of `\columnsep` between them all.

```
\newcommand*{\@Ncolumninarea}[6][all]{%
  \@colN=#2\relax
  \advance\@colN by -1\relax
  \setlength{\columnwidth}{#3}%
  \addtolength{\columnwidth}{-\@colN\columnsep}%
  \divide\columnwidth by #2\relax
  \setlength{\@ff@tmp@x}{#5}%
  \iflefttorightcolumns
  \else
    \addtolength{\@ff@tmp@x}{#3}%
    \addtolength{\@ff@tmp@x}{-\columnwidth}%
  \fi
  \setlength{\columnheight}{#4}%
  \ifffvadjust\adjustheight{\columnheight}\fi%
  \@colN=0\relax
  \loop
    \advance\@colN by 1\relax
    \newflowframe[#1]{\columnwidth}{\columnheight}{\@ff@tmp@x}{#6}%
    \iflefttorightcolumns
      \addtolength{\@ff@tmp@x}{\columnwidth}%
      \addtolength{\@ff@tmp@x}{\columnsep}%
    \else
```

```
        \addtolength{\@ff@tmp@x}{-\columnwidth}%
        \addtolength{\@ff@tmp@x}{-\columnsep}%
      \fi
    \ifnum\@colN<#2
    \repeat
  }
```

Set up something similar but have another frame (of type ⟨*type*⟩) at the top of the other frames.

`\vcolumnsep` The vertical distance between the top frames and column flow frames when created using `\Ncolumntop` etc is given by:

```
\newlength{\vcolumnsep}
\setlength{\vcolumnsep}{\columnsep}
```

`\onecolumntop` `\onecolumntop` makes one flow frame, and one ⟨*type*⟩ frame in the area specified, where the ⟨*type*⟩ frame is ⟨*H*⟩ high. The distance between the top frame and the column flow frame will be approximately `\vcolumnsep`. (The height of flow frame may be adjusted to make it an integer multiple of `\baselineskip`.)

First the special case where the area is the typeblock. Syntax:
`\onecolumntop[`⟨*pages*⟩`]{`⟨*type*⟩`}{`⟨*H*⟩`}`

```
\newcommand*{\onecolumntop}[3][all]{%
  \onecolumntopinarea[#1]{#2}{#3}{\textwidth}{\textheight}{0pt}{0pt}%
}
\@onlypreamble{\onecolumntop}
```

`\onecolumnStop` Special case for static frame. Syntax: `\onecolumnStop[`⟨*pages*⟩`]{`⟨*H*⟩`}`

```
\newcommand*{\onecolumnStop}[2][all]{%
  \onecolumntopinarea[#1]{static}{#2}{\textwidth}{\textheight}{0pt}{0pt}%
}
```

`\onecolumnDtop` Special case for dynamic frame. Syntax: `\onecolumnDtop[`⟨*pages*⟩`]{`⟨*H*⟩`}`

```
\newcommand*{\onecolumnDtop}[2][all]{%
  \onecolumntopinarea[#1]{dynamic}{#2}{\textwidth}{\textheight}{0pt}{0pt}%
}
```

`\newframe` Create a frame of given type. Syntax:
`\newframe[`⟨*pages*⟩`]{`⟨*type*⟩`}{`⟨*w*⟩`}{`⟨*h*⟩`}{`⟨*x*⟩`}{`⟨*y*⟩`}`.

```
\newcommand*{\newframe}[6][all]{%
  \ifthenelse{\equal{#2}{flow}}%
  {%
    \@n@wflowframe[#1]{#3}{#4}{#5}{#6}%
  }%
  {%
    \ifthenelse{\equal{#2}{dynamic}}%
    {%
      \@n@wdynamicframe[#1]{#3}{#4}{#5}{#6}%
    }%
```

```
      {%
        \ifthenelse{\equal{#2}{static}}%
        {%
          \@n@wstaticframe[#1]{#3}{#4}{#5}{#6}%
        }%
        {%
          \PackageError{flowfram}%
          {Unknown frame type '#2'}%
          {%
            Available frame types are: 'flow', 'static' and 'dynamic'%
          }%
        }%
      }%
    }%
  }
```

`\onecolumntopinarea` Now for a specified area. Syntax:
\onecolumntopinarea[⟨*pages*⟩]{⟨*type*⟩}{⟨*H*⟩}{⟨*w*⟩}{⟨*h*⟩}{⟨*x*⟩}{⟨*y*⟩}.

```
\newlength\@ff@staticH

\newcommand*{\onecolumntopinarea}[7][all]{%
  \setlength{\@ff@staticH}{#3}%
  \setlength{\@ff@tmp@y}{#5}%
  \addtolength{\@ff@tmp@y}{-\@ff@staticH}%
  \setlength{\columnheight}{\@ff@tmp@y}%
  \addtolength{\columnheight}{-\vcolumnsep}%
  \ifffvadjust
    \adjustheight{\columnheight}%
  \fi
  \addtolength{\@ff@tmp@y}{#7}%
  \newframe[#1]{#2}{#4}{\@ff@staticH}{#6}{\@ff@tmp@y}%
  \@n@wflowframe[#1]{#4}{\columnheight}{#6}{#7}%
}

\@onlypreamble{\onecolumntopinarea}
```

`\onecolumnStopinarea` Special case for static frame. Syntax:
\onecolumnStopinarea[⟨*pages*⟩]{⟨*H*⟩}{⟨*w*⟩}{⟨*h*⟩}{⟨*x*⟩}{⟨*y*⟩}.

```
\newcommand*{\onecolumnStopinarea}[6][all]{%
  \onecolumntopinarea[#1]{static}{#2}{#3}{#4}{#5}{#6}%
}
```

`\onecolumnDtopinarea` Special case for dynamic frame. Syntax:
\onecolumnDtopinarea[⟨*pages*⟩]{⟨*H*⟩}{⟨*w*⟩}{⟨*h*⟩}{⟨*x*⟩}{⟨*y*⟩}.

```
\newcommand*{\onecolumnDtopinarea}[6][all]{%
  \onecolumntopinarea[#1]{dynamic}{#2}{#3}{#4}{#5}{#6}%
}
```

`\twocolumntop` Now for two flow frames, with a single ⟨*type*⟩ frame above both of them. Syntax:
`\twocolumntop[`⟨*pages*⟩`]{`⟨*type*⟩`}{`⟨*H*⟩`}`

First the special case where the area is the entire typeblock:

```
\newcommand*{\twocolumntop}[3][all]{%
  \twocolumntopinarea[#1]{#2}{#3}{\textwidth}{\textheight}{0pt}{0pt}%
}
\@onlypreamble{\twocolumntop}
```

`\twocolumnStop` Special case for static frame.

```
\newcommand*{\twocolumnStop}[2][all]{%
  \@twocolumntopinarea[#1]{static}{#2}{\textwidth}{\textheight}{0pt}{0pt}%
}
```

`\twocolumnDtop` Special case for dynamic frame.

```
\newcommand*{\twocolumnDtop}[2][all]{%
  \twocolumntop[#1]{dynamic}{#2}%
}
```

Now for a general area.

`\twocolumntopinarea` Syntax:
`\twocolumntopinarea[`⟨*pages*⟩`]{`⟨*type*⟩`}{`⟨*H*⟩`}{`⟨*w*⟩`}{`⟨*h*⟩`}{`⟨*x*⟩`}{`⟨*y*⟩`}`.

```
\newcommand*{\twocolumntopinarea}{\@twocolumntopinarea}
\newcommand*{\@twocolumntopinarea}[7][all]{%
  \setlength{\@ff@staticH}{#3}%
```

work out where to put the static frame

```
  \setlength{\@ff@tmp@y}{#5}%
  \addtolength{\@ff@tmp@y}{-\@ff@staticH}%
  \setlength{\columnheight}{\@ff@tmp@y}%
  \addtolength{\@ff@tmp@y}{#7}%
  \newframe[#1]{#2}{#4}{\@ff@staticH}{#6}{\@ff@tmp@y}%
```

work out height of the flow frames

```
  \addtolength{\columnheight}{-\vcolumnsep}%
  \ifffvadjust\adjustheight{\columnheight}\fi
```

work out the widths of the flow frames

```
  \setlength{\columnwidth}{#4}%
  \addtolength{\columnwidth}{-\columnsep}%
  \divide\columnwidth by 2\relax
```

work out the offset of the right column

```
  \setlength{\@ff@tmp@x}{\columnwidth}%
  \addtolength{\@ff@tmp@x}{\columnsep}%
  \addtolength{\@ff@tmp@x}{#6}%
  \iflefttorightcolumns
    \@n@wflowframe[#1]{\columnwidth}{\columnheight}{#6}{#7}%
    \setflowframe{\c@maxflow}{margin=left}%
  \else
```

```
    \@n@wflowframe[#1]{\columnwidth}{\columnheight}{\@ff@tmp@x}{#7}%
    \setflowframe{\c@maxflow}{margin=right}%
  \fi
  \iflefttorightcolumns
    \@n@wflowframe[#1]{\columnwidth}{\columnheight}{\@ff@tmp@x}{#7}%
    \setflowframe{\c@maxflow}{margin=right}%
  \else
    \@n@wflowframe[#1]{\columnwidth}{\columnheight}{#6}{#7}%
    \setflowframe{\c@maxflow}{margin=left}%
  \fi
}
\@onlypreamble{\twocolumntopinarea}
```

`\twocolumnStopinarea` Special case for static frame.

```
\newcommand*{\twocolumnStopinarea}[6][all]{%
  \twocolumntopinarea[#1]{static}{#2}{#3}{#4}{#5}{#6}%
}
```

`\twocolumnDtopinarea` Special case for dynamic frame.

```
\newcommand*{\twocolumnDtopinarea}[6][all]{%
  \twocolumntopinarea[#1]{dynamic}{#2}{#3}{#4}{#5}{#6}%
}
```

`\Ncolumntop` Similarly for an arbitrary number of flow frames. Special case where the area is the typeblock.

Syntax:

`\Ncolumntop[`⟨*pages*⟩`]{`⟨*type*⟩`}{`⟨*n*⟩`}{`⟨*H*⟩`}`

```
\newcommand*{\Ncolumntop}[4][all]{%
  \Ncolumntopinarea[#1]{#2}{#3}{#4}{\textwidth}{\textheight}{0pt}{0pt}%
}
\@onlypreamble{\Ncolumntop}
```

`\NcolumnStop` Special case for static frame.

```
\newcommand*{\NcolumnStop}[3][all]{%
  \Ncolumntop[#1]{static}{#2}{#3}%
}
```

`\NcolumnDtop` Special case for dynamic frame.

```
\newcommand*{\NcolumnDtop}[3][all]{%
  \Ncolumntop[#1]{dynamic}{#2}{#3}%
}
```

`\Ncolumntopinarea` Again test to make sure the user requested a sensible number.

```
\newcommand*{\Ncolumntopinarea}[8][all]{%
  \ifnum#3>2\relax
    \@Ncolumntopinarea[#1]{#2}{#3}{#4}{#5}{#6}{#7}{#8}%
  \else
    \ifcase#3\relax
```

```
        \PackageError{flowfram}%
        {%
          You have requested 0 flowframes!%
        }%
        {%
          It does not make much sense to ask to create 0 flow frames%
        }%
      \or
        \onecolumntopinarea[#1]{#2}{#4}{#5}{#6}{#7}{#8}%
      \or
        \twocolumntopinarea[#1]{#2}{#4}{#5}{#6}{#7}{#8}%
      \else
        \PackageError{flowfram}%
        {%
          Can't create a negative number of flow frames!%
        }%
        {%
          You have asked for \number#3 \space flow frames
          which really doesn't make sense%
        }%
      \fi
    \fi
  }
  \@onlypreamble{\Ncolumntopinarea}
```

`\@Ncolumntopinarea` Fit the frames into specified area. Syntax:
`\Ncolumntopinarea[`⟨*pages*⟩`]{`⟨*type*⟩`}{`⟨*n*⟩`}{`⟨*H*⟩`}{`⟨*w*⟩`}{`⟨*h*⟩`}{`⟨*x*⟩`}{`⟨*y*⟩`}`.

```
\newcommand*{\@Ncolumntopinarea}[8][all]{%
  \setlength{\@ff@staticH}{#4}%
```

work out where to put the static frame

```
  \setlength{\@ff@tmp@y}{#6}%
  \addtolength{\@ff@tmp@y}{-\@ff@staticH}%
  \setlength{\columnheight}{\@ff@tmp@y}%
  \addtolength{\@ff@tmp@y}{#8}%
  \newframe[#1]{#2}{#5}{\@ff@staticH}{#7}{\@ff@tmp@y}%
```

work out height of the flow frames

```
  \addtolength{\columnheight}{-\vcolumnsep}%
```

adjust the flow frame height so that it is a multiple of `\baselineskip`

```
  \ifffvadjust
    \adjustheight{\columnheight}%
  \fi
```

work out the widths of the flow frames

```
  \@colN=#3\relax
  \advance\@colN by -1\relax
  \setlength{\columnwidth}{#5}%
  \addtolength{\columnwidth}{-\@colN\columnsep}%
  \divide\columnwidth by #3\relax
```

Set the *x* position of the first frame

```
  \setlength{\@ff@tmp@x}{#7}%
  \iflefttorightcolumns
  \else
    \addtolength{\@ff@tmp@x}{#5}%
    \addtolength{\@ff@tmp@x}{-\columnwidth}%
  \fi
  \@colN=0\relax
  \loop
    \advance\@colN by 1\relax
    \newflowframe[#1]{\columnwidth}{\columnheight}{\@ff@tmp@x}{#8}%
```

work out the offset for the next column

```
    \iflefttorightcolumns
      \addtolength{\@ff@tmp@x}{\columnwidth}%
      \addtolength{\@ff@tmp@x}{\columnsep}%
    \else
      \addtolength{\@ff@tmp@x}{-\columnwidth}%
      \addtolength{\@ff@tmp@x}{-\columnsep}%
    \fi
  \ifnum\@colN<#3
  \repeat
}
```

`\NcolumnStopinarea` Specific case for static frame.

```
\newcommand*{\NcolumnStopinarea}[7][all]{%
  \Ncolumntopinarea[#1]{static}{#2}{#3}{#4}{#5}{#6}{#7}%
}
```

`\NcolumnDtopinarea` Specific case for dynamic frame.

```
\newcommand*{\NcolumnDtopinarea}[7][all]{%
  \Ncolumntopinarea[#1]{dynamic}{#2}{#3}{#4}{#5}{#6}{#7}%
}
```

Now the same kind of thing but with the ⟨*type*⟩ frame at the bottom. Firstly, a single flow frame with a ⟨*type*⟩ frame below it.

