

Documented Source Code for flowfram.sty v2.0

Nicola L. C. Talbot

2025-11-24

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

Contents

Terms	2
1 Main Package Code	3
1.1 Package Initialisation	3
1.2 Frame Attributes	34
1.3 Flow Frames	73
1.4 Static Frames	83
1.5 Dynamic Frames	102
1.6 Determining Dimensions and Locations	113
1.7 Determining the relative location of one frame from another	120
1.8 Initialise Flow Frames	144
1.9 Output Routine	156
1.10 Switching Frames On and Off For Subsequent Pages	209
1.11 Static versions of floats	241
1.12 Standard Layouts	241
1.12.1 Column Styles	241
1.12.2 Backdrop Effects	266
1.12.3 Lines Between Frames	275
1.13 Putting Chapter Headings in Dynamic Frames	282
1.14 Thumbtabs	291
1.15 Minitocs	310
2 Support for FlowframTk	321
3 Rollback v1.17 (flowfram-2014-09-30)	360
4 Rollback v1.18 (flowfram-2025-08-23)	515
4.1 Flow Frames	523
4.2 Static Frames	543
4.3 Dynamic Frames	558

4.4	Determining Dimensions and Locations	573
4.5	Determining the relative location of one frame from another	581
4.6	Initialise Flow Frames	599
4.7	Output Routine	601
4.8	Static versions of floats	669
4.9	Standard Layouts	669
4.9.1	Column Styles	669
4.9.2	Backdrop Effects	683
4.9.3	Lines Between Frames	691
4.10	Putting Chapter Headings in Dynamic Frames	699
4.11	Thumbtabs	701
4.12	Minitocs	712

Terms

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 Main Package Code

1.1 Package Initialisation

Declare package, and identify it as a L^AT_EX 2_ε package.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback release:

```
\DeclareRelease{v1.17}{2014-09-30}{flowfram-2014-09-30.sty}
\DeclareRelease{v1.18}{2025-08-23}{flowfram-2025-08-23.sty}
\DeclareCurrentRelease{v2.0}{2025-11-24}
```

Declare package:

```
\ProvidesPackage{flowfram}[2025/11/24 v2.0 (NLCT)]
```

Version 2.0 now loaded rerunfilecheck package:

```
\RequirePackage{rerunfilecheck}
```

No longer loading xfor, ifthen or xkeyval.

```
\RequirePackage{graphics}
\RequirePackage{afterpage}
```

```
\RequirePackage{etoolbox}
```

Version 2.0: switching to L^AT_EX 3 commands but some old code still remains.

```
\ExplSyntaxOn
```

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

```
\setffdraftcolor
```

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

```
\setffdrafttypeblockcolor
```

```
\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.

```
\newlength\fflabelsep
\setlength\fflabelsep{1pt}
```

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

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

Allow user the option to show particular types of bounding boxes.

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

`\ffdraft` Set all draft settings. Version 2.0 renamed `\@ffdraft`.

```
\cs_new:Nn \__flowfram_set_draft:
{
  \showtypeblocktrue
  \showmarginstrue
  \showframebboxtrue
}
```

`\@ffnodraft` Unset all draft settings. Version 2.0 renamed `\@ffnodraft`.

```
\cs_new:Nn \__flowfram_set_final:
{
  \showtypeblockfalse
  \showmarginfalse
  \showframebboxfalse
}
```

Set the default to final:

```
\__flowfram_set_final:
```

Just in case something has redefined `\frame` (for example, `beamer`, although it would be unusual to use `flowfram` with that).

```
\cs_new_protected:Nn \flowfram_frame:n
{
  \leavevmode
  \hbox
  {
    \hskip -\@wholewidth
    \vbox
    {
      \vskip -\@wholewidth
      \hrule \@height \@wholewidth
      \hbox
      {
        \vrule \@width \@wholewidth
        #1
        \vrule \@width \@wholewidth
      }
      \hrule \@height \@wholewidth
      \vskip -\@wholewidth
    }
    \hskip -\@wholewidth
  }
}
```

```
    }  
  }
```

\flf@message Messaging system (to help debugging): Version 2.0 removed \flf@message.

```
\cs_new:Nn \__flowfram_message:n  
{  
  \flf@doifverbose  
  {  
    \msg_note:nn { flowfram } { #1 }  
  }  
}  
\cs_new:Nn \__flowfram_message:nn  
{  
  \flf@doifverbose  
  {  
    \msg_note:nnn { flowfram } { #1 } { #2 }  
  }  
}  
\cs_new:Nn \__flowfram_message:nnn  
{  
  \flf@doifverbose  
  {  
    \msg_note:nnnn { flowfram } { #1 } { #2 } { #3 }  
  }  
}  
\cs_generate_variant:Nn \__flowfram_message:nnn { nen , nee }  
\cs_new:Nn \__flowfram_message:nnnn  
{  
  \flf@doifverbose  
  {  
    \msg_note:nnnnn { flowfram } { #1 } { #2 } { #3 } { #4 }  
  }  
}  
\cs_generate_variant:Nn \__flowfram_message:nnnn { neee }  
\cs_new:Nn \__flowfram_message:nnnnn  
{  
  \flf@doifverbose  
  {  
    \msg_note:nnnnnn { flowfram } { #1 } { #2 } { #3 } { #4 } { #5 }  
  }  
}  
\cs_generate_variant:Nn \__flowfram_message:nnnnn  
{ nnnen , nnneV , nnnev }  
  
\cs_new:Nn \__flowfram_error:nnn  
{  
  \msg_error:nnnn { flowfram } { #1 } { #2 } { #3 }  
}  
\cs_generate_variant:Nn \__flowfram_error:nnn  
{ eee }
```

```

\cs_new:Nn \__flowfram_error:nnnn
{
  \msg_error:nnnnn { flowfram } { #1 } { #2 } { #3 } { #4 }
}
\cs_generate_variant:Nn \__flowfram_error:nnnn
{ eeee }
\cs_new:Nn \__flowfram_error:nnnnn
{
  \msg_error:nnnnnn { flowfram } { #1 } { #2 } { #3 } { #4 } { #5 }
}
\cs_generate_variant:Nn \__flowfram_error:nnnnn
{ eeeee }

```

\flf@doifverbose Initialise:

```

\newcommand{\flf@doifverbose}[1]{}
\char_set_catcode_space:n { '\ }
\msg_new:nnnn {flowfram} {label-defined}
{Label '#1' already defined for frame type '#2'}
{You can't assign this label, as it is already defined
for #2 frame IDN #3}
\msg_new:nnnn {flowfram} {label-undefined}
{Label '#1' is not defined for any #2 frame}
{Check that you have spelt the label correctly and that
the label corresponds to a #2 frame}
\msg_new:nnnn {flowfram} {idn-undefined}
{No #1 frame with IDN #2}
{Identification numbers start from 1 to the number of defined #1 frames}
\msg_new:nnnn {flowfram} {invalid-id}
{Invalid IDN #1}
{Identification numbers start from 1}
\msg_new:nnnn {flowfram} {attribute-type-change}
{Frame attribute #1 of type '#2' now being defined as '#3'}
{Frame attribute #1 has already been defined as type '#2' but
now this attribute is being defined as type '#3'}
\msg_new:nnnn {flowfram} {invalid-frame-type}
{Unknown frame type '#1'}
{Frame types can be one of: flow, static or dynamic}
\msg_new:nnnn {flowfram} {invalid-frame-typeid}
{Unknown frame type ID '#1'}
{Frame type IDs can be one of: 1 (flow), 2 (static) or 3 (dynamic)}
\msg_new:nnn {flowfram} {frame-unsupported-options}

```

```

{Frame type '#1' does not support option(s): #2}

\msg_new:nnnn {flowfram} {frame-unknown-tag}
{Frame type '#1' does not have #4 attribute #2 defined (IDN #3)}
{Check that you have correctly spelt the attribute '#2' and have
correctly identified the type of frame and its numeric identifier.
The other possibility is that there is a bug in the code that's
incorrectly identified the underlying data type (#4).}

\msg_new:nnn {flowfram} {non-void-box}
{Box #1 is not void. Dumping. This page: #2.
Page list: "#3". Exclusion list: "#4".
(Maybe the page list was changed after this frame was
selected or maybe you should use package option pages=absolute)}

\msg_new:nnn {flowfram} {invalid-bool}
{Invalid boolean setting '#1' for attribute '#2' applied to #3 frame IDN #4}

\msg_new:nnnn {flowfram} {invalid-pagelist}
{Invalid page list '#1'}
{A page list may be one of the keywords 'all', 'none', 'even' or
'odd', or may be an ordered comma-separated list of individual
numbers (e.g. '2'), a closed range (e.g. '12-18'), or an
open-ended range (e.g. '<3' or '>5'). Overlapping ranges are not
permitted and items must be listed in increasing numerical order
(e.g. '<4,9-12,15,>20')}

\msg_new:nnnn {flowfram} {cant-set-column}
{Can't set current column to frame '#1' (so such frame)}
{There is no frame defined with the IDN '#1'}

\msg_new:nnn {flowfram} {unequal-cols}
{Moving to flow frame of unequal width, use of
\token_to_str:N \framebreak \c_space_tl advised,
or text might not appear correctly (difference = #1,
tolerance = #2)}

\msg_new:nnn {flowfram} {no-page1-col}
{Can't find a flow frame on page 1.
Attempting to find the first page with a flow frame}

\msg_new:nnn {flowfram} {no-cols}
{No flow frames, adding #1}

\msg_new:nnn {flowfram} {no-more-cols-on-page}
{Run out of flows frames on page #1, adding new one}

\msg_new:nnn {flowfram} {misplaced-col-cmd}
{Found #1 in document environment with
column-changes=switch option set but no frame

```

```

has been identified to use with #1}

\msg_new:nnnn {flowfram} {doc-env-only}
{Command #1 only permitted in document environment}
{You can't use #1 in the preamble}

\msg_new:nnn {flowfram} {background-not-first}
{Background frame is not first static frame to be
defined. All previously defined static frames may be
obscured.}

\msg_new:nnn {flowfram} {ignoring-multiple-makethumbtabs}
{Ignoring repeated instance of \makethumbtabs}

\msg_new:nnn {flowfram} {no-thumbtabs-missing-makethumbtabs}
{No thumb tabs defined: missing \makethumbtabs}

\msg_new:nnn {flowfram} {no-thumbtabs-rerun}
{No thumb tabs defined. Rerun may be required}

\msg_new:nnn {flowfram} {cant-find-thumbtab}
{Can't find thumbtab number '#1', ttb file may not be
up-to-date}

\msg_new:nnnn {flowfram} {cant-find-thumbtab-index}
{Can't find thumbtab index number '#1'}
{Something's gone wrong. The thumbtab '#1' exists but
thumbtab index '#1' doesn't}

\msg_new:nnnn {flowfram} {invalid-thumbtab-index}
{Invalid thumbtab index '#1'}
{Thumbtab indexing starts at 1}

\msg_new:nnn {flowfram} {section-unit-not-defined}
{Sectioning type '#1' not defined}

\msg_new:nnn {flowfram} {unknown-heading-cmd}
{Unknown heading command '#1'}

\msg_new:nnn {flowfram} {unknown-shape}
{Unknown shape #1}

\msg_new:nnn {flowfram} {forbidden-cmd-in-shape}
{You can't use #1 within a paragraph shaped with #2}

\msg_new:nnnn {flowfram} {cant-continue}
{Can't continue to new frame: either not in static or dynamic frame or attempting to append content.
{\token_to_str:N \continueonframe \c_space_tl may only
be used inside 'staticcontents' or 'dynamiccontents'
environments (or their starred versions) or in the commands

```



```

\token_to_str:N \setstaticcontents \c_space_tl or
\token_to_str:N \setdynamiccontents}

\msg_new:nnnn {flowfram} {invalid-num-frames}
{You have requested #2 #1 frames!}
{The number of frames must be greater than 0}

\msg_new:nnnn {flowfram} {no-chapters}
{Chapters aren't defined}
{The document class you are using does not support chapters}

\msg_new:nnnn {flowfram} {option-too-late}
{Option #1 occurring too late (has no effect after #2)}
{If you want to set this option, you need to do so earlier}

\msg_new:nnn {flowfram} {reloc-label-already-defined}
{Relative location label '#1' multiply defined}

\msg_new:nnn {flowfram} {reloc-label-not-defined}
{Relative location label '#1' undefined}

\msg_new:nnn {flowfram} {data-changed-rerun}
{Label or frame data has changed. Rerun required}

\msg_new:nnn {flowfram} {info-set-excludelist}
{Setting exclusion list for #1 frame IDN #2 to '#3'}

\msg_new:nnn {flowfram} {info-switching-col}
{Switching to flow frame #1 on page #2}

\msg_new:nnn {flowfram} {info-setting-col}
{Setting contents of box for flow frame #1}

\msg_new:nnn {flowfram} {info-doing-col}
{Doing flow frame #1 (page #2)}

\msg_new:nnn {flowfram} {info-col-not-required}
{Flow frame #1 is not required on page #2}

\msg_new:nnn {flowfram} {info-col-list}
{List of defined flow frames: #1}

\msg_new:nnn {flowfram} {info-static-frame-hidden}
{Static frame IDN #1 is hidden (#2=true)}

\msg_new:nnn {flowfram} {info-dynamic-frame-hidden}
{Dynamic frame IDN #1 is hidden (#2=true)}

\msg_new:nnn {flowfram} {info-no-file}
{No file #1.}

```

```

\msg_new:nnn {flowfram} {info-no-rotation}
  {Frame rotation has been disable.
   Attempt to rotate frame has been ignored.}

\msg_new:nnn {flowfram} {info-found-thumbtab}
  {Found thumbtab data { #1 } { #2 } { #3 } { #4 }}

\msg_new:nnn {flowfram} {info-thumbtab}
  {Thumbtab #1 dynamic frame #2 (IDN #3)}

\msg_new:nnn {flowfram} {info-thumbtab-index}
  {Thumbtab index #1 dynamic frame #2 (IDN #3)}

\msg_new:nnn {flowfram} {info-setting-frame-attribute}
  {Setting #1 attribute for #2 frame #3 to #4}

\msg_new:nnn {flowfram} {show-frame-attributes}
  {Frame type #1 IDN #2 has the following attributes:\#3}

\msg_new:nnn {flowfram} {show-all-frames-none}
  {There are no frames of type #1}

\msg_new:nnn {flowfram} {show-all-frames}
  {Showing all #1 frames.
   Each frame will be shown separately.
   Total number of #1 frames: \#2}

\char_set_catcode_ignore:n { '\ }

```

Separate option to rotate thumbtab text from option to inhibit all frame rotation.

```

\bool_new:N \l_flowfram_allow_frame_rotation_bool
\bool_set_true:N \l_flowfram_allow_frame_rotation_bool

```

\if@ttb@rotate Settings to adjust rotation support in the thumbtabs. Version 2.0 replaced \if@ttb@rotate,

No rotation (the frame will need to be wide enough to accommodate the text or have rotation applied):

```

\cs_new:Nn \__flowfram_thumbtab_fmt:nn
  {
  \__flowfram_thumbtab_fmt_sideways:nn { #1 } { #2 }
  }

```

Rotate for right / odd pages.

```

\cs_new:Nn \__flowfram_thumbtab_fmt_sideways_oneside_right:nn
  {
  \rotatebox { -90 }
  {

```

```

        \parbox [ c ] [ \thumbtabwidth ] { #2 }
        {
            \centering #1
        }
    }
}

```

Rotate for left / even pages.

```

\cs_new:Nn \__flowfram_thumbtab_fmt_sideways_oneside_left:nn
{
    \rotatebox { 90 }
    {
        \parbox [ c ] [ \thumbtabwidth ] { #2 }
        {
            \centering #1
        }
    }
}

```

Rotate according to one or two-sided documents.

```

\cs_new:Nn \__flowfram_thumbtab_fmt_sideways:nn
{
    \if@twoside
        \int_if_odd:nTF { \c@page }
        {
            \__flowfram_thumbtab_fmt_sideways_oneside_right:nn
            { #1 } { #2 }
        }
        {
            \__flowfram_thumbtab_fmt_sideways_oneside_left:nn
            { #1 } { #2 }
        }
    }
    \else
        \__flowfram_thumbtab_fmt_sideways_oneside_right:nn
    \fi
}

```

Stack the letters vertically (this doesn't look very good).

```

\cs_new:Nn \__flowfram_thumbtab_fmt_stack:nn
{
    \parbox [ c ] [ #2 ] { \thumbtabwidth }
    {
        \centering
        \flowfram_thumbtab_stack:n { #1 }
    }
}

```

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

`\if@ttb@num` Version 2.0 removed `\if@ttb@num`.

```

\bool_new:N \l__flowfram_ttb_show_number_bool
\bool_set_false:N \l__flowfram_ttb_show_number_bool

```

```

\iff@ttb@title
    \bool_new:N \l__flowfram_ttb_show_title_bool
    \bool_set_true:N \l__flowfram_ttb_show_title_bool

```

\@ff@pages@countreg The default is relative (for backwards compatibility). Version 2.0 renamed \@ff@pages@countreg.

```

\tl_new:N \g__flowfram_pagecounter_tl
\tl_gset:Nn \g__flowfram_pagecounter_tl { \c@page }

```

\theFramePageCounter For use when the page/absolute page counter needs to be written as its actual value (not formatted) but must be deferred.

```

\newcommand \theFramePageCounter
{
    \int_eval:n { \g__flowfram_pagecounter_tl }
}

```

absolute page

```

\newcounter{absolute page}

```

\flowfram@thepagenumber

```

\newcommand \flowfram@thepagenumber { \int_use:N \c@page }

```

Constants.

```

\tl_const:Nn \c_flowfram_all_tl { all }
\tl_const:Nn \c_flowfram_odd_tl { odd }
\tl_const:Nn \c_flowfram_even_tl { even }
\tl_const:Nn \c_flowfram_none_tl { none }
\tl_const:Nn \c_flowfram_color_none_tl { { none } }
\tl_const:Nn \c_flowfram_color_black_tl { { black } }
\tl_const:Nn \c_flowfram_inner_tl { inner }
\tl_const:Nn \c_flowfram_outer_tl { outer }
\tl_const:Nn \c_flowfram_left_tl { left }
\tl_const:Nn \c_flowfram_right_tl { right }
\tl_const:Nn \c_flowfram_compute_tl { compute }
\int_const:Nn \c_flowfram_max_page_int { 100000 }
\int_const:Nn \c_flowfram_shape_type_parshape_int { 1 }
\int_const:Nn \c_flowfram_shape_type_shapepar_int { 2 }
\int_const:Nn \c_flowfram_shape_type_none_int { 0 }
\int_const:Nn \c_flowfram_frame_type_flow_int { 1 }
\int_const:Nn \c_flowfram_frame_type_static_int { 2 }
\int_const:Nn \c_flowfram_frame_type_dynamic_int { 3 }

```

\flowframecol Default frame border colour.

```

\newcommand \flowframecol { }

```

\flowframetextcol Default text colour.

```

\newcommand \flowframetextcol { }

```

`\@fr@meifdraft` Draw bounding box. Version 2.0 renamed `\@fr@meifdraft`.

```
\cs_new:Nn \__flowfram_do_if_draft:nnn
{
  \tl_set_eq:NN
    \l__flowfram_backcolor_tl
    \c_flowfram_color_none_tl
  #1
  \flowfram_frame:n { #2 }
  \tl_if_empty:nF { #3 }
  {
    \makebox [ Opt ] [ l ]
    {
      \hskip \fflabelsep
      \fflabelfont { #3 }
    }
  }
}
\cs_new:Nn \__flowfram_do_if_draft:nn
{
  \__flowfram_do_if_draft:nnn
  { \setffdraftcolor } { #1 } { #2 }
}
\cs_new:Nn \flowfram_draft_annotate:nnn
{
  [ #1 : #2 ; #3 ]
}
\cs_new:Nn \flowfram_draft_annotate:nn
{
  [ #1 : #2 ]
}
```

Colour setting commands, do nothing by default:

`\@s@tffcol` Set frame border colour. Version 2.0 renamed `\@s@tffcol`

```
\cs_new:Nn \__flowfram_set_border_color: { }
\tl_new:N \l__flowfram_bordercolor_tl
```

`\@s@tffttextcol` Set text colour. Version 2.0 renamed `\@s@tffttextcol`

```
\cs_new:Nn \__flowfram_set_text_color: { }
\tl_new:N \l__flowfram_textcolor_tl
```

`\@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. Version 2.0 renamed `\@ffbackground`.

```
\cs_new:Nn \__flowfram_background_box:n { #1 }
\tl_new:N \l__flowfram_backcolor_tl
```

\@ff@enablecolor Enable colour commands. Version 2.0 renamed \@ff@enablecolor.

```
\cs_new:Nn \__flowfram_enable_color:
{
  \tl_set_eq:NN \flowframecol \c_flowfram_color_black_tl
  \tl_set_eq:NN \flowframetextcol \c_flowfram_color_black_tl
  \cs_set:Nn \__flowfram_set_border_color:
  {
    \exp_args:Ne \__flowfram_color:n
      { \l__flowfram_bordercolor_tl }
  }
  \cs_set:Nn \__flowfram_set_text_color:
  {
    \exp_args:Ne \__flowfram_color:n
      { \l__flowfram_textcolor_tl }
  }
  \cs_set:Nn \__flowfram_background_box:n
  {
    \exp_args:Ne
      \__flowfram_background_box:nn
      { \l__flowfram_backcolor_tl } { ##1 }
  }
}

\prg_new_conditional:Nnn \__flowfram_if_no_color:n
{ T, F, TF }
{
  \tl_if_empty:nTF { #1 }
  { \prg_return_true: }
  {
    \tl_if_eq:NnTF \c_flowfram_color_none_tl { #1 }
    { \prg_return_true: }
    { \prg_return_false: }
  }
}

\cs_new:Nn \__flowfram_color:n
{
  \__flowfram_if_no_color:nF
  { #1 }
  { \color #1 }
}

\cs_new:Nn \__flowfram_background_box:nn
{
  \__flowfram_if_no_color:nTF { #1 }
  { #2 }
  {
    \group_begin:
      \dim_zero:N \fboxsep
      \colorbox #1 { #2 }
    \group_end:
  }
}
```

```
}
```

`\@ff@disablecolor` Disable colour commands. Version 2.0 renamed `\@ff@disablecolor`.

```
\cs_new:Nn \__flowfram_disable_color:
{
  \tl_clear:N \flowframtextcol
  \tl_clear:N \flowframecol
  \cs_set:Nn \__flowfram_set_border_color: { }
  \cs_set:Nn \__flowfram_set_text_color: { }
  \cs_set_eq:NN \__flowfram_background_box:n \use:n
}
```

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

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

To minimize the number of output routine kernel commands that need to be redefined in order to place all frames in their chosen locations (including header and footer frames), version 2.0 now switches on two column mode. This can cause some standard commands to switch back and forth between one column and two column mode (for example, `\tableofcontents` with the book or report class). Since `\onecolumn` and `\twocolumn` are changed by `flowfram` to create one or two column layouts, this can be problematic. So v2.0 now has the `column-changes` option, which governs what to do if `\onecolumn` or `\twocolumn` are encountered in the document. The default is to do just clear the page but don't switch the layout (if the optional argument of `\twocolumn` is present then that is simply added to the current frame).

```
\bool_new:N \g__flowfram_ignore_column_changes_bool
\bool_gset_true:N \g__flowfram_ignore_column_changes_bool
```

Determine whether or not to clear page (not applicable if switcher columns identified).

```
\bool_new:N \g__flowfram_clearpage_column_changes_bool
\bool_gset_false:N \g__flowfram_clearpage_column_changes_bool
```

Only make thumbtabs for unnumbered units if the following boolean is set.

```
\bool_new:N \g__flowfram_save_nonum_thumbtabs_bool
```

`\ifaligntoc` This is now moved into options. Instead of using `\tocandthumbtabindex`, use the `toc-thumbtab` option.

```
\newif\ifaligntoc
\aligntocfalse
\bool_new:N \l__flowfram_thumbtabs_in_toc_bool
\bool_set_false:N \l__flowfram_thumbtabs_in_toc_bool
\bool_new:N \l__flowfram_thumbtabs_span_toc_bool
\bool_set_false:N \l__flowfram_thumbtabs_span_toc_bool
```

Indicates whether or not the sectioning commands have a second optional argument.

```
\bool_new:N \l__flowfram_section_header_opt_bool
\IfClassLoadedTF { memoir }
{
  \bool_set_true:N \l__flowfram_section_header_opt_bool
}
{
  \bool_set_false:N \l__flowfram_section_header_opt_bool
}
\bool_new:N \l__flowfram_section_header_opt_thumbtab_bool
\bool_set_true:N \l__flowfram_section_header_opt_thumbtab_bool
```

Indicates whether or not the section unit first optional argument may be a key=value list (for example with KOMA's headings=optiontoheadandtoc setting).

```
\bool_new:N \l__flowfram_section_keyval_opt_bool
\bool_set_false:N \l__flowfram_section_keyval_opt_bool
```

Provide keys if the above is true:

```
\tl_new:N \l__flowfram_section_opti_tl
\tl_new:N \l__flowfram_section_optii_tl
\tl_new:N \l__flowfram_section_toc_tl
\tl_new:N \l__flowfram_section_head_tl
\keys_define:nn { flowfram / section }
{
  head .tl_set:N = \l__flowfram_section_head_tl ,
  tocentry .tl_set:N = \l__flowfram_section_toc_tl ,
  toc .tl_set:N = \l__flowfram_section_toc_tl ,
}
```

If hyperref package is loaded, automatically enable hyperlinks for thumbtab index.

`\thumbtabhyperlinkformat`

```
\NewDocumentCommand \thumbtabhyperlinkformat { m m m }
{
  \thumbtabformat { #2 } { #3 }
}

\AddToHook
{ package / hyperref / after }
{
  \RenewDocumentCommand \thumbtabhyperlinkformat { m m m }
  {
    \hyperlink { #1 } { \thumbtabformat { #2 } { #3 } }
  }
}
```

By default, only the thumbtab indexes have hyperlinks (if supported).

`\thumbtabindexformat` Thumbtab format. If hyperlinks are defined, use a hyperlink in the thumbtab index. Syntax: `\thumbtabindexformat{<link>}{<text>}{<height>}`


```

\newcommand \thumbtabindexformat [ 3 ]
{
  \thumbtabhyperlinkformat { #1 } { #2 } { #3 }
}

```

\thumbtabregularformat Format for non-index thumbtabs.

```

\newcommand \thumbtabregularformat [ 3 ]
{
  \thumbtabformat { #2 } { #3 }
}

```

Should thumbtabs be created outside the main matter?

```

\bool_new:N \g__flowfram_frontmatter_bool
\bool_gset_false:N \g__flowfram_frontmatter_bool
\bool_new:N \g__flowfram_backmatter_bool
\bool_gset_false:N \g__flowfram_backmatter_bool
\cs_if_exist:cTF { if@mainmatter }
{
  \prg_new_conditional:Nnn \flowfram_if_mainmatter:
  { T, F, TF }
  {
    \legacy_if:nTF { @mainmatter }
    { \prg_return_true: }
    { \prg_return_false: }
  }
  \cs_if_exist:NT \frontmatter
  {
    \appto \frontmatter
    {
      \bool_gset_true:N \g__flowfram_frontmatter_bool
      \bool_gset_false:N \g__flowfram_backmatter_bool
    }
  }
  \cs_if_exist:NT \backmatter
  {
    \appto \backmatter
    {
      \bool_gset_false:N \g__flowfram_frontmatter_bool
      \bool_gset_true:N \g__flowfram_backmatter_bool
      \__flowfram_write_backmatter_ttb:
      \__flowfram_backmatter_setsecnumdepth:
    }
  }
  \cs_if_exist:NT \mainmatter
  {
    \appto \mainmatter
    {
      \bool_gset_false:N \g__flowfram_frontmatter_bool
      \bool_gset_false:N \g__flowfram_backmatter_bool
    }
  }
}

```

```

    }
  }
  {
    \prg_new_conditional:Nnn \flowfram_if_mainmatter:
      { T, F, TF }
    {
      \prg_return_true:
    }
  }
}
\cs_new:Nn \__flowfram_ttb_notmainmatter_num:n
{
  \bool_if:NF \g__flowfram_frontmatter_bool
  {
    \__flowfram_ttb_num:n { #1 }
  }
}

```

change the secnumdepth counter in the backmatter if applicable

```

\cs_new:Nn \__flowfram_backmatter_setsecnumdepth:
{
  \setcounter { secnumdepth } { -1 }
}

```

Case changing command used by dynamic header.

```

\cs_new:Nn \flowfram_header_case:n { #1 }
\cs_new:Nn \flowfram_subheader_case:n { #1 }

```

Font used by dynamic header.

```

\cs_new:Nn \flowfram_header_font:n { \emph { #1 } }
\cs_new:Nn \flowfram_subheader_font:n { \flowfram_header_font:n { #1 } }
\cs_new:Nn \__flowfram_adjust_page_styles:n
{
  #1
}

```

Header separator:

```

\ExplSyntaxOff
\ifdef\chapter
{
  \newcommand{\flowframheaderchapprefix}{\@chapapp \ }
  \newcommand{\flowframheadersep}{.\ }
}
{
  \newcommand{\flowframheaderchapprefix}{}
  \newcommand{\flowframheadersep}{\quad}
}
\ExplSyntaxOn

```

Table of contents header.

```

\renewcommand \tableofcontents
{
  \__flowfram_tableofcontents:
}

```

```

}
\cs_if_exist:NTF \chapter
{
  \cs_new:Nn \__flowfram_adjusted_toc:
  {
    \chapter * { \contentsname }
    \chaptermark { \contentsname }
    \@starttoc { toc }
  }
}
{
  \cs_new:Nn \__flowfram_adjusted_toc:
  {
    \section * { \contentsname }
    \sectionmark { \contentsname }
    \@starttoc { toc }
  }
}
\cs_new:Nn \flowfram_toc:
{
  \__flowfram_adjusted_toc:
}

```

By default, ffempty style clears the header and footer.

```

\tl_new:N \l__flowfram_ffempty_style_tl
\tl_set:Nn \l__flowfram_ffempty_style_tl { \__flowfram_ps_empty: }

```

and hides the dynamic frames:

```

\bool_new:N \l__flowfram_ps_ffempty_hides_bool
\bool_set_true:N \l__flowfram_ps_ffempty_hides_bool
\cs_new:Nn \__flowfram_only_pre_makedfheaderfooter:nn { #2 }

```

Save original definition of \tableofcontents:

```

\cs_set_eq:NN \__flowfram_org_tableofcontents: \tableofcontents

```

The new L^AT_EX hook management system and sophisticated classes now make the old flowfram pre-chapter hooks somewhat redundant. These are retained for backward-compatibility but provide a way of omitting them.

```

\cs_new:Nn \__flowfram_do_addto_prechap:
{
  \__flowfram_addto_prechap:
}
\cs_new:Nn \__flowfram_addto_prechap:
{

```

First check if \chapter exists:

```

\cs_if_exist:NT \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}%

\AddToHook { cmd / chapter / before }
{
  \ffprechapterhook
  \thispagestyle { \chapterfirstpagestyle }
}

```

`\ffprechapterhook` Hook at start of chapter (before page break issued) This is redundant now that hook are available but retained for backward-compatibility.

```

\newcommand*{\ffprechapterhook}{}

```

End of test if `\chapter` defined:

```

}
}

```

Now declare the options. NB v2.0 has switched from `xkeyval` to `l3keys`.

```

\keys_define:nn { flowfram }
{

```

Should the table of contents be adjusted.

```

adjust-toc .choice: ,

```

adjust the table of contents including the header (default):

```

adjust-toc / header .code:n =
{
  \cs_set:Nn \flowfram_toc:
  {
    \__flowfram_adjusted_toc:
  }
  \renewcommand \tableofcontents
  {
    \__flowfram_tableofcontents:
  }
} ,

```

adjust the table of contents but retain original header:

```

adjust-toc / noheader .code:n =
{
  \cs_set:Nn \flowfram_toc:
  {
    \__flowfram_org_tableofcontents:
  }
  \renewcommand \tableofcontents
  {
    \__flowfram_tableofcontents:
  }
} ,

```

don't adjust the table of contents (no support for thumbtab index or minitocs):

```
adjust-toc / off .code:n =
{
  \let \tableofcontents \__flowfram_org_tableofcontents:
} ,
```

If the header and footer are converted into dynamic frames, indicate how to adjust the page styles.

```
dynamic-page-style .choice: ,
dynamic-page-style .usage:n = { preamble } ,
dynamic-page-style / adjust .code:n =
{
  \__flowfram_only_pre_makedfheaderfooter:nn
  { dynamic-page-style = adjust }
  {
    \cs_set_eq:NN \__flowfram_adjust_page_styles:n \use:n
  }
} ,
dynamic-page-style / noadjust .code:n =
{
  \__flowfram_only_pre_makedfheaderfooter:nn
  { dynamic-page-style = noadjust }
  {
    \cs_set_eq:NN \__flowfram_adjust_page_styles:n \use_none:n
  }
} ,
dynamic-header-case .choice: ,
dynamic-header-case / uc .code:n =
{
  \cs_set:Nn \flowfram_header_case:n
  {
    \MakeUppercase { ##1 }
  }
} ,
dynamic-header-case / no-change .code:n =
{
  \cs_set:Nn \flowfram_header_case:n { ##1 }
} ,
dynamic-subheader-case .choice: ,
dynamic-subheader-case / uc .code:n =
{
  \cs_set:Nn \flowfram_subheader_case:n
  {
    \MakeUppercase { ##1 }
  }
} ,
dynamic-subheader-case / no-change .code:n =
{
  \cs_set:Nn \flowfram_subheader_case:n { ##1 }
} ,
```

```

dynamic-page-style-header-font .code:n =
{
  \tl_if_empty:nTF { #1 }
  {
    \cs_set:Nn \flowfram_header_font:n { ##1 }
  }
  {
    \cs_set:Nn \flowfram_header_font:n { #1 { ##1 } }
  }
} ,
dynamic-page-style-subheader-font .code:n =
{
  \tl_if_empty:nTF { #1 }
  {
    \cs_set:Nn \flowfram_subheader_font:n { ##1 }
  }
  {
    \cs_set:Nn \flowfram_subheader_font:n { #1 { ##1 } }
  }
} ,

```

How should ffempty page style behave?

```

dynamic-empty-page-style .choice: ,
dynamic-empty-page-style / empty .code:n =
{
  \tl_set:Nn \l__flowfram_ffempty_style_tl { \__flowfram_ps_empty: }
  \bool_set_false:N \l__flowfram_ps_ffempty_hides_bool
} ,
dynamic-empty-page-style / plain .code:n =
{
  \tl_set:Nn \l__flowfram_ffempty_style_tl { \__flowfram_ps_plain: }
  \bool_set_false:N \l__flowfram_ps_ffempty_hides_bool
} ,
dynamic-empty-page-style / headings .code:n =
{
  \tl_set:Nn \l__flowfram_ffempty_style_tl { \__flowfram_ps_headings: }
  \bool_set_false:N \l__flowfram_ps_ffempty_hides_bool
} ,
dynamic-empty-page-style / myheadings .code:n =
{
  \tl_set:Nn \l__flowfram_ffempty_style_tl { \__flowfram_ps_myheadings: }
  \bool_set_false:N \l__flowfram_ps_ffempty_hides_bool
} ,
dynamic-empty-page-style / ignore .code:n =
{
  \tl_clear:N \l__flowfram_ffempty_style_tl
  \bool_set_false:N \l__flowfram_ps_ffempty_hides_bool
} ,
dynamic-empty-page-style / hide .code:n =
{

```

```

        \bool_set_true:N \l__flowfram_ps_ffempty_hides_bool
    } ,
dynamic-empty-page-style / show .code:n =
{
    \bool_set_false:N \l__flowfram_ps_ffempty_hides_bool
} ,

```

If backmatter is supported, adjust the secnumdepth counter:

```

backmatter-sections .choice: ,
backmatter-sections / no-number .code:n =
{
    \cs_set:Nn \__flowfram_backmatter_setsecnumdepth:
    {
        \setcounter { secnumdepth } { -1 }
    }
} ,
backmatter-sections / no-change .code:n =
{
    \cs_set:Nn \__flowfram_backmatter_setsecnumdepth:
    {
    }
} ,

```

Indicate if unstarred units outside of the main matter should have thumbtabs.

```

matter-thumbtabs .choice: ,
matter-thumbtabs / main-only .code:n =
{
    \cs_set:Nn \__flowfram_ttb_notmainmatter_num:n
    {
    }
} ,
matter-thumbtabs / all .code:n =
{
    \cs_set:Nn \__flowfram_ttb_notmainmatter_num:n
    {
        \__flowfram_ttb_num:n { ##1 }
    }
} ,
matter-thumbtabs / not-front .code:n =
{
    \cs_set:Nn \__flowfram_ttb_notmainmatter_num:n
    {
        \bool_if:NF \g__flowfram_frontmatter_bool
        {
            \__flowfram_ttb_num:n { ##1 }
        }
    }
} ,
matter-thumbtabs / not-back .code:n =
{
    \cs_set:Nn \__flowfram_ttb_notmainmatter_num:n

```

```

    {
      \bool_if:NF \g__flowfram_backmatter_bool
      {
        \__flowfram_ttb_num:n { ##1 }
      }
    }
  },

```

Indicate which thumbtabs should have hyperlinks (if supported).

```

thumbtab-links .choice: ,
thumbtab-links / toc-only .code:n =
{
  \renewcommand \thumbtabindexformat [ 3 ]
  {
    \thumbtabhyperlinkformat { ##1 } { ##2 } { ##3 }
  }
  \renewcommand \thumbtabregularformat [ 3 ]
  {
    \thumbtabformat { ##2 } { ##3 }
  }
} ,
thumbtab-links / all .code:n =
{
  \renewcommand \thumbtabindexformat [ 3 ]
  {
    \thumbtabhyperlinkformat { ##1 } { ##2 } { ##3 }
  }
  \renewcommand \thumbtabregularformat [ 3 ]
  {
    \thumbtabhyperlinkformat { ##1 } { ##2 } { ##3 }
  }
} ,
thumbtab-links / none .code:n =
{
  \renewcommand \thumbtabindexformat [ 3 ]
  {
    \thumbtabformat { ##2 } { ##3 }
  }
  \renewcommand \thumbtabregularformat [ 3 ]
  {
    \thumbtabformat { ##2 } { ##3 }
  }
} ,

```

Indicate whether or not unstarred units should have thumbtabs.

```

unstarred-thumbtabs .bool_gset:N =
\g__flowfram_save_nonum_thumbtabs_bool ,

```

Indicate whether or not thumbtab indexes should appear in the table of contents and, if so, whether or not to align them.

```

toc-thumbtabs .choice: ,

```



```

toc-thumbtabs / aligned .code:n =
{
  \aligntoctrue
  \bool_set_true:N \l__flowfram_thumbtabs_in_toc_bool
  \bool_set_false:N \l__flowfram_thumbtabs_span_toc_bool
},
toc-thumbtabs / unaligned .code:n =
{
  \aligntocfalse
  \bool_set_true:N \l__flowfram_thumbtabs_in_toc_bool
  \bool_set_false:N \l__flowfram_thumbtabs_span_toc_bool
},
toc-thumbtabs / false .code:n =
{
  \bool_set_false:N \l__flowfram_thumbtabs_in_toc_bool
},
toc-thumbtabs / true .code:n =
{
  \aligntocfalse
  \bool_set_true:N \l__flowfram_thumbtabs_in_toc_bool
  \bool_set_true:N \l__flowfram_thumbtabs_span_toc_bool
},
toc-thumbtabs .default:n = { true } ,
sections-extra-option .choice: ,
sections-extra-option / as-original .code:n =
{
  \bool_set_true:N \l__flowfram_section_header_opt_bool
  \bool_set_false:N \l__flowfram_section_header_opt_thumbtab_bool
},
sections-extra-option / thumbtab-only .code:n =
{
  \bool_set_false:N \l__flowfram_section_header_opt_bool
  \bool_set_true:N \l__flowfram_section_header_opt_thumbtab_bool
},
sections-extra-option / original-and-thumbtab .code:n =
{
  \bool_set_true:N \l__flowfram_section_header_opt_bool
  \bool_set_true:N \l__flowfram_section_header_opt_thumbtab_bool
},
sections-option-keyval .bool_set:N = \l__flowfram_section_keyval_opt_bool ,
column-changes .choice: ,
column-changes / ignore .code:n =
{
  \bool_gset_true:N \g__flowfram_ignore_column_changes_bool
  \bool_gset_false:N \g__flowfram_clearpage_column_changes_bool
},
column-changes / clearpage .code:n =
{

```

Clear the page but otherwise ignore the column change command.

```

\bool_gset_true:N \g__flowfram_clearpage_column_changes_bool
\bool_gset_true:N \g__flowfram_ignore_column_changes_bool
} ,
column-changes / switch .code:n =
{

```

Switch to designated frames, if set up.

```

\bool_gset_true:N \g__flowfram_clearpage_column_changes_bool
\bool_gset_false:N \g__flowfram_ignore_column_changes_bool
} ,
draft .code:n = { \__flowfram_set_draft: } ,
final .code:n = { \__flowfram_set_final: } ,
verbose .choice: ,
verbose / true .code:n =
{
\let \flf@doifverbose \@firstofone
} ,
verbose / false .code:n =
{
\let \flf@doifverbose \@gobble
} ,
verbose .default:n = { true } ,
thumbtab-text .choice: ,
thumbtab-text / normal .code:n =
{
\cs_set_eq:NN \__flowfram_thumbtab_fmt:nn \use_i:nn
} ,
thumbtab-text / stack .code:n =
{
\cs_set_eq:NN
\__flowfram_thumbtab_fmt:nn
\__flowfram_thumbtab_fmt_stack:nn
} ,
thumbtab-text / rotate-right .code:n =
{
\cs_set_eq:NN
\__flowfram_thumbtab_fmt:nn
\__flowfram_thumbtab_fmt_sideways_oneside_right:nn
} ,
thumbtab-text / rotate-left .code:n =
{
\cs_set_eq:NN
\__flowfram_thumbtab_fmt:nn
\__flowfram_thumbtab_fmt_sideways_oneside_left:nn
} ,
thumbtab-text / rotate .code:n =
{
\cs_set_eq:NN
\__flowfram_thumbtab_fmt:nn
\__flowfram_thumbtab_fmt_sideways:nn

```

```
},
```

Provide a way of inhibiting frame rotation.

```
prohibit-frame-rotation .bool_set_inverse:N =  
  \l__flowfram_allow_frame_rotation_bool ,
```

Provide old option for backward compatibility.

```
rotate .choice: ,  
rotate / true .code:n =  
{  
  \bool_set_true:N \l__flowfram_allow_frame_rotation_bool  
  \cs_set_eq:NN  
    \__flowfram_thumbtab_fmt:nn  
    \__flowfram_thumbtab_fmt_sideways_oneside_right:nn  
},  
rotate / false .code:n =  
{  
  \bool_set_false:N \l__flowfram_allow_frame_rotation_bool  
  \cs_set_eq:NN  
    \__flowfram_thumbtab_fmt:nn  
    \__flowfram_thumbtab_fmt_stack:nn  
},  
rotate .default:n = { true } ,  
norotate .code:n =  
{  
  \bool_set_false:N \l__flowfram_allow_frame_rotation_bool  
  \cs_set_eq:NN  
    \__flowfram_thumbtab_fmt:nn  
    \__flowfram_thumbtab_fmt_stack:nn  
},
```

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

```
thumbtabs .choice: ,  
thumbtabs / title .code:n =  
{  
  \bool_set_false:N \l__flowfram_ttb_show_number_bool  
  \bool_set_true:N \l__flowfram_ttb_show_title_bool  
},  
thumbtabs / number .code:n =  
{  
  \bool_set_true:N \l__flowfram_ttb_show_number_bool  
  \bool_set_false:N \l__flowfram_ttb_show_title_bool  
},  
thumbtabs / both .code:n =  
{  
  \bool_set_true:N \l__flowfram_ttb_show_number_bool  
  \bool_set_true:N \l__flowfram_ttb_show_title_bool  
},  
thumbtabs / none .code:n =  
{
```

```

        \bool_set_false:N \l__flowfram_ttb_show_number_bool
        \bool_set_false:N \l__flowfram_ttb_show_title_bool
    } ,
    thumbtabs .default:n = { title } ,
    thumbtabs .usage:n = { preamble } ,

```

Provide old options for backward compatibility:

```

ttbtitle .code:n =
    {
        \bool_set_true:N \l__flowfram_ttb_show_title_bool
    } ,
ttbtitle .usage:n = { preamble } ,
ttbnotitle .code:n =
    {
        \bool_set_false:N \l__flowfram_ttb_show_title_bool
    } ,
ttbnotitle .usage:n = { preamble } ,
ttbnum .code:n =
    {
        \bool_set_true:N \l__flowfram_ttb_show_number_bool
    } ,
ttbnum .usage:n = { preamble } ,
ttbnonum .code:n =
    {
        \bool_set_false:N \l__flowfram_ttb_show_number_bool
    } ,
ttbnonum .usage:n = { preamble } ,

```

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`.

```

pages .choice: ,
pages / relative .code:n =
    {
        \tl_gset:Nn \g__flowfram_pagecounter_tl { \c@page }
    } ,
pages / absolute .code:n =
    {
        \tl_gset:Nn \g__flowfram_pagecounter_tl { \c@absolutepage }
    } ,
pages .usage:n = { preamble } ,

```

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.

```

color .choice: ,

```

```

color / true .code:n =
{
  \__flowfram_enable_color:
} ,
color / false .code:n =
{

```

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

```

  \__flowfram_disable_color:
} ,
color .default:n = { true } ,
color .usage:n = { preamble } ,

```

Provide nocolor option for backward compatibility:

```

nocolor .code:n =
{
  \__flowfram_disable_color:
} ,
nocolor .usage:n = { preamble } ,

```

Define options that set the direction.

```

LR .code:n = { \lefttorightcolumnstrue } ,
LR .usage:n = { preamble } ,
RL .code:n = { \lefttorightcolumnsfalse } ,
RL .usage:n = { preamble } ,
adjust-pre-chapter .choice: ,
adjust-pre-chapter / true .code:n =
{
  \cs_set:Nn \__flowfram_do_addto_prechap:
  {
    \__flowfram_addto_prechap:
  }
} ,
adjust-pre-chapter / false .code:n =
{
  \cs_set:Nn \__flowfram_do_addto_prechap: { }
} ,
adjust-pre-chapter .default:n = { true } ,
adjust-pre-chapter .usage:n = { load } ,
}

```

For options that can be set after the package has loaded:

```

\NewDocumentCommand \flowframsetup { m }
{
  \keys_set:nn { flowfram } { #1 }
}

```

Set the defaults. Version 2.0 color now initialised to on. (NB original version of flowfram was released in 2005 when there was less support for colour and rotation.)

```

\flowframsetup{color}

```

Apply any options set via flowframtkutils::

```
\IfPackageLoadedT { flowframtkutils }
{
  \clist_if_empty:NF \g__flowframtk_options_clist
  {
    \keys_set:nV { flowfram } \g__flowframtk_options_clist
  }
}
```

Switch off L3 syntax:

```
\ExplSyntaxOff
```

Now load color package to support grey.

```
\RequirePackage{color}
```

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

```
\ProcessKeyOptions[flowfram]
```

```
\ExplSyntaxOn
```

Prior to v2.0, `\textwidth` and `\textheight` were used to keep track of the typeblock. This causes a problem when the output routine commands set `\vsize` to `\textheight`. Now there are two new lengths to keep track of the typeblock, which are initialised to `\textwidth` and `\textheight`. The frames may be the exact height of the typeblock or may overlap, but this provides a base for commands that need to know the default area.

```
\newlength \typeblockwidth
\newlength \typeblockheight
\newlength \typeblockoffsety

\cs_new:Nn \flowfram_update_typeblock:
{
  \global \setlength \typeblockwidth \textwidth
  \global \setlength \typeblockheight \textheight
  \dim_gset:Nn \typeblockoffsety
  {
    \topmargin
    + \headheight
    + \headsep
    + \voffset
  }
}
\flowfram_update_typeblock:
```

Ideally geometry needs to be set first.

```
\AddToHook
{ package / geometry / after }
{
  \flowfram_update_typeblock:
}
```

Add pre-chapter hook if required:

```
\_flowfram_do_addto_prechap:
```

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}
```

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

```
\newcounter{thisframe}
\AddToHook
{ package / hyperref / after }
{
  \def \theHthisframe { \theabsolutePage . \arabic {thisframe} }
}
```

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

```
\NewDocumentCommand \labelflowidn { m }
{
  \group_begin:
  \def \@currentlabel { \theHthisframe }
  \label { #1 }
  \group_end:
}
```

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}
\AddToHook
{ package / hyperref / after }
{
  \def \theHdisplayedframe { \thePage . \arabic{displayedframe} }
}
```

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

```
\NewDocumentCommand \labelflow { m }
{
  \group_begin:
  \def \@currentlabel { \theHdisplayedframe }
  \label { #1 }
  \group_end:
}
```

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

```
\newcounter{maxstatic}
```

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

```
\newcounter{maxdynamic}
```

Define some temporary variables. These have been changed to L3 integer variables in v2.0.

```
\int_new:N \l__flowfram_current_frame_int  
\int_new:N \l__flowfram_current_abspage_int  
\int_new:N \l__flowfram_current_static_int  
\int_new:N \l__flowfram_current_dynamic_int  
\int_new:N \l__flowfram_col_int  
\int_new:N \l__flowfram_tmpa_int  
\int_new:N \l__flowfram_tmpb_int  
\int_new:N \l__flowfram_id_int  
\int_new:N \l__flowfram_id_ii_int  
\int_new:N \l__flowfram_type_int  
\int_new:N \l__flowfram_type_ii_int  
\int_new:N \l__flowfram_range_start_int  
\int_new:N \l__flowfram_range_end_int  
\int_new:N \l__flowfram_shape_int  
\dim_new:N \l__flowfram_tmpa_dim  
\dim_new:N \l__flowfram_tmpb_dim  
\clist_new:N \l__flowfram_tmp_clist
```

Dimensions used for both static and dynamic frames:

```
\dim_new:N \l__flowfram_sfdf_height_dim  
\dim_new:N \l__flowfram_sfdf_width_dim  
\tl_new:N \l__flowfram_tmpa_tl  
\bool_new:N \l__flowfram_found_bool  
\tl_new:N \g__flowfram_next_tl  
  
\dim_new:N \l__flowfram_x_dim  
\dim_new:N \l__flowfram_evenx_dim  
\dim_new:N \l__flowfram_y_dim  
\dim_new:N \l__flowfram_eveny_dim  
\dim_new:N \l__flowfram_width_dim  
\dim_new:N \l__flowfram_height_dim  
\dim_new:N \l__flowfram_offset_dim  
  
\tl_new:N \l__flowfram_remainder_tl  
\tl_new:N \l__flowfram_pages_tl  
\tl_new:N \l__flowfram_next_pages_tl  
\tl_new:N \l__flowfram_exclude_pages_tl  
\clist_new:N \l__flowfram_pages_clist  
\clist_new:N \l__flowfram_even_pages_clist  
\clist_new:N \l__flowfram_odd_pages_clist  
\clist_new:N \l__flowfram_exclude_pages_clist  
\tl_new:N \l__flowfram_label_tl  
\tl_new:N \l__flowfram_frametype_tl  
\tl_new:N \l__flowfram_style_tl  
\tl_new:N \l__flowfram_parindent_tl  
\tl_new:N \l__flowfram_shape_tl
```



```

\tl_new:N \l__flowfram_valign_tl
\tl_new:N \l__flowfram_minipage_tl
\tl_new:N \l__flowfram_contents_tl

```

Empty if not set:

```

\tl_new:N \l__flowfram_width_tl
\tl_new:N \l__flowfram_height_tl
\tl_new:N \l__flowfram_x_tl
\tl_new:N \l__flowfram_y_tl
\tl_new:N \l__flowfram_evenx_tl
\tl_new:N \l__flowfram_eveny_tl
\tl_new:N \l__flowfram_oddx_tl
\tl_new:N \l__flowfram_oddy_tl
\tl_new:N \l__flowfram_offset_tl
\tl_new:N \l__flowfram_angle_tl
\tl_new:N \l__flowfram_margin_tl

```

Empty if not set otherwise true or false:

```

\tl_new:N \l__flowfram_frameflag_tl
\tl_new:N \l__flowfram_clearflag_tl
\tl_new:N \l__flowfram_hideflag_tl
\tl_new:N \l__flowfram_hidethisflag_tl

```

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

```

\newlength\sdfparindent

```

Utility command. Iterate over frame IDs in the list but the list may just be the keywords `all`, `odd` or `even` or may contain ranges. The id can be referenced in `<code>` with `\l__flowfram_id_int`. Syntax `{<clist>}{<type>}{<code>}`

```

\cs_new:Nn \__flowfram_map_idns:nnn
{
  \tl_if_eq:NnTF \c_flowfram_all_tl { #1 }
  {
    \int_step_inline:nn { \value { max #2 } }
    {
      \int_set:Nn \l__flowfram_id_int { ##1 }
      #3
    }
  }
  {
    \tl_if_eq:NnTF \c_flowfram_odd_tl { #1 }
    {
      \int_step_inline:nnn { \c_one_int } { 2 } { \value { max #2 } }
      {
        \int_set:Nn \l__flowfram_id_int { ##1 }
        #3
      }
    }
  }
  {

```

```

\tl_if_eq:NnTF \c_flowfram_even_tl { #1 }
{
  \int_step_inline:nnn { 2 } { 2 } { \value { max #2 } }
  {
    \int_set:Nn \l__flowfram_id_int { ##1 }
    #3
  }
}
{
  \clist_map_inline:nn { #1 }
  {
    \__flowfram_get_range:n { ##1 }
    \int_compare:nNnT { \l__flowfram_range_start_int } < { \c_one_int }
    {
      \int_set_eq:NN \l__flowfram_range_start_int \c_one_int
    }
    \int_compare:nNnT
    { \l__flowfram_range_end_int } > { \value { max #2 } }
    {
      \int_set_eq:Nc \l__flowfram_range_end_int { c@max #2}
    }
    \int_compare:nNnTF
    { \l__flowfram_range_start_int } < { \l__flowfram_range_end_int }
    {
      \int_step_inline:nNnTF
      { \l__flowfram_range_start_int }
      { \l__flowfram_range_end_int }
      {
        \int_set:Nn \l__flowfram_id_int { ##1 }
        #3
      }
    }
  }
  {
    \int_set_eq:NN
    \l__flowfram_id_int
    \l__flowfram_range_start_int
    #3
  }
}
}
}
}
}
\cs_generate_variant:Nn \__flowfram_map_idns:nnn { enn }

```

1.2 Frame Attributes

Convenient commands to define frames and set and access frame attributes.

Determine the frame type from its label. The integer variable provided will

be set to one of the constants if valid otherwise trigger an error and set to -1.

```

\cs_new:Nn \__flowfram_get_frame_type:Nn
{
  \int_if_exist:cTF { c_flowfram_frame_type_ #2 _int }
  {
    \int_set_eq:Nc #1 { c_flowfram_frame_type_ #2 _int }
  }
  {
    \int_set:Nn #1 { - \c_one_int }
    \msg_error:nnn { flowfram } { invalid-frame-type } { #2 }
  }
}
}
\cs_new:Nn \__flowfram_get_frame_type:n
{
  \__flowfram_get_frame_type:Nn
  \l__flowfram_type_int
  { #1 }
}

```

Store a list of all flow frame labels:

```
\seq_new:N \g__flowfram_flowlabels_seq
```

Store a list of all static frame labels:

```
\seq_new:N \g__flowfram_staticlabels_seq
```

Store a list of all dynamic frame labels:

```
\seq_new:N \g__flowfram_dynamiclabels_seq
```

Set up the keys for use with `\setflowframe`, `\setstaticframe` and `\setdynamicframe`. Version 2.0: switched to `l3keys`.

```

\keys_define:nn { flowfram / frame }
{

```

Frame width. (Dimensions are temporarily stored in a token list variable which is initialised to empty to determine whether or not the dimension has been updated.)

```

width .tl_set:N = \l__flowfram_width_tl ,
width .groups:n = { flow , static , dynamic } ,

```

Frame height.

```

height .tl_set:N = \l__flowfram_height_tl ,
height .groups:n = { flow , static , dynamic } ,

```

Frame *x* co-ordinate (odd and even pages):

```

x .tl_set:N = \l__flowfram_x_tl ,
x .groups:n = { flow , static , dynamic } ,

```

Frame *y* co-ordinate (odd and even pages):

```

y .tl_set:N = \l__flowfram_y_tl ,
y .groups:n = { flow , static , dynamic } ,

```

Frame x co-ordinate (even pages only):

```
evenx .tl_set:N = \l__flowfram_evenx_tl ,
evenx .groups:n = { flow , static , dynamic } ,
```

Frame y co-ordinate (even pages only):

```
eveny .tl_set:N = \l__flowfram_eveny_tl ,
eveny .groups:n = { flow , static , dynamic } ,
```

Frame x co-ordinate (odd pages only if twoside implemented):

```
oddx .tl_set:N = \l__flowfram_oddx_tl ,
oddx .groups:n = { flow , static , dynamic } ,
```

Frame y co-ordinate (odd pages only if twoside implemented):

```
oddy .tl_set:N = \l__flowfram_oddy_tl ,
oddy .groups:n = { flow , static , dynamic } ,
```

New identification label (IDL) for frame:

```
label .tl_set:e:N = \l__flowfram_label_tl ,
label .groups:n = { flow , static , dynamic } ,
```

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

```
border .choice: ,
border .default:n = { plain } ,
border / plain .code:n =
{
  \tl_set:Nn \l__flowfram_frameflag_tl { true }
  \tl_set:Nn \l__flowfram_frametype_tl { fbox }
} ,
border / none .code:n =
{
  \tl_set:Nn \l__flowfram_frameflag_tl { false }
} ,
border / unknown .code:n =
{
  \tl_set:Nn \l__flowfram_frameflag_tl { true }
  \tl_set:Ne \l__flowfram_frametype_tl { #1 }
} ,
border .groups:n = { flow , static , dynamic } ,
```

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

```
bordercolor .tl_set:e:N = \l__flowfram_bordercolor_tl ,
bordercolour .tl_set:e:N = \l__flowfram_bordercolor_tl ,
border-color .tl_set:e:N = \l__flowfram_bordercolor_tl ,
border-colour .tl_set:e:N = \l__flowfram_bordercolor_tl ,
bordercolor .groups:n = { flow , static , dynamic } ,
bordercolour .groups:n = { flow , static , dynamic } ,
border-color .groups:n = { flow , static , dynamic } ,
border-colour .groups:n = { flow , static , dynamic } ,
```

Frame's text colour.

```
textcolor .tl_set_e:N = \l__flowfram_textcolor_tl ,
textcolour .tl_set_e:N = \l__flowfram_textcolor_tl ,
text-color .tl_set_e:N = \l__flowfram_textcolor_tl ,
text-colour .tl_set_e:N = \l__flowfram_textcolor_tl ,
textcolor .groups:n = { flow , static , dynamic } ,
textcolour .groups:n = { flow , static , dynamic } ,
text-color .groups:n = { flow , static , dynamic } ,
text-colour .groups:n = { flow , static , dynamic } ,
```

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.

```
backcolor .tl_set_e:N = \l__flowfram_backcolor_tl ,
backcolour .tl_set_e:N = \l__flowfram_backcolor_tl ,
back-color .tl_set_e:N = \l__flowfram_backcolor_tl ,
back-colour .tl_set_e:N = \l__flowfram_backcolor_tl ,
backcolor .groups:n = { flow , static , dynamic } ,
backcolour .groups:n = { flow , static , dynamic } ,
back-color .groups:n = { flow , static , dynamic } ,
back-colour .groups:n = { flow , static , dynamic } ,
```

Page list for which the frame should appear:

```
pages .tl_set_e:N = \l__flowfram_pages_tl ,
pages .groups:n = { flow , static , dynamic } ,
```

Exclusion list:

```
excludepages .tl_set_e:N = \l__flowfram_exclude_pages_tl ,
excludepages .groups:n = { flow , static , dynamic } ,
```

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. Value may be `compute` or a length.

```
offset .tl_set:N = \l__flowfram_offset_tl ,
offset .groups:n = { flow , static , dynamic } ,
```

Angle to rotate frame (if allowed):

```
angle .code:n =
{
  \bool_if:NTF \l__flowfram_allow_frame_rotation_bool
  {
    \tl_set:Nc \l__flowfram_angle_tl { #1 }
  }
  {
    \tl_if_eq:enF { #1 } { 0 }
    {
      \msg_info:nn { flowfram } { info-no-rotation }
    }
  }
} ,
```

```
angle .groups:n = { flow , static , angle } ,
```

This key is only for flow frames:

```
margin .choice: ,  
margin / inner .code:n =  
  { \tl_set:Nn \l__flowfram_margin_tl { inner } } ,  
margin / outer .code:n =  
  { \tl_set:Nn \l__flowfram_margin_tl { outer } } ,  
margin / left .code:n =  
  { \tl_set:Nn \l__flowfram_margin_tl { left } } ,  
margin / right .code:n =  
  { \tl_set:Nn \l__flowfram_margin_tl { right } } ,  
margin .groups:n = { flow } ,
```

This key is only for static frame and dynamic frames. (Not using a boolean as the token list is initially clear to indicate no change.)

```
clear .choice: ,  
clear / true .code:n =  
  {  
    \tl_set:Nn \l__flowfram_clearflag_tl { true }  
  } ,  
clear / false .code:n =  
  {  
    \tl_set:Nn \l__flowfram_clearflag_tl { false }  
  } ,  
clear .default:n = { true } ,  
clear .groups:n = { static, dynamic } ,
```

This key is only for dynamic frames:

```
style .choice: ,  
style / none .code:n =  
  {  
    \tl_set:Nn \l__flowfram_style_tl { @firstofone }  
  } ,  
style / unknown .code:n =  
  {  
    \tl_set:Ne \l__flowfram_style_tl { #1 }  
  } ,  
style .groups:n = { dynamic } ,
```

This key is only for static frames and dynamic frames.

```
shape .tl_set:N = \l__flowfram_shape_tl ,  
shape .groups:n = { static, dynamic } ,
```

This key is only for static frames and dynamic frames.

```
parindent .tl_set:N = \l__flowfram_parindent_tl ,  
parindent .groups:n = { static, dynamic } ,
```

These keys are only for static frames and dynamic frames.

```
valign .choice: ,  
valign / c .code:n =  
  { \tl_set:Nn \l__flowfram_valign_tl { c } } ,
```

```

valign / t .code:n =
  { \tl_set:Nn \l__flowfram_valign_tl { t } } ,
valign / b .code:n =
  { \tl_set:Nn \l__flowfram_valign_tl { b } } ,
valign .groups:n = { static, dynamic } ,

```

(Not using a boolean as token list variables initially clear to indicate no change.)

```

hide .choice: ,
hide / true .code:n =
  { \tl_set:Nn \l__flowfram_hideflag_tl { true } } ,
hide / false .code:n =
  { \tl_set:Nn \l__flowfram_hideflag_tl { false } } ,
hide .default:n = { true } ,
hide .groups:n = { static, dynamic } ,
hidethis .choice: ,
hidethis / true .code:n =
  { \tl_set:Nn \l__flowfram_hidethisflag_tl { true } } ,
hidethis / false .code:n =
  { \tl_set:Nn \l__flowfram_hidethisflag_tl { false } } ,
hidethis .default:n = { true } ,
hidethis .groups:n = { static, dynamic } ,

```

The following is reserved for L^AT_EX to HTML parsers. With T_EX engines, they just write information to the aux file. The value should be the options to pass to the parser. Suggested options: `show=⟨bool⟩` (if true, put the frame contents in a div element at this point), `style={⟨css-style⟩}` (if set, add the style attribute to the div element), `class={⟨css-class⟩}` (if set, add the class attribute to the div element), `image={⟨bool⟩}` (if true, convert the frame contents to an image), `image-type={⟨mime type⟩}` (if `image=true`, the image should be this type).

```

html .tl_set:N = \l__flowfram_html_tl ,
html .groups:n = { static, dynamic } ,
}
Syntax {⟨options⟩}{⟨type⟩}{⟨idn⟩}

```

```

\cs_new:Nn \l__flowfram_set_keys_for_type:nnn
{
  \tl_clear:N \l__flowfram_frameflag_tl
  \tl_clear:N \l__flowfram_width_tl
  \tl_clear:N \l__flowfram_height_tl
  \tl_clear:N \l__flowfram_margin_tl
  \tl_clear:N \l__flowfram_x_tl
  \tl_clear:N \l__flowfram_y_tl
  \tl_clear:N \l__flowfram_frametype_tl
  \tl_clear:N \l__flowfram_bordercolor_tl
  \tl_clear:N \l__flowfram_valign_tl
  \tl_clear:N \l__flowfram_style_tl
  \tl_clear:N \l__flowfram_hideflag_tl
  \tl_clear:N \l__flowfram_hidethisflag_tl
  \tl_clear:N \l__flowfram_textcolor_tl
  \tl_clear:N \l__flowfram_clearflag_tl
}

```

```

\tl_clear:N \l__flowfram_offset_tl
\tl_clear:N \l__flowfram_pages_tl
\tl_clear:N \l__flowfram_label_tl
\tl_clear:N \l__flowfram_backcolor_tl
\tl_clear:N \l__flowfram_evenx_tl
\tl_clear:N \l__flowfram_eveny_tl
\tl_clear:N \l__flowfram_oddx_tl
\tl_clear:N \l__flowfram_oddy_tl
\tl_clear:N \l__flowfram_angle_tl
\tl_set_eq:NN \l__flowfram_exclude_pages_tl \c_novalue_tl
\tl_set_eq:NN \l__flowfram_shape_tl \c_novalue_tl
\tl_set_eq:NN \l__flowfram_parindent_tl \c_novalue_tl
\tl_clear:N \l__flowfram_html_tl
\keys_set_groups:nnnN { flowfram / frame } { #2 } { #1 }
  \l__flowfram_remainder_tl
\tl_if_empty:NF \l__flowfram_remainder_tl
{
  \__flowfram_error:eee
  { frame-unsupported-options }
  { #2 } { \l__flowfram_remainder_tl }
}

```

Set common options.

```

\tl_if_empty:NF \l__flowfram_frameflag_tl
{
  \flowfram_frame_set_bool_from_option:nnnV
  { #2 } { hasframe } { #3 }
  \l__flowfram_frameflag_tl
}
\tl_if_empty:NF \l__flowfram_width_tl
{
  \flowfram_frame_set_dim:nnnn { #2 } { width } { #3 }
  { \l__flowfram_width_tl }
}
\tl_if_empty:NF \l__flowfram_height_tl
{
  \flowfram_frame_set_dim:nnnn { #2 } { height } { #3 }
  { \l__flowfram_height_tl }
}
\tl_if_empty:NF \l__flowfram_x_tl
{
  \flowfram_frame_set_dim:nnnn { #2 } { posx } { #3 }
  { \l__flowfram_x_tl }
  \flowfram_frame_set_dim:nnnn { #2 } { evenx } { #3 }
  { \l__flowfram_x_tl }
}
\tl_if_empty:NF \l__flowfram_y_tl
{
  \flowfram_frame_set_dim:nnnn { #2 } { posy } { #3 }
  { \l__flowfram_y_tl }
}

```



```

\flowfram_frame_set_dim:nnnn { #2 } { eveny } { #3 }
  { \l__flowfram_y_tl }
}
\tl_if_empty:NF \l__flowfram_evenx_tl
{
\flowfram_frame_set_dim:nnnn { #2 } { evenx } { #3 }
  { \l__flowfram_evenx_tl }
}
\tl_if_empty:NF \l__flowfram_eveny_tl
{
\flowfram_frame_set_dim:nnnn { #2 } { eveny } { #3 }
  { \l__flowfram_eveny_tl }
}
\tl_if_empty:NF \l__flowfram_oddx_tl
{
\flowfram_frame_set_dim:nnnn { #2 } { posx } { #3 }
  { \l__flowfram_oddx_tl }
}
\tl_if_empty:NF \l__flowfram_oddy_tl
{
\flowfram_frame_set_dim:nnnn { #2 } { posy } { #3 }
  { \l__flowfram_oddy_tl }
}
\tl_if_empty:NF \l__flowfram_label_tl
{
\__flowfram_set_frame_id:nnV { #2 } { #3 } \l__flowfram_label_tl
}
\tl_if_empty:NF \l__flowfram_frametype_tl
{
\flowfram_frame_set_tl:nnnV
  { #2 } { frametype } { #3 }
  \l__flowfram_frametype_tl
}
\tl_if_empty:NF \l__flowfram_bordercolor_tl
{
\exp_after:wN \__flowfram_set_frame_color:w
  \l__flowfram_bordercolor_tl
  \q_stop { #2 } { bordercolor } { #3 }
}
\tl_if_empty:NF \l__flowfram_textcolor_tl
{
\exp_after:wN \__flowfram_set_frame_color:w
  \l__flowfram_textcolor_tl
  \q_stop { #2 } { textcolor } { #3 }
}
\tl_if_empty:NF \l__flowfram_backcolor_tl
{
\exp_after:wN \__flowfram_set_frame_color:w
  \l__flowfram_backcolor_tl
  \q_stop { #2 } { backcolor } { #3 }
}

```

```

}
\tl_if_empty:NF \l__flowfram_pages_tl
{
  \flowfram_set_frame_pagelist:nnV
  { #2 } { #3 } \l__flowfram_pages_tl
}
\exp_args:NV \tl_if_novalue:nF \l__flowfram_exclude_pages_tl
{
  \flowfram_set_frame_excludelist:nnV
  { #2 } { #3 } \l__flowfram_exclude_pages_tl
}
\tl_if_empty:NF \l__flowfram_offset_tl
{
  \flowfram_frame_set_tl:nnnV
  { #2 } { offset } { #3 }
  \l__flowfram_offset_tl
}
\tl_if_empty:NF \l__flowfram_angle_tl
{
  \flowfram_frame_set_tl:nnnV
  { #2 } { angle } { #3 }
  \l__flowfram_angle_tl
}
\tl_if_empty:NF \l__flowfram_html_tl
{
  \dim_zero:N \l__flowfram_sdf_width_dim
  \dim_zero:N \l__flowfram_sdf_height_dim
  \flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_sdf_width_dim
  { #2 } { width } { #3 }
  \flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_sdf_height_dim
  { #2 } { height } { #3 }
  \tl_put_left:Ne \l__flowfram_html_tl
  {
    width = \dim_eval:n { \l__flowfram_sdf_width_dim } ,
    height = \dim_eval:n { \l__flowfram_sdf_height_dim } ,
  }
  \exp_args:NV \__flowfram_write_html_opts:nnn
  \l__flowfram_html_tl { #2 } { \int_eval:n { #3 } }
}
}
}

```

For the benefit of L^AT_EX to HTML parsers.

```
\int_new:N \g__flowfram_html_opts_int
```

Each option set has a unique numeric identifier to help match up the information in the aux file with the setting in the L^AT_EX document. The following starts by using the preamble command but will be changed in the begin document hook to use the document command.

```
\cs_new:Nn \__flowfram_write_html_opts:nnn
{
  \__flowfram_preamble_write_html_opts:nnn { #1 } { #2 } { #3 }
}
```

The command used in the preamble:

```
\cs_new:Nn \__flowfram_preamble_write_html_opts:nnn
{
  \int_gincr:N \g__flowfram_html_opts_int
  \protected@write \@auxout
  { \let \theabsolute page \relax }
  {
    \string \flowfram@preamble@htmlopts
    { \int_use:N \g__flowfram_html_opts_int }
    { #2 } { #3 } { #1 }
    { \thepage }
    { \theabsolute page }
  }
}
```

The command used in the document:

```
\cs_new:Nn \__flowfram_doc_write_html_opts:nnn
{
  \int_gincr:N \g__flowfram_html_opts_int
  \protected@write \@auxout
  { \let \theabsolute page \relax }
  {
    \string \flowfram@doc@htmlopts
    { \int_use:N \g__flowfram_html_opts_int }
    { #2 } { #3 } { #1 }
    { \thepage }
    { \theabsolute page }
  }
}
```

`\flowfram@preamble@htmlopts` Syntax: `{<opt-id>}{<frame-type>}{<idn>}{<options>}{<page>}{<absolute-page>}`
`\newcommand*\flowfram@preamble@htmlopts[6]{}`

`\flowfram@doc@htmlopts` Syntax: `{<opt-id>}{<frame-type>}{<idn>}{<options>}{<page>}{<absolute-page>}`
`\newcommand*\flowfram@doc@htmlopts[6]{}`

`\@setframecol` Set the colour of the frame, this is a little tricky because the model may need to be specified in square brackets but the specs might not be grouped (e.g. `[rgb]1,0,0` vs `[named]red` vs `red`). First check to see if a colour model has been specified. Version 2.0 renamed `\@setframecol` `[<model>]{<specs>\q_stop}{<type>}{<attr>}{<idn>}`

```
\cs_new:Npn \__flowfram_set_frame_color:w
{
  \@ifnextchar [
    \__flowfram_set_frame_color:wnnnn
```

```

    \_flowfram_set_frame_color:wnnn
}

```

\@@setframecol A colour model has been specified. Version 2.0 renamed \@@setframecol.

```

\cs_new:Npn \_flowfram_set_frame_color:wnnnn
  [ #1 ] #2 \q_stop #3 #4 #5
{
  \flowfram_frame_set_tl:nnne
  { #3 } { #4 } { #5 }
  { [ #1 ] { #2 } }
}

```

\@@setfr@mecol A colour model has not been specified. Version 2.0 renamed \@@setfr@mecol.

```

\cs_new:Npn \_flowfram_set_frame_color:wnnn
  #1 \q_stop #2 #3 #4
{
  \flowfram_frame_set_tl:nnne
  { #2 } { #3 } { #4 }
  { { #1 } }
}

```

Syntax: {<page list>}{<width>}{<height>}{<x>}{<y>}{<label>}{<type>}

```

\cs_new_nopar:Nn \_flowfram_new_frame:nnnnnnn
{
  \int_gincr:c { c@max #7 }
}

```

Set common attributes.

```

\flowfram_frame_new_bool:nnnN
  { #7 } { hasframe } { \int_use:c { c@max #7 } } \c_false_bool
\flowfram_frame_new_dim:nnnn
  { #7 } { width } { \int_use:c { c@max #7 } } { #2 }
\flowfram_frame_new_dim:nnnn
  { #7 } { height } { \int_use:c { c@max #7 } } { #3 }
\flowfram_frame_new_dim:nnnn
  { #7 } { posx } { \int_use:c { c@max #7 } } { #4 }
\flowfram_frame_new_dim:nnnn
  { #7 } { posy } { \int_use:c { c@max #7 } } { #5 }
\flowfram_frame_new_dim:nnnn
  { #7 } { evenx } { \int_use:c { c@max #7 } } { #4 }
\flowfram_frame_new_dim:nnnn
  { #7 } { eveny } { \int_use:c { c@max #7 } } { #5 }
\flowfram_frame_new_tl:nnnn
  { #7 } { frametype } { \int_use:c { c@max #7 } } { fbox }
\flowfram_frame_new_tl:nnnn
  { #7 } { bordercolor } { \int_use:c { c@max #7 } } { \flowframecol }
\flowfram_frame_new_tl:nnnn
  { #7 } { textcolor } { \int_use:c { c@max #7 } } { \flowframetextcol }
\flowfram_frame_new_tl:nnnn
  { #7 } { bgcolor } { \int_use:c { c@max #7 } } { { none } }
\flowfram_frame_new_tl:nnnn

```

```

    { #7 } { angle } { \int_use:c { c@max #7 } } { 0 }
\clist_set:Ne \l__flowfram_pages_clist { #1 }
\exp_args:NV
\flowfram_if_valid_pagelist:nTF \l__flowfram_pages_clist
{
  \flowfram_frame_new_clist:nnnV
  { #7 } { pagelist } { \int_use:c { c@max #7 } }
  \l__flowfram_pages_clist
}
{
  \msg_error:nnn { flowfram } { invalid-pagelist } { #1 }
  \flowfram_frame_new_clist:nnnn
  { #7 } { pagelist } { \int_use:c { c@max #7 } } { all }
}

```

Page exclusion list:

```

\flowfram_frame_new_clist:nnnn
{ #7 } { excludelist } { \int_use:c { c@max #7 } } { }
\flowfram_frame_new_tl:nnnn
{ #7 } { offset } { \int_use:c { c@max #7 } } { compute }
\__flowfram_set_frame_id:nnn { #7 } { \int_use:c { c@max #7 } } { #6 }
}

```

Set the frame's label. $\langle type \rangle \langle idn \rangle \langle label \rangle$

```

\cs_new:Nn \__flowfram_set_frame_id:nnn
{
  \tl_set:Ne \l__flowfram_label_tl { #3 }
  \int_compare:nNnTF
  { #2 } > { \seq_count:c { g__flowfram_ #1 labels_seq } }
  {

```

This is a label for a new frame.