`\onecolumnbottom` Syntax:

`\onecolumnbottom[⟨pages⟩]{⟨type⟩}{⟨H⟩}`

```
\newcommand*{\onecolumnbottom}[3][all]{%
  \onecolumnbottominarea[#1]{#2}{#3}{\textwidth}{\textheight}{0pt}{0pt}%
}
```

This command may only be used in the preamble.

```
\@onlypreamble{\onecolumnbottom}
```

`\onecolumnSbottom` Special case for static frame.

```
\newcommand*{\onecolumnSbottom}[2][all]{%
  \onecolumnbottom[#1]{static}{#2}%
}
```

\onecolumnDbottom

Special case for dynamic frame.

```
\newcommand*{\onecolumnDbottom}[2][all]{%
  \onecolumnbottom[#1]{dynamic}{#2}%
}
```

General case of the above, but fit in specified area.

\onecolumnbottominarea

Syntax:
\onecolumnbottominarea[⟨*pages*⟩]{⟨*type*⟩}{⟨*H*⟩}{⟨*w*⟩}{⟨*h*⟩}{⟨*x*⟩}{⟨*y*⟩}, where ⟨*H*⟩ is the ⟨*type*⟩ frame's height. The area is defined by bottom left co-ordinates (⟨*x*⟩, ⟨*y*⟩) width ⟨*w*⟩, and height ⟨*h*⟩.

```
\newcommand*{\onecolumnbottominarea}[7][all]{%
  \setlength{\@ff@staticH}{#3}%
  \setlength{\columnheight}{#5}%
  \addtolength{\columnheight}{-\@ff@staticH}%
  \addtolength{\columnheight}{-\vcolumnsep}%
  \ifffvadjust
    \adjustheight{\columnheight}%
  \fi
  \setlength{\@ff@tmp@y}{#5}%
  \addtolength{\@ff@tmp@y}{-\columnheight}%
  \addtolength{\@ff@tmp@y}{#7}%
  \newframe[#1]{#2}{#4}{\@ff@staticH}{#6}{#7}%
  \newflowframe[#1]{#4}{\columnheight}{#6}{\@ff@tmp@y}%
}
```

Again, this command may only be used in the preamble.

```
\@onlypreamble{\onecolumnbottominarea}
```

\onecolumnSbottominarea

Special case for static frame.

```
\newcommand*{\onecolumnSbottominarea}[6][all]{%
  \onecolumnbottominarea[#1]{static}{#2}{#3}{#4}{#5}{#6}%
}
```

\onecolumnDbottominarea

Special case for dynamic frame.

```
\newcommand*{\onecolumnDbottominarea}[6][all]{%
  \onecolumnbottominarea[#1]{dynamic}{#2}{#3}{#4}{#5}{#6}%
}
```

\twocolumnbottom

Now for two flow frames side by side with a static frame underneath both of them. Firstly, the specific case where the area is the entire typeblock. Syntax:
\twocolumnbottom[⟨*pages*⟩]{⟨*type*⟩}{⟨*H*⟩}.

```
\newcommand*{\twocolumnbottom}[3][all]{%
  \twocolumnSbottominarea[#1]{#2}{#3}{\textwidth}{\textheight}{0pt}{0pt}%
}
\@onlypreamble{\twocolumnbottom}
```

`\twocolumnSbottom` Special case for static frame.

```
\newcommand*{\twocolumnSbottom}[2][all]{%
  \twocolumnbottom[#1]{static}{#2}%
}
```

`\twocolumnDbottom` Special case for dynamic frame.

```
\newcommand*{\twocolumnDbottom}[2][all]{%
  \twocolumnbottom[#1]{dynamic}{#2}%
}
```

`\twocolumnbottominarea` Now for a general area. Syntax:
`\twocolumnbottominarea[`⟨*pages*⟩`]{`⟨*type*⟩`}{`⟨*H*⟩`}{`⟨*w*⟩`}{`⟨*h*⟩`}{`⟨*x*⟩`}{`⟨*y*⟩`}`.

```
\newcommand*{\twocolumnbottominarea}[7][all]{%
  \setlength{\@ff@staticW}{#4}%
  \setlength{\@ff@staticH}{#3}%
```

work out height of the flow frames

```
  \setlength{\columnheight}{#5}%
  \addtolength{\columnheight}{-\@ff@staticH}%
  \addtolength{\columnheight}{-\vcolumnsep}%
  \ifffvadjust\adjustheight{\columnheight}\fi%
  \newframe[#1]{#2}{\@ff@staticW}{\@ff@staticH}{#6}{#7}%
```

work out the y position of the flow frames

```
  \setlength{\@ff@tmp@y}{#5}%
  \addtolength{\@ff@tmp@y}{-\columnheight}%
  \addtolength{\@ff@tmp@y}{#7}%
```

work out the widths of the flow frames

```
  \setlength{\columnwidth}{\@ff@staticW}%
  \addtolength{\columnwidth}{-\columnsep}%
  \divide\columnwidth by 2\relax
```

work out the x offset of the right column

```
  \setlength{\@ff@tmp@x}{\columnwidth}%
  \addtolength{\@ff@tmp@x}{\columnsep}%
  \addtolength{\@ff@tmp@x}{#6}%
```

Define the frames

```
  \iflefttorightcolumns
    \newflowframe[#1]{\columnwidth}{\columnheight}{#6}{\@ff@tmp@y}%
    \setflowframe{\c@maxflow}{margin=left}%
  \else
    \newflowframe[#1]{\columnwidth}{\columnheight}%
      {\@ff@tmp@x}{\@ff@tmp@y}%
    \setflowframe{\c@maxflow}{margin=right}%
  \fi
  \iflefttorightcolumns
    \newflowframe[#1]{\columnwidth}{\columnheight}%
      {\@ff@tmp@x}{\@ff@tmp@y}%
    \setflowframe{\c@maxflow}{margin=right}%
  \else
    \newflowframe[#1]{\columnwidth}{\columnheight}{#6}{\@ff@tmp@y}%
    \setflowframe{\c@maxflow}{margin=left}%
  \fi
}
\@onlypreamble{\twocolumnbottominarea}
```

\twocolumnSbottominarea Special case for static frame.

```
\newcommand*{\twocolumnSbottominarea}[6][all]{%
  \twocolumnbottominarea[#1]{static}{#2}{#3}{#4}{#5}{#6}%
}
```

\twocolumnDbottominarea Special case for dynamic frame.

```
\newcommand*{\twocolumnDbottominarea}[6][all]{%
  \twocolumnbottominarea[#1]{dynamic}{#2}{#3}{#4}{#5}{#6}%
}
```

Now for an arbitrary number of parallel flow frames with a static frame beneath all of them.

\Ncolumnbottom First make them fill the entire typeblock. Syntax: `\Ncolumnbottom[`⟨*pages*⟩`]{`⟨*type*⟩`}{`⟨*H*⟩`}`.

```
\newcommand*{\Ncolumnbottom}[4][all]{%
  \Ncolumnbottominarea[#1]{#2}{#3}{#4}{\textwidth}{\textheight}{0pt}{0pt}%
}
\@onlypreamble{\Ncolumnbottom}
```

\NcolumnSbottom Special case for static frame.

```
\newcommand*{\NcolumnSbottom}[3][all]{%
  \Ncolumnbottom[#1]{static}{#2}{#3}%
}
```

\NcolumnDbottom Special case for dynamic frame.

```
\newcommand*{\NcolumnDbottom}[3][all]{%
  \Ncolumnbottom[#1]{dynamic}{#2}{#3}%
}
```

\Ncolumnbottominarea Again check the user has requested a sensible number.

```
\newcommand*{\Ncolumnbottominarea}[8][all]{%
  \ifnum#3>2\relax
    \@Ncolumnbottominarea[#1]{#2}{#3}{#4}{#5}{#6}{#7}{#8}%
  \else
    \ifcase#3\relax
      \PackageError{flowfram}{%
      You have requested 0 flowframes!}{%
      It does not make much sense to ask to create 0 flow frames}
    \or
      \onecolumnbottominarea[#1]{#2}{#4}{#5}{#6}{#7}{#8}%
    \or
      \twocolumnbottominarea[#1]{#2}{#4}{#5}{#6}{#7}{#8}%
    \else
      \PackageError{flowfram}%
      {%
        Can't create a negative number of flow frames!%
      }%
      {%
        You have asked for \number#3 \space flow frames
        which really doesn't make sense%
      }%
    \fi
  \fi
}
\@onlypreamble{\Ncolumnbottominarea}
```

`\@NcolumnSbottominarea` An arbitrary number of columns with a static frame underneath them all, filling the specified area.

```
  \newcommand*{\@NcolumnSbottominarea}[8][all]{%
    \setlength{\@ff@staticH}{#4}%
```

work out height of the flow frames

```
    \setlength{\columnheight}{#6}%
    \addtolength{\columnheight}{-\@ff@staticH}%
    \addtolength{\columnheight}{-\vcolumnsep}%
```

adjust the flow frame height so that it is a multiple of `\baselineskip`

```
    \ifffvadjust
      \adjustheight{\columnheight}%
    \fi
    \newframe[#1]{#2}{#5}{\@ff@staticH}{#7}{#8}%
```

work out the y offset of the flow frames

```
    \setlength{\@ff@tmp@y}{#6}%
    \addtolength{\@ff@tmp@y}{-\columnheight}%
    \addtolength{\@ff@tmp@y}{#8}%
```

work out the widths of the flow frames

```
    \@colN=#3\relax
    \advance\@colN by -1\relax
    \setlength{\columnwidth}{#5}%
    \addtolength{\columnwidth}{-\@colN\columnsep}%
    \divide\columnwidth by #3\relax
```

Set the x offset of the first frame.

```
    \setlength{\@ff@tmp@x}{#7}%
    \iflefttorightcolumns
    \else
      \addtolength{\@ff@tmp@x}{#5}%
      \addtolength{\@ff@tmp@x}{-\columnwidth}%
    \fi
    \@colN=0\relax
    \loop
      \advance\@colN by 1\relax
      \newflowframe[#1]{\columnwidth}{\columnheight}%
         {\@ff@tmp@x}{\@ff@tmp@y}%
```

work out the offset for the next column

```
      \iflefttorightcolumns
        \addtolength{\@ff@tmp@x}{\columnwidth}%
        \addtolength{\@ff@tmp@x}{\columnsep}%
      \else
        \addtolength{\@ff@tmp@x}{-\columnwidth}%
        \addtolength{\@ff@tmp@x}{-\columnsep}%
      \fi
    \ifnum\@colN<#3
    \repeat
  }
```

`\NcolumnSbottominarea` Specific case for static frame.

```
  \newcommand*{\NcolumnSbottominarea}[1][all]{%
    \Ncolumnbottominarea[#1]{static}%
  }
```

`\NcolumnDbottominarea` Specific case for dynamic frame.

```
  \newcommand*{\NcolumnDbottominarea}[1][all]{%
    \Ncolumnbottominarea[#1]{dynamic}%
  }
```

`\adjustheight` Given a height `#1` (a length), adjust it so that it is a multiple of `\baselineskip`.

```
  \newcount\@ff@adjh
  \newcommand*{\adjustheight}[1]{%
```

convert to an integer

```
    \@ff@adjh=#1\relax
    \divide\@ff@adjh by \baselineskip\relax
    #1=\baselineskip\relax
    \multiply#1 by \@ff@adjh\relax
  }
```

`\adjustcolsep` Adjust the value of `\columnsep` so that the margins will fit between columns.

```
  \newcommand*{\adjustcolsep}{%
    \multiply\columnsep by 2\relax
    \addtolength{\columnsep}{\marginparwidth}%
  }
```

1.10.2 Backdrop Effects

Set up some commands to make static frames for different styles of backdrop.

`\vtwotone` Syntax:
`\vtwotone[`⟨*pages*⟩`][`⟨*xoffset*⟩`]{`⟨*W1*⟩`}{`⟨*C1*⟩`}{`⟨*L1*⟩`}{`⟨*W2*⟩`}{`⟨*C2*⟩`}{`⟨*L2*⟩`}`
where the first frame has width ⟨*W1*⟩ with background colour ⟨*C1*⟩ and label ⟨*L1*⟩. The second frame has width ⟨*W2*⟩ with background colour ⟨*C2*⟩ and label ⟨*L2*⟩. Unlike earlier commands, the x-offset is relative to the left page edge *not* the typeblock. This is because they are designed for backdrops, which tend to span the entire page. Note that the colour specs must be completely enclosed in braces. e.g. `{[gray]{0.5}}` *not* `[gray]{0.5}`.

Need a length to store the width of the static frame.

```
\newlength\@ff@staticW
```

Vertical two tone effect where the height of the static frames is equal to the paper height.

```
\newcommand*{\vtwotone}[1][all]{%
  \def\ff@pages{#1}%
  \@vtwotone
}

\newcommand*{\@vtwotone}[1][0pt]{\@@vtwotonebottom{#1}{\paperheight}}
```

`\vtwotonebottom` Vertical two tone effect along the bottom of the page, of height ⟨*H*⟩. Syntax:
`\vtwotonebottom[`⟨*pages*⟩`][`⟨*xoffset*⟩`]{`⟨*H*⟩`}{`⟨*W1*⟩`}{`⟨*C1*⟩`}{`⟨*L1*⟩`}{`⟨*W2*⟩`}{`⟨*C2*⟩`}{`⟨*L2*⟩`}`
where the first frame starts at ⟨*xoffset*⟩.

```
\newcommand*{\@@vtwotonebottom}[8]{%
  \computeleftedgeodd{\@ff@tmp@x}%
  \if@twoside
    \computeleftedgeeven{\@ff@tmp@x@even}%
  \else
    \setlength{\@ff@tmp@x@even}{\@ff@tmp@x}%
  \fi
  \computebottomedge{\@ff@tmp@y}%
  \addtolength{\@ff@tmp@x}{#1}%
  \addtolength{\@ff@tmp@x@even}{#1}%
  \@nextvband{\ff@pages}{#2}{#3}{#4}{#5}%
  \@nextvband{\ff@pages}{#2}{#6}{#7}{#8}%
}

\@onlypreamble{\vtwotone}
```

`\vtwotonebottom` Border strip along the bottom of the page

```
\newcommand*{\vtwotonebottom}[1][all]{%
  \def\ff@pages{#1}%
  \@vtwotonebottom
}

\@onlypreamble{\vtwotonebottom}

\newcommand*{\@vtwotonebottom}[2][0pt]{\@@vtwotonebottom{#1}{#2}}
```

`\vtwotonetop` Border strip along the top of the page

```
\newcommand*{\vtwotonetop}[1][all]{%
  \def\ff@pages{#1}%
  \@vtwotonetop
}

\newcommand*{\@vtwotonetop}[2][0pt]{\@@vtwotonetop{#1}{#2}}

\newcommand*{\@@vtwotonetop}[8]{%
  \computeleftedgeodd{\@ff@tmp@x}%
  \if@twoside
    \computeleftedgeeven{\@ff@tmp@x@even}%
  \else
    \setlength{\@ff@tmp@x@even}{\@ff@tmp@x}%
  \fi
  \computetopedge{\@ff@tmp@y}%
  \addtolength{\@ff@tmp@y}{-#2}%
  \addtolength{\@ff@tmp@x}{#1}%
  \addtolength{\@ff@tmp@x@even}{#1}%
  \@nextvband{\ff@pages}{#2}{#3}{#4}{#5}%
  \@nextvband{\ff@pages}{#2}{#6}{#7}{#8}%
}
```

`\@nextvband` Make next static frame. Syntax:
`\@nextvband{`⟨*pages*⟩`}{`⟨*height*⟩`}{`⟨*width*⟩`}{`⟨*colour specs*⟩`}{`⟨*label*⟩`}`
x and *y* offsets are given by `\@ff@tmp@x` and `\@ff@tmp@y`. On exit, `\@ff@tmp@x` is set to the right border.

```
\newcommand*{\@nextvband}[5]{%
  \setlength{\@ff@staticW}{#3}%
  \ifthenelse{\equal{#5}{}}%
  {%
    \newstaticframe[#1]{\@ff@staticW}{#2}{\@ff@tmp@x}{\@ff@tmp@y}%
  }%
  {%
    \newstaticframe[#1]{\@ff@staticW}{#2}{\@ff@tmp@x}{\@ff@tmp@y}[#5]%
  }%
  \expandafter\global\expandafter\setlength
   \csname @sf@\romannumeral\c@maxstatic @evenx\endcsname{%
     \@ff@tmp@x@even}%
  \@setframecol#4\end{\c@maxstatic}{backcol}{sf}%
  \addtolength{\@ff@tmp@x}{\@ff@staticW}%
  \addtolength{\@ff@tmp@x@even}{\@ff@staticW}%
}
```

`\vNtone` Similarly for N colours. Syntax:
`\vNtone[`⟨*pages*⟩`][`⟨*xoffset*⟩`]{`⟨*n*⟩`}{`⟨*W1*⟩`}{`⟨*C1*⟩`}{`⟨*L1*⟩`}`…`{`⟨*Wn*⟩`}{`⟨*Cn*⟩`}{`⟨*Ln*⟩`}`
where the first frame has width ⟨*W1*⟩ with background colour ⟨*C1*⟩ and label ⟨*L1*⟩ all the way up to the ⟨*n*⟩th frame which has width ⟨*Wn*⟩, background colour ⟨*Cn*⟩ and IDL ⟨*Ln*⟩.

Keep track of which strip we are doing.

```
\newcount\@thisstrip
```

This command needs two optional arguments, so store first optional argument, and look for the next.

```
\newcommand*{\vNtone}[1][all]{%
  \def\ff@pages{#1}%
  \@vNtone
}
```

`\@vNtone` Got the first argument, now get the next.

```
\newcommand*{\@vNtone}[2][0pt]{%
  \@@vNtone{#1}{#2}{\paperheight}%
}
```

`\@@vNtone` Vertical ⟨*n*⟩ tone aligned along the bottom of the page with height #3.

```
\newcommand*{\@@vNtone}[3]{%
  \computeleftedgeodd{\@ff@tmp@x}%
  \if@twoside
    \computeleftedgeeven{\@ff@tmp@x@even}%
  \else
    \setlength{\@ff@tmp@x@even}{\@ff@tmp@x}%
  \fi
  \computebottomedge{\@ff@tmp@y}%
  \addtolength{\@ff@tmp@x}{#1}%
  \addtolength{\@ff@tmp@x@even}{#1}%
  \@thisstrip=#2\relax
  \setlength{\@ff@staticH}{#3}%
  \@nextvNband
}
```

`\@nextvNband` Recursively do the next strip.

```
\newcommand*{\@nextvNband}{%
  \ifnum\@thisstrip>0\relax
   \let\flf@next\@@nextvNband
  \else
   \let\flf@next\relax
  \fi
  \advance\@thisstrip by -1\relax
  \flf@next
}
```

`\@@nextvNband` Do current strip, and go on to next one.

```
\newcommand*{\@@nextvNband}[3]{%
  \@nextvband{\ff@pages}{\@ff@staticH}{#1}{#2}{#3}%
  \@nextvNband
}

\@onlypreamble{\vNtone}
```

`\vNtonebottom` Border strip along the bottom of the page. Same as above but user specifies the height.

```
\newcommand*{\vNtonebottom}[1][all]{%
  \def\ff@pages{#1}%
  \@vNtonebottom
}
\@onlypreamble{\vNtonebottom}
```

`\@vNtonebottom`

```
\newcommand*{\@vNtonebottom}[3][0pt]{%
  \@@vNtone{#1}{#2}{#3}%
}
```

`\vNtonetop` Border strip along the top of the page. Again two optional arguments are required. Get first optional argument.

```
\newcommand*{\vNtonetop}[1][all]{%
  \def\ff@pages{#1}%
  \@vNtonetop
}
\@onlypreamble{\vNtonetop}
```

`\@vNtonetop` Get next optional argument.

```
\newcommand*{\@vNtonetop}[3][0pt]{%
  \@@vNtonetop{#1}{#2}{#3}%
}
```

`\@@vNtonetop` Now get on with it. Again, it has to be done recursively.

```
\newcommand*{\@@vNtonetop}[3]{%
  \computeleftedgeodd{\@ff@tmp@x}%
  \if@twoside
    \computeleftedgeeven{\@ff@tmp@x@even}%
  \else
    \setlength{\@ff@tmp@x@even}{\@ff@tmp@x}%
  \fi
  \computetopedge{\@ff@tmp@y}%
  \addtolength{\@ff@tmp@y}{-#3}%
  \addtolength{\@ff@tmp@x}{#1}%
  \addtolength{\@ff@tmp@x@even}{#1}%
  \@thisstrip=#2\relax
  \setlength{\@ff@staticH}{#3}%
  \@nextvNband%
}
```

`\htwotone` Now do horizontal strips. Syntax:
`\htwotone`[⟨*pages*⟩][⟨*y offset*⟩]{⟨*H1*⟩}{⟨*C1*⟩}{⟨*L1*⟩}{⟨*H2*⟩}{⟨*C2*⟩}{⟨*L2*⟩}

```
\newcommand*{\htwotone}[1][all]{%
  \def\ff@pages{#1}%
  \@htwotone
}
```

`\@htwotone`

```
\newcommand*{\@htwotone}[1][0pt]{\@@htwotoneleft{#1}{\paperwidth}}
```

`\@@htwotoneleft` This is all done in much the same way as the vertical strips.

```
\newcommand*{\@@htwotoneleft}[8]{%
  \computeleftedgeodd{\@ff@tmp@x}%
  \if@twoside
    \computeleftedgeeven{\@ff@tmp@x@even}%
  \else
    \setlength{\@ff@tmp@x@even}{\@ff@tmp@x}%
  \fi
  \computebottomedge{\@ff@tmp@y}%
  \addtolength{\@ff@tmp@y}{#1}%
  \@nexthband{\ff@pages}{#2}{#3}{#4}{#5}%
  \@nexthband{\ff@pages}{#2}{#6}{#7}{#8}%
}

\@onlypreamble{\htwotone}
```

`\htwotoneleft` Two tone horizontal strips along left border Syntax: `\htwotoneleft[`⟨*pages*⟩`][`⟨*y offset*⟩`]{`⟨*width*⟩`}{`⟨*H1*⟩`}{`⟨*C1*⟩`}{`⟨*L1*⟩`}{`⟨*H2*⟩`}{`⟨*C2*⟩`}{`⟨*L2*⟩`}`

```
\newcommand*{\htwotoneleft}[1][all]{%
  \def\ff@pages{#1}%
  \@htwotoneleft
}
\@onlypreamble{\htwotoneleft}
```

`\@htwotoneleft`

```
\newcommand*{\@htwotoneleft}[2][0pt]{\@@htwotoneleft{#1}{#2}}
```

`\htwotoneright` Two tone horizontal strips along right border

```
\newcommand*{\htwotoneright}[1][all]{%
  \def\ff@pages{#1}%
  \@htwotoneright
}
\@onlypreamble{\htwotoneright}
```

`\@htwotoneright`

```
\newcommand*{\@htwotoneright}[2][0pt]{\@@htwotoneright{#1}{#2}}
```

`\@@htwotoneright`

```
\newcommand*{\@@htwotoneright}[8]{%
  \computerightedgeodd{\@ff@tmp@x}%
  \if@twoside
    \computerightedgeeven{\@ff@tmp@x@even}%
  \else
    \setlength{\@ff@tmp@x@even}{\@ff@tmp@x}%
  \fi
  \computebottomedge{\@ff@tmp@y}%
  \addtolength{\@ff@tmp@y}{#1}%
  \addtolength{\@ff@tmp@x}{-#2}%
  \addtolength{\@ff@tmp@x@even}{-#2}%
  \@nexthband{\ff@pages}{#2}{#3}{#4}{#5}%
  \@nexthband{\ff@pages}{#2}{#6}{#7}{#8}%
}
```

`\hNtone` Now for ⟨*N*⟩ coloured horizontal strips

```
\newcommand*{\hNtone}[1][all]{%
  \def\ff@pages{#1}%
  \@hNtone
}
\@onlypreamble{\hNtone}
```

`\@hNtone`

```
\newcommand*{\@hNtone}[2][0pt]{%
  \@@hNtone{#1}{#2}{\paperwidth}%
}
```

`\@@hNtone`

```
\newcommand*{\@@hNtone}[3]{%
  \computeleftedgeodd{\@ff@tmp@x}%
  \if@twoside
    \computeleftedgeeven{\@ff@tmp@x@even}%
  \else
    \setlength{\@ff@tmp@x@even}{\@ff@tmp@x}%
  \fi
  \computebottomedge{\@ff@tmp@y}%
  \addtolength{\@ff@tmp@y}{#1}%
  \@thisstrip=#2\relax
  \setlength{\@ff@staticW}{#3}%
  \@nexthNband
}
```

`\hNtoneleft` Now for the N tone strips along the left border

```
\newcommand*{\hNtoneleft}[1][all]{%
  \def\ff@pages{#1}%
  \@hNtoneleft
}
\@onlypreamble{\hNtoneleft}
```

`\@hNtoneleft`

```
\newcommand*{\@hNtoneleft}[3][0pt]{%
  \@@hNtone{#1}{#2}{#3}%
}
```

`\hNtoneright` Border strip along the right border

```
\newcommand*{\hNtoneright}[1][all]{%
  \def\ff@pages{#1}%
  \@hNtoneright
}
\@onlypreamble{\hNtoneright}
```

\@hNtoneright

```
\newcommand*{\@hNtoneright}[3][0pt]{%
  \@@hNtoneright{#1}{#2}{#3}%
}
```

\@@hNtoneright

```
\newcommand*{\@@hNtoneright}[3]{%
  \computerightedgeodd{\@ff@tmp@x}%
  \if@twoside
    \computerightedgeeven{\@ff@tmp@x@even}%
  \else
    \setlength{\@ff@tmp@x@even}{\@ff@tmp@x}%
  \fi
  \computebottomedge{\@ff@tmp@y}%
  \addtolength{\@ff@tmp@y}{#1}%
  \addtolength{\@ff@tmp@x}{-#3}%
  \addtolength{\@ff@tmp@x@even}{-#3}%
  \@thisstrip=#2\relax
  \setlength{\@ff@staticW}{#3}%
  \@nexthNband
}
```

\@nexthband Make next static frame. Syntax:
`\@nexthband{`⟨*pages*⟩`}{`⟨*width*⟩`}{`⟨*height*⟩`}{`⟨*colour specs*⟩`}{`⟨*label*⟩`}`
x and *y* offsets are given by `\@ff@tmp@x` and `\@ff@tmp@y`. On exit, `\@ff@tmp@y` is set to the top border.

```
\newcommand*{\@nexthband}[5]{%
  \setlength{\@ff@staticH}{#3}%
  \ifthenelse{\equal{#5}{}}%
  {%
    \newstaticframe[#1]{#2}{\@ff@staticH}{\@ff@tmp@x}{\@ff@tmp@y}%
  }%
  {%
    \newstaticframe[#1]{#2}{\@ff@staticH}{\@ff@tmp@x}{\@ff@tmp@y}[#5]%
  }%
  \expandafter\global\expandafter
    \setlength\csname @sf@\romannumeral\c@maxstatic @evenx\endcsname
      {\@ff@tmp@x@even}%
  \@setframecol#4\end{\c@maxstatic}{backcol}{sf}%
  \addtolength{\@ff@tmp@y}{\@ff@staticH}%
}
```

\@nexthNband Get next horizontal strip recursively.

```
\newcommand*{\@nexthNband}{%
  \ifnum\@thisstrip>0\relax
   \let\flf@next\@@nexthNband
  \else
   \let\flf@next\relax
  \fi
  \advance\@thisstrip by -1\relax
  \flf@next
}
```

`\@@nexthNband`

```
\newcommand*{\@@nexthNband}[3]{%
  \@nexthband{\ff@pages}{\@ff@staticW}{#1}{#2}{#3}%
  \@nexthNband
}
```

`\makebackgroundframe` Make one big static frame that covers the entire page. This command should come before all other commands that create static frames, otherwise it will obscure all the ones defined before it. Syntax:
`\makebackgroundframe[`⟨*pages*⟩`][`⟨*label*⟩`]`.

```
\newcommand*{\makebackgroundframe}[1][all]{%
  \ifnum\c@maxstatic>0\relax
    \PackageWarning{flowfram}%
    {%
      Background frame is not first static frame to be
      defined. All previously defined static frames may be
      obscured.%
    }%
  \fi
  \computeleftedgeodd{\@ff@tmp@x}%
  \if@twoside
    \computeleftedgeeven{\@ff@tmp@x@even}%
  \else
    \setlength{\@ff@tmp@x@even}{\@ff@tmp@x}%
  \fi
  \computebottomedge{\@ff@tmp@y}%
  \newstaticframe[#1]{\paperwidth}{\paperheight}{\@ff@tmp@x}%
    {\@ff@tmp@y}%
  \expandafter\global\expandafter
    \setlength\csname @sf@\romannumeral\c@maxstatic @evenx\endcsname
      {\@ff@tmp@x@even}%
}
```

1.10.3 Lines Between Frames

`\insertvrule` Insert a static frame between two frames with a vertical rule that goes from the maximum height of the highest to the minimum height of the lowest, equidistant from both frames. Syntax:

`\insertvrule[`⟨*y top*⟩`][`⟨*y bottom*⟩`]{`⟨*frame1 type*⟩`}{`⟨*IDN1*⟩`}{`⟨*frame2 type*⟩`}`⟨*IDN2*⟩. The starred version uses IDLs instead of IDNs. The optional arguments indicate to continue above the highest point by ⟨*y top*⟩ or continue below the lowest point by ⟨*y bottom*⟩.