```

\exp_args:NnV
  \__flowfram_checkunique_frame_idl:nnF { #1 } \l__flowfram_label_tl
  {

```

Not unique so fallback on IDN (which isn't guaranteed to be unique if another frame has confusingly been given a numeric label).

```

    \tl_set:Ne \l__flowfram_label_tl { \int_eval:n { #2 } }
  }
  \seq_gput_right:cV { g__flowfram_ #1 labels_seq } \l__flowfram_label_tl
  \flowfram_frame_new_tl:nnnV
  { #1 } { label } { #2 }
  \l__flowfram_label_tl
}
{

```

This is a label for an existing flow frame. Temporarily set the corresponding sequence item to empty before checking uniqueness.

```

  \seq_set_item:cnn { g__flowfram_ #1 labels_seq } { #2 } { }
  \exp_args:NnV
  \__flowfram_checkunique_frame_idl:nnF { #1 } \l__flowfram_label_tl

```

```

    {
      \tl_set:Nc \l__flowfram_label_tl { \int_eval:n { #2 } }
    }
  \exp_args:NcnV
  \seq_gset_item:Nnn
  { g__flowfram_ #1 labels_seq }
  { #2 }
  \l__flowfram_label_tl
  \flowfram_frame_set_tl:nnnV
  { #1 } { label } { #2 }
  \l__flowfram_label_tl
}
}
\cs_generate_variant:Nn \__flowfram_set_frame_id:nnn { nne , nnV }
  Check if label is unique for given frame type. Syntax  $\langle type \rangle \{ \langle label \rangle \}$ 
\prg_new_conditional:Nnn \__flowfram_checkunique_frame_idl:nn
{ T, F, TF }
{
  \seq_if_in:cnTF { g__flowfram_ #1 labels_seq } { #2 }
  {

```

Label already exists. Find the index number to help with the error message.

```

  \exp_args:Nc
  \seq_map_indexed_inline:Nn
  { g__flowfram_ #1 labels_seq }
  {
    \tl_if_eq:nnT { ##2 } { #2 }
    {
      \msg_error:nnnnn { flowfram } { label-defined }
      { #2 } { #1 } { ##1 }
      \seq_map_break:
    }
  }
  \prg_return_false:
}
{ \prg_return_true: }
}

```

Test if label exists for given type of frame. Syntax $\langle type \rangle \{ \langle label \rangle \} \langle int-var \rangle$
 If true, the corresponding IDN is stored in the $\langle int-var \rangle$

```

\prg_new_conditional:Nnn \__flowfram_if_frame_label_exists:nnN
{ TF , T , F }
{
  \int_zero:N #3
  \exp_args:Nc \seq_map_indexed_inline:Nn
  { g__flowfram_ #1 labels_seq }
  {
    \tl_if_eq:neT { ##2 } { #2 }
    {
      \int_set:Nn #3 { ##1 }
    }
  }
}

```

```

        \seq_map_break:
      }
    }
  \int_if_zero:nTF { #3 }
  { \prg_return_false: }
  { \prg_return_true: }
}

```

The simplest way of determining if a frame identified by its number has been defined is to test if the label attribute is set. Syntax $\langle frame-type \rangle idn$

```

\prg_new_conditional:Nnn \__flowfram_if_frame_idn_exists:nn
  { p , T , F, TF }
{
  \tl_if_exist:cTF
  { g__flowfram_#1_label_ \romannumeral #2_ tl }
  { \prg_return_true: }
  { \prg_return_false: }
}

```

$\backslash @getframeid \backslash @getframeid\langle type \rangle\langle idl \rangle$

Gets the IDL for the frame of type $\langle type \rangle$ whose IDL is given by $\langle idl \rangle$. The IDN is stored in $\backslash l_flowfram_id_int$. Version 2.0 renamed $\backslash @getframeid$.

```

\cs_new:Nn \__flowfram_get_frame_id:nn
{
  \__flowfram_if_frame_label_exists:nnNF
  { #1 } { #2 } \l_flowfram_id_int
  {
    \__flowfram_error:eee { label-undefined } { #2 } { #1 }
  }
}
\cs_generate_variant:Nn \__flowfram_get_frame_id:nn { ne, nV }

```

Frame attributes.

```

\prop_new:N \g__flowfram_flow_attributes_prop
\prop_new:N \g__flowfram_static_attributes_prop
\prop_new:N \g__flowfram_dynamic_attributes_prop

```

Show flow frame attributes:

```

\cs_new:Nn \flowfram_show_flow_frame_by_idl:n
{
  \__flowfram_if_flow_label_exists:nTF { #1 }
  {
    \__flowfram_show_flow_frame:
  }
  {
    \__flowfram_error:eee { label-undefined } { #1 } { flow }
  }
}
\cs_new:Nn \flowfram_show_flow_frame_by_idn:n
{

```

```

\int_set:Nn \l__flowfram_id_int { #1 }
\tl_if_exist:cTF
  { g__flowfram_flow_label_ \romannumeral \l__flowfram_id_int _ tl }
  {
    \__flowfram_show_flow_frame:
  }
  {
    \msg_error:nnee { flowfram } { idn-undefined }
    { flow } { \int_use:N \l__flowfram_id_int }
  }
}

Show all flow frames:
\cs_new:Nn \flowfram_show_all_flow_frames:
{
  \int_if_zero:nTF { \c@maxflow }
  {
    \msg_show:nnee { flowfram } { show-all-frames-none }
    { flow }
  }
  {
    \msg_show:nnee { flowfram } { show-all-frames }
    { flow }
    {
      \exp_args:NV \msg_show_item_unbraced:n
      \c@maxflow
    }
    \int_step_inline:nn { \c@maxflow }
    {
      \int_set:Nn \l__flowfram_id_int { ##1 }
      \__flowfram_show_flow_frame:
    }
  }
}

\cs_new:Nn \__flowfram_show_flow_frame:
{
  \msg_show:nneee { flowfram } { show-frame-attributes }
  { flow } { \int_use:N \l__flowfram_id_int }
  {
    \prop_map_function:NN
    \g__flowfram_flow_attributes_prop
    \__flowfram_show_flow_attribute:nn
  }
}

\cs_new:Nn \__flowfram_show_flow_attribute:nn
{
  \exp_args:Ne \msg_show_item_unbraced:n
  {
    \__flowfram_show_attribute:nnnn
    { flow } { #1 } { \l__flowfram_id_int } { #2 }
  }
}

```



```

    }
  }
  Show static frame attributes:
\cs_new:Nn \flowfram_show_static_frame_by_idl:n
{
  \__flowfram_if_static_label_exists:nTF { #1 }
  {
    \__flowfram_show_static_frame:
  }
  {
    \__flowfram_error:eee { label-undefined } { #1 } { static }
  }
}
\cs_new:Nn \flowfram_show_static_frame_by_idn:n
{
  \int_set:Nn \l__flowfram_id_int { #1 }
  \tl_if_exist:cTF
  { g__flowfram_static_label_ \romannumeral \l__flowfram_id_int _ tl }
  {
    \__flowfram_show_static_frame:
  }
  {
    \msg_error:nnee { flowfram } { idn-undefined }
    { static } { \int_use:N \l__flowfram_id_int }
  }
}
  Show all static frames:
\cs_new:Nn \flowfram_show_all_static_frames:
{
  \int_if_zero:nTF { \c@maxstatic }
  {
    \msg_show:nnee { flowfram } { show-all-frames-none }
    { static }
  }
  {
    \msg_show:nnee { flowfram } { show-all-frames }
    { static }
    {
      \exp_args:NV \msg_show_item_unbraced:n
      \c@maxstatic
    }
  }
  \int_step_inline:nn { \c@maxstatic }
  {
    \int_set:Nn \l__flowfram_id_int { ##1 }
    \__flowfram_show_static_frame:
  }
}
}

```

```

\cs_new:Nn \__flowfram_show_static_frame:
{
  \msg_show:nnee { flowfram } { show-frame-attributes }
  { static } { \int_use:N \l__flowfram_id_int }
  {
    \prop_map_function:NN
      \g__flowfram_static_attributes_prop
      \__flowfram_show_static_attribute:nn
  }
}

```

```

\cs_new:Nn \__flowfram_show_static_attribute:nn
{
  \exp_args:Ne \msg_show_item_unbraced:n
  {
    \__flowfram_show_attribute:nnnn
    { static } { #1 } { \l__flowfram_id_int } { #2 }
  }
}

```

Show dynamic frame attributes:

```

\cs_new:Nn \flowfram_show_dynamic_frame_by_idl:n
{
  \__flowfram_if_dynamic_label_exists:nTF { #1 }
  {
    \__flowfram_show_dynamic_frame:
  }
  {
    \__flowfram_error:eee { label-undefined } { #1 } { dynamic }
  }
}

```

```

\cs_new:Nn \flowfram_show_dynamic_frame_by_idn:n
{
  \int_set:Nn \l__flowfram_id_int { #1 }
  \tl_if_exist:cTF
  { g__flowfram_dynamic_label_ \romannumeral \l__flowfram_id_int _ tl }
  {
    \__flowfram_show_dynamic_frame:
  }
  {
    \msg_error:nnee { flowfram } { idn-undefined }
    { dynamic } { \int_use:N \l__flowfram_id_int }
  }
}

```

Show all dynamic frames:

```

\cs_new:Nn \flowfram_show_all_dynamic_frames:
{
  \int_if_zero:nTF { \c@maxdynamic }
  {
    \msg_show:nne { flowfram } { show-all-frames-none }
  }
}

```

```

    { dynamic }
  }
  {
    \msg_show:nne { flowfram } { show-all-frames }
    { dynamic }
    {
      \exp_args:NV \msg_show_item_unbraced:n
      \c@maxdynamic
    }
    \int_step_inline:nn { \c@maxdynamic }
    {
      \int_set:Nn \l__flowfram_id_int { ##1 }
      \__flowfram_show_dynamic_frame:
    }
  }
}

\cs_new:Nn \__flowfram_show_dynamic_frame:
{
  \msg_show:nneee { flowfram } { show-frame-attributes }
  { dynamic } { \int_use:N \l__flowfram_id_int }
  {
    \prop_map_function:NN
    \g__flowfram_dynamic_attributes_prop
    \__flowfram_show_dynamic_attribute:nn
  }
}

\cs_new:Nn \__flowfram_show_dynamic_attribute:nn
{
  \exp_args:Ne \msg_show_item_unbraced:n
  {
    \__flowfram_show_attribute:nnnn
    { dynamic } { #1 } { \l__flowfram_id_int } { #2 }
  }
}

\cs_new:Nn \__flowfram_show_bool_attribute:nnn
{
  \bool_if:cTF
  { g__flowfram_ #1 _ #2 _ \romannumerical #3 _ bool }
  { true }
  { false }
}

\cs_new:Nn \__flowfram_show_box_attribute:nnn
{
  \box_if_empty:cTF
  { g__flowfram_ #1 _ #2 _ \romannumerical #3 _ box }
  { empty }
  { not ~ empty }
}

\cs_new:Nn \__flowfram_show_attribute:nnnn

```

```

{
#2 ~ ( #4 ) : ~
\cs_if_exist:cTF { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ #4 }
{
\cs_if_exist:cTF { __flowfram_show_ #4 _attribute:nnn }
{
\use:c { __flowfram_show_ #4 _attribute:nnn }
{ #1 } { #2 } { #3 }
}
{
\use:c { #4 _use:c }
{ g__flowfram_ #1 _ #2 _ \romannumeral #3 _ #4 }
}
}
{ undefined }
}
⟨prop-var⟩{⟨tag⟩}{⟨var type⟩}
\cs_new:Nn \__flowfram_add_frame_attribute:Nnn
{
\prop_get:NnNTF #1 { #2 } \l__flowfram_tmpa_tl
{
\l_if_eq:NnF \l__flowfram_tmpa_tl { #3 }
{
\msg_error:nneee { flowfram } {attribute-type-change}
{ #2 } { \l__flowfram_tmpa_tl } { #3 }
}
}
{
\prop_gput:Nnn #1 { #2 } { #3 }
}
}
\cs_generate_variant:Nn \__flowfram_add_frame_attribute:Nnn { cen }
Syntax: {⟨frame type⟩}{⟨tag⟩}{⟨idn⟩}
\cs_new:Nn \flowfram_frame_new_dim:nnn
{
\prop_if_exist:cTF
{ g__flowfram_ #1 _attributes_prop }
{
\dim_new:c { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ dim }
\__flowfram_add_frame_attribute:cen
{ g__flowfram_ #1 _attributes_prop }
{ #2 } { dim }
}
{
\msg_error:nne { flowfram } {invalid-frame-type} { #1 }
}
}
}
Syntax: {⟨frame type⟩}{⟨tag⟩}{⟨idn⟩}{⟨value⟩}

```

```

\cs_new:Nn \flowfram_frame_new_dim:nmmm
{
  \prop_if_exist:cTF
  { g__flowfram_ #1 _attributes_prop }
  {
    \dim_new:c { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ dim }
    \dim_gset:cn
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ dim }
    { #4 }
    \__flowfram_add_frame_attribute:cen
    { g__flowfram_ #1 _attributes_prop }
    { #2 } { dim }
  }
  {
    \msg_error:nne { flowfram } {invalid-frame-type} { #1 }
  }
}

\cs_new:Nn \flowfram_frame_use_dim:nnn
{
  \dim_use:c { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ dim }
}

Syntax: {<frame type>}{<tag>}{<idn>}{<value>}

\cs_new:Nn \flowfram_frame_set_dim:nmmm
{
  \dim_if_exist:cTF
  { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ dim }
  {
    \dim_gset:cn
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ dim }
    { #4 }
  }
  {
    \__flowfram_error:eeeeee { frame-unknown-tag }
    { #1 } { #2 } { \int_eval:n { #3 } } { dim }
  }
}

Syntax: <dim>{<frame type>}{<tag>}{<idn>}

\cs_new:Nn \flowfram_set_dim_to_frame_dim:Nnnn
{
  \dim_if_exist:cTF
  { g__flowfram_ #2 _ #3 _ \romannumeral #4 _ dim }
  {
    \dim_set_eq:Nc
    #1
    { g__flowfram_ #2 _ #3 _ \romannumeral #4 _ dim }
  }
  {
    \__flowfram_error:eeeeee { frame-unknown-tag }
  }
}

```

```

        { #2 } { #3 } { \int_eval:n { #4 } } { dim }
    }
}
Syntax: <dim>{<frame type>}{<tag>}{<idn>}
\cs_new:Nn \flowfram_gset_dim_to_frame_dim:Nnnn
{
  \dim_if_exist:cTF
    { g__flowfram_ #2 _ #3 _ \romannumeral #4 _ dim }
  {
    \dim_gset_eq:Nc
      #1
      { g__flowfram_ #2 _ #3 _ \romannumeral #4 _ dim }
    }
  {
    \__flowfram_error:eeeeee { frame-unknown-tag }
    { #2 } { #3 } { \int_eval:n { #4 } } { dim }
  }
}
Syntax: <dim>{<frame type>}{<tag>}{<idn>}
\cs_new:Nn \flowfram_set_dim_to_frame_tl:Nnnn
{
  \dim_if_exist:cTF
    { g__flowfram_ #2 _ #3 _ \romannumeral #4 _ tl }
  {
    \dim_set:Nn
      #1
      { \tl_use:c { g__flowfram_ #2 _ #3 _ \romannumeral #4 _ tl } }
    }
  {
    \__flowfram_error:eeeeee { frame-unknown-tag }
    { #2 } { #3 } { \int_eval:n { #4 } } { tl }
  }
}
Syntax: {<frame type>}{<tag1>}{<tag2>}{<idn>}
\cs_new:Nn \flowfram_frame_swap_dim:nnnn
{
  \dim_set_eq:Nc \l__flowfram_tmpa_dim
    { g__flowfram_ #1 _ #2 _ \romannumeral #4 _ dim }
  \dim_gset_eq:cc
    { g__flowfram_ #1 _ #2 _ \romannumeral #4 _ dim }
    { g__flowfram_ #1 _ #3 _ \romannumeral #4 _ dim }
  \dim_gset_eq:cN
    { g__flowfram_ #1 _ #3 _ \romannumeral #4 _ dim }
    \l__flowfram_tmpa_dim
}
Syntax: {<frame type>}{<tag>}{<idn>}
\cs_new:Nn \flowfram_frame_new_box:nnn

```

```

{
  \prop_if_exist:cTF
  { g__flowfram_ #1 _attributes_prop }
  {
    \box_new:c { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ box }
    \__flowfram_add_frame_attribute:cen
    { g__flowfram_ #1 _attributes_prop }
    { #2 } { box }
  }
  {
    \msg_error:nne { flowfram } {invalid-frame-type} { #1 }
  }
}

\cs_new:Nn \flowfram_frame_set_box_eq:nnnN
{
  \tl_if_exist:cTF
  { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ box }
  {
    \box_gset_eq:cN
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ box }
    #4
  }
  {
    \__flowfram_error:eeeeee { frame-unknown-tag }
    { #1 } { #2 } { \int_eval:n { #3 } } { box }
  }
}

\cs_new:Nn \flowfram_frame_clear_box:nnn
{
  \tl_if_exist:cTF
  { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ box }
  {
    \box_gclear:c { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ box }
  }
  {
    \__flowfram_error:eeeeee { frame-unknown-tag }
    { #1 } { #2 } { \int_eval:n { #3 } } { box }
  }
}

\cs_new:Nn \flowfram_frame_use_box:nnn
{
  \tl_if_exist:cTF
  { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ box }
  {
    \leavevmode
    \box_use:c { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ box }
  }
  {
    \__flowfram_error:eeeeee { frame-unknown-tag }
  }
}

```

```

        { #1 } { #2 } { \int_eval:n { #3 } } { box }
    }
}
\cs_new:Nn \flowfram_frame_use_drop_box:nnn
{
  \tl_if_exist:cTF
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ box }
    {
      \box_use_drop:c { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ box }
    }
    {
      \__flowfram_error:eeeeee { frame-unknown-tag }
      { #1 } { #2 } { \int_eval:n { #3 } } { box }
    }
}
\prg_new_conditional:Nnn \flowfram_frame_if_box_empty:nnn
{ T, F, TF }
{
  \exp_args:Nc \if_box_empty:N
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ box }
  \prg_return_true:
  \else:
  \prg_return_false:
  \fi:
}
Syntax: {<frame type>}{<tag>}{<idn>}
\cs_new:Nn \flowfram_frame_new_tl:nnn
{
  \prop_if_exist:cTF
    { g__flowfram_ #1 _attributes_prop }
    {
      \tl_new:c { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ tl }
      \__flowfram_add_frame_attribute:cen
        { g__flowfram_ #1 _attributes_prop }
        { #2 } { tl }
    }
    {
      \msg_error:nne { flowfram } {invalid-frame-type} { #1 }
    }
}
Syntax: {<frame type>}{<tag>}{<idn>}{<value>}
\cs_new:Nn \flowfram_frame_new_tl:nnnn
{
  \prop_if_exist:cTF
    { g__flowfram_ #1 _attributes_prop }
    {
      \tl_new:c { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ tl }
      \tl_gset:cn

```



```

    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ t1 }
    { #4 }
    \__flowfram_add_frame_attribute:cen
    { g__flowfram_ #1 _attributes_prop }
    { #2 } { t1 }
  }
  {
    \msg_error:nne { flowfram } {invalid-frame-type} { #1 }
  }
}
\cs_generate_variant:Nn \flowfram_frame_new_tl:nnnn
{ nnnn , nnnV , nnne }
\cs_new:Nn \flowfram_frame_use_tl:nnn
{
  \tl_use:c { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ t1 }
}
Syntax: {<frame type>}{<tag>}{<idn>}{<value>}
\cs_new:Nn \flowfram_frame_set_tl:nnnn
{
  \tl_if_exist:cTF
  { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ t1 }
  {
    \__flowfram_message:nnen
    { info-setting-frame-attribute }
    { #2 } { #1 } { \int_eval:n { #3 } } { #4 }
    \tl_gset:cn
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ t1 }
    { #4 }
  }
  {
    \__flowfram_error:eeee{ frame-unknown-tag }
    { #1 } { #2 } { \int_eval:n { #3 } } { t1 }
  }
}
\cs_generate_variant:Nn \flowfram_frame_set_tl:nnnn { nnne , nnnV }
Syntax: {<frame type>}{<tag>}{<idn>}
\cs_new:Nn \flowfram_frame_clear_tl:nnn
{
  \tl_if_exist:cTF
  { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ t1 }
  {
    \__flowfram_message:nnen
    { info-setting-frame-attribute }
    { #2 } { #1 } { \int_eval:n { #3 } } { }
    \tl_gclear:c
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ t1 }
  }
  {

```

```

    \__flowfram_error:eeeeee { frame-unknown-tag }
      { #1 } { #2 } { \int_eval:n { #3 } } { t1 }
  }
}
Syntax: {<frame type>}{<tag>}{<idn>}{<value>}
\cs_new:Nn \flowfram_frame_put_right_tl:nnnn
{
  \tl_if_exist:cTF
    { g__flowfram_ #1 _ #2 _ \romannumerals #3 _ t1 }
  {
    \tl_gput_right:cn
      { g__flowfram_ #1 _ #2 _ \romannumerals #3 _ t1 }
      { #4 }
    \__flowfram_message:nnnev
      { info-setting-frame-attribute }
      { #2 } { #1 } { \int_eval:n { #3 } }
      { g__flowfram_ #1 _ #2 _ \romannumerals #3 _ t1 }
  }
  {
    \__flowfram_error:eeeeee { frame-unknown-tag }
      { #1 } { #2 } { \int_eval:n { #3 } } { t1 }
  }
}
\cs_generate_variant:Nn \flowfram_frame_put_right_tl:nnnn { nnne , nnnV }
Syntax: <tl>{<frame type>}{<tag>}{<idn>}
\cs_new:Nn \flowfram_set_tl_to_frame_tl:Nnnn
{
  \tl_if_exist:cTF
    { g__flowfram_ #2 _ #3 _ \romannumerals #4 _ t1 }
  {
    \tl_set_eq:Nc
      #1
      { g__flowfram_ #2 _ #3 _ \romannumerals #4 _ t1 }
  }
  {
    \__flowfram_error:eeeeee { frame-unknown-tag }
      { #2 } { #3 } { \int_eval:n { #4 } } { t1 }
  }
}
Syntax: {<frame type>}{<tag>}{<idn>}{<tl>}
\cs_new:Nn \flowfram_set_frame_tl_to_tl:Nnnn
{
  \tl_if_exist:cTF
    { g__flowfram_ #1 _ #2 _ \romannumerals #3 _ t1 }
  {
    \__flowfram_message:nnneV
      { info-setting-frame-attribute }
      { #2 } { #1 } { \int_eval:n { #3 } } #4
  }
}

```

```

        \tl_gset_eq:cN
        { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ t1 }
        #4
    }
    {
        \__flowfram_error:eeeeee { frame-unknown-tag }
        { #2 } { #3 } { \int_eval:n { #4 } } { t1 }
    }
}

\prg_new_conditional:Nnn \flowfram_if_tl_eq_frame_tl:Nnnn
{ T, F, TF }
{
    \tl_if_eq:NcTF
    #1
    { g__flowfram_ #2 _ #3 _ \romannumeral #4 _ t1 }
    { \prg_return_true: }
    { \prg_return_false: }
}

\prg_new_conditional:Nnn \flowfram_if_tl_eq_frame_tl:nnnn
{ T, F, TF }
{
    \tl_if_eq:cnTF
    { g__flowfram_ #2 _ #3 _ \romannumeral #4 _ t1 }
    { #1 }
    { \prg_return_true: }
    { \prg_return_false: }
}

\prg_new_conditional:Nnn \flowfram_if_frame_tl_exist:nnn
{ T, F, TF }
{
    \tl_if_exist:cTF
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ t1 }
    { \prg_return_true: }
    { \prg_return_false: }
}

\prg_new_conditional:Nnn \flowfram_if_frame_tl_empty:nnn
{ T, F, TF }
{
    \tl_if_empty:cTF
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ t1 }
    { \prg_return_true: }
    { \prg_return_false: }
}

Syntax: {<frame type>}{<tag>}{<idn>}{<value>}

\cs_new:Nn \flowfram_frame_new_clist:nnnn
{
    \prop_if_exist:cTF

```

```

{ g__flowfram_ #1 _attributes_prop }
{
  \clist_new:c { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ clist }
  \clist_gset:cn
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ clist }
    { #4 }
  \__flowfram_add_frame_attribute:cen
    { g__flowfram_ #1 _attributes_prop }
    { #2 } { clist }
}
{
  \msg_error:nne { flowfram } {invalid-frame-type} { #1 }
}
}
\cs_generate_variant:Nn \flowfram_frame_new_clist:nnnn
{ nnne , nnnV }
\cs_new:Nn \flowfram_frame_use_clist:nnn
{
  \clist_use:c { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ clist }
}
Syntax: {<frame type>}{<tag>}{<idn>}{<value>}
\cs_new:Nn \flowfram_frame_set_clist:nnnn
{
  \clist_if_exist:cTF
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ clist }
    {
      \__flowfram_message:nnnen
        { info-setting-frame-attribute }
        { #2 } { #1 } { \int_eval:n { #3 } } { #4 }
      \clist_gset:cn
        { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ clist }
        { #4 }
    }
    {
      \__flowfram_error:eeee { frame-unknown-tag }
      { #1 } { #2 } { \int_eval:n { #3 } } { clist }
    }
}
\cs_generate_variant:Nn \flowfram_frame_set_clist:nnnn { nnne , nnnV }
Syntax: {<frame type>}{<tag>}{<idn>}{<value>}
\cs_new:Nn \flowfram_frame_put_right_clist:nnnn
{
  \clist_if_exist:cTF
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ clist }
    {
      \clist_gput_right:cn
        { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ clist }
        { #4 }
    }
}

```

```

    \__flowfram_message:nnnev
    { info-setting-frame-attribute }
    { #2 } { #1 } { \int_eval:n { #3 } }
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ clist }
  }
  {
    \__flowfram_error:eeeeee { frame-unknown-tag }
    { #1 } { #2 } { \int_eval:n { #3 } } { clist }
  }
}
\cs_generate_variant:Nn \flowfram_frame_put_right_clist:nnnn { nnne , nnnV }
Syntax: {<frame type>}{<tag>}{<idn>}{<clist var>}
\cs_new:Nn \flowfram_frame_concat_clist:nnnN
{
  \clist_if_exist:cTF
  { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ clist }
  {
    \exp_args:Ncc
    \clist_gconcat:NNN
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ clist }
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ clist }
    #4
    \__flowfram_message:nnnev
    { info-setting-frame-attribute }
    { #2 } { #1 } { \int_eval:n { #3 } }
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ clist }
  }
  {
    \__flowfram_error:eeeeee { frame-unknown-tag }
    { #1 } { #2 } { \int_eval:n { #3 } } { clist }
  }
}
\cs_generate_variant:Nn \flowfram_frame_concat_clist:nnnN { nnnnc }
Syntax: <clist>{<frame type>}{<tag>}{<idn>}
\cs_new:Nn \flowfram_set_clist_to_frame_clist:Nnnn
{
  \clist_if_exist:cTF
  { g__flowfram_ #2 _ #3 _ \romannumeral #4 _ clist }
  {
    \clist_set_eq:Nc
    #1
    { g__flowfram_ #2 _ #3 _ \romannumeral #4 _ clist }
  }
  {
    \__flowfram_error:eeeeee { frame-unknown-tag }
    { #2 } { #3 } { \int_eval:n { #4 } } { clist }
  }
}
}

```

```

\prg_new_conditional:Nnn \flowfram_if_tl_eq_frame_clist:Nnnn
  { T, F, TF }
  {
    \tl_if_eq:cNTF
      { g__flowfram_ #2 _ #3 _ \romannumeral #4 _ clist }
      #1
    { \prg_return_true: }
    { \prg_return_false: }
  }

```

Syntax: $\{\langle frame\ type\rangle\}\{\langle tag\rangle\}\{\langle idn\rangle\}\{\langle code\rangle\}$

```

\cs_new:Nn \flowfram_frame_clist_map_inline:nnnn
  {
    \clist_if_exist:cTF
      { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ clist }
      {
        \clist_map_inline:cn
          { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ clist }
          { #4 }
        }
      {
        \__flowfram_error:eeeeee { frame-unknown-tag }
        { #1 } { #2 } { \int_eval:n { #3 } } { clist }
      }
  }

```

```

\prg_new_conditional:Nnn \flowfram_if_in_frame_clist:nnnn
  { T, F, TF }
  {
    \clist_if_in:cnTF
      { g__flowfram_ #2 _ #3 _ \romannumeral #4 _ clist }
      { #1 }
    { \prg_return_true: }
    { \prg_return_false: }
  }

```

$\{\langle page\ num\rangle\}\{\langle frame\ type\rangle\}\{\langle idn\rangle\}$

```

\prg_new_conditional:Nnn \flowfram_if_frame_excludedpage:nnn
  { T, F, TF}
  {
    \flowfram_if_in_frame_clist:nnnnTF
      { #1 } { #2 } { excludelist } { #3 }
    { \prg_return_true: }
    { \prg_return_false: }
  }

```

Syntax: $\{\langle page\ num\rangle\}\{\langle frame\ type\rangle\}\{\langle idn\rangle\}$ Check if given page is in the page list for a frame.

```

\prg_new_conditional:Nnn \flowfram_if_in_frame_pagelist:nnn
  { T, F, TF}
  {

```


$\langle int-var \rangle \{ \langle type \rangle \} \{ \langle idn \rangle \}$ Gets the starting page and stores in $\langle int-var \rangle$.

```
\cs_new:Nn \flowfram_frame_get_start_page:Nnn
{
  \flowfram_if_frame_allpages:nnTF { #2 } { #3 }
  {
    \int_set_eq:NN #1 \c_one_int
  }
  {
    \flowfram_if_frame_nopages:nnTF { #2 } { #3 }
    {
      \int_set:Nn #1
        { \c_flowfram_max_page_int + \c_one_int }
    }
    {
      \flowfram_if_frame_oddpages_only:nnTF { #2 } { #3 }
      {
        \int_set_eq:NN #1 \c_one_int
      }
      {
        \flowfram_if_frame_evenpages_only:nnTF { #2 } { #3 }
        {
          \int_set:Nn #1 { 2 }
        }
      }
    }
  }
}
```

Get the first range in the page list.

```
\exp_args:Ne
\__flowfram_get_range:n
{
  \clist_item:cn
  { g__flowfram_ #2 _ pagelist _ \romannumerical #3 _ clist }
  { \c_one_int }
}
\int_if_zero:nT { #1 }
{
  \int_set_eq:NN #1 \c_one_int
}
}
}
}
}
```

Gets the starting page (stored in $\backslash l_flowfram_range_start_int$).
 $\{ \langle type \rangle \} \{ \langle idn \rangle \}$

```
\cs_new:Nn \__flowfram_frame_get_start_page:nn
{
  \flowfram_frame_get_start_page:Nnn
  \l_flowfram_range_start_int { #1 } { #2 }
}
```

```

\cs_new:Nn \flowfram_set_frame_pagelist:nnn
{
  \flowfram_if_valid_pagelist:nTF { #3 }
  {
    \flowfram_frame_set_clist:nnnn
    { #1 } { pagelist } { #2 } { #3 }
  }
  {
    \msg_error:nnn { flowfram } { invalid-pagelist } { #3 }
  }
}
\cs_generate_variant:Nn \flowfram_set_frame_pagelist:nnn
{ nne , nnV }

Check if value is a valid page list.
\prg_new_conditional:Nnn \flowfram_if_valid_pagelist:n
{ T, F, TF }
{
  \tl_if_blank:nTF { #1 }
  { \prg_return_false: }
  {
    \flowfram_if_all_or_odd:nTF { #1 }
    { \prg_return_true: }
    {
      \flowfram_if_none_or_even:nTF { #1 }
      { \prg_return_true: }
      {
        \bool_set_false:N \l__flowfram_found_bool
        \int_set_eq:NN \l__flowfram_tmpa_int \c_zero_int
        \clist_map_inline:nn { #1 }
        {
          \__flowfram_get_range:n { ##1 }
          \int_if_zero:nT { \l__flowfram_range_start_int }
          {
            \int_set_eq:NN \l__flowfram_range_start_int \c_one_int
          }
          \bool_lazy_or:nnT
          {
            \int_compare_p:nNn
            { \l__flowfram_range_start_int } > { \l__flowfram_range_end_int }
          }
          {
            \bool_not_p:n
            {
              \int_compare_p:nNn
              { \l__flowfram_range_start_int } > { \l__flowfram_tmpa_int }
            }
          }
        }
      }
      {
        \clist_map_break:n
      }
    }
  }
}

```



```

\flowfram_if_tl_eq_frame_clist:NnnnTF
  \c_flowfram_even_tl
  { #1 } { pagelist } { #2 }
  { \prg_return_true: }
  { \prg_return_false: }
}
If frame's page list is set to all:
\prg_new_conditional:Nnn \flowfram_if_frame_allpages:nn
{ p , T , F , TF }
{
  \flowfram_if_tl_eq_frame_clist:NnnnTF
  \c_flowfram_all_tl
  { #1 } { pagelist } { #2 }
  { \prg_return_true: }
  { \prg_return_false: }
}
If frame's page list is set to none:
\prg_new_conditional:Nnn \flowfram_if_frame_nopages:nn
{ p , T , F , TF }
{
  \flowfram_if_tl_eq_frame_clist:NnnnTF
  \c_flowfram_none_tl
  { #1 } { pagelist } { #2 }
  { \prg_return_true: }
  { \prg_return_false: }
}
\prg_new_conditional:Nnn \flowfram_if_all_or_odd:N
{ p, T, F, TF }
{
  \bool_lazy_or:nnTF
  {
    \tl_if_eq_p:NN #1 \c_flowfram_all_tl
  }
  {
    \tl_if_eq_p:NN #1 \c_flowfram_odd_tl
  }
  { \prg_return_true: }
  { \prg_return_false: }
}
\prg_new_conditional:Nnn \flowfram_if_all_or_odd:n
{ T, F, TF }
{
  \tl_if_eq:NnTF \c_flowfram_all_tl { #1 }
  { \prg_return_true: }
  {
    \tl_if_eq:NnTF \c_flowfram_odd_tl { #1 }
    { \prg_return_true: }
    { \prg_return_false: }
  }
}

```

```

    }
  }
\prg_new_conditional:Nnn \flowfram_if_none_or_even:N
{ p, T, F, TF }
{
  \bool_lazy_or:nnTF
  {
    \tl_if_eq_p:NN #1 \c_flowfram_none_tl
  }
  {
    \tl_if_eq_p:NN #1 \c_flowfram_even_tl
  }
  { \prg_return_true: }
  { \prg_return_false: }
}
\prg_new_conditional:Nnn \flowfram_if_none_or_even:n
{ T, F, TF }
{
  \tl_if_eq:NnTF \c_flowfram_none_tl { #1 }
  { \prg_return_true: }
  {
    \tl_if_eq:NnTF \c_flowfram_even_tl { #1 }
    { \prg_return_true: }
    { \prg_return_false: }
  }
}
}

```

`\@ff@getrange` Find out what kind of range the argument is (for example, a single number 24 or a closed range 30-40 or an open range, <10 or >25) and set integer variables to start and end. Version 2.0 renamed command.

```

\cs_new:Nn \__flowfram_get_range:n
{
  \flowfram_if_all_or_odd:nTF { #1 }
  {
    \int_set_eq:NN \l__flowfram_range_start_int \c_one_int
    \int_set_eq:NN
      \l__flowfram_range_end_int
      \c_flowfram_max_page_int
  }
  {
    \tl_if_eq:nnTF { #1 } { even }
    {
      \int_set:Nn \l__flowfram_range_start_int { 2 }
      \int_set_eq:NN
        \l__flowfram_range_end_int
        \c_flowfram_max_page_int
    }
    {
      \tl_if_eq:nnTF { #1 } { none }
    }
  }
}

```

```

    {
      \int_zero:N \l__flowfram_range_start_int
      \int_zero:N \l__flowfram_range_end_int
    }
    {
      \__flowfram_get_range:w #1 - \q_nil \q_stop
    }
  }
}
\cs_generate_variant:Nn \__flowfram_get_range:n { e }

```

\@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.

```

\cs_new:Npn \__flowfram_get_range:w #1 - #2 \q_stop
{
  \quark_if_nil:nTF { #2 }
  {
    \tl_if_head_eq_meaning:nNTF { #1 } <
    {
      \int_zero:N \l__flowfram_range_start_int
      \int_set:Nn \l__flowfram_range_end_int
      {
        \tl_tail:n { #1 } - \c_one_int
      }
    }
    {
      \tl_if_head_eq_meaning:nNTF { #1 } >
      {
        \int_set:Nn \l__flowfram_range_start_int
        {
          \tl_tail:n { #1 } + \c_one_int
        }
        \int_set_eq:NN
        \l__flowfram_range_end_int
        \c_flowfram_max_page_int
      }
      {
        \int_set:Nn \l__flowfram_range_start_int { #1 }
        \int_set:Nn \l__flowfram_range_end_int { #1 }
      }
    }
  }
}
{
  \int_set:Nn \l__flowfram_range_start_int { #1 }
  \__flowfram_get_end_range:w #2 - \q_stop
}
}

```

Discard trailing content.