`\ffcolumnseprule` This has changed in v1.09. Define `\ffcolumnseprule` and use instead of `\columnseprule`

```
\newlength\ffcolumnseprule
\setlength{\ffcolumnseprule}{2pt}
```

`\ffruledeclarations` This can be redefined to use declarations that affect how the rule appears. For example, it can be used to set the colour of the rule.

```
\newcommand*{\ffruledeclarations}{}
```

`\insertvrule` Determine whether or not the starred version is being used.

```
\newcommand*{\insertvrule}{\@ifstar\@sinsertvrule\@insertvrule}
```

`\@insertvrule` Two optional arguments required.

```
\newcommand*{\@insertvrule}[1][0pt]{%
  \@ifnextchar[{\@@insertvrule[#1]}{\@@insertvrule[#1][0pt]}%
}
```

Need some lengths:

```
\newlength\@ff@left@x
\newlength\@ff@left@y
\newlength\@ff@left@evenx
\newlength\@ff@left@eveny
\newlength\@ff@left@width
\newlength\@ff@left@height
```

`\@@insertvrule` Arguments all accounted for. Convert the frame type into a number to make life easier

```
\def\@@insertvrule[#1][#2]#3#4#5#6{%
  \ifthenelse{\equal{#3}{flow}}%
  {%
    \def\@ff@type@i{1}%
  }%
  {%
    \ifthenelse{\equal{#3}{static}}%
    {%
      \def\@ff@type@i{2}%
    }%
    {%
      \ifthenelse{\equal{#3}{dynamic}}%
      {%
        \def\@ff@type@i{3}%
      }%
      {%
```

```
        \PackageError{flowfram}%
        {Unknown frame type '#3'}%
        {%
           Available frame types are: 'flow', 'static'
           or 'dynamic'%
        }%
      }%
    }%
  }%
  \ifthenelse{\equal{#5}{flow}}%
  {%
    \def\@ff@type@ii{1}%
  }%
  {%
    \ifthenelse{\equal{#5}{static}}%
    {%
      \def\@ff@type@ii{2}%
    }%
    {%
      \ifthenelse{\equal{#5}{dynamic}}%
      {%
        \def\@ff@type@ii{3}%
      }%
      {%
        \PackageError{flowfram}%
        {Unknown frame type '#5'}%
        {%
          Available frame types are: 'flow', 'static'
          or 'dynamic'%
        }%
      }%
    }%
  }%
  \@@insert@vrule{#1}{#2}{\@ff@type@i}{#4}{\@ff@type@ii}{#6}%
}
```

`\@@insert@vrule` Insert a new static frame between the two specified frames. Check to make sure which one is on the left and which one is on the right. Syntax:
`\@@insert@vrule{`⟨*y top*⟩`}{`⟨*y bottom*⟩`}{`⟨*type ID*⟩`}{`⟨*IDN*⟩`}{`⟨*type ID*⟩`}{`⟨*IDN*⟩`}`.

```
\newcommand*{\@@insert@vrule}[6]{%
  \@ff@getdim{#3}{#4}%
  \setlength{\@ff@left@x}{\ffareax}%
  \setlength{\@ff@left@y}{\ffareay}%
  \setlength{\@ff@left@width}{\ffareawidth}%
  \setlength{\@ff@left@height}{\ffareaheight}%
  \@ff@getdim{#5}{#6}%
  \ifnum\@ff@left@x>\ffareax\relax
    \@ff@swaplen{\@ff@left@x}{\ffareax}%
    \@ff@swaplen{\@ff@left@y}{\ffareax}%
```

```
      \@ff@swaplen{\@ff@left@evenx}{\ffareaevenx}%
      \@ff@swaplen{\@ff@left@eveny}{\ffareaevenx}%
      \@ff@swaplen{\@ff@left@width}{\ffareawidth}%
      \@ff@swaplen{\@ff@left@height}{\ffareaheight}%
    \fi
    \setlength{\@ff@tmp@x}{\@ff@left@x}
    \addtolength{\@ff@tmp@x}{\@ff@left@width}%
    \setlength{\@ff@staticW}{\ffareax}%
    \addtolength{\@ff@staticW}{-\@ff@tmp@x}%
    \setlength{\@ff@staticH}{\@ff@left@y}%
    \addtolength{\@ff@staticH}{\@ff@left@height}%
    \setlength{\@ff@tmp@y}{\ffareay}%
    \addtolength{\@ff@tmp@y}{\ffareaheight}%
    \ifnum\@ff@tmp@y>\@ff@staticH
      \setlength{\@ff@staticH}{\@ff@tmp@y}%
    \fi
    \ifnum\@ff@left@y<\ffareay\relax
      \setlength{\@ff@tmp@y}{\@ff@left@y}%
    \else
      \setlength{\@ff@tmp@y}{\ffareay}%
    \fi
    \addtolength{\@ff@staticH}{-\@ff@tmp@y}%
    \newstaticframe{\@ff@staticW}{\@ff@staticH}%
      {\@ff@tmp@x}{\@ff@tmp@y}%
    \addtolength{\@ff@staticH}{#1}%
    \addtolength{\@ff@staticH}{#2}%
    \setstaticcontents{\c@maxstatic}{%
    \ffruledeclarations
    \ffvrule{#2}{\ffcolumnseprule}{\@ff@staticH}}%
    \ifcase#3\relax
      \or \edef\@ff@pages{\csname @ff@pages@\romannumeral#4\endcsname}%
      \or \edef\@ff@pages{\csname @sf@pages@\romannumeral#4\endcsname}%
      \or \edef\@ff@pages{\csname @df@pages@\romannumeral#4\endcsname}%
    \fi
    \setstaticframe{\c@maxstatic}{pages=\@ff@pages}%
```

Check the difference between odd and even page co-ordinates and shift new frame in same direction. (Assumes the two original frames stay in the same relative position.)

```
    \addtolength{\@ff@tmp@x}{\@ff@left@evenx}%
    \addtolength{\@ff@tmp@x}{-\@ff@left@x}%
    \addtolength{\@ff@tmp@y}{\@ff@left@eveny}%
    \addtolength{\@ff@tmp@y}{-\@ff@left@y}%
    \setstaticframe{\c@maxstatic}{evenx=\@ff@tmp@x,eveny=\@ff@tmp@y}%
  }
```

`\ffvrule` `\ffvrule{`⟨*offset*⟩`}{`⟨*width*⟩`}{`⟨*height*⟩`}`

Draws the rule for `\insertvrule`

```
  \newcommand*{\ffvrule}[3]{%
  \hfill \rule[-#1]{#2}{#3}\hfill\mbox{}%
}
```

`\@sinsertvrule` Starred version. Two optional arguments required.

```
\newcommand*{\@sinsertvrule}[1][0pt]{%
  \@ifnextchar[{\@@sinsertvrule[#1]}{\@@sinsertvrule[#1][0pt]}%
}
```

`\@@sinsertvrule` Find out the frame types and their IDN.

```
\def\@@sinsertvrule[#1][#2]#3#4#5#6{%
  \ifthenelse{\equal{#3}{flow}}%
  {%
    \def\@ff@type@i{1}%
    \@flowframeid{#4}%
    \@ff@tmpN=\ff@id
  }%
  {%
    \ifthenelse{\equal{#3}{static}}%
    {%
      \def\@ff@type@i{2}\@staticframeid{#4}\@ff@tmpN=\ff@id
    }%
    {%
      \ifthenelse{\equal{#3}{dynamic}}%
      {%
        \def\@ff@type@i{3}%
        \@dynamicframeid{#4}%
        \@ff@tmpN=\ff@id
      }%
      {%
        \PackageError{flowfram}%
        {Unknown frame type ’#3’}%
        {%
          Available frame types are: ’flow’, ’static’
          or ’dynamic’%
        }%
      }%
    }%
  }%
  \ifthenelse{\equal{#5}{flow}}%
  {%
    \def\@ff@type@ii{1}\@flowframeid{#6}%
  }%
  {%
    \ifthenelse{\equal{#5}{static}}%
    {%
      \def\@ff@type@ii{2}%
      \@staticframeid{#6}%
    }%
    {%
```

```
      \ifthenelse{\equal{#5}{dynamic}}%
      {%
        \def\@ff@type@ii{3}%
        \@dynamicframeid{#6}%
      }%
      {%
        \PackageError{flowfram}%
        {Unknown frame type '#5'}%
        {%
          Available frame types are: 'flow', 'static'
          or 'dynamic'%
        }%
      }%
    }%
  }%
  \@@insert@vrule{#1}{#2}{\@ff@type@i}{\@ff@tmpN}%
    {\@ff@type@ii}{\ff@id}%
}
```

`\inserthrule` Now for a horizontal rule. Syntax similar to `\insertvrule`. Determine whether or not the starred version is being used.

```
\newcommand*{\inserthrule}{\@ifstar\@sinserthrule\@inserthrule}
```

`\@inserthrule` Two optional arguments required.

```
\newcommand*{\@inserthrule}[1][0pt]{%
  \@ifnextchar[{\@@inserthrule[#1]}{\@@inserthrule[#1][0pt]}%
}
```

`\@@inserthrule` Arguments all accounted for. Convert the frame type into a number to make life easier

```
\def\@@inserthrule[#1][#2]#3#4#5#6{%
  \ifthenelse{\equal{#3}{flow}}%
  {%
    \def\@ff@type@i{1}%
  }%
  {%
    \ifthenelse{\equal{#3}{static}}%
    {%
      \def\@ff@type@i{2}%
    }%
    {%
      \ifthenelse{\equal{#3}{dynamic}}%
      {%
        \def\@ff@type@i{3}}%
      {%
        \PackageError{flowfram}%
        {Unknown frame type '#3'}%
        {%
          Available frame types are: 'flow', 'static'
          or 'dynamic'%
        }%
      }%
    }%
  }%
  \ifthenelse{\equal{#5}{flow}}%
  {%
    \def\@ff@type@ii{1}%
  }%
  {%
    \ifthenelse{\equal{#5}{static}}%
    {%
      \def\@ff@type@ii{2}%
    }%
    {%
      \ifthenelse{\equal{#5}{dynamic}}%
      {%
        \def\@ff@type@ii{3}%
      }%
      {%
        \PackageError{flowfram}%
        {Unknown frame type '#5'}%
        {%
          Available frame types are: 'flow', 'static'
          or 'dynamic'%
        }%
      }%
    }%
  }%
  \@@insert@hrule{#1}{#2}{\@ff@type@i}{#4}{\@ff@type@ii}{#6}%
}
```

`\@@insert@hrule` Insert a new static frame between the two specified frames. Check to make sure which one is on the top and which one is on the bottom. Syntax: `\@@insert@hrule{`⟨*x left*⟩`}{`⟨*x right*⟩`}{`⟨*type ID*⟩`}{`⟨*IDN*⟩`}{`⟨*type ID*⟩`}{`⟨*IDN*⟩`}`.

```
\newcommand*{\@@insert@hrule}[6]{%
  \@ff@getdim{#3}{#4}%
  \setlength{\@ff@left@x}{\ffareax}%
  \setlength{\@ff@left@y}{\ffareay}%
  \setlength{\@ff@left@width}{\ffareawidth}%
  \setlength{\@ff@left@height}{\ffareaheight}%
  \@ff@getdim{#5}{#6}%
  \ifnum\@ff@left@y>\ffareay\relax
    \@ff@swaplen{\@ff@left@x}{\ffareax}%
    \@ff@swaplen{\@ff@left@y}{\ffareay}%
    \@ff@swaplen{\@ff@left@width}{\ffareawidth}%
    \@ff@swaplen{\@ff@left@height}{\ffareaheight}%
  \fi
  \setlength{\@ff@tmp@y}{\@ff@left@y}%
  \addtolength{\@ff@tmp@y}{\@ff@left@height}%
  \setlength{\@ff@staticH}{\ffareay}%
  \addtolength{\@ff@staticH}{-\@ff@tmp@y}%
  \setlength{\@ff@staticW}{\@ff@left@x}%
  \addtolength{\@ff@staticW}{\@ff@left@width}%
  \setlength{\@ff@tmp@x}{\ffareax}%
  \addtolength{\@ff@tmp@x}{\ffareawidth}%
  \ifnum\@ff@tmp@x>\@ff@staticW\relax
    \setlength{\@ff@staticW}{\@ff@tmp@x}%
  \fi
  \ifnum\@ff@left@x<\ffareax\relax
    \setlength{\@ff@tmp@x}{\@ff@left@x}%
  \else
    \setlength{\@ff@tmp@x}{\ffareax}%
  \fi
  \addtolength{\@ff@staticW}{-\@ff@tmp@x}%
  \newstaticframe{\@ff@staticW}{\@ff@staticH}%
    {\@ff@tmp@x}{\@ff@tmp@y}%
  \addtolength{\@ff@staticW}{#1}%
  \addtolength{\@ff@staticW}{#2}%
  \setstaticcontents{\c@maxstatic}%
  {%
    \ffruledeclarations
    \ffhrule{#1}{\@ff@staticW}{\ffcolumnseprule}%
  }%
  \ifcase#3\relax
    \or \edef\@ff@pages{\csname @ff@pages@\romannumeral#4\endcsname}%
    \or \edef\@ff@pages{\csname @sf@pages@\romannumeral#4\endcsname}%
    \or \edef\@ff@pages{\csname @df@pages@\romannumeral#4\endcsname}%
  \fi
  \setstaticframe{\c@maxstatic}{pages=\@ff@pages}%
  \addtolength{\@ff@tmp@x}{\@ff@left@evenx}%
  \addtolength{\@ff@tmp@x}{-\@ff@left@x}%
  \addtolength{\@ff@tmp@y}{\@ff@left@eveny}%
  \addtolength{\@ff@tmp@y}{-\@ff@left@y}%
  \setstaticframe{\c@maxstatic}{evenx=\@ff@tmp@x,eveny=\@ff@tmp@y}%
}
```