```

\cs_new:Npn \__flowfram_get_end_range:w #1 - #2 \q_stop
{
  \int_set:Nn \l__flowfram_range_end_int { #1 }
}
Syntax: {<frame type>}{<tag>}{<idn>}
\cs_new:Nn \flowfram_frame_new_bool:nnn
{
  \prop_if_exist:cTF
  { g__flowfram_ #1 _attributes_prop }
  {
    \bool_new:c { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ bool }
    \__flowfram_add_frame_attribute:cen
    { g__flowfram_ #1 _attributes_prop }
    { #2 } { bool }
  }
  {
    \msg_error:nne { flowfram } {invalid-frame-type} { #1 }
  }
}
Syntax: {<frame type>}{<tag>}{<idn>}{<value>}
\cs_new:Nn \flowfram_frame_new_bool:nnnN
{
  \prop_if_exist:cTF
  { g__flowfram_ #1 _attributes_prop }
  {
    \bool_new:c { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ bool }
    \bool_gset_eq:cN
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ bool }
    #4
    \__flowfram_add_frame_attribute:cen
    { g__flowfram_ #1 _attributes_prop }
    { #2 } { bool }
  }
  {
    \msg_error:nne { flowfram } {invalid-frame-type} { #1 }
  }
}
\prg_new_conditional:Nnn \flowfram_if_frame_bool:nnn
{ T, F, TF }
{
  \bool_if_exist:cTF
  { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ bool }
  {
    \bool_if:cTF
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ bool }
    { \prg_return_true: }
    { \prg_return_false: }
  }
}

```

```

    {
      \__flowfram_error:eeeeee { frame-unknown-tag }
      { #1 } { #2 } { \int_eval:n { #3 } } { bool }
      \prg_return_false:
    }
  }

```

Syntax: $\langle\text{frame type}\rangle\langle\text{tag}\rangle\langle\text{idn}\rangle$

`\cs_new:Nn \flowfram_frame_set_bool_true:nnn`

```

{
  \bool_if_exist:cTF
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ bool }
  {
    \bool_gset_true:c { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ bool }
  }
  {
    \__flowfram_error:eeeeee { frame-unknown-tag }
    { #1 } { #2 } { \int_eval:n { #3 } } { bool }
  }
}

```

Syntax: $\langle\text{frame type}\rangle\langle\text{tag}\rangle\langle\text{idn}\rangle$

`\cs_new:Nn \flowfram_frame_set_bool_false:nnn`

```

{
  \bool_if_exist:cTF
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ bool }
  {
    \bool_gset_false:c { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ bool }
  }
  {
    \__flowfram_error:eeeeee { frame-unknown-tag }
    { #1 } { #2 } { \int_eval:n { #3 } } { bool }
  }
}

```

Syntax: $\langle\text{frame type}\rangle\langle\text{tag}\rangle\langle\text{idn}\rangle\langle\text{choice value}\rangle$ The $\langle\text{choice value}\rangle$ should be either true or false.

`\cs_new:Nn \flowfram_frame_set_bool_from_option:nnnn`

```

{
  \bool_if_exist:cTF
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ bool }
  {
    \tl_if_eq:nnTF { #4 } { true }
    {
      \bool_gset_true:c { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ bool }
    }
  }
  {
    \tl_if_eq:nnTF { #4 } { false }
    {
      \bool_gset_false:c { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ bool }
    }
  }
}

```



```

        {
          \msg_error:nnnnnn { flowfram } { invalid-bool }
            { #4 } { #2 } { #1 } { #3 }
        }
    }
}
{
  \__flowfram_error:eeeeee { frame-unknown-tag }
    { #1 } { #2 } { \int_eval:n { #3 } } { bool }
}
}
\cs_generate_variant:Nn \flowfram_frame_set_bool_from_option:nnnn
{ nnnV , nnne }

```

Frames may only be defined in the preamble although the output routine will create one if there are none available.

```

\cs_new:Nn \__flowfram_only_preamble:Nn { #2 }
\cs_new:Nn \__flowfram_forbidden:Nn
{
  \msg_error:nnn { flowfram } { misplaced-col-cmd } { #1 }
}

```

Document only commands:

```

\cs_new:Nn \__flowfram_only_document_env:Nn
{
  \msg_error:nnn { flowfram } { doc-env-only } { #1 }
}

```

1.3 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.

```

\NewDocumentCommand \flowframeshowlayout { }
{
  \finishthispage
  \group_begin:
    \__flowfram_set_draft:
    \mbox{}
    \finishthispage
    \clearpage
  \group_end:
}

```

```
}
```

`\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
\usedframebreaktrue
\NewDocumentCommand \framebreak { 0{4} }
{
  \global \usedframebreaktrue
  \group_begin:
    \dim_zero:N \parfillskip
    \pagebreak [ #1 ]
    \dim_zero:N \parskip
    \par
    \noindent
  \group_end:
}
```

`\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.)

```
\NewDocumentCommand \finishthispage { }
{
  \ifvmode
    \int_set_eq:NN \l__flowfram_current_frame_int \c@thisframe
    \int_set_eq:NN \l__flowfram_current_abspage_int \c@absolutepage
    \dim_compare:nNnT { \pagetotal } < { \topskip }
    {
      \hbox { }
    }
    \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.

```
\int_compare:nNnT
{ \l__flowfram_current_abspage_int } = { \c@absolutepage }
{
  \bool_while_do:nn
  {
    \bool_not_p:n
    {
      \int_compare_p:nNn
```

```

        { \l__flowfram_current_frame_int } > { \c@maxflow }
      }
    }
  {
    \__flowfram_check_if_this_page:nn
      { \g__flowfram_pagecounter_tl } { \l__flowfram_current_frame_int }
    \bool_if:NF \g__flowfram_not_this_frame_bool
      {
        \int_gset:Nn \c@thisframe { \l__flowfram_current_frame_int }
        \hbox { }
        \newpage
      }
    \int_incr:N \l__flowfram_current_frame_int
  }
}
\fi
}

```

`\cleardoublepage` Modify the definition of `\cleardoublepage`. This may or may not be defined.

```

\providecommand\cleardoublepage{}
\RenewDocumentCommand \cleardoublepage { }
{
  \clearpage
  \if@twoside
    \ifodd \c@page
      \else
        \hbox { }
        \clearpage
      \fi
    \fi
}

```

`\cleartoevenpage`

```

\NewDocumentCommand \cleartoevenpage { }
{
  \clearpage
  \if@twoside
    \ifodd \c@page
      \hbox { }
      \clearpage
    \fi
  \fi
}

```

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

```

\preto\newpage{\global\usedframebreaktrue}

```

Need to prevent `\@topnewpage` from pushing things out of whack as `\twocolumn` now behaves differently.

```

\long\def\@topnewpage[#1]{#1}

```

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.

```
\NewDocumentCommand \globalreversemargin { }
{
  \global \mparbottom \z@
  \global \@reversemargintrue
}
```

`\globalnormalmargin`

```
\NewDocumentCommand \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` (to the right or left) 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.) The setting is stored in `\l_flowfram_margin_tl`. Version 2.0 renamed `\@getmarginpos`.

```
\cs_new:Nn \_flowfram_get_margin_pos:n
{
  \tl_if_eq:NnTF \c_flowfram_inner_tl { #1 }
  {
    \legacy_if:nTF { @twoside }
    {
      \int_if_odd:nTF { \c@page }
      {
        \tl_set_eq:NN
        \l_flowfram_margin_tl
        \c_flowfram_left_tl
      }
      {
        \tl_set_eq:NN
        \l_flowfram_margin_tl
        \c_flowfram_right_tl
      }
    }
  }
  {
    \tl_set_eq:NN
    \l_flowfram_margin_tl
    \c_flowfram_left_tl
  }
}
```

```

\tl_if_eq:NnTF \c_flowfram_outer_tl { #1 }
{
  \legacy_if:nTF { @twoside }
  {
    \int_if_odd:nTF { \c@page }
    {
      \tl_set_eq:NN
      \l__flowfram_margin_tl
      \c_flowfram_right_tl
    }
    {
      \tl_set_eq:NN
      \l__flowfram_margin_tl
      \c_flowfram_left_tl
    }
  }
  {
    \tl_set_eq:NN
    \l__flowfram_margin_tl
    \c_flowfram_right_tl
  }
}
{
  \tl_set:Nn \l__flowfram_margin_tl { #1 }
}
}
\cs_generate_variant:Nn \__flowfram_get_margin_pos:n { e }

```

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

```

\NewDocumentCommand \setmargin { }
{
  \__flowfram_get_margin_pos:e
  {
    \flowfram_frame_use_tl:nnn { flow } { margin } { \c@thisframe }
  }
  \tl_if_eq:NNTF \l__flowfram_margin_tl \c_flowfram_left_tl
  {
    \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.

```
\NewDocumentCommand \newflowframe
  { s O{all} m m m m O { \int_use:N \c@maxflow } }
{
  \__flowfram_only_preamble:Nn \newflowframe
  {
    \__flowfram_new_flow:nnnnnn
    { #2 } { #3 } { #4 } { #5 } { #6 } { #7 }
  }
}
```

If the starred version was used, enable the border frame setting.

```
\IfBooleanT { #1 }
{
  \flowfram_frame_set_bool_true:nnnn { flow } { hasframe } { \c@maxflow }
}
}
```

`\@newflowframe` Version 2.0 removed `\@newflowframe`

`\@snewflowframe` Starred version sets boolean flag to indicate a border Version 2.0 removed `\@snewflowframe`.

`\@newflowframe` Version 2.0 removed `\@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. Version 2.0 removed

```
\@@newflowframe
  {<page list>}{<width>}{<height>}{<x>}{<y>}
\cs_new_nopar:Nn \__flowfram_new_flow:nnnnn
{
  \__flowfram_new_flow:nnnnnn
  { #1 } { #2 } { #3 } { #4 } { #5 } { \int_use:N \c@maxflow }
}
  {<page list>}{<width>}{<height>}{<x>}{<y>}{<label>}
\cs_new:Nn \__flowfram_new_flow:nnnnnn
{
  \__flowfram_new_flow:nnnnnn
  { #1 } { #2 } { #3 } { #4 } { #5 } { #6 } { flow }
}
```

Define content box.

```
\flowfram_frame_new_box:nnn { flow } { column } { \c@maxflow }
```

Define attributes specific to flow frames.

```
\flowfram_frame_new_tl:nnnn { flow } { margin } { \c@maxflow } { right }
```

Initialise if first (frame may have been created on the fly by the output routine).

```

\int_if_zero:nT { \c@thisframe }
{
  \__flowfram_frame_get_start_page:nn { flow } { \c@maxflow }
  \int_compare:nNnT
    { \l__flowfram_range_start_int } = { \c_one_int }
  {
    \int_gset_eq:NN \c@thisframe \c@maxflow
    \__flowfram_set_column:n { \c@maxflow }
    \global\usedframebreaktrue
  }
}
}

```

`\@s@tflowframeid` If square brackets occur after `\newflowframe`, take the contents to be the label, otherwise the label will be the flow frame number. Version 2.0 removed `\@s@tflowframeid`.

`\@ff@checkuniqueidl` `{\flow-id}{\label}` Check proposed label for flow frame is unique. Version 2.0 removed `\@ff@checkuniqueidl`.

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

```

\newcommand*\getflowlabel[1]{%
  \flowfram_frame_use_tl:nnn { flow } { label } { #1 }
}

```

`\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.

```

\NewDocumentCommand \getflowid { m m }
{
  \__flowfram_get_flow_id:e { #2 }
  \tl_set:Ne #1 { \int_use:N \l__flowfram_id_int }
}

```

`\@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 identify them by their IDN rather than their IDL. The IDN will be stored in `\l__flowfram_id_int`. Version 2.0 renamed `\@flowframeid`.

```

\cs_new:Nn \__flowfram_get_flow_id:n
{
  \__flowfram_if_flow_label_exists:nF { #1 }
  {
    \__flowfram_error:eee { label-undefined } { #1 } { flow }
  }
}
\cs_generate_variant:Nn \__flowfram_get_flow_id:n { e, V }

\prg_new_conditional:Nnn \__flowfram_if_flow_label_exists:n
{ TF , T , F }

```

```

{
  \__flowfram_if_frame_label_exists:nnNTF
    { flow } { #1 } \l__flowfram_id_int
  { \prg_return_true: }
  { \prg_return_false: }
}

```

`\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.

```

\NewDocumentCommand \setallflowframes { m }
{
  \int_step_inline:nn { \c@maxflow }
  {
    \__flowfram_set_flow_by_idn:nn { ##1 } { #1 }
  }
}

```

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

```

\NewDocumentCommand \setflowframe { s m m }
{
  \IfBooleanTF { #1 }
  {
    \exp_args:Ne \clist_map_inline:nn { #2 }
    {
      \__flowfram_if_flow_label_exists:nTF { ##1 }
      {
        \__flowfram_set_flow_by_idn:nn { \l__flowfram_id_int } { #3 }
      }
      {
        \__flowfram_error:eee { label-undefined } { ##1 } { flow }
      }
    }
  }
  {
    \__flowfram_map_idns:enn { #2 } { flow }
    {
      \__flowfram_set_flow_by_idn:nn { \l__flowfram_id_int } { #3 }
    }
  }
}

```

`\@ssetflowframe` Version 2.0 removed `\@ssetflowframe`.

`\@setflowframe` Version 2.0 removed `\@setflowframe`.

`\@@setflowframe` This is the command that actually sets the values for the flow frame whose IDN is specified by the first parameter. Version 2.0 renamed from `\@@setflowframe`.

```

\cs_new:Nn \__flowfram_set_flow_by_idn:nn

```



```
{
  \_flowfram_set_keys_for_type:nnn { #2 } { flow } { #1 }
```

Option specific to flow frames:

```
\tl_if_empty:NF \l__flowfram_margin_tl
{
  \flowfram_frame_set_tl:nnnV
  { flow } { margin } { #1 }
  \l__flowfram_margin_tl
}
}
```

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

```
\NewDocumentCommand \flowsetpagelist { m m }
{
  \flowfram_set_frame_pagelist:nne { flow } { #1 } { #2 }
}
```

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

```
\NewDocumentCommand \flowsetexclusion { m m }
{
  \flowfram_set_frame_excludelist:nne { flow } { #1 } { #2 }
}
```

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

```
\NewDocumentCommand \flowaddexclusion { m m }
{
  \flowfram_concat_frame_excludelist:nne
  { flow } { #1 } { #2 }
}
```

`\@@flowframeswapcoords` Swap odd and even offsets for a given flow frame. Do the main stuff for a given flow frame IDN. Version 2.0 renamed `\@@flowframeswapcoords`.

```
\cs_new:Nn \_flowfram_flow_swap_coords:n
{
  \flowfram_frame_swap_dim:nnnn
  { flow } { posx } { evenx } { #1 }
  \flowfram_frame_swap_dim:nnnn
  { flow } { posy } { eveny } { #1 }
}
```

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

```
\NewDocumentCommand \ffswapoddeven { s m }
{
  \IfValueTF { #1 }
  {
    \exp_args:Ne \clist_map_inline:nn { #2 }
    {
      \_flowfram_get_flow_id:n { ##1 }
    }
  }
}
```

```

        \_flowfram_flow_swap_coords:n { \l_flowfram_id_int }
      }
    {
      \_flowfram_map_idns:enn { #2 } { flow }
      {
        \_flowfram_flow_swap_coords:n { \l_flowfram_id_int }
      }
    }
  }
}

```

`\@sflowframeswapcoords` Version 2.0 removed `\@sflowframeswapcoords`.

`\@flowframeswapcoords` Version 2.0 removed `\@flowframeswapcoords`.

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

`\flowframex`

```

\newcommand*\flowframex[1]{%
  \flowfram_frame_use_dim:nnn { flow } { posx } { #1 }
}

```

`\flowframey`

```

\newcommand*\flowframey[1]{%
  \flowfram_frame_use_dim:nnn { flow } { posy } { #1 }
}

```

`\flowframeevenx`

```

\newcommand*\flowframeevenx[1]{%
  \flowfram_frame_use_dim:nnn { flow } { evenx } { #1 }
}

```

`\flowframeeveny`

```

\newcommand*\flowframeeveny[1]{%
  \flowfram_frame_use_dim:nnn { flow } { eveny } { #1 }
}

```

`\flowframewidth`

```

\newcommand*\flowframewidth[1]{%
  \flowfram_frame_use_dim:nnn { flow } { width } { #1 }
}

```

`\flowframeheight`

```

\newcommand*\flowframeheight[1]{%
  \flowfram_frame_use_dim:nnn { flow } { height } { #1 }
}

```

1.4 Static Frames

`\newstaticframe` Now deal with setting up the static frames. This is similar to the flow frames, except it has an associated L^AT_EX savebox rather than a T_EX box. Syntax:

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

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

```
\NewDocumentCommand \newstaticframe
  { s O{all} m m m m 0 { \int_use:N \c@maxstatic } }
{
  \__flowfram_new_static:nnnnnn
  { #2 } { #3 } { #4 } { #5 } { #6 } { #7 }
  \IfBooleanT { #1 }
  {
    \flowfram_frame_set_bool_true:nnnn
    { static } { hasframe } { \c@maxstatic }
  }
}

\cs_new:Nn \__flowfram_new_static:nnnnn
{
  \__flowfram_new_static:nnnnnn
  { #1 } { #2 } { #3 } { #4 } { #5 } { \int_use:N \c@maxstatic }
}
```

`\@n@wstaticframe` Version 2.0 removed `\@n@wstaticframe`.

`\@snewstaticframe` Version 2.0 removed `\@snewstaticframe`.

`\@newstaticframe` Version 2.0 removed `\@newstaticframe`.

`\@@newstaticframe` Now set up the static frame: Version 2.0 renamed `\@@newstaticframe`.

```
\cs_new_nopar:Nn \__flowfram_new_static:nnnnnn
{
  \__flowfram_new_frame:nnnnnnn
  { #1 } { #2 } { #3 } { #4 } { #5 } { #6 } { static }
}
```

Define content box.

```
\flowfram_frame_new_box:nnn { static } { content } { \c@maxstatic }
```

Define attributes specific to static frames.

```
\flowfram_frame_new_bool:nnnN
  { static } { clear } { \c@maxstatic } \c_false_bool
\flowfram_frame_new_bool:nnnN
  { static } { hide } { \c@maxstatic } \c_false_bool
\flowfram_frame_new_bool:nnnN
  { static } { hidethis } { \c@maxstatic } \c_false_bool
```

Inner position argument for minipage environment:

```
\flowfram_frame_new_tl:nnnn
  { static } { valign } { \c@maxstatic } { c }
```

Paragraph shaping declaration.

```
\flowfram_frame_new_tl:nnnn
  { static } { shape } { \c@maxstatic } { \relax }
```

Paragraph indentation defaults to `\sdfparindent`. This is stored as a token list variable rather than a dimension to allow it to pick up changes if `\sdfparindent` is changed afterwards.

```
\flowfram_frame_new_tl:nnnn
  { static } { parindent } { \c@maxstatic } { \sdfparindent }
}
```

`\@s@tstaticframeid` Removed `\@s@tstaticframeid`

`\@sf@checkuniqueidl` Removed `\@sf@checkuniqueidl`

`\getstaticlabel` `\getstaticlabel{<idl>}` Gets the IDL for the static frame identified by its IDN.

```
\newcommand*{\getstaticlabel}[1]{%
  \flowfram_frame_use_tl:nnn { static } { label } { #1 }
}
```

`\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.

```
\NewDocumentCommand \getstaticid { m m }
{
  \__flowfram_get_static_id:e { #2 }
  \tl_set:Ne #1 { \int_use:N \l__flowfram_id_int }
}
```

`\@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 `\l__flowfram_id_int`. Version 2.0 renamed `\@staticframeid`.

```
\cs_new:Nn \__flowfram_get_static_id:n
{
  \__flowfram_if_static_label_exists:nF { #1 }
  {
    \__flowfram_error:eee { label-undefined } { #1 } { static }
  }
}
\cs_generate_variant:Nn \__flowfram_get_static_id:n { e, V }
\prg_new_conditional:Nnn \__flowfram_if_static_label_exists:n
{ TF , T , F }
{
  \__flowfram_if_frame_label_exists:nnNTF
  { static } { #1 } \l__flowfram_id_int
  { \prg_return_true: }
  { \prg_return_false: }
}
```

Make it easier to get the x and y values for static frames.

```
\staticframex
\newcommand*\staticframex[1]{%
  \flowfram_frame_use_dim:nnn { static } { posx } { #1 }
}

\staticframey
\newcommand*\staticframey[1]{%
  \flowfram_frame_use_dim:nnn { static } { posy } { #1 }
}

\staticframeevenx
\newcommand*\staticframeevenx[1]{%
  \flowfram_frame_use_dim:nnn { static } { evenx } { #1 }
}

\staticframeeveny
\newcommand*\staticframeeveny[1]{%
  \flowfram_frame_use_dim:nnn { static } { eveny } { #1 }
}

\staticframewidth
\newcommand*\staticframewidth[1]{%
  \flowfram_frame_use_dim:nnn { static } { width } { #1 }
}

\staticframeheight
\newcommand*\staticframeheight[1]{%
  \flowfram_frame_use_dim:nnn { static } { height } { #1 }
}

\setallstaticframes Modify the settings for all the static frames:
\NewDocumentCommand \setallstaticframes { m }
{
  \int_step_inline:nn { \c@maxstatic }
  {
    \__flowfram_set_static_by_idn:nn { ##1 } { #1 }
  }
}

\setstaticframe Modify the settings for the specified static frames:
\NewDocumentCommand \setstaticframe { s m m }
{
  \IfBooleanTF { #1 }
  {
    \exp_args:Ne \clist_map_inline:nn { #2 }
    {
      \__flowfram_if_static_label_exists:nTF { ##1 }
    }
  }
}
```

```

    {
      \__flowfram_set_static_by_idn:nn { \l__flowfram_id_int } { #3 }
    }
    {
      \__flowfram_error:eee { label-undefined } { ##1 } { static }
    }
  }
}
{
  \__flowfram_map_idns:enn { #2 } { static }
  {
    \__flowfram_set_static_by_idn:nn { \l__flowfram_id_int } { #3 }
  }
}
}

```

`\@ssetstaticframe` Version 2.0 removed `\@ssetstaticframe`.

`\@setstaticframe` Version 2.0 removed `\@setstaticframe`.

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

Version 2.0 renamed `\@@setstaticframe`.

```

\cs_new:Nn \__flowfram_set_static_by_idn:nn
{
  \__flowfram_set_keys_for_type:nnn { #2 } { static } { #1 }
}

```

Option specific to static frames:

```

\tl_if_empty:NF \l__flowfram_clearflag_tl
{
  \flowfram_frame_set_bool_from_option:nnnV
  { static } { clear } { #1 }
  \l__flowfram_clearflag_tl
}
\tl_if_empty:NF \l__flowfram_hideflag_tl
{
  \flowfram_frame_set_bool_from_option:nnnV
  { static } { hide } { #1 }
  \l__flowfram_hideflag_tl
}
\tl_if_empty:NF \l__flowfram_hidethisflag_tl
{
  \flowfram_frame_set_bool_from_option:nnnV
  { static } { hidethis } { #1 }
  \l__flowfram_hidethisflag_tl
}
\exp_args:NV \tl_if_novalue:nF \l__flowfram_shape_tl
{
  \flowfram_frame_set_tl:nnnV
  { static } { shape } { #1 } \l__flowfram_shape_tl
}

```

```

\exp_args:NV \tl_if_novalue:nF \l__flowfram_parindent_tl
{
  \flowfram_frame_set_tl:nnnV
  { static } { parindent } { #1 } \l__flowfram_parindent_tl
}
\tl_if_empty:NF \l__flowfram_valign_tl
{
  \flowfram_frame_set_tl:nnnV
  { static } { valign } { #1 }
  \l__flowfram_valign_tl
}
}

```

`\FLFsimpar` Simulate paragraph break inside `\shapepar`. Version 2.0: the command is now called `\FLFsimpar` to reduce the chances of a name clash.

```

\NewDocumentCommand \FLFsimpar { }
{
  \hfill \newline
  \hspace* { \parindent }
}

```

`\simpar` Provide old command for backward-compatibility if it doesn't clash.

```

\providecommand*\simpar{\FLFsimpar}

```

`\ffpshpar` Provide means to allow `\parshape` to be carried over a paragraph break. Deprecated as from v2.0. Does nothing now. Marked for removal.

```

\newcommand \ffpshpar { }

```

`\FLForgpar` Version 2.0: no longer used.

```

\let\FLForgpar\par

```

Provide a means to have section headings within `\parshape`.

`\@ff@parshape` Version 2.0 renamed `\@ff@parshape`.

```

\tl_new:N \l__flowfram_parshape_tl
\tl_set:Nn \l__flowfram_parshape_tl { \parshape = 0 }

```

`\@ff@sectionhead` Version 2.0 removed `\@ff@sectionhead`.

`\@s@ff@heading` Version 2.0 removed `\@s@ff@heading`.

`\@ff@heading` Version 2.0 removed `\@ff@heading`.

```

\tl_new:N \l__flowfram_current_pre_section_tl
\tl_new:N \@@flowfram@current@pre@section@title
\tl_new:N \l__flowfram_current_post_section_tl

```

If a document class supports an optional argument for sectioning units, re-define to include the second argument in square brackets.

```

\cs_new:Nn \flowfram_nonum_section:Nnn

```

```

{
  #1 * { #3 }
}
\cs_generate_variant:Nn \flowfram_nonum_section:Nnn { cVn }

```

\addtocontents

```

\renewcommand \addtocontents [2]
{
  \protected@write \@auxout
  {
    \let \label \@gobble@om
    \let \index \@gobble@som
    \let \glossary \@gobble@om
    \let \@@flowfram@current@pre@section@title \empty
    \let \theFramePageCounter \relax
  }
  {
    \string \@writefile { #1 } { #2 }
  }
}

```

\FlowFramSectionUnit Do the original sectioning command but take into account the current paragraph shape and thumb tab settings. This has an extra optional argument for the thumbtab title. NB memoir also has an extra optional argument for the header so include option to allow for this. The KOMA-Script classes now support a key=value optional argument, so also need an option to allow for that.

```

\bool_new:N \l__flowfram_starred_section_bool
\tl_new:N \l__flowfram_section_thumbtab_title_tl
\NewDocumentCommand \FlowFramSectionUnit { m s o o m }
{
  \tl_clear:N \l__flowfram_section_toc_tl
  \tl_clear:N \l__flowfram_section_head_tl
  \tl_clear:N \l__flowfram_section_opti_tl
  \tl_clear:N \l__flowfram_section_optii_tl
  \tl_clear:N \l__flowfram_section_thumbtab_title_tl
  \cs_if_exist:cTF { __flowfram_org_ #1 : }
  {

```

Do initial hook.

```
\l__flowfram_current_pre_section_tl
```

Initialise header and toc variables:

```

\IfValueTF { #3 }
{
  \tl_if_empty:nTF { #3 }
  {
    \tl_set:Nn \l__flowfram_section_opti_tl { #5 }
    \tl_set:Nn \l__flowfram_section_toc_tl { #5 }
  }
}

```



```

\l_flowfram_section_opti_tl { #3 }
\l_flowfram_section_toc_tl { #3 }
}
\bool_if:NT \l_flowfram_section_keyval_opt_bool
{

```

The first optional argument may be a key=value list.

```

\l_flowfram_section_keyval_opt_tl = { #3 }
{
\keys_set:known:n { flowfram / section } { #3 }
}
}
{
\l_flowfram_section_opti_tl { #5 }
}
\l_flowfram_section_toc_tl
{
\l_flowfram_section_toc_tl { #5 }
}
\IfValueT { #4 }
{

```

Second optional argument has been provided.

```

\l_flowfram_section_header_opt_bool
{

```

Second optional argument should be passed to original command.

```

\l_flowfram_section_optii_tl { #4 }
}
\l_flowfram_section_header_opt_thumbtab_bool
{

```

Second optional argument should be used as the thumbtab title.

```

\l_flowfram_section_thumbtab_title_tl { #4 }
}
{
\l_flowfram_section_thumbtab_title_tl
\l_flowfram_section_toc_tl
}
}
\l_flowfram_section_head_tl
{
\l_flowfram_section_head_tl
\l_flowfram_section_toc_tl
}
\l_flowfram_section_thumbtab_title_tl
{
\l_flowfram_section_thumbtab_title_tl

```

```

        \l__flowfram_section_thumbtab_title_tl
        \l__flowfram_section_head_tl
    }

```

Starred?

```

\IfBooleanTF { #2 }
{
    \bool_set_true:N \l__flowfram_starred_section_bool
    \__flowfram_if_df_chapter:nTF { #1 }
    {

```

Ensure arguments are expanded in case they contain placeholder commands, such as `\glossarytitle`.

```

        \exp_args:Nee
        \__flowfram_df_chapter_star:nn
        { \l__flowfram_section_opti_tl }
        { #5 }
    }
}
{
    \flowfram_nonum_section:cVn
    { __flowfram_org_ #1 : }
    \l__flowfram_section_opti_tl
    {
        @@flowfram@current@pre@section@title
        #5
    }
}
}
{
    \bool_set_false:N \l__flowfram_starred_section_bool
    \__flowfram_if_df_chapter:nTF { #1 }
    {

```

Ensure arguments are expanded in case they contain placeholder commands, such as `\glossarytitle`.

```

        \tl_if_empty:NTF \l__flowfram_section_optii_tl
        {
            \exp_args:Nee
            \__flowfram_df_chapter:nn
            { \l__flowfram_section_opti_tl }
            { #5 }
        }
        {
            \exp_args:Neee
            \__flowfram_df_chapter:nnn
            { \l__flowfram_section_opti_tl }
            { \l__flowfram_section_optii_tl }
            { #5 }
        }
    }
}
{

```

```

        \_flowfram_use_original_sec:cVn
        { \_flowfram_org_ #1 : }
        \l_flowfram_section_opti_tl
        \l_flowfram_section_optii_tl
        { #5 }
    }
}
\cs_if_exist:cF { \_flowfram_org_end_ #1 : }
{
    \_flowfram_post_section_unit:n { #1 }
}
\l_flowfram_current_post_section_tl
}
{
    \msg_error:nne { flowfram } { unknown-heading-cmd }
    { #1 }
}
\@afterheading
}

```

Use original section command:

```

\cs_new:Nn \_flowfram_use_original_sec:Nnnn
{
    \tl_if_empty:nTF { #3 }
    {
        #1 [ #2 ]
        {
            \@@flowfram@current@pre@section@title
            #4
        }
    }
    {
        #1 [ #2 ] [ #3 ]
        {
            \@@flowfram@current@pre@section@title
            #4
        }
    }
}
\cs_generate_variant:Nn \_flowfram_use_original_sec:Nnnn
{ cVn , cVnn , cnnn }
\cs_new:Nn \_flowfram_post_section_unit:n
{

```

Add thumbtab if applicable.

```

    \_flowfram_add_thumbtab:n { #1 }

```

Do minitoc if supported.

```

    \bool_if:NF \l_flowfram_starred_section_bool
    {

```

```

    \__flowfram_minitoc_post_section_unit:n { #1 }
  }
}

```

Does nothing until redefined by `\enablethumbtabs`:

```

\cs_new:Nn \__flowfram_add_thumbtab:n { }
\cs_new:Nn \__flowfram_actual_add_thumbtab:n
{
  \tl_if_eq:eeT \g__flowfram_thumbtab_type_tl { #1 }
  {
    \bool_if:NTF \l__flowfram_starred_section_bool
    {
      \__flowfram_ttb_nonum:n { \l__flowfram_section_thumbtab_title_tl }
    }
    {
      \flowfram_if_mainmatter:TF
      {
        \__flowfram_ttb_num:n
        { \l__flowfram_section_thumbtab_title_tl }
      }
      {
        \__flowfram_ttb_notmainmatter_num:n
        { \l__flowfram_section_thumbtab_title_tl }
      }
    }
  }
}
}

```

Hook for minitocs (redefined by `\enableminitoc`):

```

\cs_new:Nn \__flowfram_minitoc_post_section_unit:n
{
}

```

Adjust sectioning commands definitions.

```

\cs_new:Nn \__flowfram_provide_section_unit:n
{
  \cs_if_exist:cT { #1 }
  {
    \cs_set_eq:cc { __flowfram_org_ #1 : } { #1 }
    \cs_set:cpn { #1 }
    {
      \FlowFramSectionUnit { #1 }
    }
  }
}

```

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.)

```

\cs_if_exist:cT { @end #1 }

```



```

}
{
  \tl_if_head_eq_meaning:nNTF { #1 } \Shapepar
  {
    \int_set_eq:NN
      \l__flowfram_shape_int
      \c_flowfram_shape_type_shapepar_int
  }
  {
    \tl_if_head_eq_meaning:nNTF { #1 } \relax
    {
      \int_set_eq:NN
        \l__flowfram_shape_int
        \c_flowfram_shape_type_none_int
    }
    {
      \msg_error:nnn { flowfram } { unknown-shape } { #1 }
      \int_set_eq:NN
        \l__flowfram_shape_int
        \c_flowfram_shape_type_shapepar_int
    }
  }
}
}
}
}
\cs_generate_variant:Nn \__flowfram_get_shape:n { V }

\cs_new:Nn \__flowfram_disabled_sec:NN
{
  \msg_error:nmmm { flowfram } { forbidden-cmd-in-shape } { #1 } { #2 }
  \@gobble@som
}

```

\@ff@disablesec Disable sectioning commands

```

\newcommand*{\@ff@disablesec}{%
  \def\section{
    \__flowfram_disabled_sec:NN \section \shapepar
  }
  \def\subsection{
    \__flowfram_disabled_sec:NN \subsection \shapepar
  }
  \def\subsubsection{
    \__flowfram_disabled_sec:NN \subsubsection \shapepar
  }
  \def\paragraph{
    \__flowfram_disabled_sec:NN \paragraph \shapepar
  }
  \def\subparagraph{
    \__flowfram_disabled_sec:NN \subparagraph \shapepar
  }
}

```

```
}
```

`staticcontents` (*env.*) Set the contents of the static frame given by its IDN. Syntax: `\begin{staticcontents}{<idn>}`. Version 2.0 added optional argument to also set frame options at the same time. (All settings are global so this shouldn't be localised by the environment.)

```
\newbox\staticframe
\NewDocumentEnvironment { staticcontents } { o m }
{
  \IfValueT { #1 }
  {
    \__flowfram_set_static_by_idn:nn { #2 } { #1 }
  }
  \RenewDocumentCommand \continueonframe { o m }
  {
    \__flowfram_static_continue_idn:nn { ##1 } { ##2 }
  }
  \__flowfram_static_contents_begin:n { #2 }
}
{
  \__flowfram_static_contents_end:
  \ignorespacesafterend
}
```

`staticcontents*` (*env.*) Set the contents of the static frame given by its IDL. Syntax: `\begin{staticcontents*}{<label>}`. Version 2.0 added optional argument to also set frame options at the same time. (All settings are global so this shouldn't be localised by the environment.)

```
\NewDocumentEnvironment { staticcontents* } { o m }
{
  \__flowfram_get_static_id:n { #2 }
  \IfValueT { #1 }
  {
    \__flowfram_set_static_by_idn:nn
    { \l__flowfram_id_int }
    { #1 }
  }
  \RenewDocumentCommand \continueonframe { o m }
  {
    \__flowfram_static_continue_idl:nn { ##1 } { ##2 }
  }
  \__flowfram_static_contents_begin:n { \l__flowfram_id_int }
}
{
  \__flowfram_static_contents_end:
  \ignorespacesafterend
}
```

`\@beginstaticcontents` Begin `staticcontents` stuff. Version 2.0 renamed `\@beginstaticcontents`.

```

\cs_new:Nn \__flowfram_static_contents_begin:n
{
  \int_zero:N \l__flowfram_current_static_int
  \box_if_exist:cTF
    { g__flowfram_static_content_ \romannumeral #1 _box }
  {
    \int_set:Nn \l__flowfram_current_static_int { #1 }
    \tl_set:Ne \l__flowfram_minipage_tl
      {
        \exp_not:N \begin { minipage }
        [ c ]
        [
          \staticframeheight { #1 }
        ]
        [
          \flowfram_frame_use_tl:nnn
            { static } { valign } { #1 }
        ]
        {
          \staticframewidth { #1 }
        }
      }
    \flowfram_set_tl_to_frame_tl:Nnnn
      \l__flowfram_parindent_tl
      { static } { parindent } { #1 }
    \flowfram_set_tl_to_frame_tl:Nnnn
      \l__flowfram_parshape_tl
      { static } { shape } { #1 }
    \__flowfram_get_shape:V \l__flowfram_parshape_tl
    \int_case:nnF { \l__flowfram_shape_int }
      {
        { \c_flowfram_shape_type_none_int }
        No shape (nothing extra to do)
        { }
        { \c_flowfram_shape_type_parshape_int }
        {
          \parshape:
            \tl_put_right:NV
              \l__flowfram_minipage_tl
              \l__flowfram_parshape_tl
            \tl_put_right:Nn \l__flowfram_minipage_tl
              {
                \let \par \FLFsimpar
                \__flowfram_set_shaped_section_headings:
              }
          }
        }
      }
}

```


`\shapepar` or `\Shapepar`:

```

\l_flowfram_minipage_tl
{
  \@ff@disablesec
}
\l_flowfram_minipage_tl
\l_flowfram_parshape_tl
}
\UseHook { flowfram / staticbox / before }
\begin { lrbox } { \staticframe }
\flowfram_set_tl_to_frame_tl:Nnnn
\l_flowfram_textcolor_tl
{ static } { textcolor } { #1 }
\__flowfram_set_text_color:
\noindent
\l_flowfram_minipage_tl
\dim_set:Nn \parindent { \l_flowfram_parindent_tl }
}
{
\msg_error:nnnn { flowfram } { idn-undefined } { static } { #1 }
}
}

```

Hooks used before and after `lrbox` environment when static contents are set. Since the box is set within the `staticcontents` or `staticcontents*` environments, it will be within a scoped context.

```

\NewMirroredHookPair
{ flowfram / staticbox / before }
{ flowfram / staticbox / after }

```

`\@endstaticcontents` End `staticcontents` stuff. Version 2.0 renamed `\@endstaticcontents`.

```

\cs_new:Nn \__flowfram_static_contents_end:
{
\int_if_zero:nF { \l_flowfram_current_static_int }
{
\endgraf

```

Grouping has been removed as it interferes with `\shapepar` but the environment already provides scoping.

```

\end { minipage }
\end { lrbox }
\UseHook { flowfram / staticbox / after }
\flowfram_frame_set_box_eq:nnnN
{ static } { content } { \l_flowfram_current_static_int }
\staticframe
}
}

```

`\setstaticcontents` Provide a command version. Syntax: `\setstaticcontents[<options>]{<idn>}` `{<text>}`. Version 2.0 added optional argument to also set frame options at the same time.