`\ffhrule` `\ffhrule{`⟨*offset*⟩`}{`⟨*width*⟩`}{`⟨*height*⟩`}`

Draws the rule for `\inserthrule`

```
\newcommand*{\ffhrule}[3]{%
  \hspace*{-#1}\rule{#2}{#3}%
}
```

`\@sinserthrule` Starred version. Two optional arguments required.

```
\newcommand*{\@sinserthrule}[1][0pt]{%
  \@ifnextchar[{\@@sinserthrule[#1]}{\@@sinserthrule[#1][0pt]}%
}
```

`\@@sinserthrule` Find out the frame types and their IDN.

```
\def\@@sinserthrule[#1][#2]#3#4#5#6{%
  \ifthenelse{\equal{#3}{flow}}%
  {%
    \def\@ff@type@i{1}%
    \@flowframeid{#4}%
    \@ff@tmpN=\ff@id
  }%
  {%
    \ifthenelse{\equal{#3}{static}}%
    {%
      \def\@ff@type@i{2}%
      \@staticframeid{#4}%
      \@ff@tmpN=\ff@id
    }%
    {%
      \ifthenelse{\equal{#3}{dynamic}}%
      {%
        \def\@ff@type@i{3}%
        \@dynamicframeid{#4}%
        \@ff@tmpN=\ff@id
      }%
      {%
        \PackageError{flowfram}%
        {Unknown frame type '#3'}%
        {%
          Available frame types are: 'flow', 'static'
          or 'dynamic'%
        }%
      }%
    }%
  }%
  \ifthenelse{\equal{#5}{flow}}%
  {%
    \def\@ff@type@ii{1}%
    \@flowframeid{#6}%
  }%
  {%
    \ifthenelse{\equal{#5}{static}}%
    {%
      \def\@ff@type@ii{2}%
      \@staticframeid{#6}%
    }%
    {%
      \ifthenelse{\equal{#5}{dynamic}}%
      {%
        \def\@ff@type@ii{3}%
        \@dynamicframeid{#6}%
      }%
```

```
        {%
          \PackageError{flowfram}%
          {Unknown frame type '#5'}%
          {%
            Available frame types are: 'flow', 'static'
            or 'dynamic'%
          }%
        }%
      }%
    }%
    \@@insert@hrule{#1}{#2}{\@ff@type@i}{\@ff@tmpN}%
      {\@ff@type@ii}{\ff@id}%
  }
```

1.11 Putting Chapter Headings in Dynamic Frames

`\dfchaphead` Provide facility to make chapter headings appear in specified dynamic frame. I originally called this macro `\putchapterheadingsindynamicframe` which was descriptive, but overly long, so I changed it to the rather more cryptic name `\dfchaphead`. If the starred form is used, the frame is identified by IDL, the unstarred form identifies the frame IDN.

```
\newcommand*{\dfchaphead}{%
  \@ifstar\@sdynamicchap\@dynamicchap
}
```

Define style for the chapter heading. These commands are should only be used when `\dfchaphead` has been used.

`\DFchapterstyle`

```
\newcommand{\DFchapterstyle}[1]{#1}
```

`\DFschapterstyle`

```
\newcommand{\DFschapterstyle}[1]{#1}
```

`\@dynamicchap` Unstarred version.

```
\newcommand{\@dynamicchap}[1]{%
  \@ifundefined{chapter}%
  {%
    \PackageError{flowfram}%
    {Chapters aren't defined}%
    {%
      The document class you are using does not
      define chapters%
    }%
  }%
  {%
```

Store current chapter head definitions for starred and unstarred versions

```
      \let\@ff@OLDmakechapterhead\@makechapterhead
      \let\@ff@OLDmakeschapterhead\@makeschapterhead
```

Define user commands that can be redefined to modify the chapter head style (in the event that the user is using a class that doesn't provide an easy means to do this.)

```
      \renewcommand{\DFchapterstyle}[1]{\@ff@OLDmakechapterhead{##1}}%
      \renewcommand{\DFschapterstyle}[1]{\@ff@OLDmakeschapterhead{##1}}%
```

Redefine chapter heads so that they put their contents in the requested dynamic frame. First the unstarred version:

```
      \xdef\@makechapterhead##1{%
        \noexpand\@setdynamiccontents{\number#1}%
        {%
          \noexpand\DFchapterstyle{##1}%
        }%
      }%
```

Now the starred version:

```
      \xdef\@makeschapterhead##1{%
        \noexpand\@setdynamiccontents{\number#1}%
        {%
          \noexpand\DFschapterstyle{##1}%
        }%
      }%
    }%
  }
```

`\@sdynamicchap` Starred form.

```
  \newcommand{\@sdynamicchap}[1]{%
    \@dynamicframeid{#1}%
    \@dynamicchap{\ff@id}%
  }
```

There is no facility for placing other sectional types in dynamic frames. This is because, either (1) the sectioning command does not start a new page, in which case there is no way of telling where exactly the new section will start, and having a section title in some other location on the page is ambiguous, and would really confuse the reader, or (2) in the case of `\part` in report or book class files, the title appears on a page of its own, so where is the point in putting it in a dynamic frame?

1.12 Thumbtabs

Define counter to keep track of total number of thumbtabs.

```
  \newcounter{maxthumbtabs}
```

`\defaultthumbtabtype` Check to see if chapters are defined, if they are make thumbtabs correspond to chapters, otherwise make thumbtabs correspond to sections.

```
\@ifundefined{chapter}%
{%
  \newcommand*{\defaultthumbtabtype}{section}%
}%
{%
  \newcommand*{\defaultthumbtabtype}{chapter}%
}
```

\@ttb@type Section type to assign to thumbtabs.

```
\newcommand*{\@ttb@type}{\defaultthumbtabtype}
```

\makethumbtabs Make the thumbtabs. Read in information from `.ttb` file, and open it for output. Syntax:

`\makethumbtabs[`⟨*y offset*⟩`]{`⟨*height*⟩`}[`⟨*sec type*⟩`]`.

First check to see if there is a second optional argument.

```
\newcommand*{\makethumbtabs}[2][0pt]{%
  \@ifnextchar[%
  {\@makethumbtabs[#1]{#2}}%
  {%
    \@makethumbtabs[#1]{#2}[\defaultthumbtabtype]%
  }%
}
```

\@makethumbtabs Now all arguments are known, first redefine the appropriate sectioning command, then input the ttb file, and create the thumbtabs.

```
\def\@makethumbtabs[#1]#2[#3]{%
  \@ifundefined{#3}%
  {%
    \PackageError{flowfram}%
    {%
      Unknown section type '#3'%
    }%
    {}%
  }%
  {%
    \renewcommand{\@ttb@type}{#3}%
    \ifthenelse{\equal{#3}{chapter}}%
    {%
      \@makethumbchapter
    }%
    {%
      \ifthenelse{\equal{#3}{part}}%
      {\@makethumbpart}%
      {%
        \@makethumbsection{#3}%
      }%
    }%
  }%
  \@starttoc{ttb}%
  \@dothumbtabs{#1}{#2}%
}
```

`\@makethumbchapter` If thumbtabs correspond to chapters, redefine `\@chapter` so that each unstarred chapter writes an entry to the `.ttb` file.

```
\newcommand{\@makethumbchapter}{%
  \let\@ttb@old@chapter\@chapter
  \def\@chapter[##1]##2{%
    \@ttb@old@chapter[##1]{##2}%
    \addtocontents{ttb}{\protect\thumbtab
      {\thepage}{\thechapter}{##1}{chapter.\thechapter}}%
    \@afterheading
  }%
}
```

`\@makethumbpart` For parts in books or reports, the thumbtab needs to be saved after the part counter has been incremented, but before the page break so that the page number and part numbers are correct. If `\@endpart` is not defined, then the document class probably does not start a new page after `\part`. (This can't be guaranteed for non standard class files, but there's nothing that can be done about that.) If this happens, just redefine `\@part`, and hope for the best.

```
\newcommand{\@makethumbpart}{%
  \let\@ttb@old@part\@part
  \@ifundefined{@endpart}%
  {%
    \def\@part[##1]##2{%
      \@ttb@old@part[##1]{##2}%
      \addtocontents{ttb}{\protect\thumbtab
        {\thepage}{\thepart}{##1}{part.\thepage}}%
      \@afterheading
    }%
  }%
  {%
    \let\@ttb@old@endpart\@endpart
    \def\@part[##1]##2{%
      \def\@parttitle{##1}%
      \@ttb@old@part[##1]{##2}%
    }%
    \def\@endpart{%
      \addtocontents{ttb}%
      {%
       \protect\thumbtab{\thepage}%
         {\thepart}{\@parttitle}{part.\thepage}%
      }%
      \@ttb@old@endpart
    }%
  }%
}
```

`\@makethumbsection` Thumbtabs defined for one of the remaining standard sectioning commands. Since these commands use `\@startsection`, it is necessary to redefine `\@sect` to add the thumbtab information to the .ttb file.

```
\newcommand*{\@makethumbsection}[1]{%
  \let\@ttb@old@sect=\@sect
  \def\@sect##1##2##3##4##5##6[##7]##8{%
    \@ttb@old@sect{##1}{##2}{##3}{##4}{##5}{##6}[##7]{##8}%
    \ifthenelse{\equal{##1}{#1}}%
    {%
      \addtocontents{ttb}%
      {%
        \protect\thumbtab{\thepage}{\csname the#1\endcsname}%
          {##7}{#1.\csname the#1\endcsname}%
      }%
      \@afterheading
    }%
    {}%
  }%
}
```

`\thumbtab` The thumbtab file, `.ttb`, will have a series of `\thumbtab` commands, when this file is read in, just store the information for now.

```
\newcommand{\thumbtab}[4]{%
  \stepcounter{maxthumbtabs}%
  \expandafter
    \gdef\csname thumbtab@pages@\romannumeral\c@maxthumbtabs\endcsname{#1}%
  \expandafter
    \gdef\csname thumbtab@num@\romannumeral\c@maxthumbtabs\endcsname{#2}%
  \expandafter
    \gdef\csname thumbtab@title@\romannumeral\c@maxthumbtabs\endcsname{#3}%
  \expandafter
    \gdef\csname thumbtab@link@\romannumeral\c@maxthumbtabs\endcsname{#4}%
}
```

`\@dothumbtabs` Once the thumbtab information has been read in and stored in the thumbtab macros, create the thumbtabs using this information. First need to work out the page ranges between each thumbtab. If the following thumbtab starts on the same page as the previous one, leave the page varable as a single number (this may happen if the thumbtabs correspond to sections rather than chapters). If the following thumbtab starts on a different page to the one before it, the preceding thumbtab page variable so be a range from its own initial page up to the page before the next thumbtab starts. The final thumbtab has an open ended range. This final thumbtab will continue to be displayed until cancelled by `\disablethumbtabs`.

Syntax: `\@dothumbtabs{`⟨*y offset*⟩`}{`⟨*height*⟩`}`.

```
\newcommand*{\@dothumbtabs}[2]{%
  \@colN=0\relax
  \whiledo{\@colN<\c@maxthumbtabs}%
  {%
    \advance\@colN by 1\relax
    \edef\ff@pages{%
      \csname thumbtab@pages@\romannumeral\@colN\endcsname}%
    \ifnum\@colN=\c@maxthumbtabs
      \expandafter
        \xdef\csname thumbtab@pages@\romannumeral\@colN\endcsname{%
          \ff@pages,>\ff@pages}%
    \else
      \advance\@colN by 1\relax
      \edef\ff@endpage{%
        \csname thumbtab@pages@\romannumeral\@colN\endcsname}%
      \advance\@colN by -1\relax
      \@ff@tmpN=\ff@endpage\relax
      \advance\@ff@tmpN by -1\relax
      \ifnum\@ff@tmpN>\ff@pages
        \expandafter
          \xdef\csname thumbtab@pages@\romannumeral\@colN\endcsname{%
            \ff@pages-\number\@ff@tmpN}%
      \fi
    \fi
  }%
  \@@dothumbtabs{#1}{#2}%
}
```

`\thumbtabwidth` Default thumbtab width.

```
\newlength{\thumbtabwidth}
\setlength{\thumbtabwidth}{1cm}
```

`\thumbtabindexformat` Thumbtab format. If hyperlinks have been defined, use a hyperlink in the thumbtab index. Syntax: `\thumbtabindexformat{`⟨*link*⟩`}{`⟨*text*⟩`}{`⟨*height*⟩`}`

```
\@ifundefined{hyperlink}%
{%
  \newcommand{\thumbtabindexformat}[3]{%
    \thumbtabformat{#2}{#3}%
  }%
}%
{%
  \newcommand{\thumbtabindexformat}[3]{%
    \hyperlink{#1}{\thumbtabformat{#2}{#3}}%
  }%
}
```

`\thumbtabformat` Individual thumbtab format. If rotating has been disabled, stack the letters vertically (this doesn't look very good). Syntax: `\thumbtabformat{`⟨*text*⟩`}{`⟨*height*⟩`}`

```
\newcommand{\thumbtabformat}[2]{%
  \if@ttb@rotate
    \rotatebox{-90}%
```

```
      {%
        \parbox[c][\thumbtabwidth]{#2}{%
          \centering#1%
        }%
      }%
    \else
      \parbox[c][#2]{\thumbtabwidth}{%
        \centering\@ttb@stack{#1}%
      }%
    \fi
  }
```