```

\NewDocumentCommand \setstaticcontents { s o m +m }
{
  \IfBooleanTF { #1 }
  {
    \__flowfram_get_static_id:n { #3 }
    \IfValueT { #2 }
    {
      \__flowfram_set_static_by_idn:nn
        { \l__flowfram_id_int }
        { #2 }
    }
    \begin { staticcontents } { \l__flowfram_id_int }
      #4
    \end { staticcontents }
  }
  {
    \IfValueT { #2 }
    {
      \__flowfram_set_static_by_idn:nn { #3 } { #2 }
    }
    \begin { staticcontents } { #3 }
      #4
    \end { staticcontents }
  }
}

```

`\@sstaticconts` Version 2.0 removed `\@sstaticconts`.

`\@staticconts` Version 2.0 removed `\@staticconts`.

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

```

\NewDocumentCommand \staticsetpagelist { m m }
{
  \flowfram_set_frame_pagelist:nne { static } { #1 } { #2 }
}

```

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

```

\NewDocumentCommand \staticsetexclusion { m m }
{
  \flowfram_set_frame_excludelist:nne { static } { #1 } { #2 }
}

```

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

```

\NewDocumentCommand \staticaddexclusion { m m }
{
  \flowfram_concat_frame_excludelist:nne

```

```

    { static } { #1 } { #2 }
  }

```

`\@staticframeswapcoords` Version 2.0 removed `\@staticframeswapcoords`.

`\sfswapoddeven` Swap odd and even offsets for a given static frame. Allow user to specify static frame either by IDN or IDL:

```

\NewDocumentCommand \sfswapoddeven { s m }
{
  \IfValueTF { #1 }
  {
    \exp_args:Ne \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_static_id:n { ##1 }
      \__flowfram_static_swap_coords:n { \l__flowfram_id_int }
    }
  }
  {
    \__flowfram_map_idns:enn { #2 } { static }
    {
      \__flowfram_static_swap_coords:n { \l__flowfram_id_int }
    }
  }
}

```

`\@sstaticframeswapcoords` Version 2.0 removed `\@sstaticframeswapcoords`.

`\@staticframeswapcoords` Version 1.18 renamed `\@staticframeswapcoords`.

```

\cs_new:Nn \__flowfram_static_swap_coords:n
{
  \flowfram_frame_swap_dim:nnnn
  { static } { posx } { evenx } { #1 }
  \flowfram_frame_swap_dim:nnnn
  { static } { posy } { eveny } { #1 }
}

```

`\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. Version 2.0 now allows `\continueonframe` in `\setdynamiccontents` but not in `\appenddynamiccontents`.

```

\NewDocumentCommand \continueonframe { 0{ } m }
{
  \msg_error:nn { flowfram } { cant-continue }
}

```

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

\@staticscontinueonframe Static starred version uses IDL Version 2.0 replaced \@staticscontinueonframe.

```
\cs_new:Nn \__flowfram_static_continue_idl:nn
{
  \IfValueTF { #1 }
  {
    \ffcontinuedtextlayout { #1 }
  }
  {
    \ffcontinuedtextlayout
    {
      \flowfram_continue_on_static_frame_idl:Nn
      \l__flowfram_current_static_int
      { #2 }
    }
  }
  \exp_args:NnNV
  \end { staticcontents* }
  \__flowfram_set_post_continued:nnn
  \l__flowfram_current_static_int
  { \l__flowfram_current_static_int }
  { \c_flowfram_frame_type_static_int }
  \tl_put_left:Nn \l__flowfram_postcontinued_tl
  { \begin { staticcontents* } { #2 } }
  \l__flowfram_postcontinued_tl
}
```

\@staticcontinueonframe Static unstarred version uses IDN Version 2.0 replaced \@staticcontinueonframe.

```
\cs_new:Nn \__flowfram_static_continue_idn:nn
{
  \IfValueTF { #1 }
  {
    \ffcontinuedtextlayout { #1 }
  }
  {
    \ffcontinuedtextlayout
    {
      \flowfram_continue_on_static_frame_idn:Nn
      \l__flowfram_current_static_int
      { #2 }
    }
  }
  \exp_args:NnNV
  \end { staticcontents }
  \__flowfram_set_post_continued:nnn
  \l__flowfram_current_static_int
  { \l__flowfram_current_static_int }
  { \c_flowfram_frame_type_static_int }
  \tl_put_left:Nn \l__flowfram_postcontinued_tl
  { \begin { staticcontents } { #1 } }
  \l__flowfram_postcontinued_tl
}
```

```

}

\tl_new:N \l__flowfram_postcontinued_tl
\cs_new:Nn \__flowfram_set_post_continued:nnn
{
  \int_case:nnF { #3 }
  {
    { \c_flowfram_frame_type_static_int }
    {
      \tl_set:Nn \l__flowfram_postcontinued_tl
      {
        \ffstaticpostcontinued { #1 } { #2 }
      }
    }
    { \c_flowfram_frame_type_dynamic_int }
    {
      \tl_set:Nn \l__flowfram_postcontinued_tl
      {
        \ffdynamicpostcontinued { #1 } { #2 }
      }
    }
  }
}
\cs_new:Nn \flowfram_continue_on_static_frame_idn:Nn
{
  \ffdefaultstaticcontinuetext { #1 } { #2 }
}

```

The first argument is the IDN of the current frame and the second is the label of the next frame.

```

\cs_new:Nn \flowfram_continue_on_static_frame_idl:Nn
{
  \__flowfram_get_static_id:n { #2 }
  \flowfram_continue_on_static_frame_idn:Nn #1 { \l__flowfram_id_int }
}

```

```
\ffdefaultstaticcontinuetext
```

```
\newcommand \ffdefaultstaticcontinuetext [ 2 ] { \ffdefaultcontinuetext }
```

```
\ffdefaultdynamiccontinuetext
```

```
\newcommand \ffdefaultdynamiccontinuetext [ 2 ] { \ffdefaultcontinuetext }
```

```
\ffdefaultcontinuetext
```

```
\newcommand \ffdefaultcontinuetext { }
```

```
\ffstaticpostcontinued
```

```
\newcommand \ffstaticpostcontinued [ 2 ]
{
  \ffdefaultpostcontinued
}
```

\ffdynamictopcontinued

```
\newcommand \ffdynamictopcontinued [ 2 ]
{
  \ffdefaultpostcontinued
}
```

\ffdefaultpostcontinued



Hook for start of next frame:

```
\newcommand{\ffdefaultpostcontinued}{%
  \par
  \noindent
  \ignorespaces
}
```

\ffcontinuedtextlayout Displays the continued text used by \continueonframe.

```
\newcommand{\ffcontinuedtextlayout}[1]{%
  \group_begin:
  \dim_zero:N \parfillskip
  \par
  \hfill
  \ffcontinuedtextfont { #1 }
  \par
  \group_end:
}
```

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

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

1.5 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⟩]
\NewDocumentCommand \newdynamicframe
  { s O{all} m m m m O { \int_use:N \c@maxdynamic } }
{
  \__flowfram_new_dynamic:nnnnn
  { #2 } { #3 } { #4 } { #5 } { #6 } { #7 }
}
```

```

\IfBooleanT { #1 }
{
  \flowfram_frame_set_bool_true:nnnn
  { dynamic } { hasframe } { \c@maxdynamic }
}
}

\cs_new:Nn \__flowfram_new_dynamic:nnnnn
{
  \__flowfram_new_dynamic:nnnnnn
  { #1 } { #2 } { #3 } { #4 } { #5 } { \int_use:N \c@maxdynamic }
}

```

`\@n@wdynamicframe` Version 2.0 removed `\@n@wdynamicframe`.

`\@snewdynamicframe` Version 2.0 removed `\@snewdynamicframe`.

`\@newdynamicframe` Version 2.0 removed `\@newdynamicframe`.

`\@@newdynamicframe` Create new dynamic frame: Version 2.0 renamed `\@@newdynamicframe`.

```

\cs_new_nopar:Nn \__flowfram_new_dynamic:nnnnnn
{
  \__flowfram_new_frame:nnnnnnnn
  { #1 } { #2 } { #3 } { #4 } { #5 } { #6 } { dynamic }
}

```

Define content token list.

```
\flowfram_frame_new_tl:nnn { dynamic } { content } { \c@maxdynamic }
```

Define attributes specific to dynamic frames.

```

\flowfram_frame_new_bool:nnnN
{ dynamic } { clear } { \c@maxdynamic } \c_false_bool
\flowfram_frame_new_bool:nnnN
{ dynamic } { hide } { \c@maxdynamic } \c_false_bool
\flowfram_frame_new_bool:nnnN
{ dynamic } { hidethis } { \c@maxdynamic } \c_false_bool

```

Inner position argument for `\parbox`:

```
\flowfram_frame_new_tl:nnnn
{ dynamic } { valign } { \c@maxdynamic } { t }
```

Paragraph shaping declaration.

```
\flowfram_frame_new_tl:nnnn
{ dynamic } { shape } { \c@maxdynamic } { \relax }
```

Paragraph indentation defaults to `\sdfparindent`. This is stored as a token list variable rather than a dimension to allow it to pick up changes if `\sdfparindent` is changed.

```
\flowfram_frame_new_tl:nnnn
{ dynamic } { parindent } { \c@maxdynamic } { \sdfparindent }
```

Formatting style csname.

```

\flowfram_frame_new_tl:nnnn
{ dynamic } { style } { \c@maxdynamic } { @firstofone }
}

```

`\s@tdynamicframeid` Version 2.0 removed `\s@tdynamicframeid`.

`\df@checkuniqueidl` Version 2.0 removed `\df@checkuniqueidl`.

`\getdynamiclabel` `\getdynamiclabel{<idl>}` Gets the IDL for the dynamic frame identified by its IDN.

```
\newcommand*\getdynamiclabel}[1]{%
  \flowfram_frame_use_tl:nnn { dynamic } { label } { #1 }
}
```

`\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.

```
\NewDocumentCommand \getdynamicid { m m }
{
  \__flowfram_get_dynamic_id:e { #2 }
  \tl_set:N#1 { \int_use:N \l__flowfram_id_int }
}
```

`\@dynamicframeid` Determine the IDN of the dynamic frame from its label. The IDN is stored in `\l__flowfram_id_int`. Version 2.0 renamed `\@dynamicframeid`.

```
\cs_new:Nn \__flowfram_get_dynamic_id:n
{
  \__flowfram_if_dynamic_label_exists:nF { #1 }
  {
    \__flowfram_error:eee { label-undefined } { #1 } { dynamic }
  }
}
\cs_generate_variant:Nn \__flowfram_get_dynamic_id:n { e, V }

\prg_new_conditional:Nnn \__flowfram_if_dynamic_label_exists:n
{ TF , T , F }
{
  \__flowfram_if_frame_label_exists:nnNTF
  { dynamic } { #1 } \l__flowfram_id_int
  { \prg_return_true: }
  { \prg_return_false: }
}
```

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

`\dynamicframex`

```
\newcommand*\dynamicframex}[1]{%
  \flowfram_frame_use_dim:nnn { dynamic } { posx } { #1 }
}
```

`\dynamicframey`

```
\newcommand*\dynamicframey}[1]{%
  \flowfram_frame_use_dim:nnn { dynamic } { posy } { #1 }
}
```



```

\dynamicframeevenx
  \newcommand*\dynamicframeevenx[1]{%
    \flowfram_frame_use_dim:nnn { dynamic } { evenx } { #1 }
  }

\dynamicframeeveny
  \newcommand*\dynamicframeeveny[1]{%
    \flowfram_frame_use_dim:nnn { dynamic } { eveny } { #1 }
  }

\dynamicframewidth
  \newcommand*\dynamicframewidth[1]{%
    \flowfram_frame_use_dim:nnn { dynamic } { width } { #1 }
  }

\dynamicframeheight
  \newcommand*\dynamicframeheight[1]{%
    \flowfram_frame_use_dim:nnn { dynamic } { height } { #1 }
  }

\setalldynamicframes Change the settings for all the dynamic frames:
  \NewDocumentCommand \setalldynamicframes { m }
  {
    \int_step_inline:nn { \c@maxdynamic }
    {
      \__flowfram_set_dynamic_by_idn:nn { ##1 } { #1 }
    }
  }

\setdynamicframe Change the settings for specified dynamic frames: The starred version iterates
through comma-separated list of labels.
  \NewDocumentCommand \setdynamicframe { s m m }
  {
    \IfBooleanTF { #1 }
    {
      \exp_args:Ne \clist_map_inline:nn { #2 }
      {
        \__flowfram_if_dynamic_label_exists:nTF { ##1 }
        {
          \__flowfram_set_dynamic_by_idn { \l__flowfram_id_int } { #3 }
        }
        {
          \__flowfram_error:eee { label-undefined } { ##1 } { dynamic }
        }
      }
    }
    {
      \__flowfram_map_idns:enn { #2 } { dynamic }
      {

```

```

        \__flowfram_set_dynamic_by_idn:nn { \l__flowfram_id_int } { #3 }
      }
    }
  }

```

`\@ssetdynamicframe` Version 2.0 removed `\@ssetdynamicframe`.

`\@setdynamicframe` Version 2.0 removed `\@setdynamicframe`.

`\@@setdynamicframe` Change the setting for the dynamic frame given by its IDN. Version 2.0 renamed `\@@setdynamicframe`.

```

\cs_new:Nn \__flowfram_set_dynamic_by_idn:nn
{
  \__flowfram_set_keys_for_type:nnn { #2 } { dynamic } { #1 }
}

```

Option specific to dynamic frames:

```

\tl_if_empty:NF \l__flowfram_style_tl
{
  \flowfram_frame_set_tl:nnnV
  { dynamic } { style } { #1 } \l__flowfram_style_tl
}
\tl_if_empty:NF \l__flowfram_clearflag_tl
{
  \flowfram_frame_set_bool_from_option:nnnV
  { dynamic } { clear } { #1 }
  \l__flowfram_clearflag_tl
}
\tl_if_empty:NF \l__flowfram_hideflag_tl
{
  \flowfram_frame_set_bool_from_option:nnnV
  { dynamic } { hide } { #1 }
  \l__flowfram_hideflag_tl
}
\tl_if_empty:NF \l__flowfram_hidethisflag_tl
{
  \flowfram_frame_set_bool_from_option:nnnV
  { dynamic } { hidethis } { #1 }
  \l__flowfram_hidethisflag_tl
}
\exp_args:NV \tl_if_novalue:nF \l__flowfram_shape_tl
{
  \flowfram_frame_set_tl:nnnV
  { dynamic } { shape } { #1 } \l__flowfram_shape_tl
}
\exp_args:NV \tl_if_novalue:nF \l__flowfram_parindent_tl
{
  \flowfram_frame_set_tl:nnnV
  { dynamic } { parindent } { #1 } \l__flowfram_parindent_tl
}
\tl_if_empty:NF \l__flowfram_valign_tl

```

```

    {
      \flowfram_frame_set_tl:nnnV
        { dynamic } { valign } { #1 }
      \l__flowfram_valign_tl
    }
  }
}

```

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

```

\NewDocumentCommand \dynamicsetpagelist { m m }
{
  \flowfram_set_frame_pagelist:nne { dynamic } { #1 } { #2 }
}

```

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

```

\NewDocumentCommand \dynamicsetexclusion { m m }
{
  \flowfram_set_frame_excludelist:nne { dynamic } { #1 } { #2 }
}

```

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

```

\NewDocumentCommand \dynamicaddexclusion { m m }
{
  \flowfram_concat_frame_excludelist:nne
    { dynamic } { #1 } { #2 }
}

```

`\@@dynamicframeswapcoords` Swap odd and even offsets for a given dynamic frame. Do the main stuff for a given dynamic frame IDN. Version 2.0 removed `\@@dynamicframeswapcoords`.

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

```

\NewDocumentCommand \dfswapoddeven { s m }
{
  \IfValueTF { #1 }
  {
    \exp_args:Ne \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_dynamic_id:n { ##1 }
      \__flowfram_dynamic_swap_coords:n { \l__flowfram_id_int }
    }
  }
  {
    \__flowfram_map_idns:enn { #2 } { dynamic }
    {
      \__flowfram_dynamic_swap_coords:n { \l__flowfram_id_int }
    }
  }
}

```

`\@dynamicframeswapcoords` Version 2.0 removed `\@dynamicframeswapcoords`.

`\@dynamicframeswapcoords` Version 2.0 renamed `\@dynamicframeswapcoords`.

```
\cs_new:Nn \__flowfram_dynamic_swap_coords:n
{
  \flowfram_frame_swap_dim:nnnn
  { dynamic } { posx } { evenx } { #1 }
  \flowfram_frame_swap_dim:nnnn
  { dynamic } { posy } { eveny } { #1 }
}
```

Set the contents of a dynamic frame.

`dynamiccontents` (*env.*) Syntax: `\begin{dynamiccontents}<idn>` Set the contents of a dynamic frame with an environment that captures its body.

```
\NewDocumentEnvironment { dynamiccontents } { o m +b }
{
  \IfValueT { #1 }
  {
    \__flowfram_set_dynamic_by_idn:nn { #2 } { #1 }
  }
  \setdynamiccontents { #2 } { #3 }
}
{ }
```

`dynamiccontents*` (*env.*) Starred version references the frame by its label.

```
\NewDocumentEnvironment { dynamiccontents* } { o m +b }
{
  \__flowfram_get_dynamic_id:n { #2 }
  \IfValueT { #1 }
  {
    \__flowfram_set_dynamic_by_idn:nn
    { \l__flowfram_id_int } { #1 }
  }
  \setdynamiccontents
  { \l__flowfram_id_int } { #3 }
}
{ }
```

`\@dynamicctok` Version 2.0 removed `\@dynamicctok`.

`\xdynamiccontents` Version 2.0 removed `\xdynamiccontents`.

`\@flf@get@body` Version 2.0 removed `\@flf@get@body`.

`\ifdfcontinued` Version 2.0 removed `\ifdfcontinued`.

`\@flf@checkcontinued` Version 2.0 removed `\@flf@checkcontinued`.

`\@lempy` Version 2.0 removed `\@lempy`.

`\flf@getcontargs` Version 2.0 removed `\flf@getcontargs`.

\@flf@getcontargs Version 2.0 removed \@flf@getcontargs.

\@flf@find@end Version 2.0 removed \@flf@find@end.

\endxdynamiccontents Version 2.0 removed \endxdynamiccontents.

\@flf@endxdynamiccontents Version 2.0 removed \@flf@endxdynamiccontents.

\setdynamiccontents

```
\NewDocumentCommand \setdynamiccontents { s o m +m }
{
  \group_begin:
  \IfBooleanTF { #1 }
  {
    \__flowfram_get_dynamic_id:n { #3 }
    \IfValueT { #2 }
    {
      \__flowfram_set_dynamic_by_idn:nn
        { \l__flowfram_id_int } { #2 }
    }
    \RenewDocumentCommand \continueonframe { o m }
    {
      \__flowfram_dynamic_continue_idl:nn { ##1 } { ##2 }
    }
    \__flowfram_set_dynamic_contents_from_body:nn
      { \l__flowfram_id_int } { #4 }
  }
  {
    \IfValueT { #2 }
    {
      \__flowfram_set_dynamic_by_idn:nn { #3 } { #2 }
    }
    \RenewDocumentCommand \continueonframe { O{} m }
    {
      \__flowfram_dynamic_continue_idn:nn { ##1 } { ##2 }
    }
    \__flowfram_set_dynamic_contents_from_body:nn
      { #3 } { #4 }
  }
  \group_end:
}

\cs_new:Nn \__flowfram_dynamic_continue_idn:nn
{
  \IfValueTF { #1 }
  {
    \__flowfram_append_dynamic_contents_idn:nn
      { \l__flowfram_current_dynamic_int }
      { \ffcontinuedtextlayout { #1 } }
  }
  {

```

Expansion required variables may have different values when the frame contents are displayed.

```

\exp_args:Nne
\__flowfram_append_dynamic_contents_idn:nn
{ \l__flowfram_current_dynamic_int }
{
  \exp_not:N \ffcontinuedtextlayout
  {
    \exp_not:N \flowfram_continue_on_dynamic_frame_idn:nn
    { \int_use:N \l__flowfram_current_dynamic_int }
    { \int_eval:n { #2 } }
  }
}
}
\exp_args:NV
\__flowfram_set_post_continued:nnn
\l__flowfram_current_dynamic_int
{ \l__flowfram_current_dynamic_int }
{ \c_flowfram_frame_type_dynamic_int }
\int_set:Nn \l__flowfram_current_dynamic_int { #2 }
\exp_after:wN
\__flowfram_set_dynamic_contents_from_body:w
\l__flowfram_postcontinued_tl
}
\cs_new:Nn \__flowfram_dynamic_continue_idl:nn
{
  \IfValueTF { #1 }
  {
    \__flowfram_append_dynamic_contents_idn:nn
    { \l__flowfram_current_dynamic_int }
    { \ffcontinuedtextlayout { #1 } }
  }
  {

```

Expansion required variables may have different values when the frame contents are displayed.

```

\exp_args:Nne
\__flowfram_append_dynamic_contents_idn:nn
{ \l__flowfram_current_dynamic_int }
{
  \exp_not:N \ffcontinuedtextlayout
  {
    \exp_not:N \flowfram_continue_on_dynamic_frame_idl:nn
    { \int_use:N \l__flowfram_current_dynamic_int }
    { #2 }
  }
}
}
\exp_args:NV

```

```

    \__flowfram_set_post_continued:nnn
      \l__flowfram_current_dynamic_int
      { \l__flowfram_current_dynamic_int }
      { \c_flowfram_frame_type_dynamic_int }
    \__flowfram_get_dynamic_id:n { #2 }
    \int_set_eq:Nn
      \l__flowfram_current_dynamic_int
      \l__flowfram_id_int
    \exp_after:wN
    \__flowfram_set_dynamic_contents_from_body:w
      \l__flowfram_postcontinued_tl
  }
\cs_new:Nn \flowfram_continue_on_dynamic_frame_idn:nn
{
  \ffdefaultdynamiccontinuetext { #1 } { #2 }
}
\cs_new:Nn \flowfram_continue_on_dynamic_frame_idl:nn
{
  \__flowfram_get_dynamic_id:n { #2 }
  \ffdefaultdynamiccontinuetext { #1 } { \l__flowfram_id_int }
}
\cs_new:Nn \__flowfram_set_dynamic_contents_from_body:nn
{
  \int_set:Nn \l__flowfram_current_dynamic_int { #1 }
  \__flowfram_set_dynamic_contents_from_body:w
  #2 \continueonframe \q_nil \q_stop
}
\cs_new:Npn \__flowfram_set_dynamic_contents_from_body:w
#1 \continueonframe #2 \q_stop
{
  \__flowfram_set_dynamic_contents:nn
  { \l__flowfram_current_dynamic_int }
  { #1 }
  \tl_if_head_eq_meaning:nNTF { #2 } [
  {
    \__flowfram_dynamic_continued_opt_conts:wn #2 \q_stop
  }
  {
    \quark_if_nil:nF { #2 }
    {
      \__flowfram_dynamic_continued_conts:wn
      #2
      \q_stop
    }
  }
}
\cs_new:Npn \__flowfram_dynamic_continued_opt_conts:wn
[ #1 ] #2 #3 \q_stop

```

```

{
  \continueonframe [ #1 ] { #2 }
  #3 \q_stop
}
\cs_new:Npn \__flowfram_dynamic_continued_conts:wn
#1 #2 \q_stop
{
  \continueonframe { #1 }
  #2 \q_stop
}

```

`\@ssetdynamiccontents` Starred version: identify dynamic frame by its IDL: Version 2.0 removed `\@ssetdynamiccontents`.

`\@setdynamiccontents` Set the contents of a dynamic frame identified by its IDN: Version 2.0 renamed `\@setdynamiccontents`.

```

\cs_new:Nn \__flowfram_set_dynamic_contents:nn
{
  \flowfram_frame_set_tl:nnnn
  { dynamic } { content } { #1 } { #2 }
}

```

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

```

\NewDocumentCommand \appenddynamiccontents { s o m +m }
{
  \IfBooleanTF { #1 }
  {
    \__flowfram_get_dynamic_id:n { #3 }
    \IfValueT { #2 }
    {
      \__flowfram_set_dynamic_by_idn:nn
      { \l__flowfram_id_int } { #2 }
    }
    \__flowfram_append_dynamic_contents_idn:nn { \l__flowfram_id_int } { #4 }
  }
  {
    \IfValueT { #2 }
    {
      \__flowfram_set_dynamic_by_idn:nn { #3 } { #2 }
    }
    \__flowfram_append_dynamic_contents_idn:nn { #3 } { #4 }
  }
}

```

`\@sappenddynamic` Append to dynamic frame identified by its IDL. Version 2.0 renamed `\@sappenddynamic`.

```

\cs_new:Nn \__flowfram_append_dynamic_contents_idl:nn
{

```



```

    \__flowfram_get_dynamic_id:n { #1 }
    \__flowfram_append_dynamic_contents_idn:nn
      { \l__flowfram_id_int } { #2 }
  }

```

`\@appenddynamic` Append to dynamic frame identified by its IDN. Version 2.0 renamed `\@appenddynamic`.

```

\cs_new:Nn \__flowfram_append_dynamic_contents_idn:nn
{
  \flowfram_frame_put_right_tl:nxxx
    { dynamic } { content } { #1 } { #2 }
}

```

`\flf@ta` Version 2.0 removed `\flf@ta`.

`\flf@tb` Version 2.0 removed `\flf@tb`.

`\@ff@addtolist` Version 2.0 removed `\@ff@addtolist`.

1.6 Determining Dimensions and Locations

`\compuleftedgeodd` Compute the position of the leftmost 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.

```

\NewDocumentCommand \compuleftedgeodd { m }
{
  \dim_set:Nn #1 { -1in - \hoffset - \oddsidemargin }
}

```

`\compuleftedgeeven` Now for the even pages

```

\NewDocumentCommand \compuleftedgeeven { m }
{
  \dim_set:Nn #1 { -1in - \hoffset - \evensidemargin }
}

```

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

```

\NewDocumentCommand \computetopedge { m }
{
  \dim_set:Nn #1
  {
    \typeblockheight
    + \typeblockoffsety
    + 1in
  }
}

```

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

```

\NewDocumentCommand \computebottomedge { m }

```

```

{
  \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.

```

\NewDocumentCommand \computerightedgeodd { m }
{
  \computeleftedgeodd { #1 }
  \addtolength { #1 } { \paperwidth }
}

```

`\computerightedgeeven` Now for the even pages.

```

\NewDocumentCommand \computerightedgeeven { m }
{
  \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.

```

\NewDocumentCommand \computeflowframearea { s m }
{
  \setlength \ffareax { \paperwidth }
  \setlength \ffareay { \paperheight }
  \dim_zero:N \l__flowfram_x_dim
  \dim_zero:N \l__flowfram_y_dim
  \IfBooleanTF { #1 }
  {
    \exp_args:Ne \clist_map_inline:nn { #1 }
    {
      \__flowfram_if_flow_label_exists:nTF { ##1 }
      {
        \__flowfram_compute_flow_area_by_idn:n { \l__flowfram_id_int }
      }
      {
        \__flowfram_error:eee { label-undefined } { ##1 } { flow }
      }
    }
  }
}

```

```

}
{
  \__flowfram_map_idns:enn { #2 } { flow }
  {
    \__flowfram_compute_flow_area_by_idn:n { \l__flowfram_id_int }
  }
}
\dim_set:Nn \ffareawidth
{
  \l__flowfram_x_dim -\ffareax
}
\dim_set:Nn \ffareaheight
{
  \l__flowfram_y_dim - \ffareay
}
}

\cs_new:Nn \__flowfram_compute_flow_area_by_idn:n
{
  \dim_compare:nNnT
  { \ffareax } > { \flowframex { #1 } }
  {
    \dim_set:Nn \ffareax { \flowframex { #1 } }
  }
  \dim_compare:nNnT
  { \ffareay } > { \flowframey { #1 } }
  {
    \dim_set:Nn \ffareay { \flowframey { #1 } }
  }
  \dim_set:Nn \l__flowfram_offset_dim
  {
    \flowframex { #1 }
    + \flowframewidth { #1 }
  }
  \dim_compare:nNnT
  { \l__flowfram_x_dim } < { \l__flowfram_offset_dim }
  {
    \dim_set_eq:NN \l__flowfram_x_dim \l__flowfram_offset_dim
  }
  \dim_set:Nn \l__flowfram_offset_dim
  {
    \flowframey { #1 }
    + \flowframeheight { #1 }
  }
  \dim_compare:nNnT
  { \l__flowfram_y_dim } < { \l__flowfram_offset_dim }
  {
    \dim_set_eq:NN \l__flowfram_y_dim \l__flowfram_offset_dim
  }
}
}

```

\@scomputeffarea Version 2.0 removed \@scomputeffarea.

\@computeffarea Version 2.0 removed \@computeffarea.

\@ff@swaplen Swap the values of two dimensions. Version 2.0 renamed \@ff@swaplen.

```
\cs_new:Nn \__flowfram_swap_dim:NN
{
  \dim_set_eq:NN \l__flowfram_tmpa_dim #1
  \dim_set_eq:NN #1 #2
  \dim_set_eq:NN #2 \l__flowfram_tmpa_dim
}
```

\@ff@getdim Get the dimensions for the given type of frame. The first parameter should be a number indicating type of frame : 1 (flow), 2 (static), 3 (dynamic). The second number is its IDN. Values are stored in \ffareax, \ffareay, \ffareawidth and \ffareaheight. Version 2.0 renamed \@ff@getdim.

```
\cs_new:Nn \__flowfram_get_frame_bounds_by_typeid:nn
{
  \int_case:nnF { #1 }
  {
    { \c_flowfram_frame_type_flow_int }
    {
      \__flowfram_get_flow_bounds:n { #2 }
    }
    { \c_flowfram_frame_type_static_int }
    {
      \__flowfram_get_static_bounds:n { #2 }
    }
    { \c_flowfram_frame_type_dynamic_int }
    {
      \__flowfram_get_dynamic_bounds:n { #2 }
    }
  }
  {
    \msg_error:nne { flowfram } { invalid-frame-typeid }
    { \int_eval:n { #1 } }
  }
}
```

Identify by type label.

```
\cs_new:Nn \__flowfram_get_frame_bounds_by_type:nn
{
  \flowfram_set_dim_to_frame_dim:Nnnn
  \ffareax { #1 } { posx } { #2 }
  \flowfram_set_dim_to_frame_dim:Nnnn
  \ffareay { #1 } { posy } { #2 }
  \flowfram_set_dim_to_frame_dim:Nnnn
  \ffareaevenx { #1 } { evenx } { #2 }
  \flowfram_set_dim_to_frame_dim:Nnnn
}
```

```

\ffareaeveny { #1 } { eveny } { #2 }
\flowfram_set_dim_to_frame_dim:Nnnn
\ffareawidth { #1 } { width } { #2 }
\flowfram_set_dim_to_frame_dim:Nnnn
\ffareaheight { #1 } { height } { #2 }
}

```

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

```

\cs_new:Nn \__flowfram_get_frame_even_bounds_by_typeid:nn
{
  \int_case:nnF { #1 }
  {
    { \c_flowfram_frame_type_flow_int }
    {
      \__flowfram_get_flow_even_bounds:n { #2 }
    }
    { \c_flowfram_frame_type_static_int }
    {
      \__flowfram_get_static_even_bounds:n { #2 }
    }
    { \c_flowfram_frame_type_dynamic_int }
    {
      \__flowfram_get_dynamic_even_bounds:n { #2 }
    }
  }
  {
    \msg_error:nne { flowfram } { invalid-frame-typeid }
    { \int_eval:n { #1 } }
  }
}

```

```

\cs_new:Nn \__flowfram_get_frame_even_bounds_by_type:nn
{
  \flowfram_set_dim_to_frame_dim:Nnnn
  \ffareaax { #1 } { evenx } { #2 }
  \flowfram_set_dim_to_frame_dim:Nnnn
  \ffareaay { #1 } { eveny } { #2 }
  \flowfram_set_dim_to_frame_dim:Nnnn
  \ffareaawidth { #1 } { width } { #2 }
  \flowfram_set_dim_to_frame_dim:Nnnn
  \ffareaheight { #1 } { height } { #2 }
}

```

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

```

\NewDocumentCommand \getstaticbounds { s m }
{
  \IfBooleanTF { #1 }

```

```

    {
      \__flowfram_get_static_id:n { #2 }
      \__flowfram_get_static_bounds:n { \l__flowfram_id_int }
    }
    {
      \__flowfram_get_static_bounds:n { #2 }
    }
  }

```

`\@sgetstaticbounds` Version 2.0 removed `\@sgetstaticbounds`.

`\@getstaticbounds` Unstarred version (specify by IDN): Version 2.0 renamed `\@getstaticbounds`.

```

\cs_new:Nn \__flowfram_get_static_bounds:n
{
  \__flowfram_get_frame_bounds_by_type:nn { static } { #1 }
}

```

`\getstaticevenbounds` Even pages.

```

\NewDocumentCommand \getstaticevenbounds { s m }
{
  \IfBooleanTF { #1 }
  {
    \__flowfram_get_static_id:n { #2 }
    \__flowfram_get_static_even_bounds:n { \l__flowfram_id_int }
  }
  {
    \__flowfram_get_static_even_bounds:n { #2 }
  }
}

```

`\@sgetstaticevenbounds` Version 2.0 removed `\@sgetstaticevenbounds`.

`\@getstaticevenbounds` Get static bounds by IDN. Version 2.0 renamed `\@getstaticevenbounds`.

```

\cs_new:Nn \__flowfram_get_static_even_bounds:n
{
  \__flowfram_get_frame_even_bounds_by_type:nn { static } { #1 }
}

```

`\getflowbounds` Next flow frames:

```

\NewDocumentCommand \getflowbounds { s m }
{
  \IfBooleanTF { #1 }
  {
    \__flowfram_get_flow_id:e { #2 }
    \__flowfram_get_flow_bounds:n { \l__flowfram_id_int }
  }
  {
    \__flowfram_get_flow_bounds:n { #2 }
  }
}

```

`\@sgetflowbounds` Version 2.0 removed `\@sgetflowbounds`.

`\@getflowbounds` Get flow bounds by IDN). Version 2.0 renamed `\@getflowbounds`.

```
\cs_new:Nn \__flowfram_get_flow_bounds:n
{
  \__flowfram_get_frame_bounds_by_type:nn { flow } { #1 }
}
```

`\getflowevenbounds` Even pages:

```
\NewDocumentCommand \getflowevenbounds { s m }
{
  \IfBooleanTF { #1 }
  {
    \__flowfram_get_flow_id:e { #1 }
    \__flowfram_get_flow_even_bounds:n { \l__flowfram_id_int }
  }
  {
    \__flowfram_get_flow_even_bounds:n { #2 }
  }
}
```

`\@sgetflowevenbounds` Version 2.0 removed `\@sgetflowevenbounds`.

`\@getflowevenbounds` Get bounds by IDN. Version 2.0 renamed `\@getflowevenbounds`.

```
\cs_new:Nn \__flowfram_get_flow_even_bounds:n
{
  \__flowfram_get_frame_even_bounds_by_type:nn { flow } { #1 }
}
```

`\getdynamicbounds` Next dynamic frames:

```
\NewDocumentCommand \getdynamicbounds { s m }
{
  \IfBooleanTF { #1 }
  {
    \__flowfram_get_dynamic_id:n { #1 }
    \__flowfram_get_dynamic_bounds:n { \l__flowfram_id_int }
  }
  {
    \__flowfram_get_dynamic_bounds:n { #2 }
  }
}
```

`\@sgetdynamicbounds` Version 2.0 removed `\@sgetdynamicbounds`.

`\@getdynamicbounds` Get bounds by IDN): Version 2.0 renamed `\@getdynamicbounds`.

```
\cs_new:Nn \__flowfram_get_dynamic_bounds:n
{
  \__flowfram_get_frame_bounds_by_type:nn { dynamic } { #1 }
}
```

`\getdynamicsevenbounds` Even pages:

```
\NewDocumentCommand \getdynamicsevenbounds { s m }
{
  \IfBooleanTF { #1 }
  {
    \__flowfram_get_dynamic_id:n { #1 }
    \__flowfram_get_dynamic_even_bounds:n { \l__flowfram_id_int }
  }
  {
    \__flowfram_get_dynamic_even_bounds:n { #2 }
  }
}
```

`\@sgetdynamicsevenbounds` Version 2.0 removed `\@sgetdynamicsevenbounds`.

`\@getdynamicsevenbounds` Get bounds by IDN). Version 2.0 renamed `\@getdynamicsevenbounds`.

```
\cs_new:Nn \__flowfram_get_dynamic_even_bounds:n
{
  \__flowfram_get_frame_even_bounds_by_type:nn { dynamic } { #1 }
}
```

1.7 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.