`\@flf@subsp` Substitute spaces for `\space`. Stores resulting text in `\@flf@subsptext` which should be set to empty before use.

```
\def\@flf@subsp#1 #2{%
  \expandafter\flf@ta\expandafter{\@flf@subsptext}%
  \flf@tb{#1}%
  \edef\@flf@subsptext{\the\flf@ta\the\flf@tb}%
  \def\@flf@tmp{#2}%
  \ifx\@flf@tmp\@nnil
    \let\@flf@donextsubsp=\@gobble
  \else
    \expandafter\flf@ta\expandafter{\@flf@subsptext}%
    \edef\@flf@subsptext{\the\flf@ta\noexpand\space}%
    \let\@flf@donextsubsp=\@flf@subsp
  \fi
  \@flf@donextsubsp{#2}%
}
```

`\@ttb@stack` Stack letters vertically. Any spaces first need to be substituted with `\space`, otherwise they will be ignored.

```
\newcommand{\@ttb@stack}[1]{%
  \def\@flf@subsptext{}%
  \expandafter\@flf@subsp#1 \@nil\relax
  \begin{tabular}{l}%
  \expandafter\@@ttb@stack\@flf@subsptext\@nil\relax
  \end{tabular}%
}
```

`\@@ttb@stack`

```
\def\@@ttb@stack#1#2{%
  \def\@flf@tmp{#1}%
  \ifx\@flf@tmp\@nnil
    \let\flf@next\relax
  \else
    #1\\%
    \def\@flf@tmp{#2}%
    \ifx\@nnil#2\relax
      \let\flf@next\@gobble
    \else
      \let\flf@next\@@ttb@stack
    \fi
  \fi
  \flf@next{#2}%
}
```

`\@greyscale` Count register to compute the grey scale.

```
  \newcount\@greyscale
```

`\@@dothumbtabs` Once the page range have been sorted, create the dynamic frames associated with each thumbtab. Thumbtabs will initially have a grey background, but this can be changed by the user. Each thumbtab is given an IDL `thumbtab`⟨*n*⟩ where ⟨*n*⟩ is the index of the thumbtab (starting from 1 for the topmost thumbtab.) Each frame in the thumbtab index is given an IDL `thumbtabindex`⟨*n*⟩, where ⟨*n*⟩ is as before.

```
  \newcommand{\@@dothumbtabs}[2]{%
    \setlength{\@ff@tmp@y}{\textheight}%
    \addtolength{\@ff@tmp@y}{-#2}%
    \addtolength{\@ff@tmp@y}{-#1}%
    \computerightedgeodd{\@ff@tmp@x}%
    \addtolength{\@ff@tmp@x}{-\thumbtabwidth}%
    \computeleftedgeeven{\@ff@tmp@x@even}%
    \@ff@tmpN=0\relax
    \whiledo{\@ff@tmpN<\c@maxthumbtabs}%
    {%
      \advance\@ff@tmpN by 1\relax
      \@greyscale=\@ff@tmpN\relax
      \multiply\@greyscale by 60\relax
      \divide\@greyscale by \c@maxthumbtabs
      \advance\@greyscale by 25\relax
      \edef\@ff@greyscale{0.\number\@greyscale}%
```

Thumbtab

```
      \newdynamicframe[none]{\thumbtabwidth}{#2}%
        {\@ff@tmp@x}{\@ff@tmp@y}[thumbtab\number\@ff@tmpN]%
      \expandafter\global\expandafter
        \setlength\csname @df@\romannumeral\c@maxdynamic @evenx\endcsname
          {\@ff@tmp@x@even}%
```

set the contents of the dynamic frame

```
      \ifthenelse{\boolean{@ttb@title}\and\boolean{@ttb@num}}%
      {%
        \expandafter
          \xdef\csname @dynamicframe@\romannumeral\c@maxdynamic\endcsname{%
            \noexpand\thumbtabformat
            {%
              \csname thumbtab@num@\romannumeral\@ff@tmpN\endcsname\
              \csname thumbtab@title@\romannumeral\@ff@tmpN\endcsname
            }%
            {#2}%
          }%
      }%
      {%
        \if@ttb@title
          \expandafter
            \xdef\csname @dynamicframe@\romannumeral\c@maxdynamic\endcsname{%
              \noexpand\thumbtabformat
              {%
                \csname thumbtab@title@\romannumeral\@ff@tmpN\endcsname
              }%
              {#2}%
            }%
        \fi
        \if@ttb@num
          \expandafter
            \xdef\csname @dynamicframe@\romannumeral\c@maxdynamic\endcsname{%
              \noexpand\thumbtabformat
              {%
                \csname thumbtab@num@\romannumeral\@ff@tmpN\endcsname
              }%
              {#2}%
            }%
      \fi
      }%
      \expandafter
        \xdef\csname @df@backcol@\romannumeral\c@maxdynamic\endcsname
          {[gray]{\@ff@greyscale}}
```

Thumbtab index

```
  \newdynamicframe[none]{\thumbtabwidth}{#2}%
    {\@ff@tmp@x}{\@ff@tmp@y}[thumbtabindex\number\@ff@tmpN]%
  \expandafter\global\expandafter
    \setlength\csname @df@\romannumeral\c@maxdynamic @evenx\endcsname
      {\@ff@tmp@x@even}%
  \expandafter
```

set the contents of the dynamic frame

```
  \ifthenelse{\boolean{@ttb@title}\and\boolean{@ttb@num}}%
  {%
    \expandafter
      \xdef\csname @dynamicframe@\romannumeral\c@maxdynamic\endcsname{%
        \noexpand\thumbtabindexformat
        {%
          \csname thumbtab@link@\romannumeral\@ff@tmpN\endcsname
        }%
        {%
          \csname thumbtab@num@\romannumeral\@ff@tmpN\endcsname\
          \csname thumbtab@title@\romannumeral\@ff@tmpN\endcsname
          }%
          {#2}%
        }%
    }%
    {%
      \if@ttb@title
        \expandafter
        \xdef\csname @dynamicframe@\romannumeral\c@maxdynamic\endcsname{%
          \noexpand\thumbtabindexformat
          {%
            \csname thumbtab@link@\romannumeral\@ff@tmpN\endcsname
          }%
          {%
            \csname thumbtab@title@\romannumeral\@ff@tmpN\endcsname
          }%
          {#2}%
        }%
      \fi
      \if@ttb@num
        \expandafter
            \xdef\csname @dynamicframe@\romannumeral\c@maxdynamic\endcsname{%
              \noexpand\thumbtabindexformat
              {%
                \csname thumbtab@link@\romannumeral\@ff@tmpN\endcsname
              }%
              {%
                \csname thumbtab@num@\romannumeral\@ff@tmpN\endcsname
              }%
              {#2}%
            }%
      \fi
    }%
    \expandafter
      \xdef\csname @df@backcol@\romannumeral\c@maxdynamic\endcsname
        {[gray]{\@ff@greyscale}}
    \addtolength{\@ff@tmp@y}{-#2}%
  }%
}%
```

`\enablethumbtabs`

Enable thumbtabs. Once the IDN is obtained for the first thumbtab, the rest can be found by incrementing the number by 2 (the frames in between correspond to the thumbtab index.)

```
\newcommand*{\enablethumbtabs}{%
  \ifnum\c@maxthumbtabs>0\relax
    \@ff@tmpN=0\relax
    \@dynamicframeid{thumbtab1}%
    \whiledo{\@ff@tmpN<\c@maxthumbtabs}%
    {%
      \advance\@ff@tmpN by 1\relax
```

thumbtab

```
      \edef\@ff@pages{\csname thumbtab@pages@\romannumeral\@ff@tmpN\endcsname}%
      \@@setdynamicframe{\ff@id}{pages=\@ff@pages}%
      \advance\ff@id by 2\relax
    }%
  \else
    \PackageWarning{flowfram}{No thumb tabs defined}%
  \fi
}
```

`\disablethumbtabs` Disable all thumbtabs.

```
\newcommand*{\disablethumbtabs}{%
  \ifnum\c@maxthumbtabs>0\relax
    \@ff@tmpN=0\relax
    \@dynamicframeid{thumbtab1}%
    \whiledo{\@ff@tmpN<\c@maxthumbtabs}%
    {%
      \advance\@ff@tmpN by 1\relax
```

Thumbtab:

```
      \expandafter\xdef\csname @df@pages@\romannumeral\ff@id\endcsname
        {none}%
      \advance\ff@id by 1\relax
```

Thumbtab index:

```
      \expandafter\xdef\csname @df@pages@\romannumeral\ff@id\endcsname
        {none}%
      \advance\ff@id by 1\relax
    }%
  \fi
}
```

`\thumbtabindex` Show thumbtab index on current page. The `\@ff@doafter` bit circumvents the problem of duplicate page numbers, as the table of contents is quite frequently on page i while the first chapter starts on page 1.

```
\newcommand*{\thumbtabindex}{%
  \ifnum\c@maxthumbtabs>0\relax
    \@ff@tmpN=0\relax
    \@dynamicframeid{thumbtabindex1}%
    \whiledo{\@ff@tmpN<\c@maxthumbtabs}%
    {%
      \advance\@ff@tmpN by 1\relax
      \expandafter
        \xdef\csname @df@pages@\romannumeral\ff@id\endcsname{\c@page}%
      \edef\@ff@doafter{%
        \noexpand\afterpage
        {%
          \noexpand\setdynamicframe{\number\ff@id}{pages=none}%
        }%
      }%
      \@ff@doafter
      \advance\ff@id by 2\relax
    }%
  \fi
}
```

`\setthumbtab` Modify the settings for all the thumbtabs (including thumbtab index). Since the thumbtabs are dynamic frames you could just use `\setdynamicframe`, however, the thumbtabs will not be generated on the first run, as there will be no information in the ttb file, so `\setdynamicframe` would generate an error. `\setthumbtab` will only give a warning message if it can not find the thumbtab. The argument #1 is the index of the thumbtab (starting from 1), the second argument #2 is the frame settings.

```
\newcommand{\setthumbtab}[2]{%
  \ifthenelse{\equal{#1}{all}}%
  {%
    \@ff@tmpN=0\relax
    \whiledo{\@ff@tmpN<\c@maxthumbtabs}%
    {%
      \advance\@ff@tmpN by 1\relax
      \@setthumbtab{\@ff@tmpN}{#2}%
    }%
  }%
  {%
    \@for\@ttb@id:=#1\do{\@setthumbtab{\@ttb@id}{#2}}%
  }%
}
```

`\@setthumbtab` Set individual thumbtab and its index tab.

```
\newcommand{\@setthumbtab}[2]{%
```

Check if this thumbtab exists

```
  \ifthenelse{\(\c@maxthumbtabs<#1\) \or \(#1<1\)}%
  {%
    \PackageWarning{flowfram}%
    {%
      Can’t find thumbtab number ’#1’, ttb file may not be
      up-to-date%
    }%
  }%
  {%
    \@dynamicframeid{thumbtab\number#1}%
    \@@setdynamicframe{\ff@id}{#2}%
    \@dynamicframeid{thumbtabindex\number#1}%
    \@@setdynamicframe{\ff@id}{#2}%
  }%
}
```

`\setthumbtabindex` Only change settings for the thumbtab index. This can take a comma-separated number list.

```
\newcommand{\setthumbtabindex}[2]{%
  \ifthenelse{\equal{#1}{all}}%
  {%
    \@ff@tmpN=0\relax
    \whiledo{\@ff@tmpN<\c@maxthumbtabs}%
    {%
      \advance\@ff@tmpN by 1\relax
      \@setthumbtabindex{\@ff@tmpN}{#2}%
    }%
  }%
  {%
    \@for\@ttb@id:=#1\do{\@setthumbtabindex{\@ttb@id}{#2}}%
  }%
}
```

`\@setthumbtabindex` Change setting for individual thumbtab index entry.

```
\newcommand{\@setthumbtabindex}[2]{%
```

Check if this thumbtab exists

```
  \ifthenelse{\(\c@maxthumbtabs<#1\) \or \(#1<1\)}%
  {%
    \PackageWarning{flowfram}%
    {%
      Can’t find thumbtab number ‘\number#1’,
      ttb file may not be up-to-date%
    }%
  }%
  {%
    \@dynamicframeid{thumbtabindex\number#1}%
    \@@setdynamicframe{\ff@id}{#2}%
  }%
}
```

`\tocandhumbtabindex` Do both the table of contents and the thumbtab index

```
\newcommand*{\tocandthumbtabindex}{%
  \aligntoctrue
  \tableofcontents
  \thumbtabindex
  \aligntocfalse
}
```

1.13 Minitocs

`\@ttb@minitoctype` Sectioning type for the minitoc, by default it is the same as the thumbtabs

```
\newcommand*{\@ttb@minitoctype}{\@ttb@type}
```

`\@starttoc` In order to align the table of contents with the thumbtabs, or to use minitocs, the toc information must be stored, rather than simply input. Therefore, modify `\@starttoc` so that it can store the contents of the file. `\if@storetoc` is used to determine whether to store the contents, or act as normal.

```
\let\@ttb@old@starttoc\@starttoc
\newif\if@storetoc
\@storetocfalse
\renewcommand*{\@starttoc}[1]{%
  \if@storetoc
    \@ttb@storetoc{#1}%
  \else
    \@ttb@old@starttoc{#1}%
  \fi
}
```

`\@ttb@storetoc` store the contents of the file with the given extension

```
\newcommand*{\@ttb@storetoc}[1]{%
  \begingroup
    \makeatletter
    \@storefileconts{\jobname.#1}%
    \if@filesw
      \expandafter\newwrite\csname tf@#1\endcsname
      \immediate\openout\csname tf@#1\endcsname\jobname.#1\relax
    \fi
    \@nobreakfalse
  \endgroup
}
```

`\@storefileconts` Store the contents of named file, if it exists.

```
\newcommand*{\@storefileconts}[1]{%
  \IfFileExists{#1}%
  {%
    \@@storefileconts\@filef@und
  }%
  {%
    \PackageInfo{flowfram}{No file #1.}%
  }%
}
```

store the number of units corresponding to the thumbtab type, and minitoc units. These will usually have the same value, but this is not always guaranteed.

`\c@maxtocunits` Total number of toc units

```
\newcount\c@maxtocunits
```

`\c@maxminitoc` Total number of minitoc units

```
\newcount\c@maxminitoc
```

`\@@storefileconts` Read each line in from the file, and add to the contents list.

```
\newcommand{\@@storefileconts}[1]{%
  \@ifundefined{\@ttb@minitoctype}%
  {%
    \@ttb@minitoclevel=6\relax
  }%
  {%
    \expandafter\@ttb@minitoclevel\expandafter
      =\csname @ttb@\@ttb@minitoctype @level\endcsname
  }%
  \newread\@ttb@toc
  \openin\@ttb@toc=#1\relax
  \c@maxtocunits=0\relax
  \c@maxminitoc=0\relax
  \whiledo{\not\boolean{eof}\@ttb@toc}%
  {%
    \read\@ttb@toc to\tocline
    \@addtotoclist{\tocline}{\c@maxtocunits}%
  }%
  \closein\@ttb@toc
}
```

`\@addtotoclist` Before each line is added to the contents list, it is first checked to see whether the line starts with `\contentsline`. If it does, then check to see if the sectioning type corresponds to the thumbtab level. If it does, then start a new list. There will be `\c@maxtocunits` lists, each one corresponding to each thumbtab group. In addition, each contents line needs to be added to the minitoclists, but only if the sectioning type level is greater than `\@ttb@minitoctype`. The number of minitoc lists is given by `\c@maxminitoc`.

```
\newif\if@contsline
\newcount\@ttb@level
\newcount\@ttb@minitoclevel

\newcommand{\@addtotoclist}[2]{%
  \expandafter\@checkcontentsline#1\end
  \if@contsline
    \expandafter\@gettype#1\end
    \ifthenelse{\equal{\@ttb@contstype}{\@ttb@type}}%
    {%
      \global\advance#2 by 1\relax
    }%
    {}%
  \fi
  \@ifundefined{@toc@\romannumeral#2}%
  {%
    \flf@ta=\expandafter{#1}%
    \expandafter\xdef\csname @toc@\romannumeral#2\endcsname{\the\flf@ta}%
  }%
  {%
    \flf@ta=\expandafter{#1}%
    \flf@tb=\expandafter\expandafter\expandafter
      {\csname @toc@\romannumeral#2\endcsname}%
    \expandafter\xdef\csname @toc@\romannumeral#2\endcsname{%
      \the\flf@tb\the\flf@ta}%
  }%
```

now do minitoc stuff. If the sectioning type is unknown, assume it comes last

```
  \if@minitoc
    \if@contsline
      \@ifundefined{\@ttb@contstype}%
      {\@ttb@level=6}%
      {%
        \@ttb@level=\csname @ttb@\@ttb@contstype @level\endcsname
      }%
      \relax
      \ifnum\@ttb@level=\@ttb@minitoclevel
        \global\advance\c@maxminitoc by 1\relax
        \expandafter
           \gdef\csname @minitoc@\romannumeral\c@maxminitoc\endcsname{}%
      \else
        \ifnum\@ttb@level>\@ttb@minitoclevel
          \flf@ta=\expandafter{#1}\relax
          \flf@tb=\expandafter\expandafter\expandafter
           {\csname @minitoc@\romannumeral\c@maxminitoc\endcsname}\relax
          \expandafter
            \xdef\csname @minitoc@\romannumeral\c@maxminitoc\endcsname{%
             \the\flf@tb\the\flf@ta}
        \fi
      \fi
    \fi
  \fi
}
```

Is there already a way of determining the sectioning level from its name?

```
\def\@ttb@part@level{-1}
\def\@ttb@chapter@level{0}
\def\@ttb@section@level{1}
\def\@ttb@subsection@level{2}
\def\@ttb@subsubsection@level{3}
\def\@ttb@paragraph@level{4}
\def\@ttb@subparagraph@level{5}
```

`\@checkcontentsline` Check to see if line starts with `\contentsline`

```
\long\def\@checkcontentsline#1#2\end{%
  \ifx#1\contentsline
    \@contslinetrue
  \else
    \@contslinefalse
  \fi
}
```

`\@gettype` Given that the line starts with `\contentsline`, extract the first argument of `\contentsline` to get the sectioning type.

```
\def\@gettype\contentsline#1#2\end{%
  \def\@ttb@contstype{#1}%
}
```

`\tableofcontents` Modify `\tableofcontents`. It first stores the contents of the toc file, and then, either simply prints it on the page (so it appears no different to the standard `\tableofcontents`), or it prints it out so that each thumbtab unit has the same height as the thumbtabs. Note: this assumes that the actual table of contents starts at the same height as the thumbtabs. The thumbtab vertical position may need to be adjusted to compensate for space taken up by the contents title.

```
\newif\ifaligntoc
\aligntocfalse
```

Save original definition of `\tableofcontents`

```
\let\@ttb@old@tableofcontents\tableofcontents
```

Redefine `\tableofcontents`

```
\renewcommand{\tableofcontents}{%
  \@storetoctrue
  \@ttb@old@tableofcontents
  \ifaligntoc
    \@printalignedtoc
  \else
    \@printtoc
  \fi
  \@storetocfalse
  \global\c@minitoc=0\relax
}
```

`\beforeminitocskip` Vertical space to add before minitoc.

```
\newlength\beforeminitocskip
\setlength{\beforeminitocskip}{0pt}
```

`\afterminitocskip` Vertical space to add after minitoc.

```
\newlength\afterminitocskip
\setlength{\afterminitocskip}{\baselineskip}
```

`\dominitoc` Do the minitoc for unit `#1`. Check first that minitocs have been enabled.

```
\newcommand*{\dominitoc}[1]{%
  \if@minitoc
    \@dominitoc{#1}%
  \fi
}
\newcommand*{\@dominitoc}[1]{\@@dominitoc{#1}}
```

`\minitocstyle` Style in which to display the minitoc.

```
\newcommand{\minitocstyle}[1]{%
  \normalfont\normalsize\normalcolor
  #1%
}
```

`\@@dominitoc` Now do the actual minitoc for unit #1.

```
\newcommand*{\@@dominitoc}[1]{%
  {%
    \minitocstyle
    {%
      \vskip\beforeminitocskip
      \csname @minitoc@\romannumeral#1\endcsname
    }%
  }%
  \vskip\afterminitocskip
}
```

`\appenddfminitoc` Modify `\dominitoc` so that the minitoc is appended to specified dynamic frame. Starred version uses dynamic frame IDL, unstarred version uses dynamic frame IDN. I originally called this macro `\appendminitoctodynamicframe` but decided it was too long, for I've opted instead for a slightly more cryptic name.

```
\newcommand*{\appenddfminitoc}{%
  \renewcommand{\beforeminitocskip}{\baselineskip}%
  \@ifstar\@sappendminitocdf\@appendminitocdf
}
```

`\@sappendminitocdf` Starred version

```
\newcommand*{\@sappendminitocdf}[1]{%
  \renewcommand{\@dominitoc}[1]{%
    \@sappenddynamic{#1}{\@@dominitoc{##1}}%
  }%
}
```

`\@appendminitocdf` Unstarred version

```
\newcommand*{\@appendminitocdf}[1]{%
  \renewcommand{\@dominitoc}[1]{%
    \@appenddynamic{#1}{\@@dominitoc{##1}}%
  }%
}
```

`\@printtoc` Do the table of contents, which has been stored in `\c@maxtocunits` macros. (or possibly `\c@maxtocunits + 1`, if information was added before the first group—which corresponds to `\@colN=0`.)

```
\newcommand*{\@printtoc}{%
  \@colN=0\relax
  \csname @toc@\romannumeral\@colN\endcsname
  \whiledo{\@colN<\c@maxtocunits}%
  {%
    \advance\@colN by 1\relax
    \csname @toc@\romannumeral\@colN\endcsname
  }%
}
```

`\@printalignedtoc` Print the table of contents so that each unit is has vertical height the same as the height of the thumbtabs. Note that you may have to adjust the vertical offset of the thumbtabs (in `\makethumbtabs`) in order to make them correctly aligned.

```
\newcommand{\@printalignedtoc}{%
  \@ff@tmpN=0\relax
  \@ifundefined{@toc@\romannumeral\@ff@tmpN}%
  {}%
  {%
    \csname @toc@\romannumeral\@ff@tmpN\endcsname
    \par\noindent\hrulefill
  }%
  \whiledo{\@ff@tmpN<\c@maxtocunits}%
  {%
    \advance\@ff@tmpN by 1\relax
    \ifnum\@ff@tmpN>\c@maxthumbtabs
      \csname @toc@\romannumeral\@ff@tmpN\endcsname
    \else
      \@dynamicframeid{thumbtabindex\number\@ff@tmpN}%
      \expandafter\expandafter\expandafter
        \@ff@getstaticpos\csname @df@dim@\romannumeral\ff@id\endcsname
      \vbox to \@ff@tmp@y
      {%
        \noindent\parbox{\linewidth}%
        {%
          \csname @toc@\romannumeral\@ff@tmpN\endcsname
        }%
        \vfill
        \par\noindent\hrulefill
      }%
    \fi
  }%
}
```

`\enableminitoc` Make mini tocs appear at the start of given sectional unit.

```
\newcounter{minitoc}
\newif\if@minitoc
\@minitocfalse

\newcommand*{\enableminitoc}[1][\@ttb@type]{%
  \@minitoctrue
  \setcounter{minitoc}{0}%
  \@ifundefined{#1}%
```

```
    {%
      \PackageError{flowfram}{Sectioning type ‘#1’ not defined}{}%
    }%
    {%
      \renewcommand{\@ttb@minitoctype}{#1}%
      \ifthenelse{\equal{#1}{chapter}}%
      {%
        \@makeminitocchapter
      }%
      {%
        \ifthenelse{\equal{#1}{part}}%
        {\@makeminitocpart}%
        {%
          \@makeminitocsection{#1}%
        }%
      }%
    }%
  }
```

This command should only appear in the preamble. (This ensures that it is used before `\tableofcontents`.

```
\@onlypreamble{\enableminitoc}
```

`\@makeminitocchapter` If minitocs are associated with chapters, redefine `\@chapter` so that the minitoc appears after the chapter heading.

```
\newcommand{\@makeminitocchapter}{%
  \let\@mtoc@old@chapter\@chapter
  \def\@chapter[##1]##2{%
    \@mtoc@old@chapter[##1]{##2}%
    \stepcounter{minitoc}%
    \dominitoc{\c@minitoc}%
    \@afterheading
  }%
}
```

`\@makeminitocpart` Again, for parts. As before, need to redefine `\@endpart` if it exists, otherwise redefine `\@part`.

```
\newcommand{\@makeminitocpart}{%
  \@ifundefined{@endpart}%
  {%
    \let\@mtoc@old@part\@part
    \def\@part[##1]##2{%
      \@mtoc@old@part[##1]{##2}%
      \stepcounter{minitoc}%
      \dominitoc{\c@minitoc}%
      \@afterheading
    }%
  }%
  {%
```

```
    \let\@mtoc@old@endpart\@endpart
    \def\@endpart{%
      \stepcounter{minitoc}%
      \dominitoc{\c@minitoc}%
      \@mtoc@old@endpart
    }%
  }%
}
```

`\@makeminitocsection` Now for the remaining sectional units.

```
\newcommand{\@makeminitocsection}[1]{%
  \let\@mtoc@old@sect=\@sect
  \def\@sect##1##2##3##4##5##6[##7]##8{%
    \@mtoc@old@sect{##1}{##2}{##3}{##4}{##5}{##6}[##7]{##8}%
    \ifthenelse{\equal{##1}{#1}}%
    {%
      \stepcounter{minitoc}%
      \dominitoc{\c@minitoc}%
      \@afterheading
    }%
    {}%
  }%
}
```

Index

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

Symbols
\@@@ff@getrange 99
\@@dofooter 154
\@@doheader 153
\@@dominitoc 206
\@@dothumbtabs 196
\@@dynamicframeswapcoords 55
\@@ff@chckifthispg 136
\@@ff@getrange 99
\@@hNtone 178
\@@hNtoneright 179
\@@htwotoneleft 177
\@@htwotoneright 177
\@@insert@hrule 186
\@@insert@vrule 182
\@@inserthrule 185
\@@insertvrule 181
\@@newdynamicframe 46
\@@newflowframe 15
\@@newstaticframe 31
\@@nexthNband 180
\@@nextvNband 175
\@@setdynamicframe 51
\@@setflowframe 24
\@@setfr@mecol 31
\@@setframecol 31
\@@setstaticframe 36
\@@sinserthrule 188
\@@sinsertvrule 184
\@@staticframeswapcoords 44
\@@storefileconts 203
\@@ttb@stack 195
\@@vNtone 175
\@@vNtonetop 176
\@NcolumnSbottominarea 171
\@Ncolumninarea 161
\@Ncolumntopinarea 166
\@addtotoclist 203
\@appenddynamic 60
\@appendminitocdf 206
\@checkcontentsline 204
\@computeffarea 63
\@dblfloatplacement 158
\@df@chckifthispg 137
\@df@checkuniqueidl 48
\@df@setoffset 144
\@doclearpage 91
\@docolbbox 139
\@docolbox 138
\@dodfparbox 150
\@dodynamicbbox 151
\@dodynamicbox 151
\@dostaticbbox 147
\@dostaticbox 146
\@dothefoot 92
\@dothehead 92
\@dothumbtabs 193
\@dynamicchap 189
\@dynamicframeid 49
\@dynamicframeswapcoords 56
\@dynamicswitchoffnext 114
\@dynamicswitchoffnextodd 116
\@dynamicswitchoffnextoddonly 122
\@dynamicswitchoffnextonly 122
\@dynamicswitchonnext 112
\@dynamicswitchonnextodd 113
\@dynamicswitchonnextoddonly 120
\@dynamicswitchonnextonly 118
\@ff@@getrange 98
\@ff@addtolist 60
\@ff@box 139
\@ff@chckifthispg 136
\@ff@checkifmoreframes 95
\@ff@checknumrange 97
\@ff@checkpages 97
\@ff@checkthispage 97
\@ff@checkuniqueidl 16
\@ff@disablecolor 8
\@ff@disablesec 41
\@ff@do@allframes 156
\@ff@doalldynamics 155
\@ff@doalldynamicsbbox 155
\@ff@doallflowframes 154
\@ff@doallflowframesbbox 154
\@ff@doallstatics 155
\@ff@doallstaticsbbox 155
\@ff@dotypeblock 155
\@ff@drawmargins 146
\@ff@enablecolor 7
\@ff@fbox 139