`\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.

```
\NewDocumentCommand \checkifframeabove { s m m m m }
{
  \IfBooleanTF { #1 }
  {
    \__flowframe_check_if_frame_above_by_idl:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
}
```



```

    }
    {
      \__flowframe_check_if_frame_above_by_idn:nnnn
      { #2 } { #3 } { #4 } { #5 }
    }
  }
}

```

`\@scheckifframeabove` Check by IDL. Version 2.0 renamed `\@scheckifframeabove`.

```

\cs_new:Nn \__flowframe_check_if_frame_above_by_idl:nnnn
{
  \int_if_odd:nTF { \c@page }
  {
    \__flowframe_check_odd_if_frame_above_by_idl:nnnn
    { #1 } { #2 } { #3 } { #4 }
  }
  {
    \__flowframe_check_even_if_frame_above_by_idl:nnnn
    { #1 } { #2 } { #3 } { #4 }
  }
}

```

`\@checkifframeabove` Check by IDL. Version 2.0 renamed `\@checkifframeabove`.

```

\cs_new:Nn \__flowframe_check_if_frame_above_by_idn:nnnn
{
  \__flowframe_check_if_frame_above_by_idn:nnnn
  { \c@page } { #1 } { #2 } { #3 } { #4 }
}

```

Supply page number in first argument:

```

\cs_new:Nn \__flowframe_check_if_frame_above_by_idn:nnnnn
{
  \int_if_odd:nTF { #1 }
  {
    \__flowframe_check_odd_if_frame_above_by_idn:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
  {
    \__flowframe_check_even_if_frame_above_by_idn:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
}

```

`\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.

```

\NewDocumentCommand \oddcheckifframeabove { s m m m m }
{

```

```

\IfBooleanTF { #1 }
{
  \__flowframe_check_odd_if_frame_above_by_idl:nnnn
  { #2 } { #3 } { #4 } { #5 }
}
{
  \__flowframe_check_odd_if_frame_above_by_idn:nnnn
  { #2 } { #3 } { #4 } { #5 }
}
}

```

\@soddcheckifframeabove Check by IDL. Version 2.0 renamed \@soddcheckifframeabove.

```

\cs_new:Nn \__flowframe_check_odd_if_frame_above_by_idl:nnnn
{
  \__flowfram_get_frame_id:nn { #1 } { #2 }
  \__flowfram_get_frame_bounds_by_type:nn { #1 } { \l__flowfram_id_int }
  \dim_set_eq:NN \l__flowfram_tmpa_dim \ffareay
  \__flowfram_get_frame_id:nn { #3 } { #4 }
  \__flowfram_get_frame_bounds_by_type:nn { #3 } { \l__flowfram_id_int }
  \dim_add:Nn \ffareay { \ffareaheight }
  \dim_compare:nNnTF { \l__flowfram_tmpa_dim } > { \ffareay }
  {
    \FLFabovetrue
  }
  {
    \FLFabovefalse
  }
}

```

\@oddcheckifframeabove Check by IDN. Version 2.0 renamed \@oddcheckifframeabove.

```

\cs_new:Nn \__flowframe_check_odd_if_frame_above_by_idn:nnnn
{
  \__flowfram_get_frame_bounds_by_type:nn { #1 } { #2 }
  \dim_set_eq:NN \l__flowfram_tmpa_dim \ffareay
  \__flowfram_get_frame_bounds_by_type:nn { #3 } { #4 }
  \dim_add:Nn \ffareay { \ffareaheight }
  \dim_compare:nNnTF { \l__flowfram_tmpa_dim } > { \ffareay }
  {
    \FLFabovetrue
  }
  {
    \FLFabovefalse
  }
}

```

\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.

```
\NewDocumentCommand \checkifframebelow { s m m m m }
{
  \IfBooleanTF { #1 }
  {
    \__flowframe_check_if_frame_below_by_idl:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
  {
    \__flowframe_check_if_frame_below_by_idn:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
}
```

`\@scheckifframebelow` Check by IDL. Version 2.0 renamed `\@scheckifframebelow`.

```
\cs_new:Nn \__flowframe_check_if_frame_below_by_idl:nnnn
{
  \int_if_odd { \c@page }
  {
    \__flowframe_check_odd_if_frame_below_by_idl:nnnn
    { #1 } { #2 } { #3 } { #4 }
  }
  {
    \__flowframe_check_even_if_frame_below_by_idl:nnnn
    { #1 } { #2 } { #3 } { #4 }
  }
}
```

`\@checkifframebelow` Check by IDL. Version 2.0 renamed `\@checkifframebelow`.

```
\cs_new:Nn \__flowframe_check_if_frame_below_by_idn:nnnn
{
  \__flowframe_check_if_frame_below_by_idn:nnnnn
  { \c@page } { #1 } { #2 } { #3 } { #4 }
}
\cs_new:Nn \__flowframe_check_if_frame_below_by_idn:nnnnn
{
  \int_if_odd:nTF { #1 }
  {
    \__flowframe_check_odd_if_frame_below_by_idn:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
  {
    \__flowframe_check_even_if_frame_below_by_idn:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
}
```

`\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.

```
\NewDocumentCommand \oddcheckifframebelow { s m m m m }
{
  \IfBooleanTF { #1 }
  {
    \__flowframe_check_odd_if_frame_below_by_idl:nmmm
    { #2 } { #3 } { #4 } { #5 }
  }
  {
    \__flowframe_check_odd_if_frame_below_by_idn:nmmm
    { #2 } { #3 } { #4 } { #5 }
  }
}
```

`\@soddcheckifframebelow` Check by IDL. Version 2.0 renamed `\@soddcheckifframebelow`.

```
\cs_new:Nn \__flowframe_check_odd_if_frame_below_by_idl:nmmm
{
  \__flowfram_get_frame_id:nn { #1 } { #2 }
  \__flowfram_get_frame_bounds_by_type:nn { #1 } { \l__flowfram_id_int }
  \dim_add:Nn \ffareay { \ffareaheight }
  \dim_set_eq:NN \l__flowfram_tmpa_dim \ffareay
  \__flowfram_get_frame_id:nn { #3 } { #4 }
  \__flowfram_get_frame_bounds_by_type:nn { #3 } { \l__flowfram_id_int }
  \dim_compare:nNnTF { \l__flowfram_tmpa_dim } < { \ffareay }
  {
    \FLFbelowtrue
  }
  {
    \FLFbelowfalse
  }
}
```

`\@oddcheckifframebelow` Check by IDN. Version 2.0 renamed `\@oddcheckifframebelow`.

```
\cs_new:Nn \__flowframe_check_odd_if_frame_below_by_idn:nmmm
{
  \__flowfram_get_frame_bounds_by_type:nn { #1 } { #2 }
  \dim_add:Nn \ffareay { \ffareaheight }
  \dim_set_eq:NN \l__flowfram_tmpa_dim \ffareay
  \__flowfram_get_frame_bounds_by_type:nn { #3 } { #4 }
  \dim_compare:nNnTF { \l__flowfram_tmpa_dim } < { \ffareay }
  {
    \FLFbelowtrue
  }
  {
  }
```

```

        \FLFbelowfalse
    }
}

```

`\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.

```

\NewDocumentCommand \checkifframeleft { s m m m m m }
{
  \IfBooleanTF { #1 }
  {
    \__flowframe_check_if_frame_left_by_idl:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
  {
    \__flowframe_check_if_frame_left_by_idn:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
}

```

`\@scheckifframeleft` Check by IDL. Version 2.0 renamed `\@scheckifframeleft`.

```

\cs_new:Nn \__flowframe_check_if_frame_left_by_idl:nnnn
{
  \int_if_odd:nTF { \c@page }
  {
    \__flowframe_check_odd_if_frame_left_by_idl:nnnn
    { #1 } { #2 } { #3 } { #4 }
  }
  {
    \__flowframe_check_even_if_frame_left_by_idl:nnnn
    { #1 } { #2 } { #3 } { #4 }
  }
}

```

`\@checkifframeleft` Check by IDN. Version 2.0 renamed `\@checkifframeleft`.

```

\cs_new:Nn \__flowframe_check_if_frame_left_by_idn:nnnn
{
  \__flowframe_check_if_frame_left_by_idn:nnnnn
  { \c@page } { #1 } { #2 } { #3 } { #4 }
}
\cs_new:Nn \__flowframe_check_if_frame_left_by_idn:nnnnn
{
  \int_if_odd:nTF { #1 }
  {

```

```

    \__flowframe_check_odd_if_frame_left_by_idn:nnnn
      { #2 } { #3 } { #4 } { #5 }
  }
  {
    \__flowframe_check_even_if_frame_left_by_idn:nnnn
      { #2 } { #3 } { #4 } { #5 }
  }
}

```

`\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.

```

\NewDocumentCommand \oddcheckifframeleft { s m m m m }
{
  \IfBooleanTF { #1 }
  {
    \__flowframe_check_odd_if_frame_left_by_idl:nnnn
      { #2 } { #3 } { #4 } { #5 }
  }
  {
    \__flowframe_check_odd_if_frame_left_by_idn:nnnn
      { #2 } { #3 } { #4 } { #5 }
  }
}

```

`\@soddcheckifframeleft` Check by IDL. Version 2.0 renamed `\@soddcheckifframeleft`.

```

\cs_new:Nn \__flowframe_check_odd_if_frame_left_by_idl:nnnn
{
  \__flowfram_get_frame_id:nn { #1 } { #2 }
  \__flowfram_get_frame_bounds_by_type:nn { #1 } { \l__flowfram_id_int }
  \dim_add:Nn \ffareax { \ffareawidth }
  \dim_set_eq:NN \l__flowfram_tmpa_dim \ffareax
  \__flowfram_get_frame_id:nn { #3 } { #4 }
  \__flowfram_get_frame_bounds_by_type:nn { #3 } { \l__flowfram_id_int }
  \dim_compare:nNnTF { \l__flowfram_tmpa_dim } < { \ffareax }
  {
    \FLFlefttrue
  }
  {
    \FLFleftfalse
  }
}

```

`\@oddcheckifframeleft` Check by IDN. Version 2.0 renamed `\@oddcheckifframeleft`.

```

\cs_new:Nn \__flowframe_check_odd_if_frame_left_by_idn:nnnn
{

```

```

    \__flowfram_get_frame_bounds_by_type:nn { #1 } { #2 }
    \dim_add:Nn \ffareax { \ffareawidth }
    \dim_set_eq:NN \l__flowfram_tmpa_dim \ffareax
    \__flowfram_get_frame_bounds_by_type:nn { #3 } { #4 }
    \dim_compare:nNnTF { \l__flowfram_tmpa_dim } < { \ffareax }
    {
      \FLlefttrue
    }
    {
      \FLleftfalse
    }
  }
}

```

`\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.

```

\NewDocumentCommand \checkifframeright { s m m m m }
{
  \IfBooleanTF { #1 }
  {
    \__flowframe_check_if_frame_right_by_idl:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
  {
    \__flowframe_check_if_frame_right_by_idn:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
}

```

`\scheckifframeright` Check by IDL. Version 2.0 renamed `\@scheckifframeright`.

```

\cs_new:Nn \__flowframe_check_if_frame_right_by_idl:nnnn
{
  \int_if_odd:nTF { \c@page }
  {
    \__flowframe_check_odd_if_frame_right_by_idl:nnnn
    { #1 } { #2 } { #3 } { #4 }
  }
  {
    \__flowframe_check_even_if_frame_right_by_idl:nnnn
    { #1 } { #2 } { #3 } { #4 }
  }
}

```

`\@checkifframeright` Check by IDN. Version 2.0 renamed `\@checkifframeright`.

```

\cs_new:Nn \__flowframe_check_if_frame_right_by_idn:nnnn
{
  \__flowframe_check_if_frame_right_by_idn:nnnnn
  { \c@page } { #1 } { #2 } { #3 } { #4 }
}
\cs_new:Nn \__flowframe_check_if_frame_right_by_idn:nnnnn
{
  \int_if_odd:nTF { #1 }
  {
    \__flowframe_check_odd_if_frame_right_by_idn:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
  {
    \__flowframe_check_even_if_frame_right_by_idn:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
}
}

```

`\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.

```

\NewDocumentCommand \oddcheckifframeright { s m m m m }
{
  \IfBooleanTF { #1 }
  {
    \__flowframe_check_odd_if_frame_right_by_idl:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
  {
    \__flowframe_check_odd_if_frame_right_by_idn:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
}
}

```

`\@soddcheckifframeright` Check by IDL. Version 2.0 renamed `\@soddcheckifframeright`.

```

\cs_new:Nn \__flowframe_check_odd_if_frame_right_by_idl:nnnn
{
  \__flowfram_get_frame_id:nn { #1 } { #2 }
  \__flowfram_get_frame_bounds_by_type:nn { #1 } { \l__flowfram_id_int }
  \dim_set_eq:NN \l__flowfram_tmpa_dim \ffareax
  \__flowfram_get_frame_id:nn { #3 } { #4 }
  \__flowfram_get_frame_bounds_by_type:nn { #3 } { \l__flowfram_id_int }
  \dim_add:Nn \ffareax { \ffareawidth }
  \dim_compare:nNnTF { \l__flowfram_tmpa_dim } > { \ffareax }
  {
    \FLFrighthtrue
  }
}

```



```

    }
    {
      \FLFrighfalse
    }
  }

```

`\@oddcheckiframeright` Check by IDN. Version 2.0 renamed `\@oddcheckiframeright`.

```

\cs_new:Nn \__flowframe_check_odd_if_frame_right_by_idn:nnnn
{
  \__flowfram_get_frame_bounds_by_type:nn { #1 } { #2 }
  \dim_set_eq:NN \l__flowfram_tmpa_dim \ffareax
  \__flowfram_get_frame_bounds_by_type:nn { #3 } { #4 }
  \dim_add:Nn \ffareax { \ffareawidth }
  \dim_compare:nNnTF { \l__flowfram_tmpa_dim } > { \ffareax }
  {
    \FLFrightrue
  }
  {
    \FLFrighfalse
  }
}

```

`\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.

```

\NewDocumentCommand \evencheckifframeabove { s m m m m }
{
  \IfBooleanTF { #1 }
  {
    \__flowframe_check_even_if_frame_above_by_idl:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
  {
    \__flowframe_check_even_if_frame_above_by_idn:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
}

```

`\@sevencheckifframeabove` Check by IDL. Version 2.0 renamed `\@sevencheckifframeabove`.

```

\cs_new:Nn \__flowframe_check_even_if_frame_above_by_idl:nnnn
{
  \__flowfram_get_frame_id:nn { #1 } { #2 }
  \__flowfram_get_frame_even_bounds_by_type:nn { #1 } { \l__flowfram_id_int }
  \dim_set_eq:NN \l__flowfram_tmpa_dim \ffareay
  \__flowfram_get_frame_id:nn { #3 } { #4 }
  \__flowfram_get_frame_even_bounds_by_type:nn { #3 } { \l__flowfram_id_int }
  \dim_add:Nn \ffareay { \ffareaheight }
}

```

```

\dim_compare:nNnTF { \l__flowfram_tmpa_dim } > { \ffareay }
{
  \FLFabove>true
}
{
  \FLFabove>false
}
}

```

\@evencheckifframeabove Check by IDN. Version 2.0 renamed \@evencheckifframeabove.

```

\cs_new:Nn \__flowframe_check_even_if_frame_above_by_idn:nnnn
{
  \__flowfram_get_frame_even_bounds_by_type:nn { #1 } { #2 }
  \dim_set_eq:NN \l__flowfram_tmpa_dim \ffareay
  \__flowfram_get_frame_even_bounds_by_type:nn { #3 } { #4 }
  \dim_add:Nn \ffareay { \ffareaheight }
  \dim_compare:nNnTF { \l__flowfram_tmpa_dim } > { \ffareay }
  {
    \FLFabove>true
  }
  {
    \FLFabove>false
  }
}

```

\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.

```

\NewDocumentCommand \evencheckifframebelow { s m m m m }
{
  \IfBooleanTF { #1 }
  {
    \__flowframe_check_even_if_frame_below_by_idl:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
  {
    \__flowframe_check_even_if_frame_below_by_idn:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
}

```

\@sevencheckifframebelow Check by IDL. Version 2.0 renamed \@sevencheckifframebelow.

```

\cs_new:Nn \__flowframe_check_even_if_frame_below_by_idl:nnnn
{
  \__flowfram_get_frame_id:nn { #1 } { #2 }
  \__flowfram_get_frame_even_bounds_by_type:nn { #1 } { \l__flowfram_id_int }
  \dim_add:Nn \ffareay { \ffareaheight }
}

```

```

\dim_set_eq:NN \l__flowfram_tmpa_dim \ffareay
\__flowfram_get_frame_id:nn { #3 } { #4 }
\__flowfram_get_frame_even_bounds_by_type:nn { #3 } { \l__flowfram_id_int }
\dim_compare:nNnTF { \l__flowfram_tmpa_dim } < { \ffareay }
{
  \FLFbelowtrue
}
{
  \FLFbelowfalse
}
}

```

\@evencheckifframebelow Check by IDN. Version 2.0 renamed \@evencheckifframebelow.

```

\cs_new:Nn \__flowframe_check_even_if_frame_below_by_idn:nnnn
{
  \__flowfram_get_frame_even_bounds_by_type:nn { #1 } { #2 }
  \dim_add:Nn \ffareay { \ffareaheight }
  \dim_set_eq:NN \l__flowfram_tmpa_dim \ffareay
  \__flowfram_get_frame_even_bounds_by_type:nn { #3 } { #4 }
  \dim_compare:nNnTF { \l__flowfram_tmpa_dim } < { \ffareay }
  {
    \FLFbelowtrue
  }
  {
    \FLFbelowfalse
  }
}

```

\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.

```

\NewDocumentCommand \evencheckifframeleft { s m m m m }
{
  \IfBooleanTF { #1 }
  {
    \__flowframe_check_even_if_frame_left_by_idl:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
  {
    \__flowframe_check_even_if_frame_left_by_idn:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
}

```

\@sevencheckifframeleft Check by IDL. Version 2.0 renamed \@sevencheckifframeleft.

```

\cs_new:Nn \__flowframe_check_even_if_frame_left_by_idl:nnnn
{

```

```

    \__flowfram_get_frame_id:nn { #1 } { #2 }
    \__flowfram_get_frame_even_bounds_by_type:nn { #1 } { \l__flowfram_id_int }
    \dim_add:Nn \ffareax { \ffareawidth }
    \dim_set_eq:NN \l__flowfram_tmpa_dim \ffareax
    \__flowfram_get_frame_id:nn { #3 } { #4 }
    \__flowfram_get_frame_even_bounds_by_type:nn { #3 } { \l__flowfram_id_int }
    \dim_compare:nNnTF { \l__flowfram_tmpa_dim } < { \ffareax }
    {
      \FLFlefttrue
    }
    {
      \FLFleftfalse
    }
  }
}

```

`\@evencheckifframeleft` Check by IDN. Version 2.0 renamed `\@evencheckifframeleft`.

```

\cs_new:Nn \__flowframe_check_even_if_frame_left_by_idn:nmmm
{
  \__flowfram_get_frame_even_bounds_by_type:nn { #1 } { #2 }
  \dim_add:Nn \ffareax { \ffareawidth }
  \dim_set_eq:NN \l__flowfram_tmpa_dim \ffareax
  \__flowfram_get_frame_even_bounds_by_type:nn { #3 } { #4 }
  \dim_compare:nNnTF { \l__flowfram_tmpa_dim } < { \ffareax }
  {
    \FLFlefttrue
  }
  {
    \FLFleftfalse
  }
}

```

`\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.

```

\NewDocumentCommand \evencheckifframeright { s m m m m }
{
  \IfBooleanTF { #1 }
  {
    \__flowframe_check_even_if_frame_right_by_idl:nmmm
    { #2 } { #3 } { #4 } { #5 }
  }
  {
    \__flowframe_check_even_if_frame_right_by_idn:nmmm
    { #2 } { #3 } { #4 } { #5 }
  }
}

```

`\@sevencheckiframeright` Check by IDL. Version 2.0 renamed `\@sevencheckiframeright`.

```
\cs_new:Nn \__flowframe_check_even_if_frame_right_by_idl:nnnn
{
  \__flowfram_get_frame_id:nn { #1 } { #2 }
  \__flowfram_get_frame_even_bounds_by_type:nn { #1 } { \l__flowfram_id_int }
  \dim_set_eq:NN \l__flowfram_tmpa_dim \ffareax
  \__flowfram_get_frame_id:nn { #3 } { #4 }
  \__flowfram_get_frame_even_bounds_by_type:nn { #3 } { \l__flowfram_id_int }
  \dim_add:Nn \ffareax { \ffareawidth }
  \dim_compare:nNnTF { \l__flowfram_tmpa_dim } > { \ffareax }
  {
    \FLFrighttrue
  }
  {
    \FLFrightfalse
  }
}
```

`\@evencheckiframeright` Check by IDN. Version 2.0 renamed `\@evencheckiframeright`.

```
\cs_new:Nn \__flowframe_check_even_if_frame_right_by_idn:nnnn
{
  \__flowfram_get_frame_even_bounds_by_type:nn { #1 } { #2 }
  \dim_set_eq:NN \l__flowfram_tmpa_dim \ffareax
  \__flowfram_get_frame_even_bounds_by_type:nn { #3 } { #4 }
  \dim_add:Nn \ffareax { \ffareawidth }
  \dim_compare:nNnTF { \l__flowfram_tmpa_dim } > { \ffareax }
  {
    \FLFrighttrue
  }
  {
    \FLFrightfalse
  }
}
```

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{<type1>}{<id1>}{<type2>}{<id2>}`

`\relativeframelocation`

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.

```
\NewDocumentCommand \relativeframelocation { s m m m m }
{
  \IfBooleanTF { #1 }
  {
    \__flowfram_relative_frame_location_by_idl:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
  {
    \__flowfram_relative_frame_location_by_idn:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
}
```

`\@srelativeframelocation` Check by IDL. Version 2.0 renamed `\@srelativeframelocation`.

```
\cs_new:Nn \__flowfram_relative_frame_location_by_idl:nnnn
{
  \__flowfram_get_frame_id:nn { #1 } { #2 }
  \int_set_eq:NN \l__flowfram_tmpa_int \l__flowfram_id_int
  \__flowfram_get_frame_id:nn { #3 } { #4 }
  \int_set_eq:NN \l__flowfram_tmpb_int \l__flowfram_id_int
  \__flowfram_relative_frame_location_by_idn:nVnV
  { #1 } \l__flowfram_tmpa_int
  { #3 } \l__flowfram_tmpb_int
}
```

`\@relativeframelocation` Check by IDN. Version 2.0 renamed `\@relativeframelocation`.

```
\cs_new:Nn \__flowfram_relative_frame_location_by_idn:nnnn
{
  \__flowfram_relative_frame_location_by_idn:nnnn
  { \c@page } { #1 } { #2 } { #3 } { #4 }
}
```

```

}
\cs_generate_variant:Nn \__flowfram_relative_frame_location_by_idn:nnnn
{ nVnV }

```

For a specific page:

```

\cs_new:Nn \__flowfram_relative_frame_location_by_idn:nnnnn
{
  \__flowframe_check_if_frame_above_by_idn:nnnnn
  { #1 } { #2 } { #3 } { #4 } { #5 }
  \__flowframe_check_if_frame_below_by_idn:nnnnn
  { #1 } { #2 } { #3 } { #4 } { #5 }
  \__flowframe_check_if_frame_left_by_idn:nnnnn
  { #1 } { #2 } { #3 } { #4 } { #5 }
  \__flowframe_check_if_frame_right_by_idn:nnnnn
  { #1 } { #2 } { #3 } { #4 } { #5 }
  \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
}

```

```
\SaveRelativeFrameLocation{<label>}{<type1>}{<ID1>}{<type2>}{<ID2>}
```

\SaveRelativeFrameLocation

Save in aux file for next run.

```
\NewDocumentCommand \SaveRelativeFrameLocation { s m m m m m }
{
  \IfBooleanTF { #1 }
  {
    \__flowfram_get_frame_id:nn { #3 } { #4 }
    \int_set_eq:NN \l__flowfram_tmpa_int \l__flowfram_id_int
    \__flowfram_get_frame_id:nn { #5 } { #6 }
    \int_set_eq:NN \l__flowfram_tmpb_int \l__flowfram_id_int
  }
  {
    \int_set:Nn \l__flowfram_tmpa_int { #4 }
    \int_set:Nn \l__flowfram_tmpb_int { #6 }
  }
}
```

Obtain the dimensions of the first frame:

```
\flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_x_dim
  { #3 } { posx } { \l__flowfram_tmpa_int }
\flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_evenx_dim
  { #3 } { evenx } { \l__flowfram_tmpa_int }
\flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_y_dim
  { #3 } { posy } { \l__flowfram_tmpa_int }
\flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_eveny_dim
  { #3 } { eveny } { \l__flowfram_tmpa_int }
\flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_width_dim
  { #3 } { width } { \l__flowfram_tmpa_int }
\flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_height_dim
  { #3 } { height } { \l__flowfram_tmpa_int }
```

Store the first frame values:

```
\tl_set:Ne \l__flowfram_x_tl
  { \dim_use:N \l__flowfram_x_dim }
\tl_set:Ne \l__flowfram_y_tl
  { \dim_use:N \l__flowfram_y_dim }
\tl_set:Ne \l__flowfram_evenx_tl
  { \dim_use:N \l__flowfram_evenx_dim }
\tl_set:Ne \l__flowfram_eveny_tl
  { \dim_use:N \l__flowfram_eveny_dim }
\tl_set:Ne \l__flowfram_width_tl
  { \dim_use:N \l__flowfram_width_dim }
```



```

\tl_set:Ne \l__flowfram_height_tl
  { \dim_use:N \l__flowfram_height_dim }

```

Obtain the dimensions of the first frame:

```

\flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_x_dim
  { #5 } { posx } { \l__flowfram_tmpb_int }
\flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_evenx_dim
  { #5 } { evenx } { \l__flowfram_tmpb_int }
\flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_y_dim
  { #5 } { posy } { \l__flowfram_tmpb_int }
\flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_eventy_dim
  { #5 } { eventy } { \l__flowfram_tmpb_int }
\flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_width_dim
  { #5 } { width } { \l__flowfram_tmpb_int }
\flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_height_dim
  { #5 } { height } { \l__flowfram_tmpb_int }

```

Write the information to the aux file (the actual page value is required not \thepage):

```

\protected@write \@auxout
{ \let \flowfram@thepagenumber \relax }
{
  \string \@flowfram@saverelativeloc
  { #2 }
  { \flowfram@thepagenumber }
  {
    { \l__flowfram_x_tl }
    { \l__flowfram_y_tl }
    { \l__flowfram_evenx_tl }
    { \l__flowfram_eventy_tl }
    { \l__flowfram_width_tl }
    { \l__flowfram_height_tl }
  }
  {
    { \dim_use:N \l__flowfram_x_dim }
    { \dim_use:N \l__flowfram_y_dim }
    { \dim_use:N \l__flowfram_evenx_dim }
    { \dim_use:N \l__flowfram_eventy_dim }
    { \dim_use:N \l__flowfram_width_dim }
    { \dim_use:N \l__flowfram_height_dim }
  }
}
\tl_if_exist:cF { g__flowfram_saved_rel_loc_ #2 _tl }
{
  \@flowframe@warn@rerun
}

```

```

}
}

```

Extract the dimension arguments:

```

\cs_new:Nn \__flowfram_fetch_odd_dimensions:nnnnnn
{
  \dim_set:Nn \l__flowfram_x_dim { #1 }
  \dim_set:Nn \l__flowfram_y_dim { #2 }
  \dim_set:Nn \l__flowfram_width_dim { #5 }
  \dim_set:Nn \l__flowfram_height_dim { #6 }
}
\cs_new:Nn \__flowfram_fetch_even_dimensions:nnnnnn
{
  \dim_set:Nn \l__flowfram_x_dim { #3 }
  \dim_set:Nn \l__flowfram_y_dim { #4 }
  \dim_set:Nn \l__flowfram_width_dim { #5 }
  \dim_set:Nn \l__flowfram_height_dim { #6 }
}

```

```

\@flowfram@saverelativeloc{<label>}{<page>}{<frame 1
dimensions>}{<frame 2 dimensions>}

```

```

\@flowfram@saverelativeloc

```

```

\newcommand\@flowfram@saverelativeloc [ 4 ]
{
  \tl_if_exist:cTF { g__flowfram_saved_rel_loc_ #1 _tl }
  {
    \msg_warning:nnn { flowfram } { relloc-label-already-defined }
    { #1 }
  }
  {
    \tl_const:cn { c__flowfram_saved_data_rel_loc_ #1 _tl }
    { { #2 } { #3 } { #4 } }
    \tl_new:c { g__flowfram_saved_rel_loc_ #1 _tl }
    \int_if_odd:nTF { #2 }
    {
      \__flowfram_fetch_odd_dimensions:nnnnnn #3
    }
    {
      \__flowfram_fetch_even_dimensions:nnnnnn #3
    }
    \dim_set_eq:NN \ffareax \l__flowfram_x_dim
    \dim_set_eq:NN \ffareay \l__flowfram_y_dim
    \dim_set_eq:NN \ffareawidth \l__flowfram_width_dim
    \dim_set_eq:NN \ffareaheight \l__flowfram_height_dim
    \int_if_odd:nTF { #2 }
    {
      \__flowfram_fetch_odd_dimensions:nnnnnn #4
    }
  }
}

```

```

{
  \__flowfram_fetch_even_dimensions:nnnnnn #4
}
\FLFabovefalse
\FLFbelowfalse
\FLFleftfalse
\FLFrightfalse
\dim_compare:nNnTF
  { \ffareax } > { \l__flowfram_x_dim + \l__flowfram_width_dim }
  {
    \FLFrighttrue
  }
  {
    \dim_compare:nNnT
      { \ffareax + \ffareawidth } < { \l__flowfram_x_dim }
      {
        \FLFlefttrue
      }
    }
  }
\dim_compare:nNnTF
  { \ffareay } > { \l__flowfram_y_dim + \l__flowfram_height_dim }
  {
    \FLFabovetrue
  }
  {
    \dim_compare:nNnT
      { \ffareay + \ffareaheight } < { \l__flowfram_y_dim }
      {
        \FLFbelowtrue
      }
    }
  }
\ifFLFabove
  \ifFLFleft
    \tl_gset:cn { g__flowfram_saved_rel_loc_ #1 _tl }
    { \FFaboveleft }
  \else
    \ifFLFright
      \tl_gset:cn { g__flowfram_saved_rel_loc_ #1 _tl }
      { \FFaboveright }
    \else
      \tl_gset:cn { g__flowfram_saved_rel_loc_ #1 _tl }
      { \FFabove }
    \fi
  \fi
\else
  \ifFLFbelow
    \ifFLFleft
      \tl_gset:cn { g__flowfram_saved_rel_loc_ #1 _tl }
      { \FFbelowleft }
    \else

```

```

        \ifFLFright
        \tl_gset:cn { g__flowfram_saved_rel_loc_ #1 _tl }
        { \FFbelowright }
        \else
        \tl_gset:cn { g__flowfram_saved_rel_loc_ #1 _tl }
        { \FFbelow }
        \fi
    \fi
\else
\ifFLFleft
\tl_gset:cn { g__flowfram_saved_rel_loc_ #1 _tl }
{ \FFleft }
\else
\ifFLFright
\tl_gset:cn { g__flowfram_saved_rel_loc_ #1 _tl }
{ \FFright }
\else
\tl_gset:cn { g__flowfram_saved_rel_loc_ #1 _tl }
{ \FFoverlap }
\fi
\fi
\fi
\fi
\bool_new:c { g__flowfram_saved_rel_loc_ #1 _above_bool }
\bool_new:c { g__flowfram_saved_rel_loc_ #1 _below_bool }
\bool_new:c { g__flowfram_saved_rel_loc_ #1 _left_bool }
\bool_new:c { g__flowfram_saved_rel_loc_ #1 _right_bool }
\ifFLFabove
\bool_gset_true:c { g__flowfram_saved_rel_loc_ #1 _above_bool }
\fi
\ifFLFbelow
\bool_gset_true:c { g__flowfram_saved_rel_loc_ #1 _below_bool }
\fi
\ifFLFleft
\bool_gset_true:c { g__flowfram_saved_rel_loc_ #1 _left_bool }
\fi
\ifFLFright
\bool_gset_true:c { g__flowfram_saved_rel_loc_ #1 _right_bool }
\fi
}
}
\AddToHook { begindocument / end }
{
\cs_gset_eq:NN \@flowfram@saverelativeloc \@flowfram@checkrelativeloc
}
\@flowfram@checkrelativeloc
\newcommand\@flowfram@checkrelativeloc [ 4 ]
{
\tl_if_exist:cT { c__flowfram_saved_data_rel_loc_ #1 _tl }

```

```

    {
      \tl_if_eq:cnF { c__flowfram_saved_data_rel_loc_ #1 _tl }
        { { #2 } { #3 } { #4 } }
      {
        \@flowframe@warn@rerun
      }
    }
  }
}

```

\@flowframe@warn@rerun

```

\newcommand \@flowframe@warn@rerun
{
  \AtEndDocument
  {
    \msg_warning:nn { flowfram } { data-changed-rerun }
  }
  \global \let \@flowframe@warn@rerun \relax
}

```

\RefSavedRelativeLocation

```

\NewDocumentCommand \RefSavedRelativeLocation { m }
{
  \tl_if_exist:cTF { g__flowfram_saved_rel_loc_ #1 _tl }
  {
    \tl_use:c { g__flowfram_saved_rel_loc_ #1 _tl }
  }
  {
    ??
    \msg_warning:nnn { flowfram } { relloc-label-not-defined }
      { #1 }
  }
}

```

\IfSavedRelativeLocationEq{<label>}{<value>}{<true>}
{<false>}

\IfSavedRelativeLocationEq

```

\NewDocumentCommand \IfSavedRelativeLocationEq { m m m m }
{
  \tl_if_exist:cTF { g__flowfram_saved_rel_loc_ #1 _tl }
  {
    \tl_if_eq:cnTF { g__flowfram_saved_rel_loc_ #1 _tl } { #2 }
      { #3 } { #4 }
  }
  { #4 }
}

```

\IfSavedRelativeLocationAbove

```
\IfSavedRelativeLocationAbove{<label>}{<true>}{<false>}
```

```
\newcommand \IfSavedRelativeLocationAbove [ 3 ]
{
  \bool_if_exist:cTF { g__flowfram_saved_rel_loc_ #1 _above_bool }
  {
    \bool_if:cTF { g__flowfram_saved_rel_loc_ #1 _above_bool }
    { #2 } { #3 }
  }
  { #3 }
}
```

\IfSavedRelativeLocationBelow

```
\IfSavedRelativeLocationBelow{<label>}{<true>}{<false>}
```

```
\newcommand \IfSavedRelativeLocationBelow [ 3 ]
{
  \bool_if_exist:cTF { g__flowfram_saved_rel_loc_ #1 _below_bool }
  {
    \bool_if:cTF { g__flowfram_saved_rel_loc_ #1 _below_bool }
    { #2 } { #3 }
  }
  { #3 }
}
```

\IfSavedRelativeLocationLeft

```
\IfSavedRelativeLocationLeft{<label>}{<true>}{<false>}
```

```
\newcommand \IfSavedRelativeLocationLeft [ 3 ]
{
  \bool_if_exist:cTF { g__flowfram_saved_rel_loc_ #1 _left_bool }
  {
    \bool_if:cTF { g__flowfram_saved_rel_loc_ #1 _left_bool }
    { #2 } { #3 }
  }
  { #3 }
}
```

\IfSavedRelativeLocationRight

```
\IfSavedRelativeLocationRight{<label>}{<true>}{<false>}
```

```
\newcommand \IfSavedRelativeLocationRight [ 3 ]
{
  \bool_if_exist:cTF { g__flowfram_saved_rel_loc_ #1 _right_bool }
  {
    \bool_if:cTF { g__flowfram_saved_rel_loc_ #1 _right_bool }
    { #2 } { #3 }
  }
}
```

```

}
{ #3 }
}

```

Short cut commands for frames of the same type.

```

\reldynamicloc \reldynamicloc{<id1>}{<id2>}
\NewDocumentCommand \reldynamicloc { s m m }
{
  \IfBooleanTF { #1 }
  {
    \__flowfram_relative_frame_location_by_idl:nnnn
    { dynamic } { #2 } { dynamic } { #3 }
  }
  {
    \__flowfram_relative_frame_location_by_idn:nnnn
    { dynamic } { #2 } { dynamic } { #3 }
  }
}

```

\@sreldynamicloc Version 2.0 removed \@sreldynamicloc.

\@reldynamicloc Version 2.0 removed \@reldynamicloc.

```

\relstaticloc \relstaticloc{<id1>}{<id2>}
\NewDocumentCommand \relstaticloc { s m m }
{
  \IfBooleanTF { #1 }
  {
    \__flowfram_relative_frame_location_by_idl:nnnn
    { static } { #2 } { static } { #3 }
  }
  {
    \__flowfram_relative_frame_location_by_idn:nnnn
    { static } { #2 } { static } { #3 }
  }
}

```

\@srelstaticloc Version 2.0 removed \@srelstaticloc.

\@relstaticloc Version 2.0 removed \@relstaticloc.

```

\relflowloc \relflowloc{<id1>}{<id2>}
\NewDocumentCommand \relflowloc { s m m }
{
  \IfBooleanTF { #1 }
  {
    \__flowfram_relative_frame_location_by_idl:nnnn
    { flow } { #2 } { flow } { #3 }
  }
}

```

```

    {
      \__flowfram_relative_frame_location_by_idn:nnnn
      { flow } { #2 } { flow } { #3 }
    }
  }
}

```

`\@sreflowloc` Version 2.0 removed `\@sreflowloc`.

`\@reflowloc` Version 2.0 removed `\@reflowloc`.

1.8 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.

```

\NewDocumentCommand \setinitialframe { m }
{
  \int_gset:Nn \c@thisframe { #1 }
  \global \usedframebreaktrue
  \__flowfram_set_column:n { #1 }
}

```

`\if@setfr@mes` Scratch conditional. TODO remove ??

```

\newif\if@setfr@mes
\@setfr@mesfalse

```

`\setframes` Set the initial frame.

```

\NewDocumentCommand \setframes { }
{
  \int_if_zero:nT { \c@thisframe }
  {
    \msg_warning:nn { flowfram } { no-page1-col }
    \int_gset_eq:NN \g__flowfram_next_frame_int \c_one_int
    \int_gset_eq:NN \g__flowfram_current_page_int \c_one_int
    \__flowfram_get_next_column:N \g__flowfram_next_frame_int
  }
}

```

Shipout pages without flow frames.

```

\int_gdecr:N \g__flowfram_current_page_int
\bool_while_do:nn
{ \int_compare_p:nNn { \g__flowfram_current_page_int } > { \c_zero_int } }
{
  \int_gdecr:N \g__flowfram_current_page_int
  \setbox \@outputbox
  \vbox_to_ht:nn { \typeblockheight }
  {
    \__flowfram_do_all_frames:
  }
  \@outputpage
}
\int_gset_eq:NN \c@thisframe \g__flowfram_next_frame_int

```



```

}
\__flowfram_set_column:n { \c@thisframe }
\@flowfram@update@col@count { \c@thisframe }
\@setfr@mesttrue
\flowfram_set_tl_to_frame_tl:Nnnn
  \l__flowfram_textcolor_tl
  { flow } { textcolor } { \c@thisframe }
\__flowfram_set_text_color:
}

```

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

```

\NewDocumentCommand \emulatetwocolumn { 0{} }
{
  \finishthispage
  \setallflowframes { pages = none }
  \settoheight \l__flowfram_sdf_height_dim { #1 }
  \settodepth \l__flowfram_tmpa_dim { #1 }
  \addtolength \l__flowfram_sdf_height_dim
    { \l__flowfram_tmpa_dim }
  \dim_compare:nNnTF
    { \l__flowfram_sdf_height_dim } > { \c_zero_dim }
  {
    \__flowfram_two_column_with_top_in_area:nnnnnnn
      { \int_eval:n { \g__flowfram_pagecounter_tl } }
      { static }
      { \dim_use:N \l__flowfram_sdf_height_dim }
      { \typeblockwidth } { \typeblockheight } { Opt } { Opt }
    \int_gset:Nn \c@thisframe { \c@maxflow - \c_one_int }
    \exp_args:Ne \__flowfram_two_columns:n
      { > \int_eval:n { \g__flowfram_pagecounter_tl } }
    \setstaticcontents { \c@maxstatic } { #1 }
  }
  {
    \__flowfram_two_columns:n { all }
    \int_gset:Nn \c@thisframe { \c@maxflow - \c_one_int }
  }
  \__flowfram_set_column:n { \c@thisframe }
  \relax
}

```

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

```

\NewDocumentCommand \emulateonecolumn { 0{} }
{
  \finishthispage
  \setallflowframes { pages = none }
  \settoheight \l__flowfram_sdf_height_dim { #1 }
  \settodepth \l__flowfram_tmpa_dim { #1 }

```

```

\addtolength \l__flowfram_sdf_height_dim { \l__flowfram_tmpa_dim }
\dim_compare:nNnTF
  { \l__flowfram_sdf_height_dim } > { \c_zero_dim }
  {
    \__flowfram_one_column_with_top_in_area:nnnnnn
      { \int_eval:n { \g__flowfram_pagecounter_tl } }
      { static }
      { \dim_use:N \l__flowfram_sdf_height_dim }
      { \typeblockwidth } { \typeblockheight } { Opt } { Opt }
    \int_gset:Nn \c@thisframe { \c@maxflow - \c_one_int }
    \__flowfram_new_flow:nnnnn
      { > \int_eval:n { \g__flowfram_pagecounter_tl } }
      { \typeblockwidth } { \typeblockheight } { Opt } { Opt }
    \setstaticcontents { \c@maxstatic } { #1 }
  }
  {
    \__flowfram_one_column:n { all }
    \int_gset:Nn \c@thisframe { \c@maxflow - \c_one_int }
  }
  \__flowfram_set_column:n { \c@thisframe }
  \relax
}

```

If no flow frames have been defined, create a single flow frame the size of the typeblock, and initialise the frames.

```

\AtBeginDocument
{
  \__flowfram_doc_init:
}
\cs_new:Nn \__flowfram_doc_init:
{

```

Save original definition of \@starttoc in case it's changed by another package loaded after flowfram:

```

\cs_set_eq:NN \__flowfram_org_starttoc: \@starttoc

```

Initialise absolute page.

```

\int_gset_eq:NN \c@absolute page \c_one_int
\int_if_zero:nT { \c@maxflow }
{
  \if@twocolumn
    \msg_warning:nnn { flowfram } { no-cols } { 2 }
    \__flowfram_two_columns:n { all }
  \else
    \msg_warning:nnn { flowfram } { no-cols } { 1 }
    \__flowfram_one_column:n { all }
  \fi
}
\setframes
\cs_set_eq:NN \__flowfram_only_document_env:Nn \use_ii:nn

```

```

\cs_set_eq:NN
  \__flowfram_only_preamble:Nn
  \__flowfram_forbidden:Nn
\cs_set_eq:NN \onecolumn \__flowfram_doc_onecolumn:
\RenewDocumentCommand \twocolumn { o }
{
  \__flowfram_doc_twocolumn:n { ##1 }
}
\cs_set_eq:NN
  \__flowfram_write_html_opts:nnn
  \__flowfram_doc_write_html_opts:nnn
}

```

The behaviour of `\onecolumn` in the document environment:

```

\cs_new:Nn \__flowfram_doc_onecolumn:
{
  \bool_if:NTF \g__flowfram_ignore_column_changes_bool
  {
    \bool_if:NT \g__flowfram_clearpage_column_changes_bool
    {
      \clearpage
    }
  }
  {
    \flowfram_set_one_col:
  }
}

```

The behaviour of `\twocolumn` in the document environment:

```

\cs_new:Nn \__flowfram_doc_twocolumn:n
{
  \IfValueTF { #1 }
  {
    \bool_if:NTF \g__flowfram_ignore_column_changes_bool
    {
      \bool_if:NT \g__flowfram_clearpage_column_changes_bool
      {
        \clearpage
      }
      #1
    }
    {
      \flowfram_set_two_col_header:n { #1 }
    }
  }
  {
    \bool_if:NTF \g__flowfram_ignore_column_changes_bool
    {
      \bool_if:NT \g__flowfram_clearpage_column_changes_bool
      {
        \clearpage
      }
    }
  }
}

```

```

    }
  }
  {
    \flowfram_set_two_col:
  }
}
}

```

If a class or package is being used that automatically inserts `\onecolumn` or `\twocolumn`, identify which frames should be switched on or off.

```

\tl_new:N \g__flowfram_one_col_id_tl
\tl_new:N \g__flowfram_two_col_i_id_tl
\tl_new:N \g__flowfram_two_col_ii_id_tl
\tl_new:N \g__flowfram_two_col_header_id_tl
\tl_new:N \g__flowfram_two_col_header_type_tl
\tl_new:N \g__flowfram_headed_two_col_i_id_tl
\tl_new:N \g__flowfram_headed_two_col_ii_id_tl

```

`\SetOneColumnFrame` Identify which column should be used if `\onecolumn` is encountered:

```

\NewDocumentCommand \SetOneColumnFrame { s m }
{
  \tl_if_empty:nTF { #2 }
  {
    \tl_gclear:N \g__flowfram_one_col_id_tl
  }
  {
    \IfBooleanTF { #1 }
    {
      \__flowfram_get_flow_id:n { #2 }
      \tl_gset:Ne \g__flowfram_one_col_id_tl
        { \int_use:N \l__flowfram_id_int }
    }
    {
      \bool_lazy_or:nnTF
      {
        \int_compare_p:nNn { #2 } < { \c_one_int }
      }
      {
        \int_compare_p:nNn { #2 } > { \c_maxflow }
      }
      {
        \msg_error:nnnn { flowfram } { idn-undefined }
          { flow } { #2 }
      }
    }
    {
      \tl_gset:Ne \g__flowfram_one_col_id_tl
        { \int_eval:n { #2 } }
    }
  }
}

```

```

    }
}

```

`\SetTwoColumnFrames` Identify which column should be used if `\twocolumn` is encountered. Syntax:
`[\header-type] [\header-id] {\col-1} [\header-col1] {\col-2} [\header-col-2]`

```

\NewDocumentCommand \SetTwoColumnFrames { s o o m o m o }

```

```

{
  \IfValueTF { #2 }
  {

```

Header frame of given type.

```

    \cs_if_exist:cTF { __flowfram_get_ #2 _id:n }
    {
      \tl_gset:Ne \g__flowfram_two_col_header_type_tl { #2 }
      \IfValueT { #3 }
      {
        \tl_if_empty:nTF { #3 }
        {
          \tl_gclear:N \g__flowfram_two_col_header_id_tl
        }
        {
          \IfBooleanTF { #1 }
          {
            \use:c { __flowfram_get_ #2 _id:n } { #3 }
            \tl_gset:Ne \g__flowfram_two_col_header_id_tl
              { \int_use:N \l__flowfram_id_int }
          }
          {
            \bool_lazy_or:nnTF
            {
              \int_compare_p:nNn { #3 } < { \c_one_int }
            }
            {
              \int_compare_p:nNn { #3 } > { \int_use:c { c@max #2 } }
            }
            {
              \msg_error:nnnn { flowfram } { idn-undefined }
                { #2 } { #3 }
            }
            {
              \tl_gset:Ne \g__flowfram_two_col_header_id_tl
                { \int_eval:n { #3 } }
            }
          }
        }
      }
    }
  }
  {
    \msg_error:nnn { flowfram } { invalid-frame-type } { #2 }
  }
}

```

}

First flow frame.

```
\tl_if_empty:nTF { #4 }
{
  \tl_gclear:N \g__flowfram_two_col_i_id_tl
}
{
  \IfBooleanTF { #1 }
  {
    \__flowfram_get_flow_id:n { #4 }
    \tl_gset:Ne \g__flowfram_two_col_i_id_tl
      { \int_use:N \l__flowfram_id_int }
  }
  {
    \bool_lazy_or:nnTF
    {
      \int_compare_p:nNn { #4 } < { \c_one_int }
    }
    {
      \int_compare_p:nNn { #4 } > { \c@maxflow }
    }
    {
      \msg_error:nnnn { flowfram } { idn-undefined }
      { flow } { #4 }
    }
  }
  {
    \tl_gset:Ne \g__flowfram_two_col_i_id_tl
      { \int_eval:n { #4 } }
  }
}
}
```

First flow frame on headed page.

```
\IfValueT { #5 }
{
  \tl_if_empty:nTF { #5 }
  {
    \tl_gclear:N \g__flowfram_headed_two_col_i_id_tl
  }
  {
    \IfBooleanTF { #1 }
    {
      \__flowfram_get_flow_id:n { #5 }
      \tl_gset:Ne \g__flowfram_headed_two_col_i_id_tl
        { \int_use:N \l__flowfram_id_int }
    }
    {
      \bool_lazy_or:nnTF
      {
        \int_compare_p:nNn { #5 } < { \c_one_int }
      }
    }
  }
}
```

```

    }
    {
      \int_compare_p:nNn { #5 } > { \c@maxflow }
    }
    {
      \msg_error:nnnn { flowfram } { idn-undefined }
      { flow } { #5 }
    }
    {
      \tl_gset:Ne \g__flowfram_headed_two_col_ii_id_tl
      { \int_eval:n { #5 } }
    }
  }
}

```

Second flow frame.

```

\tl_if_empty:nTF { #6 }
{
  \tl_gclear:N \g__flowfram_two_col_ii_id_tl
}
{
  \IfBooleanTF { #1 }
  {
    \__flowfram_get_flow_id:n { #6 }
    \tl_gset:Ne \g__flowfram_two_col_ii_id_tl
    { \int_use:N \l__flowfram_id_int }
  }
  {
    \bool_lazy_or:nnTF
    {
      \int_compare_p:nNn { #6 } < { \c_one_int }
    }
    {
      \int_compare_p:nNn { #6 } > { \c@maxflow }
    }
    {
      \msg_error:nnnn { flowfram } { idn-undefined }
      { flow } { #6 }
    }
    {
      \tl_gset:Ne \g__flowfram_two_col_ii_id_tl
      { \int_eval:n { #6 } }
    }
  }
}
}

```

Second flow frame on headed page.

```

\IfValueT { #7 }
{
  \tl_if_empty:nTF { #7 }

```



```

\l_gput_right:Nn \g__flowfram_output_adjust_frames_tl
{
  \flowsetpagelist { ##1 }
  {
    pages =
    { > \int_eval:n { \g__flowfram_pagecounter_tl } }
  }
}
}
{
\bool_if:NF \g__flowfram_not_this_frame_bool
{
  \l_gput_right:Nn \g__flowfram_output_adjust_frames_tl
  {
    \flowsetpagelist { ##1 } { pages = none }
  }
}
}
}
}
\clearpage
}

```

Two columns with no header.

```

\cs_new:Nn \flowfram_set_two_col:
{
  \bool_lazy_or:nnTF
  {
    \tl_if_empty_p:N \g__flowfram_two_col_i_id_tl
  }
  {
    \tl_if_empty_p:N \g__flowfram_two_col_ii_id_tl
  }
  {
    \msg_warning:nnn { flowfram } { misplaced-col-cmd } { \twocolumn }
  }
  {
    \int_step_inline:nn { \c@maxflow }
    {

```

Is this frame currently on or off on this page?

```

  \__flowfram_check_if_this_page:nn
  { \g__flowfram_pagecounter_tl } { ##1 }
\bool_lazy_or:nnTF
{
  \int_compare_p:nNn
  { ##1 } = { \g__flowfram_two_col_i_id_tl }
}
{
  \int_compare_p:nNn

```

```

        { ##1 } = { \g__flowfram_two_col_ii_id_tl }
    }
    {
    \bool_if:NT \g__flowfram_not_this_frame_bool
    {
    \tl_gput_right:Nn \g__flowfram_output_adjust_frames_tl
    {
    \flowsetpagelist { ##1 }
    {
    pages =
    { > \int_eval:n { \g__flowfram_pagecounter_tl } }
    }
    }
    }
    }
    {
    \bool_if:NF \g__flowfram_not_this_frame_bool
    {
    \tl_gput_right:Nn \g__flowfram_output_adjust_frames_tl
    {
    \flowsetpagelist { ##1 } { pages = none }
    }
    }
    }
    }
    }
    \clearpage
}

```

Two columns with header.

```

\cs_new:Nn \flowfram_set_two_col_header:n
{
\tl_if_empty:NTF \g__flowfram_two_col_header_id_tl
{
\flowfram_set_two_col:
#1
}
{
\bool_lazy_and:nnTF
{
\tl_if_empty_p:N \g__flowfram_headed_two_col_i_id_tl
}
{
\tl_if_empty_p:N \g__flowfram_headed_two_col_ii_id_tl
}
{
\flowfram_set_two_col:
}
{
\int_step_inline:nn { \c@maxflow }

```

```

{
Is this frame currently on or off on this page?
  \__flowfram_check_if_this_page:nn
  { \g__flowfram_pagecounter_tl } { ##1 }
\bool_lazy_or:nnTF
{
  \int_compare_p:nNn
  { ##1 } = { \g__flowfram_two_col_i_id_tl }
}
{
  \int_compare_p:nNn
  { ##1 } = { \g__flowfram_two_col_ii_id_tl }
}
{
\bool_if:NT \g__flowfram_not_this_frame_bool
{
  \tl_gput_right:Nn \g__flowfram_output_adjust_frames_tl
  {
    \flowsetpagelist { ##1 }
    {
      pages =
      {
        >
        \int_eval:n
        {
          \g__flowfram_pagecounter_tl
          + \c_one_int
        }
      }
    }
  }
}
}
{
\bool_lazy_or:nnTF
{
  \int_compare_p:nNn
  { ##1 } = { \g__flowfram_headed_two_col_i_id_tl }
}
{
  \int_compare_p:nNn
  { ##1 } = { \g__flowfram_headed_two_col_ii_id_tl }
}
{
\bool_if:NT \g__flowfram_not_this_frame_bool
{
  \tl_gput_right:Nn \g__flowfram_output_adjust_frames_tl
  {
    \flowsetpagelist { ##1 }
  }
}
}
}

```

```

        {
          pages =
            { \int_eval:n { \g__flowfram_pagecounter_tl } }
        }
      }
    }
  {
    \bool_if:NF \g__flowfram_not_this_frame_bool
    {
      \tl_gput_right:Nn \g__flowfram_output_adjust_frames_tl
      {
        \flowsetpagelist { ##1 } { pages = none }
      }
    }
  }
}
}
\clearpage
}
\cs_if_exist:cTF
{ set \g__flowfram_two_col_header_type_tl contents }
{
  \use:c
  { set \g__flowfram_two_col_header_type_tl contents }
  { \g__flowfram_two_col_header_id_tl }
  { #1 }
}
{
  #1
}
}
}
}

```

1.9 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}

```

`\columnheight` Version 2.0 removed `\columnheight`

`\@setcol` Set up the output box so it has the correct dimensions for specified flow frame.

This is used by the output routine. Version 2.0 renamed \@setcol.

```

\cs_new:Nn \__flowfram_set_column:n
{
  \bool_lazy_or:nnTF
  {
    \int_compare_p:nNn { \c@maxflow } < { #1 }
  }
  {
    \int_compare_p:nNn { #1 } < { \c_one_int }
  }
  {
    \msg_error:nne { flowfram } { cant-set-column }
    { \int_eval:n { #1 } }
  }
  {
    \exp_args:Nnee
    \__flowfram_message:nnn
    { info-switching-col }
    { \int_eval:n { #1 } }
    { \int_eval:n { \g__flowfram_pagecounter_tl } }
  }
  \dim_gset:Nn
  \columnwidth
  { \flowfram_frame_use_dim:nnn { flow } { width } { #1 } }
  \legacy_if:nF { usedframebreak }
  {
    \dim_set_eq:NN \l__flowfram_tmpb_dim \columnwidth
    \dim_sub:Nn \l__flowfram_tmpb_dim { \hsize }
    \dim_compare:nNnT { \l__flowfram_tmpb_dim } < { \c_zero_dim }
    {
      \dim_set:Nn \l__flowfram_tmpb_dim { - \l__flowfram_tmpb_dim }
    }
    \dim_compare:nNnT { \l__flowfram_tmpb_dim } > { \fftolerance }
    {
      \msg_warning:nnee { flowfram } { unequal-cols }
      { \dim_use:N \l__flowfram_tmpb_dim }
      { \the \fftolerance }
    }
  }
}
\global \usedframebreakfalse
\global \hsize \columnwidth
\flowfram_gset_dim_to_frame_dim:Nnnn
\textheight { flow } { height } { #1 }
\global \vsize \textheight
\global \@colht \textheight
\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.

```

\int_compare:nNnTF { \@listdepth } > { \c_zero_int }
{
  \dim_compare:nNnT { \linewidth } > { \columnwidth }
  {
    \dim_gset_eq:NN \linewidth \columnwidth
  }
}
{
  \dim_gset_eq:NN \linewidth \columnwidth
}

\setmargin
}
\stepcounter { displayedframe }
}

```

Version 2.0 no longer overriding \output.

\@docclearpage Version 2.0 no longer redefining \@docclearpage.

\flowfram@header

```

\newcommand \flowfram@header [2]
{
  \__flowfram_header:n
  {
    \color@hbox
    \g__flowfram_headercolor_tl
    \hbox_to_wd:nn { #1 } { #2 }
    \color@endbox
  }
}

```

Boxes may interfere with the dynamic frame settings, so provide an alternative.
This may be redefined to \flowfram@header if preferred:

\flowfram@dynamicheader

```

\newcommand \flowfram@dynamicheader [2]
{
  \g__flowfram_headercolor_tl
  \__flowfram_header:n { #2 }
}

\tl_new:N \g__flowfram_headercolor_tl
\tl_gset:Nn \g__flowfram_headercolor_tl { \normalcolor }
\cs_new:Nn \__flowfram_header:n
{
  \pdfannot@link@off@@
  \UseTaggingSocket
  {build/page/header}
  { }
  { #1 }
}

```

```

        \pdfannot@link@on@@
    }

\flowfram@footer
\newcommand \flowfram@footer [2]
{
  \_flowfram_footer:n
  {
    \color@hbox
    \g_flowfram_footercolor_t1
    \hbox_to_wd:nn { #1 } { #2 }
    \color@endbox
  }
}

```

Boxes may interfere with the dynamic frame settings, so provide an alternative. This may be redefined to `\flowfram@footer` if preferred:

```

\flowfram@dynamicfooter
\newcommand \flowfram@dynamicfooter [2]
{
  \g_flowfram_footercolor_t1
  \_flowfram_footer:n { #2 }
}

\tl_new:N \g_flowfram_footercolor_t1
\tl_gset:Nn \g_flowfram_footercolor_t1 { \normalcolor }

\cs_new:Nn \_flowfram_footer:n
{
  \pdfannot@link@off@@
  \UseTaggingSocket {build/page/footer}
  { }
  { #1 }
  \pdfannot@link@on@@
}

```

`\@dothehead` First define macro to do the header. This will be modified if it is turned into a dynamic frame. Version 2.0 added tagging socket.

```

\newcommand{\@dothehead}{
  \vbox_to_ht:nn { \headheight }
  {
    \vfil
    \flowfram@header { \typeblockwidth }
    {
      \@thehead
    }
  }
}

```

\@dothefoot Same again for the footer.

```
\newcommand{\@dothefoot}{
  \vbox:n
  {
    \flowfram@footer { \typeblockwidth }
    {
      \@thefoot
    }
  }
}
\newcommand{\@ddynamicthehead}{-}
\newcommand{\@ddynamicthefoot}{-}
```

Column spanning floats conflict with flowfram's output routine so make starred floats behave like unstarred floats.

```
\cs_set_eq:NN \__flowfram_org_dbfloat: \@dbfloat
\cs_set_eq:NN \@dbfloat \@float
\cs_set_eq:NN \__flowfram_org_enddbfloat: \enddbfloat
\cs_set_eq:NN \enddbfloat \endfloat
```

\FlowFramRestoreOR Version 2.0: provide a way of restoring the output routine.

```
\cs_set_eq:NN \__flowfram_org_opcol: \@opcol
\cs_set_eq:NN \__flowfram_org_outputdblcol: \@outputdblcol
\int_new:N \g__flowfram_saved_thisframe_int
\NewDocumentCommand \FlowFramRestoreOR { }
{
  \__flowfram_only_document_env:Nn \FlowFramRestoreOR
  {
    \finishthispage
    \int_gset_eq:NN \g__flowfram_saved_thisframe_int \c@thisframe
    \cs_set_eq:NN \@dbfloat \__flowfram_org_dbfloat:
    \cs_set_eq:NN \enddbfloat \__flowfram_org_enddbfloat:
    \cs_set_eq:NN \@opcol \__flowfram_org_opcol:
    \cs_set_eq:NN \@outputdblcol \__flowfram_org_outputdblcol:
    \let\@outputpage\@flowfram@org@output@page
    \renewcommand \@flowfram@build@page@after
    {
      \stepcounter{absolutepage}%
    }
    \cs_set_eq:NN \onecolumn \__flowfram_org_onecolumn:
    \cs_set_eq:NN \twocolumn \__flowfram_org_twocolumn:
    \int_gset_eq:NN \col@number \c_one_int
    \global \@twocolumnfalse
    \global \textwidth \typeblockwidth
    \global \textheight \typeblockheight
    \global \columnwidth \textwidth
    \global \hsize \textwidth
    \global \vsize \textheight
    \global \@colht \vsize
```



```

    \global \@colroom \@colht
  }
}

```

`\FlowFramUnrestoreOR` And switch back again.

```

\NewDocumentCommand \FlowFramUnrestoreOR { }
{
  \__flowfram_only_document_env:Nn \FlowFramUnrestoreOR
  {
    \clearpage
    \cs_set_eq:NN \@opcol \__flowfram_op_col:
    \cs_set_eq:NN \@outputdblcol \__flowfram_output_dbl_col:
    \let\@outputpage\@flowfram@output@page
    \cs_set_eq:NN \@dblfloat \@float
    \cs_set_eq:NN \enddblfloat \endfloat
    \renewcommand \@flowfram@build@page@before { }
    \renewcommand \@flowfram@build@page@after { }
    \cs_set_eq:NN \onecolumn \__flowfram_doc_onecolumn:
    \RenewDocumentCommand \twocolumn { o }
    {
      \__flowfram_doc_twocolumn:n { ##1 }
    }
    \int_gzero:N \c@displayedframe
    \int_gset_eq:NN \c@thisframe \g__flowfram_saved_thisframe_int
    \int_gset_eq:NN \g__flowfram_next_frame_int \g__flowfram_saved_thisframe_int
    \__flowfram_set_column:n { \c@thisframe }
    \@flowfram@update@col@count{\c@thisframe}%
  }
}

```

`\@opcol` Version 2.0 ensure marks stuff is included but always use `\@outputdblcol`.

```

\renewcommand * \@opcol { \__flowfram_op_col: }

\cs_new:Nn \__flowfram_op_col:
{
  \if@twocolumn
    \@expl@@@mark@update@dblcol@structures@@
  \else
    \@expl@@@mark@update@singlecol@structures@@
  \fi
  \@outputdblcol
  \global \@mparbottom \z@
  \global \@textfloatsheight \z@
  \@floatplacement
}

```

`\@outputpage` The header and footer code need to be removed as they may be placed in a dynamic frame instead and repositioned. Version 2.0 updated.

```

\ExplSyntaxOff

```

Save original so that it can be restored.

```
\let\@@flowfram@org@output@page\@outputpage
\def\@outputpage{\@@flowfram@output@page}
```

Define flowfram's version of \@outputpage:

```
\newcommand \@@flowfram@output@page{%
  \UseHook {build/page/before}%
  \begingroup
  \let \protect \noexpand
  \language\document@default@language
  \@resetactivechars
  \nfss@catcodes
  \catcode'\$\thr@@
  \catcode'\_8\relax
  \catcode'\_10\relax
  \catcode'\^~I10\relax
  \catcode'\~13\relax
  \catcode'\^~M5\relax
  \global\let\@@if@newlist@if@newlist
  \global\@newlistfalse
  \UseHook {build/page/reset}%
  \shipout \vbox{%
    \set@typeset@protect
    \aftergroup \endgroup
    \aftergroup \set@typeset@protect
    \if@specialpage
      \global \@specialpagefalse
      \ifcsdef {ps@special\@specialstyle}%
        {\@nameuse {ps@special\@specialstyle}}%
        {\@nameuse {ps@\@specialstyle}}%
      \fi
    \if@twoside
      \ifodd\count\z@
        \let \@thehead \@oddhead
        \let \@thefoot \@oddfoot
        \let \@themargin \oddsidemargin
      \else
        \let \@thehead \@evenhead
        \let \@thefoot \@evenfoot
        \let \@themargin \evensidemargin
      \fi
    \else
      \let \@thehead \@oddhead
      \let \@thefoot \@oddfoot
      \let \@themargin \oddsidemargin
    \fi
  }
```

Omit block relating to header, footer and margin.

```
\reset@font
\normalsize
```

```

\normalsfcodes
\let \label \@gobble@with@sphack@om
\let \index \@gobble@with@sphack@som
\let \glossary \@gobble@with@sphack@om
\let \@@flowfram@current@pre@section@title \empty
\baselineskip \z@skip
\lineskip \z@skip
\lineskiplimit \z@
  \@beginvdi
  \vskip \typeblockoffsety
  \moveright\@themargin \vbox {%
Omit header.
  \box \@outputbox
Omit footer.
  }%
}%
\global \let \if@newlist \@@if@newlist
\@flowfram@set@next@vsize
\global \textwidth \hsize
\global \textheight \vsize
\global \@colht \vsize
\global \@colroom \@colht
\stepcounter{page}%
\stepcounter{absolute page}%
\setcounter{displayedframe}{0}%
\@flowfram@update@col@count{\c@thisframe}%
\UseHook {build/page/after}%
}
\ExplSyntaxOn
If OR is restored, provide a means to still display the dynamic frames and
update absolute page.
\AddToHook {build/page/before}
{
  \@flowfram@build@page@before
}
\AddToHook {build/page/after}
{
  \@flowfram@build@page@after
}
No extra action unless OR is restored.
\newcommand\@flowfram@build@page@before{}
\newcommand\@flowfram@build@page@after{}
Set \vsize for next frame.
\newcommand\@flowfram@set@next@vsize
{
  \flowfram_gset_dim_to_frame_dim:Nnnn \vsize
  { flow } { height } { \g__flowfram_next_frame_int }

```

}

`\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`.

Keep track of which frames have the header and footer. These will be zero if feature not implemented. NB the even IDs are only set by `flowframtkutils`.

```
\int_new:N \g__flowfram_dynamic_header_int
\int_new:N \g__flowfram_dynamic_even_header_int
\int_new:N \g__flowfram_dynamic_footer_int
\int_new:N \g__flowfram_dynamic_even_footer_int
```

Create dynamic frames for the header and footer.

```
\NewDocumentCommand \makedfheaderfooter { }
{
```

Create dynamic frames at the standard location:

```
__flowfram_new_dynamic:nnnnnn
{ all } { \typeblockwidth } { \headheight }
{ Opt } { \typeblockheight + \headsep } { header }
\int_set_eq:NN \g__flowfram_dynamic_header_int
\c@maxdynamic
__flowfram_set_dynamic_by_idn:nn { \c@maxdynamic } { valign=c }
__flowfram_new_dynamic:nnnnnn
{ all } { \typeblockwidth } { \headheight }
{ Opt } { -\footskip } { footer }
\int_set_eq:NN \g__flowfram_dynamic_footer_int
\c@maxdynamic
__flowfram_set_dynamic_by_idn:nn { \c@maxdynamic } { valign=c }
\gdef \@dothehead { }
\gdef \@dothefoot { }
\gdef \@dodynamicthehead
{
  __flowfram_set_dynamic_contents:nn
  { \g__flowfram_dynamic_header_int }
  {
    \flowfram_dynamic_header:n
    { \g__flowfram_dynamic_header_int }
  }
}
\gdef \@dodynamicthefoot
{
  __flowfram_set_dynamic_contents:nn
  { \g__flowfram_dynamic_footer_int }
  {
    \flowfram_dynamic_footer:n
    { \g__flowfram_dynamic_footer_int }
  }
}
}
```

```

\tl_clear:N \g__flowfram_headercolor_tl
\tl_clear:N \g__flowfram_footercolor_tl
\__flowfram_adjust_page_styles:n
{
  \let \ps@myheadings \ps@ffmyheadings
  \let \ps@headings \ps@ffheadings
  \let \ps@plain \ps@ffplain
  \let \ps@empty \ps@ffempty
}
\pagestyle { ffheadings }
\cs_set:Nn \__flowfram_only_pre_makedfheaderfooter:nn
{
  \msg_error:nnn { flowfram } { option-too-late } { #1 } { \makedfheaderfooter }
}
}

```

This should only be done in the preamble.

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

SetDynamicHeadFootAttributes Set dynamic header and footer frame attributes if they have been defined. Does nothing otherwise.

```

\NewDocumentCommand \FlowFramSetDynamicHeadFootAttributes { m }
{
  \int_compare:nNnT
  { \g__flowfram_dynamic_header_int } > { \c_zero_int }
  {
    \__flowfram_set_dynamic_by_idn:nn
    { \g__flowfram_dynamic_header_int } { #1 }
  }
  \int_compare:nNnT
  { \g__flowfram_dynamic_even_header_int } > { \c_zero_int }
  {
    \__flowfram_set_dynamic_by_idn:nn
    { \g__flowfram_dynamic_even_header_int } { #1 }
  }
  \int_compare:nNnT
  { \g__flowfram_dynamic_footer_int } > { \c_zero_int }
  {
    \__flowfram_set_dynamic_by_idn:nn
    { \g__flowfram_dynamic_footer_int } { #1 }
  }
  \int_compare:nNnT
  { \g__flowfram_dynamic_even_footer_int } > { \c_zero_int }
  {
    \__flowfram_set_dynamic_by_idn:nn
    { \g__flowfram_dynamic_even_footer_int } { #1 }
  }
}

```

The argument is the dynamic frame's IDN. The box is added because in

some cases without it an infinite shrinkage error occurs.

```

\cs_new:Nn \flowfram_dynamic_header:n
{
  \flowfram_set_dim_to_frame_dim:Nnnn
    \l__flowfram_sfd_width_dim
    { dynamic } { width } { #1 }
  \if@twoside
    \ifodd\c@page
      \flowfram@dynamicheader
        { \l__flowfram_sfd_width_dim }
        { \@oddhead }
    \else
      \flowfram@dynamicheader
        { \l__flowfram_sfd_width_dim }
        { \@evenhead }
    \fi
  \else
    \flowfram@dynamicheader
      { \l__flowfram_sfd_width_dim }
      { \@thehead }
  \fi
}

```

For the benefit of flowframtkutils, where there may be separate odd and even header dynamic frames:

```

\cs_new:Nn \flowfram_dynamic_odd_header:n
{
  \flowfram_set_dim_to_frame_dim:Nnnn
    \l__flowfram_sfd_width_dim
    { dynamic } { width } { #1 }
  \if@twoside
    \ifodd\c@page
      \flowfram@dynamicheader
        { \l__flowfram_sfd_width_dim }
        { \@oddhead }
    \else
      \flowfram@dynamicheader
        { \l__flowfram_sfd_width_dim }
        { }
    \fi
  \else
    \flowfram@dynamicheader
      { \l__flowfram_sfd_width_dim }
      { \@thehead }
  \fi
}

\cs_new:Nn \flowfram_dynamic_even_header:n
{
  \flowfram_set_dim_to_frame_dim:Nnnn

```

```

        \l__flowfram_sfdf_width_dim
        { dynamic } { width } { #1 }
\if@twoside
  \ifodd\c@page
    \flowfram@dynamicheader
    { \l__flowfram_sfdf_width_dim }
    { }
  \else
    \flowfram@dynamicheader
    { \l__flowfram_sfdf_width_dim }
    { \@evenhead }
  \fi
\else
  \flowfram@dynamicheader
  { \l__flowfram_sfdf_width_dim }
  { }
\fi
}
\cs_new:Nn \flowfram_dynamic_footer:n
{
  \flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_sfdf_width_dim
  { dynamic } { width } { #1 }
\if@twoside
  \ifodd\c@page
    \flowfram@dynamicfooter
    { \l__flowfram_sfdf_width_dim } { \@oddfoot }
  \else
    \flowfram@dynamicfooter
    { \l__flowfram_sfdf_width_dim } { \@evenfoot }
  \fi
\else
  \flowfram@dynamicfooter
  { \l__flowfram_sfdf_width_dim } { \@thefoot }
\fi
}
\cs_new:Nn \flowfram_dynamic_odd_footer:n
{
  \flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_sfdf_width_dim
  { dynamic } { width } { #1 }
\if@twoside
  \ifodd\c@page
    \flowfram@dynamicfooter
    { \l__flowfram_sfdf_width_dim } { \@oddfoot }
  \else
    \flowfram@dynamicfooter
    { \l__flowfram_sfdf_width_dim } { }
  \fi
}

```

```

\else
  \flowfram@dynamicfooter
  { \l__flowfram_sfdf_width_dim } { \@thefoot }
\fi
}
\cs_new:Nn \flowfram_dynamic_even_footer:n
{
  \flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_sfdf_width_dim
  { dynamic } { width } { #1 }
  \if@twoside
  \ifodd\c@page
    \flowfram@dynamicfooter
    { \l__flowfram_sfdf_width_dim } { }
  \else
    \flowfram@dynamicfooter
    { \l__flowfram_sfdf_width_dim } { \@evenfoot }
  \fi
  \else
  \flowfram@dynamicfooter
  { \l__flowfram_sfdf_width_dim } { }
  \fi
}

```

`\dfOddHeaderStyle`

```
\newcommand \dfOddHeaderStyle [ 1 ] { \hfill #1 }
```

`\dfEvenHeaderStyle`

```
\newcommand \dfEvenHeaderStyle [ 1 ] { #1 \hfill }
```

`\dfOddFooterStyle`

```
\newcommand \dfOddFooterStyle [ 1 ] { \hfill #1 \hfill }
```

`\dfEvenFooterStyle`

```
\newcommand \dfEvenFooterStyle [ 1 ] { \hfill #1 \hfill }
```

Some classes have page styles that cause a problem with the dynamic header and footer so provide a simple page style. The dynamic frame's attributes can be adjusted to changing the formatting.

```

\cs_if_exist:NTF \chapter
{
  \cs_new:Nn \__flowfram_ps_myheadings:
  {
    \def \@oddfoot { \dfOddFooterStyle { \thepage } }
    \def \@evenfoot { \dfEvenFooterStyle { \thepage } }
    \def \@oddhead { \dfOddHeaderStyle { \rightmark } }
    \def \@evenhead { \dfEvenHeaderStyle { \leftmark } }
    \let \@mkboth \@gobbletwo
    \let \chaptermark \@gobble
  }
}

```



```

\let \sectionmark \@gobble
}
\cs_new:Nn \__flowfram_ps_headings:
{
\def \@oddfoot { \dfOddFooterStyle { \thepage } }
\def \@evenfoot { \dfEvenFooterStyle { \thepage } }
\def \@oddhead { \dfOddHeaderStyle { \rightmark } }
\def \@evenhead { \dfEvenHeaderStyle { \leftmark } }
\let \@mkboth \markboth
\def \chaptermark ##1
{
\markboth
{
\flowframchapterheader { ##1 }
}
{
\flowfram_initial_right_mark:n
{
\flowframchapterheader { ##1 }
}
}
}
\def \sectionmark ##1
{
\markright
{
\flowframsectionheader { ##1 }
}
}
}
\newcommand \flowframchapterheader [ 1 ]
{
\flowfram_header_font:n
{
\flowfram_header_case:n
{
\ifnum \c@secnumdepth >\m@ne
\if@mainmatter
\flowframheaderchapprefix
\thechapter
\flowframheadersep
\fi
\fi
#1
}
}
}
\newcommand \flowframsectionheader [ 1 ]
{
\flowfram_subheader_font:n

```

```

    {
      \flowfram_subheader_case:n
      {
        \ifnum \c@secnumdepth >\z@
          \thesection
          \flowframheadersep
        \fi
        #1
      }
    }
  }
}
{
\cs_new:Nn \__flowfram_ps_myheadings:
{
  \def \@oddfoot { \dfOddFooterStyle { \thepage } }
  \def \@evenfoot { \dfEvenFooterStyle { \thepage } }
  \def \@oddhead { \dfOddHeaderStyle { \rightmark } }
  \def \@evenhead { \dfEvenHeaderStyle { \leftmark } }
  \let \@mkboth \@gobbletwo
  \let \sectionmark \@gobble
  \let \subsectionmark \@gobble
}
\cs_new:Nn \__flowfram_ps_headings:
{
  \def \@oddfoot { \dfOddFooterStyle { \thepage } }
  \def \@evenfoot { \dfEvenFooterStyle { \thepage } }
  \def \@oddhead { \dfOddHeaderStyle { \rightmark } }
  \def \@evenhead { \dfEvenHeaderStyle { \leftmark } }
  \let \@mkboth \markboth
  \def \sectionmark ##1
  {
    \markboth
    {
      \flowframsectionheader { ##1 }
    }
    {
      \flowfram_initial_right_mark:n
      {
        \flowframsectionheader { ##1 }
      }
    }
  }
  \def \subsectionmark ##1
  {
    \markright
    {
      \flowframsubsectionheader { ##1 }
    }
  }
}

```

```

}
\newcommand \flowframsectionheader [ 1 ]
{
  \flowfram_header_font:n
  {
    \flowfram_header_case:n
    {
      \ifnum \c@secnumdepth >\m@ne
        \thesection
        \flowframheadersep
        \fi
      #1
    }
  }
}
\newcommand \flowframsubsectionheader [ 1 ]
{
  \flowfram_subheader_font:n
  {
    \flowfram_subheader_case:n
    {
      \int_compare:nNnT { \c@secnumdepth } > { 2 }
      {
        \thesubsection
        \flowframheadersep
      }
      #1
    }
  }
}
}

```

At the start of a chapter/section the following governs whether or not to initialise the right mark to the same as the left mark.

```
\cs_new:Nn \flowfram_initial_right_mark:n { #1 }
```

`\ps@ffmyheadings`

```

\newcommand \ps@ffmyheadings
{
  \__flowfram_ps_myheadings:
  \bool_if:NT \l__flowfram_ps_ffempty_hides_bool
  {
    \FlowFramSetDynamicHeadFootAttributes
    { hide = false , hidethis = false }
  }
}

```

`\ps@ffheadings`

```

\newcommand \ps@ffheadings
{

```

```

    \__flowfram_ps_headings:
    \bool_if:NT \l__flowfram_ps_ffempty_hides_bool
    {
      \FlowFramSetDynamicHeadFootAttributes
        { hide = false , hidethis = false }
    }
  }

\ps@ffplain
\newcommand \ps@ffplain
{
  \__flowfram_ps_plain:
  \bool_if:NT \l__flowfram_ps_ffempty_hides_bool
  {
    \FlowFramSetDynamicHeadFootAttributes
      { hide = false , hidethis = false }
  }
}

\cs_new:Nn \__flowfram_ps_plain:
{
  \def \@oddfoot { \dfOddFooterStyle { \thepage } }
  \def \@evenfoot { \dfEvenFooterStyle { \thepage } }
  \def \@oddhead { }
  \def \@evenhead { }
  \let \@mkboth \@gobbletwo
  \let \sectionmark \@gobble
  \let \subsectionmark \@gobble
}

\ps@ffempty
\newcommand \ps@ffempty
{
  \l__flowfram_ffempty_style_tl
  \bool_if:NT \l__flowfram_ps_ffempty_hides_bool
  {
    \FlowFramSetDynamicHeadFootAttributes { hide }
  }
}

\cs_new:Nn \__flowfram_ps_empty:
{
  \def \@oddfoot { }
  \def \@evenfoot { }
  \def \@oddhead { }
  \def \@evenhead { }
  \let \@mkboth \@gobbletwo
  \let \sectionmark \@gobble
  \let \subsectionmark \@gobble
}

```

`\ps@specialffempty` Used with `\thispagestyle`:

```
\newcommand \ps@specialffempty
{
  \l__flowfram_ffempty_style_tl
  \bool_if:NT \l__flowfram_ps_ffempty_hides_bool
  {
    \FlowFramSetDynamicHeadFootAttributes { hidethis }
  }
}
```

`\ps@specialeempty` Used with `\thispagestyle`:

```
\newcommand \ps@specialeempty { \ps@empty }
```

`\footnotecolor` Does nothing now. Marked for removal.

```
\newcommand \footnotecolor { }
```

`\@makecol` Version 2.0 removed redefinition of `\@makecol`. This means that footnotes will now be the default colour rather than the text colour associated with the current flow frame. Use `\color` in the preamble to set the default colour.