\@ff@getdim 64
\@ff@getevendim 65
\@ff@getrange 98
\@ff@getrangegreater 98
\@ff@getrangeless 98
\@ff@getshape 40
\@ff@getstaticpos 146
\@ff@heading 40
\@ff@output@adjustframes 99
\@ff@pages@countreg 7
\@ff@parshape 39
\@ff@s@t@Ovalboxoffset 142
\@ff@s@t@defaultoffset 142
\@ff@s@t@doubleboxoffset 141
\@ff@s@t@ovalboxoffset 141
\@ff@sectionhead 39
\@ff@setoffset 142
\@ff@setsecthead 40
\@ff@swaplen 64
\@ffbackground 4
\@ffdraft 4
\@ffnodraft 4
\@flf@subsp 195
\@flowframeid 17
\@flowframeswapcoords 29
\@flowswitchoffnext 102
\@flowswitchoffnextodd 104
\@flowswitchoffnextoddonly 111
\@flowswitchoffnextonly 110
\@flowswitchonnext 100
\@flowswitchonnextodd 101
\@flowswitchonnextoddonly 108
\@flowswitchonnextonly 106
\@fr@meifdraft 4
\@g@tnextcol 135
\@getdynamicbounds 68
\@getdynamicevenbounds 69
\@getflowbounds 68
\@getflowevenbounds 68
\@getframeid 49
\@getmarginpos 13
\@getstaticbounds 67
\@getstaticevenbounds 68
\@gettype 205
\@greyscale 196
\@hNtone 178
\@hNtoneleft 178
\@hNtoneright 179
\@htwotone 177
\@htwotoneleft 177
\@htwotoneright 177
\@inserthrule 185
\@insertvrule 181
\@makecol 94
\@makeminitocchapter 208
\@makeminitocpart 208
\@makeminitocsection 209
\@makethumbchapter 192
\@makethumbpart 192
\@makethumbsection 193
\@makethumbtabs 191
\@n@wflowframe 14
\@n@wstaticframe 31
\@newdynamicframe 46
\@newflowframe 14
\@newstaticframe 31
\@nexthNband 179
\@nexthband 179
\@nextvNband 175
\@nextvband 174
\@onecolumn 159
\@onecolumninarea 159
\@opcol 95
\@outputdblcol 156
\@outputpage 93
\@printalignedtoc 207
\@printtoc 206
\@putcolbbox 141
\@putcolbox 140
\@putdynamicbbox 153
\@putdynamicbox 152
\@putmarginbox 145
\@putstaticbbox 148
\@putstaticbox 147
\@resetdyn@mics 149
\@resetst@tics 149
\@s@ff@heading 39
\@s@tdynamicframeid 48
\@s@tffcol 7
\@s@tfftextcol 7
\@s@tflowframeid 16
\@s@tfr@mes 154
\@s@tstaticframeid 33
\@sappenddynamic 60
\@sappendminitocdf 206
\@scomputeffarea 62
\@sdynamicchap 190
\@sdynamicframeswapcoords 56
\@sdynamicswitchoffnext 114
\@sdynamicswitchoffnextodd 114
\@sdynamicswitchoffnextoddonly 122
\@sdynamicswitchoffnextonly 121
\@sdynamicswitchonnext 111
\@sdynamicswitchonnextodd 112
\@sdynamicswitchonnextoddonly 118

\@sdynamicswitchonnextonly 117
\@setcol 90
\@setcolbox 138
\@setdynamiccontents 59
\@setdynamicframe 50
\@setflowframe 23
\@setframecol 30
\@setstaticframe 35
\@setthumbtab 200
\@setthumbtabindex 201
\@sf@chckifthispg 137
\@sf@checkuniqueidl 33
\@sf@setoffset 143
\@sflowframeswapcoords 29
\@sflowswitchoffnext 102
\@sflowswitchoffnextodd 103
\@sflowswitchoffnextoddonly 110
\@sflowswitchoffnextonly 110
\@sflowswitchonnext 99
\@sflowswitchonnextodd 100
\@sflowswitchonnextoddonly 106
\@sflowswitchonnextonly 105
\@sgetdynamicbounds 68
\@sgetdynamicevenbounds 69
\@sgetflowbounds 68
\@sgetflowevenbounds 68
\@sgetstaticbounds 67
\@sgetstaticevenbounds 67
\@sinserthrule 187
\@sinsertvrule 184
\@snewdynamicframe 46
\@snewflowframe 14
\@snewstaticframe 31
\@ssetdynamiccontents 59
\@ssetdynamicframe 50
\@ssetflowframe 23
\@ssetstaticframe 35
\@sstaticconts 43
\@sstaticframeswapcoords 45
\@sstaticswitchoffnext 125
\@sstaticswitchoffnextodd 126
\@sstaticswitchoffnextoddonly 134
\@sstaticswitchoffnextonly 133
\@sstaticswitchonnext 123
\@sstaticswitchonnextodd 124
\@sstaticswitchonnextoddonly 130
\@sstaticswitchonnextonly 129
\@starttoc 202
\@staticconts 43
\@staticframeid 34
\@staticframeswapcoords 45
\@staticswitchoffnext 126
\@staticswitchoffnextodd 128
\@staticswitchoffnextoddonly 134
\@staticswitchoffnextonly 134
\@staticswitchonnext 123
\@staticswitchonnextodd 125
\@staticswitchonnextoddonly 132
\@staticswitchonnextonly 129
\@storefileconts 202
\@ttb@minitoctype 201
\@ttb@stack 195
\@ttb@storetoc 202
\@ttb@type 191
\@twocolumn 159
\@twocolumninarea 160
\@vNtone 175
\@vNtonebottom 176
\@vNtonetop 176

A

absolutepage (counter) 7
\adjustcolsep 172
\adjustheight 172
\afterminitocskip 205
\appenddfminitoc 206
\appenddynamiccontents 59

B

\beforeminitocskip 205

C

\c@absolutepage 7
\c@maxminitoc 202
\c@maxtocunits 202
\c@page 7
\chapter 9
\chapterfirstpagestyle 9
\checkifframeabove 69
\checkifframebelow 71
\checkifframeleft 73
\checkifframeright 75
\cleardoublepage 12
\clearpage 12

color (option) 7
\computebottomedge 62
\computeflowframearea 62
\computeleftedgeeven 61
\computeleftedgeodd 61
\computerightedgeeven 62
\computerightedgeodd 62
\computetopedge 61
\continueonframe 60

D

\defaultthumbtabtype 190
\dfchaphead 189
\DFchapterstyle 189
\DFschapterstyle 189
\dfswapoddeven 56
\disablethumbtabs 199
displayedframe (counter) 10
document (environment) 2
\dominitoc 205
draft (option) 4
\dynamicaddexclusion 55
dynamiccontents (environment) 57, 60, 61
dynamiccontents* (environment) 59
\dynamicframeevenx 50
\dynamicframeeveny 50
\dynamicframex 49
\dynamicframey 50
\dynamicsetexclusion 55
\dynamicsetpagelist 54
\dynamicswitchoffnext 113
\dynamicswitchoffnextodd 114
\dynamicswitchoffnextoddonly 122
\dynamicswitchoffnextonly 121
\dynamicswitchonnext 111
\dynamicswitchonnextodd 112
\dynamicswitchonnextoddonly 118
\dynamicswitchonnextonly 117

E

\emulateonecolumn 88
\emulatetwocolumn 88
\enableminitoc 207
\enablethumbtabs 198
environments:
    document 2
    dynamiccontents 57, 60, 61
    dynamiccontents* 59
    staticcontents 41, 60, 61
    staticcontents* 42
    staticfigure 158
    statictable 158
\evencheckifframeabove 78
\evencheckifframebelow 79
\evencheckifframeleft 81
\evencheckifframeright 82

F

\ff@col 7
\FFabove 84
\FFaboveleft 84
\FFaboveright 84
\ffaddtoadjustframeshook 135
\FFbelow 84
\FFbelowleft 84
\FFbelowright 84
\ffcolumnseprule 181
\ffcontinuedtextfont 61
\ffcontinuedtextlayout 61
\ffhrule 187
\fflabelfont 3
\fflabelsep 3, *3*
\FFleft 84
\FFoverlap 84
\ffprechapterhook 9
\ffpshpar 39
\ffruledeclarations 181
\ffswapoddeven 28, 29
\fftolerance 89
\ffvrule 183
final (option) 4
\finishthispage 12
\flf@doifverbose 5
\flf@message 5
\flowaddexclusion 28
flowfram counters:
    absolutepage 7
    displayedframe 10
    maxdynamic 11
    maxflow 10
    maxstatic 10
    thisframe 10
\flowframecol 7
\flowframeevenx 30
\flowframeeveny 30
\flowframeheight 30
\flowframerule 11
\flowframesep 11
\flowframeshowlayout 11, *11*
\flowframetextcol 7
\flowframewidth 30
\flowframex 30

\flowframey 30
\flowsetexclusion 28
\flowsetpagelist 28
\flowswitchoffnext 101
\flowswitchoffnextodd 102
\flowswitchoffnextoddonly 110
\flowswitchoffnextonly 109
\flowswitchonnext 99
\flowswitchonnextodd 100
\flowswitchonnextoddonly 106
\flowswitchonnextonly 105
\footnotecolor . . 94
\framebreak 11

G

\getdynamicbounds 68
\getdynamicevenbounds 68
\getdynamicid . . . 49
\getdynamiclabel 48
\getflowbounds . . 68
\getflowevenbounds 68
\getflowid 17
\getflowlabel . . . 17
\getstaticbounds 67
\getstaticevenbounds 67
\getstaticid 33
\getstaticlabel . 33
\globalnormalmargin 13
\globalreversemargin 13

H

\hNtone 178
\hNtoneleft 178
\hNtoneright . . . 178
\htwotone 176
\htwotoneleft . . 177
\htwotoneright . 177

I

\if@ttb@num 6
\if@ttb@title 6
\iflefttorightcolumns 8
\inserthrule . . . 185
\insertvrule *180*, 181

L

\labelflow 10
\labelflowidn . . . 10
LR (option) 8

M

\makebackgroundframe 180
\makedfheaderfooter 93
\makethumbtabs . 191
maxdynamic (counter) . . . 11
maxflow (counter) . 10
maxstatic (counter) 10
\minitocstyle . . 206

N

\Ncolumn 160
\Ncolumnbottom . 170
\Ncolumnbottominarea 170
\NcolumnDbottom 170
\NcolumnDbottominarea 172
\NcolumnDtop . . . 165
\NcolumnDtopinarea 167
\Ncolumninarea . 160
\NcolumnSbottom 170
\NcolumnSbottominarea 172
\NcolumnStop . . . 165
\NcolumnStopinarea 167
\Ncolumntop 165
\Ncolumntopinarea 165
\newdynamicframe 46
\newflowframe . . . 14
\newframe 162
\newpage 12
\newstaticframe . 31
\normalcolor 8
norotate (option) . . 5

O

\oddcheckifframeabove 70
\oddcheckifframebelow 72
\oddcheckifframeleft 74
\oddcheckifframeright 76
\onecolumn 158
\onecolumnbottom 167
\onecolumnbottominarea 168
\onecolumnDbottom 168
\onecolumnDbottominarea 168
\onecolumnDtop . 162
\onecolumnDtopinarea 163
\onecolumninarea 159
\onecolumnSbottom 167
\onecolumnSbottominarea 168
\onecolumnStop . 162
\onecolumnStopinarea 163
\onecolumntop . . 162
\onecolumntopinarea 163

P

package options:
- color
  - false 8
  - true 7–9
- color 7
- draft 4
- final 5
- final 4
- LR 8
- nocolor 7

  norotate, 5
  norotate, 5
  pages, 7
  RL, 8
  rotate, 5
  thumbtabs, 6
  thumbtabs, 6
  ttbnonum, 6
  ttbnonum, 6
  ttbnotitle, 6
  ttbnotitle, 6
  ttbnum, 6
  ttbnum, 6
  ttbtitle, 6
  ttbtitle, 6
  verbose, 5
\pagebreak, 12
pages (option), 7
\parshape, 42, 150

R

\relativeframelocation, 84
\reldynamicloc, 86
\relflowloc, 87
\relstaticloc, 86
RL (option), 8
rotate (option), 5
\rotateframe, 5

S

\sdfparindent, 11
\setalldynamicframes, 50
\setallflowframes, 23
\setallstaticframes, 35
\setdynamiccontents, 59
\setdynamicframe, 50
\setflowframe, 23
\setframes, 87
\setinitialframe, 87
\setmargin, 14
\setstaticcontents, 43
\setstaticframe, 35
\setthumbtab, 200
\setthumbtabindex, 201
\sfswapoddeven, 45
\Shapepar, 43
\shapepar, 43, 150
\simpar, 39
\staticaddexclusion, 44
staticcontents (environment), 41, 60, 61
staticcontents* (environment), 42
staticfigure (environment), 158
\staticframeevenx, 34
\staticframeeveny, 34
\staticframex, 34
\staticframey, 34
\staticsetexclusion, 44
\staticsetpagelist, 44
\staticswitchoffnext, 125
\staticswitchoffnextodd, 126
\staticswitchoffnextoddonly, 134
\staticswitchoffnextonly, 133
\staticswitchonnext, 123
\staticswitchonnextodd, 124
\staticswitchonnextoddonly, 130
\staticswitchonnextonly, 129
statictable (environment), 158

T

\tableofcontents, 205
thisframe (counter), 10
\thumbtab, 193
\thumbtabformat, 194
\thumbtabindex, 199
\thumbtabindexformat, 194
thumbtabs (option), 6
\thumbtabwidth, 194
\tocandhumbtabindex, 201
ttbnonum (option), 6
ttbnotitle (option), 6
ttbnum (option), 6
ttbtitle (option), 6
\twocolumn, 159
\twocolumnbottom, 168
\twocolumnbottominarea, 169
\twocolumnDbottom, 169
\twocolumnDbottominarea, 170
\twocolumnDtop, 164
\twocolumnDtopinarea, 165
\twocolumninarea, 160
\twocolumnSbottom, 169
\twocolumnSbottominarea, 170
\twocolumnStop, 164
\twocolumnStopinarea, 165
\twocolumntop, 164
\twocolumntopinarea, 164

V

\vcolumnsep, 162
verbose (option), 5
\vNtone, 174
\vNtonebottom, 176
\vNtonetop, 176
\vtwotone, 173
\vtwotonebottom, 173
\vtwotonetop, 174