`\if@ff@moreframes` Version 2.0 replaced legacy conditional `\if@ff@moreframes` with `l3bool`.

```
\bool_new:N \g__flowfram_more_frames_bool
```

`\@ff@checkifmoreframes` Check to see if there are more flow frames defined, and set `\g__flowfram_more_frames_bool` as appropriate. This involves iterating through all flow frames, and through each frame's page list. Version 2.0 renamed `\@ff@checkifmoreframes`.

```
\cs_new:Nn \__flowfram_check_if_more_frames:n
{
  \bool_gset_false:N \g__flowfram_more_frames_bool
  \int_set:Nn \l__flowfram_col_int { #1 }
  \bool_while_do:nn
  {
    \bool_lazy_and_p:nn
    {
      \int_compare_p:nNn { \l__flowfram_col_int } < { \c@maxflow }
    }
    {
      \bool_not_p:n
      {
        \bool_if_p:N \g__flowfram_more_frames_bool
      }
    }
  }
}
```

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

```
\exp_args:Ne
\flowfram_if_in_frame_clist:nnnnF
```

```

    { \int_eval:n { \g__flowfram_pagecounter_tl } }
    { flow } { excludelist } { \l__flowfram_col_int }
  {

```

This page is in this not in frame's exclusion list so now check if in frame's page list.

```

    \exp_args:Ne
    \flowfram_if_frame_has_morepages:nnnT
    { \int_eval:n { \g__flowfram_pagecounter_tl } }
    { flow } { \l__flowfram_col_int }
  {
    \bool_gset_true:N \g__flowfram_more_frames_bool
  }

```

If found a frame, break out of loop.

```

  }
}
\bool_if:NF \g__flowfram_more_frames_bool
{

```

```

  \int_set:Nn \l__flowfram_tmpa_int
  { \g__flowfram_pagecounter_tl }

```

Look ahead up to a maximum of 4 pages.

```

\int_zero:N \l__flowfram_tmpb_int
\bool_do_while:nn
{
  \int_compare_p:nNn { \l__flowfram_tmpb_int } < { 4 }
}
{
  \int_incr:N \l__flowfram_tmpa_int
  \int_zero:N \l__flowfram_col_int
  \bool_while_do:nn
  {
    \int_compare_p:nNn { \l__flowfram_col_int } < { \c@maxflow }
  }
  {
    \int_incr:N \l__flowfram_col_int
  }
}

```

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

```

\exp_args:NV \flowfram_if_frame_excludedpage:nnnF
\l__flowfram_tmpa_int
{ flow } { \l__flowfram_col_int }
{

```

This page is not in this frame's exclusion list.

```

\exp_args:NV
\flowfram_if_frame_has_morepages:nnnT
\l__flowfram_tmpa_int
{ flow } { \l__flowfram_col_int }
{
  \bool_gset_true:N \g__flowfram_more_frames_bool
}

```

```

    }
    If found a frame, break out of loop.
    \bool_if:NT \g__flowfram_more_frames_bool
    {
      \int_set_eq:NN \l__flowfram_col_int \c@maxflow
    }
  }
}
\bool_if:NTF \g__flowfram_more_frames_bool
{
  \int_set:Nn \l__flowfram_tmpb_int { 4 }
}
{
  \int_incr:N \l__flowfram_tmpb_int
}
}
}
}
}

```

`\@ff@checkpages` Version 2.0 removed `\@ff@checkpages`

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.

`\@ff@checkthispage` Version 2.0 removed `\@ff@checkthispage`

`\@ff@checknumrange` Version 2.0 removed `\@ff@checknumrange`.

`\c@ffrangenum` Version 2.0 removed `\ffrangenum`.

`\@ff@getrangeless` Version 2.0 removed `\@ff@getrangeless`.

`\@ff@getrangegreater` Version 2.0 removed `\@ff@getrangegreater`.

`\@ff@getrange` Version 2.0 removed `\@ff@getrange`.

`\@@ff@getrange` Version 2.0 removed `\@ff@getrange`.

`\@ff@output@adjustframes` Provide a hook to adjust frame settings in the output routine. Version 2.0 renamed `\@ff@output@adjustframes`

```
\tl_new:N \g__flowfram_output_adjust_frames_tl
```

`\ffaddtoadjustframeshook` Add stuff to the output hook.

```

\NewDocumentCommand \ffaddtoadjustframeshook { m }
{
  \tl_gput_right:Nn \g__flowfram_output_adjust_frames_tl { #1 }
}

```

`\iff@nwpg` Conditional to keep track of whether or not a new page is required. Version 2.0 replaced legacy conditional with `l3bool`.

```
%\newif\iff@nwpg
\bool_new:N \g__flowfram_new_page_bool
```

`\if@notthiscol` Version 2.0 replaced legacy conditional `\if@notthiscol` with `l3bool`.

```
\bool_new:N \g__flowfram_not_this_frame_bool
```

`\c@curpg` Keep track of current page. Version 2.0 renamed `\c@curpg`.

```
\int_new:N \g__flowfram_current_page_int
```

`\@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.) Version 2.0 renamed `\@g@tnextcol`. Get the next available column, which may be on this page or the next. The integer variable is the current frame index which will be updated. Also, `\g__flowfram_current_page_int` will be updated if there are no further frames available on the current page.

```
\cs_new:Nn \__flowfram_get_next_column:N
{
```

Do any frame adjustments:

```
\g__flowfram_output_adjust_frames_tl
```

Now clear the hook:

```
\tl_gclear:N \g__flowfram_output_adjust_frames_tl
```

Now check for any more frames.

```
\__flowfram_check_if_more_frames:n { \c@thisframe }
\bool_if:NF \g__flowfram_more_frames_bool
{
```

No more frames, add new frame

```
\msg_warning:nne { flowfram } { no-more-cols-on-page }
{ \int_eval:n { \g__flowfram_pagecounter_tl } }
\flf@doifverbose
{
\tl_clear:N \l__flowfram_tmpa_tl
\seq_map_indexed_inline:nn \g__flowfram_flowlabels_seq
{
\tl_put_right:Ne \l__flowfram_tmpa_tl
{
\exp_not:N \MessageBreak
##1 ( ##2 )
Pages: ~
\flowfram_frame_use_clist:nnn
{ flow } { pagelist } { ##1 }
Exclusions: ~
\flowfram_frame_use_clist:nnn
{ flow } { excludelist } { ##1 }
}
}
```



```

    }
    \msg_info:nnV { flowfram } { info-col-list }
    \l__flowfram_tmpa_tl
  }

```

No flow frames available. Create a single flow frame.

```

  \__flowfram_one_column:n { all }
  \int_gset_eq:NN #1 \c@maxflow
}
\bool_gset_true:N \g__flowfram_not_this_frame_bool
\bool_gset_false:N \g__flowfram_new_page_bool
\int_set_eq:NN \l__flowfram_col_int #1

\int_gset:Nn \g__flowfram_current_page_int
{ \g__flowfram_pagecounter_tl }
\bool_while_do:Nn \g__flowfram_not_this_frame_bool
{
  \int_compare:nNnTF
    { \l__flowfram_col_int } = { \c@maxflow }
  {

```

Reached the end of the page. Try the next one.

```

  \int_set_eq:NN \l__flowfram_col_int \c_one_int
  \bool_gset_true:N \g__flowfram_new_page_bool
  \int_gincr:N \g__flowfram_current_page_int
}
{

```

Move on to the next flow frame on this page.

```

  \int_incr:N \l__flowfram_col_int
}
\__flowfram_check_if_this_page:nn
{ \g__flowfram_current_page_int } { \l__flowfram_col_int }
}
\int_gset_eq:NN #1 \l__flowfram_col_int
}

```

`\@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. Version 2.0 renamed `\@ff@chckifthispg`.

```

\cs_new:Nn \__flowfram_check_if_this_page:nn
{

```

First set up some variables.

```

  \bool_gset_false:N \g__flowfram_not_this_frame_bool
  \flowfram_set_clist_to_frame_clist:Nnnn
  \l__flowfram_exclude_pages_clist
  { flow } { excludelist } { #2 }

```

Iterate over exclusion list.

```

  \clist_map_inline:Nn
  \l__flowfram_exclude_pages_clist

```

```

{
  \int_compare:nNnT { ##1 } = { #1 }
  {
    \bool_gset_true:N \g__flowfram_not_this_frame_bool
    \clist_map_break:
  }
}
\bool_if:NF \g__flowfram_not_this_frame_bool
{

```

Not in the excluded list. Now check if in the page list range.

```

  \bool_gset_true:N \g__flowfram_not_this_frame_bool
  \flowfram_set_clist_to_frame_clist:Nnnn
  \l__flowfram_pages_clist
  { flow } { pagelist } { #2 }
  \__flowfram_check_if_this_page:n { #1 }
}
}

```

\@@ff@chckifthispg Now go ahead and check. Version 2.0 renamed \@@ff@chckifthispg.

```

\cs_new:Nn \__flowfram_check_if_this_page:n
{
  \tl_if_eq:NNF \l__flowfram_pages_clist \c_flowfram_none_tl
  {
    \tl_if_eq:NNTF \l__flowfram_pages_clist \c_flowfram_all_tl
    {
      \bool_gset_false:N \g__flowfram_not_this_frame_bool
    }
    {
      \tl_if_eq:NNTF \l__flowfram_pages_clist \c_flowfram_odd_tl
      {
        \int_if_odd:nT { #1 }
        {
          \bool_gset_false:N \g__flowfram_not_this_frame_bool
        }
      }
    }
    {
      \tl_if_eq:NNTF \l__flowfram_pages_clist \c_flowfram_even_tl
      {
        \int_if_even:nT { #1 }
        {
          \bool_gset_false:N \g__flowfram_not_this_frame_bool
        }
      }
    }
  }
}

```

check through list of page numbers

```

  \clist_map_inline:Nn \l__flowfram_pages_clist
  {
    \flowfram_if_in_pagerange:nnT { #1 } { ##1 }
  }

```



```

\exp_args:Nne \__flowfram_message:nn
  { info-setting-col } { \int_eval:n { #1 } }
\flowfram_frame_set_box_eq:nnnN
  { flow } { column } { #1 } \@outputbox
}

```

\rotateframe

```

\NewDocumentCommand \rotateframe { m m }
{
  \int_if_zero:nTF { #1 }
  { #2 }
  {
    \rotatebox { #1 } { #2 }
  }
}

```

\@docolbox Put flow frame on the page with the correct border, if it has one. Version 2.0 renamed \@docolbox.

```

\cs_new:Nn \__flowfram_do_flow_box:n
{
  \exp_args:Nnee
  \__flowfram_message:nnn
  { info-doing-col }
  { \int_eval:n { #1 } }
  { \int_eval:n { \g__flowfram_pagecounter_tl } }
\flowfram_set_tl_to_frame_tl:Nnnn
  \l__flowfram_frametype_tl
  { flow } { frametype } { #1 }

```

Frame colour

```

\flowfram_set_tl_to_frame_tl:Nnnn
  \l__flowfram_bordercolor_tl
  { flow } { bordercolor } { #1 }

```

Text colour

```

\flowfram_set_tl_to_frame_tl:Nnnn
  \l__flowfram_textcolor_tl
  { flow } { textcolor } { #1 }

```

Background colour

```

\flowfram_set_tl_to_frame_tl:Nnnn
  \l__flowfram_backcolor_tl
  { flow } { backcolor } { #1 }

```

Compute offset for this frame

```

\__flowfram_flow_calc_offset:n {#1}%

```

Rotate frame if required

```

\exp_args:Ne \rotateframe
{
  \flowfram_frame_use_tl:nnn { flow } { angle } { #1 }
}

```

```

}
{
Check if frame has a border
\flowfram_if_frame_bool:nnnTF
{ flow } { hasframe } { #1 }
{
Put the required border around the frame
\__flowfram_bordered_box:nnnn
{
\flowfram_frame_use_dim:nnn { flow } { width } { #1 }
}
{
\flowfram_frame_use_dim:nnn { flow } { height } { #1 }
}
{
\flowfram_frame_use_drop_box:nnn
{ flow } { column } { #1 }
}
{
\use:c { \l__flowfram_frametype_tl }
}
}
{
Do the frame without a border
\__flowfram_noborder_box:nnn
{
\flowfram_frame_use_dim:nnn { flow } { width } { #1 }
}
{
\flowfram_frame_use_dim:nnn { flow } { height } { #1 }
}
{
\flowfram_frame_use_drop_box:nnn { flow } { column } { #1 }
}
}
}
}
}

```

\@docolbbox Do the bounding box for given flow frame. Version 2.0 renamed \@docolbbox.

```

\cs_new:Nn \__flowfram_do_flow_bounding_box:n
{
\__flowfram_flow_calc_offset:n { #1 }
\tl_clear:N \l__flowfram_bordercolor_tl
\tl_clear:N \l__flowfram_textcolor_tl
\__flowfram_do_if_draft:nn
{
\__flowfram_noborder_box:nnn
{

```

```

        \flowfram_frame_use_dim:nnn { flow } { width } { #1 }
      }
    {
      \flowfram_frame_use_dim:nnn { flow } { height } { #1 }
    }
    {
      \flowfram_frame_use_drop_box:nnn { flow } { column } { #1 }
    }
  }
  {
    \flowfram_draft_annotate:nnn
    { F } { \int_eval:n { #1 } } { \getflowlabel { #1 } }
  }
}

```

`\@ff@fbox` Put the T_EX box #3 of width #1 and height #2, and frame making command specified by #4. Version 2.0 renamed `\@ff@fbox`.

```

\cs_new:Nn \__flowfram_bordered_box:nnnn
{
  \group_begin:
  \dim_set_eq:NN \fboxsep \flowframesep
  \dim_set_eq:NN \fboxrule \flowframerule
  \__flowfram_set_border_color:
  \kern \l__flowfram_offset_dim
  #4 { \__flowfram_noborder_box:nnn { #1 } { #2 } { #3 } }
  \group_end:
}

```

`\@ff@box` Put the T_EX box #3 of width #1 and height #2 on the page. Version 2.0 renamed `\@ff@box`.

```

\cs_new:Nn \__flowfram_noborder_box:nnn
{
  \group_begin:
  \__flowfram_background_box:n
  {
    \vbox_to_ht:nn { #2 }
    {
      \hbox_to_wd:nn { #1 }
      { \hss { \__flowfram_set_text_color: #3 } \hss }
      \vss
      \kern\z@
    }
  }
  \group_end:
}

```

`\@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. Version 2.0 renamed `\@putcolbox`.

```

\cs_new:Nn \__flowfram_put_flow_box:n
{
  \__flowfram_check_if_this_page:nn
  { \g__flowfram_pagecounter_tl } { #1 }

  \bool_if:NT \g__flowfram_not_this_frame_bool
  {
    \flowfram_frame_if_box_empty:nnnF
    { flow } { column } { #1 }
    {
      \msg_warning:nneeee
      { flowfram } { non-void-box }
      { \int_eval:n { #1 } }
      { \int_use:N \g__flowfram_pagecounter_tl }
      { \flowfram_frame_use_clist:nnn { flow } { pagelist } { #1 } }
      { \flowfram_frame_use_clist:nnn { flow } { excludelist } { #1 } }
      \bool_gset_false:N \g__flowfram_not_this_frame_bool
    }
  }
  \bool_if:NTF \g__flowfram_not_this_frame_bool
  {
    \exp_args:Nnee
    \__flowfram_message:nnn
    { info-col-not-required }
    { \int_eval:n { #1 } }
    { \int_eval:n { \g__flowfram_pagecounter_tl } }
  }
  {
    \@killglue
    \legacy_if:nTF { @twoside }
    {
      \int_if_odd:nTF { \c@page }
      {
        \box_move_up:nn
        {
          \flowfram_frame_use_dim:nnn { flow } { posy } { #1 }
        }
        {
          \hbox_to_zero:n
          {
            \kern
            \flowfram_frame_use_dim:nnn { flow } { posx } { #1 }
            \__flowfram_do_flow_box:n { #1 }
            \hss
          }
        }
      }
    }
  }
  {
    \box_move_up:nn
  }
}

```

```

    {
      \flowfram_frame_use_dim:nnn { flow } { eveny } { #1 }
    }
    {
      \hbox_to_zero:n
      {
        \kern
          \flowfram_frame_use_dim:nnn { flow } { evenx } { #1 }
        \__flowfram_do_flow_box:n { #1 }
        \hss
      }
    }
  }
}
{
  \box_move_up:nn
  {
    \flowfram_frame_use_dim:nnn { flow } { posy } { #1 }
  }
  {
    \hbox_to_zero:n
    {
      \kern
        \flowfram_frame_use_dim:nnn { flow } { posx } { #1 }
      \__flowfram_do_flow_box:n { #1 }
      \hss
    }
  }
}
}
}
}

```

\@putcolbbox Same for flow frame bounding box: Version 2.0 renamed \@putcolbbox.

```

\cs_new:Nn \__flowfram_put_flow_bounding_box:n
{
  \__flowfram_check_if_this_page:nn
  { \g__flowfram_pagecounter_tl } { #1 }
  \bool_if:NF \g__flowfram_not_this_frame_bool
  {
    \@killglue
    \legacy_if:nTF { @twoside }
    {
      \int_if_odd:nTF { \c@page }
      {
        \box_move_up:nn
        {
          \flowfram_frame_use_dim:nnn { flow } { posy } { #1 }
        }
      }
    }
  }
}

```



```

\cs_new:Nn \__flowfram_set_doublebox_offset:
{
  \dim_set:Nn \l__flowfram_offset_dim
  {
    - \flowframesep
    -3.75 \flowframerule
    - 0.5pt
  }
}

```

\ff@s@t@ovalboxoffset Compute offset for \ovalbox. Version 2.0 renamed \ff@s@t@ovalboxoffset

```

\cs_new:Nn \__flowfram_set_ovalbox_offset:
{
  \dim_set:Nn \l__flowfram_offset_dim
  {
    - \flowframesep
    - \fontdimen 8 \tenln
  }
}

```

\ff@s@t@Oovalboxoffset Compute offset for \Oovalbox: Version 2.0 renamed \ff@s@t@Oovalboxoffset

```

\cs_new:Nn \__flowfram_set_Oovalbox_offset:
{
  \dim_set:Nn \l__flowfram_offset_dim
  {
    - \flowframesep
    - \fontdimen 8 \tenlnw
  }
}

```

\ff@s@t@defaultoffset Compute default offset: Version 2.0 renamed \ff@s@t@defaultoffset

```

\cs_new:Nn \__flowfram_set_default_offset:
{
  \dim_set:Nn \l__flowfram_offset_dim
  {
    - \flowframesep
    - \flowframerule
  }
}

```

\ff@setoffset Compute offset for flow frame #1. Stores offset value in \l__flowfram_offset_tl.
Version 2.0 renamed \ff@setoffset.

```

\cs_new:Nn \__flowfram_flow_calc_offset:n
{
  \flowfram_if_tl_eq_frame_tl:NnnnTF
  \c_flowfram_compute_tl
  { flow } { offset } { #1 }
  {
    \flowfram_if_frame_bool:nnnTF { flow } { hasframe } { #1 }
  }
}

```

```

    {
      \flowfram_set_tl_to_frame_tl:Nnnn
        \l__flowfram_frametype_tl
        { flow } { frametype } { #1 }
      \cs_if_exist_use:cF
        { __flowfram_set_ \l__flowfram_frametype_tl _offset: }
        {
          \__flowfram_set_default_offset:
        }
    }
    {
      \dim_zero:N \l__flowfram_offset_dim
    }
  }
  {
    \flowfram_set_dim_to_frame_tl:Nnnn
      \l__flowfram_offset_dim
      { flow } { offset } { #1 }
  }
}

```

\@sf@setoffset Compute offset for static frame #1. Stores offset value in \l__flowfram_offset_tl.
Version 2.0 renamed \@sf@setoffset.

```

\cs_new:Nn \__flowfram_static_calc_offset:n
{
  \flowfram_if_tl_eq_frame_tl:NnnnTF
    \c_flowfram_compute_tl
    { static } { offset } { #1 }
  {
    \flowfram_if_frame_bool:nnnTF { static } { hasframe } { #1 }
    {
      \flowfram_set_tl_to_frame_tl:Nnnn
        \l__flowfram_frametype_tl
        { static } { frametype } { #1 }
      \cs_if_exist_use:cF
        { __flowfram_set_ \l__flowfram_frametype_tl _offset: }
        {
          \__flowfram_set_default_offset:
        }
    }
  }
  {
    \dim_zero:N \l__flowfram_offset_dim
  }
}
{
  \flowfram_set_dim_to_frame_tl:Nnnn
    \l__flowfram_offset_dim
    { static } { offset } { #1 }
}
}

```

`\@df@setoffset` Compute offset for dynamic frame #1. Stores offset value in `\l__flowfram_offset_tl`.
Version 2.0 renamed `\@sf@setoffset`.

```

\cs_new:Nn \__flowfram_dynamic_calc_offset:n
{
  \flowfram_if_tl_eq_frame_tl:NnnnTF
  \c_flowfram_compute_tl
  { dynamic } { offset } { #1 }
  {
    \flowfram_if_frame_bool:nnnTF { dynamic } { hasframe } { #1 }
    {
      \flowfram_set_tl_to_frame_tl:Nnnn
      \l__flowfram_frametype_tl
      { dynamic } { frametype } { #1 }
      \cs_if_exist_use:cF
      { __flowfram_set_ \l__flowfram_frametype_tl _offset: }
      {
        \__flowfram_set_default_offset:
      }
    }
  }
  {
    \dim_zero:N \l__flowfram_offset_dim
  }
}
{
  \flowfram_set_dim_to_frame_tl:Nnnn
  \l__flowfram_offset_dim
  { dynamic } { offset } { #1 }
}
}

```

`\@putmarginbox` Draw box representing the margin for flow frame #1. Version 2.0 renamed
`\@putmarginbox`.

```

\cs_new:Nn \__flowframe_put_margin_box:n
{
  \__flowfram_check_if_this_page:nn
  { \g__flowfram_pagecounter_tl } { #1 }
  \bool_if:NF \g__flowfram_not_this_frame_bool
  {
    \@killglue
    \legacy_if:nTF { @twoside }
    {
      \int_if_odd:nTF { \c@page }
      {
        \flowfram_set_dim_to_frame_dim:Nnnn
        \l__flowfram_x_dim
        { flow } { posx } { #1 }
        \flowfram_set_dim_to_frame_dim:Nnnn
        \l__flowfram_y_dim
        { flow } { posy } { #1 }
      }
    }
  }
}

```

```

}
{
  \flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_x_dim
  { flow } { evenx } { #1 }
  \flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_y_dim
  { flow } { eveny } { #1 }
}
}
{
  \flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_x_dim
  { flow } { posx } { #1 }
  \flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_y_dim
  { flow } { posy } { #1 }
}
\__flowfram_get_margin_pos:e
{
  \flowfram_frame_use_tl:nnn { flow } { margin } { #1 }
}
\tl_if_eq:NNTF \l__flowfram_margin_tl \c_flowfram_left_tl
{
  \dim_add:Nn \l__flowfram_x_dim
  {
    - \marginparwidth
    - \marginparsep
  }
}
{
  \dim_add:Nn \l__flowfram_x_dim
  {
    \flowfram_frame_use_dim:nnn { flow } { width } { #1 }
    + \marginparsep
  }
}
\box_move_up:nn { \l__flowfram_y_dim }
{
  \hbox_to_zero:n
  {
    \kern \l__flowfram_x_dim
    \__flowfram_do_if_draft:nn
    {
      \__flowfram_noborder_box:nnn { \marginparwidth }
      {
        \flowfram_frame_use_dim:nnn { flow } { height } { #1 }
      }
      { }
    }
  }
}

```

```

        {
        \flowfram_draft_annotate:nn { M } { \int_eval:n { #1 } }
        }
        \hss
    }
}
\ignorespaces
}

```

`\@ff@drawmargins` Draw all the margins associated with the flow frames defined on the current page. Version 2.0 renamed `\@ff@drawmargins`.

```

\cs_new:Nn \__flowfram_draw_margins:
{
  \int_step_inline:nn { \c@maxflow }
  {
    \makebox [ \c_zero_dim ] [ l ]
    {
      \__flowframe_put_margin_box:n { ##1 }
    }
  }
}

```

`\@ff@getstaticpos` Version 2.0 removed `\@ff@getstaticpos`.

`\@dostaticbox` Display the savebox associated with static frame #1

```

\cs_new:Nn \__flowfram_do_static_box:n
{
  \int_set:Nn \l__flowfram_current_static_int { #1 }
}

```

Get the frame border type.

```

\flowfram_set_tl_to_frame_tl:Nnnn
\l__flowfram_frametype_tl
{ static } { frametype } { #1 }

```

Get the frame border colour.

```

\flowfram_set_tl_to_frame_tl:Nnnn
\l__flowfram_bordercolor_tl
{ static } { bordercolor } { #1 }

```

Note that it's too late to apply the text colour as that's set when the contents are set. Get the background colour.

```

\flowfram_set_tl_to_frame_tl:Nnnn
\l__flowfram_backcolor_tl
{ static } { backcolor } { #1 }

```

Calculate the offset to take into account the border.

```

\__flowfram_static_calc_offset:n {#1}%

```

Rotate the frame by the designated angle.

```

\exp_args:Ne \rotateframe

```

```

{
\flowfram_frame_use_tl:nnn { static } { angle } { #1 }
}
{
\flowfram_if_frame_bool:nnnTF
{ static } { hasframe } { #1 }
{
\__flowfram_bordered_box:nnnn
{
\flowfram_frame_use_dim:nnn { static } { width } { #1 }
}
{
\flowfram_frame_use_dim:nnn { static } { height } { #1 }
}
{
\flowfram_frame_use_box:nnn
{ static } { content } { #1 }
}
{
\use:c { \l__flowfram_frametype_tl }
}
}
{
\__flowfram_noborder_box:nnn
{
\flowfram_frame_use_dim:nnn { static } { width } { #1 }
}
{
\flowfram_frame_use_dim:nnn { static } { height } { #1 }
}
{
\flowfram_frame_use_box:nnn
{ static } { content } { #1 }
}
}
}
}
}

```

\@dostaticbbox Now for the bounding box: Version 2.0 renamed \@dostaticbbox

```

\cs_new:Nn \__flowfram_do_static_bounding_box:n
{
\tl_clear:N \l__flowfram_bordercolor_tl
\__flowfram_static_calc_offset:n { #1 }
\__flowfram_do_if_draft:nn
{
\__flowfram_noborder_box:nnn
{
\flowfram_frame_use_dim:nnn { static } { width } { #1 }
}
{

```

```

        \flowfram_frame_use_dim:nnn { static } { height } { #1 }
      }
    {
      \flowfram_frame_use_box:nnn
        { static } { content } { #1 }
    }
  }
{
  \flowfram_draft_annotate:nnn
  { S } { \int_eval:n { #1 } }
  {
    \flowfram_frame_use_tl:nnn { static } { label } { #1 }
  }
}
}
}

```

\@putstaticbox Put the static box #1 at its given position, with its associated border. Version 2.0 renamed \@putstaticbox

```

\cs_new:Nn \__flowfram_put_static_box:n
{

```

Check the 'hide' and 'hidethis' attributes

```

\flowfram_if_frame_bool:nnnTF
{ static } { hidethis } { #1 }
{
  \__flowfram_message:nen
    { info-static-frame-hidden } { \int_eval:n { #1 } } { hidethis }
  \bool_gset_true:N \g__flowfram_not_this_frame_bool
  \flowfram_frame_set_bool_false:nnn
    { static } { hidethis } { #1 }
}
{
  \flowfram_if_frame_bool:nnnTF
  { static } { hide } { #1 }
  {
    \__flowfram_message:nen
      { info-static-frame-hidden } { \int_eval:n { #1 } } { hide }
    \bool_gset_true:N \g__flowfram_not_this_frame_bool
  }
}

```

Neither 'hide' nor 'hidethis' have been set so check the page list.

```

  \__flowfram_static_check_if_this_page:n { #1 }
}
}
\bool_if:NF \g__flowfram_not_this_frame_bool
{
  \@killglue
  \legacy_if:nTF { @twoside }
  {

```



```

\int_if_odd:nTF { \c@page }
{
  \box_move_up:nn
  {
    \flowfram_frame_use_dim:nnn { static } { posy } { #1 }
  }
  {
    \hbox_to_zero:n
    {
      \kern
      \flowfram_frame_use_dim:nnn { static } { posx } { #1 }
      \__flowfram_do_static_box:n { #1 }
      \hss
    }
  }
}
{
  \box_move_up:nn
  {
    \flowfram_frame_use_dim:nnn { static } { eveny } { #1 }
  }
  {
    \hbox_to_zero:n
    {
      \kern
      \flowfram_frame_use_dim:nnn { static } { evenx } { #1 }
      \__flowfram_do_static_box:n { #1 }
      \hss
    }
  }
}
{
  \box_move_up:nn
  {
    \flowfram_frame_use_dim:nnn { static } { posy } { #1 }
  }
  {
    \hbox_to_zero:n
    {
      \kern
      \flowfram_frame_use_dim:nnn { static } { posx } { #1 }
      \__flowfram_do_static_box:n { #1 }
      \hss
    }
  }
}
}
}
}

```

\@putstaticbbox Now for the bounding box: Version 2.0 renamed \@putstaticbbox

```
\cs_new:Nn \__flowfram_put_static_bounding_box:n
{
  \flowfram_if_frame_bool:nnnTF
  { static } { hidethis } { #1 }
  {
    \bool_gset_true:N \g__flowfram_not_this_frame_bool
    \flowfram_frame_set_bool_false:nnn
    { static } { hidethis } { #1 }
  }
  {
    \flowfram_if_frame_bool:nnnTF
    { static } { hide } { #1 }
    {
      \bool_gset_true:N \g__flowfram_not_this_frame_bool
    }
    {
      \__flowfram_static_check_if_this_page:n { #1 }
    }
  }
  \bool_if:NF \g__flowfram_not_this_frame_bool
  {
    \@killglue
    \legacy_if:nTF { @twoside }
    {
      \int_if_odd:nTF { \c@page }
      {
        \box_move_up:nn
        {
          \flowfram_frame_use_dim:nnn { static } { posy } { #1 }
        }
        {
          \hbox_to_zero:n
          {
            \kern
            \flowfram_frame_use_dim:nnn { static } { posx } { #1 }
            \__flowfram_do_static_bounding_box:n { #1 }
            \hss
          }
        }
      }
    }
  }
  \box_move_up:nn
  {
    \flowfram_frame_use_dim:nnn { static } { eveny } { #1 }
  }
  {
    \hbox_to_zero:n
  }
}
```

```

        {
            \kern
            \flowfram_frame_use_dim:nnn { static } { evenx } { #1 }
            \__flowfram_do_static_bounding_box:n { #1 }
            \hss
        }
    }
}
{
\box_move_up:nn
{
\flowfram_frame_use_dim:nnn { static } { posy } { #1 }
}
{
\hbox_to_zero:n
{
\kern
\flowfram_frame_use_dim:nnn { static } { posx } { #1 }
\__flowfram_do_static_bounding_box:n { #1 }
\hss
}
}
}
\ignorespaces
}
}

```

`\@resetst@tics` Clear the contents of all the static frames that have the clear option set. Version 2.0 renamed `\@resetst@tics`

```

\cs_new:Nn \__flowfram_reset_statics:
{
\int_step_inline:nn { \c@maxstatic }
{
Has the clear flag been set?
\flowfram_if_frame_bool:nnnT
{ static } { clear } { ##1 }
{
Set the contents of the box to empty
\flowfram_frame_clear_box:nnn
{ static } { content } { ##1 }
}
}
}

```

`\@resetdyn@mics` Clear the contents of the dynamic frames that have the clear option set. Version 2.0 renamed `\@resetdyn@mics`

```

\cs_new:Nn \__flowfram_reset_dynamics:

```

```

{
  \int_step_inline:nn { \c@maxdynamic }
  {
    Has the clear flag been set?
      \flowfram_if_frame_bool:nnnT
      { dynamic } { clear } { ##1 }
      {
        Clear the contents.
          \flowfram_frame_clear_tl:nnn
          { dynamic } { content } { ##1 }
        }
      }
  }
}

```

Dynamic frame content is now put inside a minipage instead of a `\parbox` (`\shapepar` has problems in a `\parbox`).

```

\cs_new:Nn \__flowfram_parbox:nnnn
{
  \begin { minipage } [ c ] [ #1 ] [ #2 ] { #3 }
  #4
  \end { minipage }
}
\cs_generate_variant:Nn \__flowfram_parbox:nnnn { eeem }

```

`\@dodfparbox` Display contents of dynamic box (contents stored in `\l__flowfram_contents_tl`, style given by `\l__flowfram_style_tl`): Version 2.0 renamed `\@dodfparbox`.

```

\cs_new:Nn \__flowfram_do_dynamic_content:n
{
  \flowfram_set_tl_to_frame_tl:Nnnn
  \l__flowfram_parindent_tl
  { dynamic } { parindent } { #1 }
  \flowfram_set_tl_to_frame_tl:Nnnn
  \l__flowfram_parshape_tl
  { dynamic } { shape } { #1 }
  \__flowfram_get_shape:V \l__flowfram_parshape_tl
  \int_case:nnF { \l__flowfram_shape_int }
  {
    { \c_flowfram_shape_type_parshape_int }
    {
      \parshape
        \__flowfram_parbox:eeem
        {
          \dynamicframeheight { #1 }
        }
        {
          \flowfram_frame_use_tl:nnn
          { dynamic } { valign } { #1 }
        }
      }
    }
  }
}

```

```

    {
      \dynamicframewidth { #1 }
    }
  {
    \dim_set:Nn \parindent { \l__flowfram_parindent_tl }
    \use:c { \l__flowfram_style_tl }
    {
      \__flowfram_set_shaped_section_headings:
      \let \par \FLFsimpar
      \l__flowfram_parshape_tl
      \l__flowfram_contents_tl
      \endgraf
    }
  }
}
{ \c_flowfram_shape_type_shapepar_int }
{
\shapepar or \Shapepar
  \__flowfram_parbox:eeen
  {
    \dynamicframeheight { #1 }
  }
  {
    \flowfram_frame_use_tl:nnn
    { dynamic } { valign } { #1 }
  }
  {
    \dynamicframewidth { #1 }
  }
  {
    \dim_set:Nn \parindent { \l__flowfram_parindent_tl }
    \use:c { \l__flowfram_style_tl }
    {
      \@ff@disablesec
      \let \par \FLFsimpar
      \l__flowfram_parshape_tl
      \l__flowfram_contents_tl
      \endgraf
    }
  }
}
}
{
no shape
  \__flowfram_parbox:eeen
  {
    \dynamicframeheight { #1 }
  }
  {

```

```

\flowfram_frame_use_tl:nnn
{ dynamic } { valign } { #1 }
}
{
\dynamicframewidth { #1 }
}
{
\dim_set:Nn \parindent { \l__flowfram_parindent_tl }
\use:c { \l__flowfram_style_tl }

```

This will be picked up as an argument if the style is a text-block command but won't add extra grouping otherwise:

```

\l__flowfram_contents_tl
}
}
}

```

`\@dodynamicbox` Typeset the dynamic box with its associated border. Version 2.0 renamed

```

\@dodynamicbox
\cs_new:Nn \__flowfram_do_dynamic_box:n
{
\int_set:Nn \l__flowfram_current_dynamic_int { #1 }

```

Get the frame border type.

```

\flowfram_set_tl_to_frame_tl:Nnnn
\l__flowfram_frametype_tl
{ dynamic } { frametype } { #1 }

```

Get the frame border colour.

```

\flowfram_set_tl_to_frame_tl:Nnnn
\l__flowfram_bordercolor_tl
{ dynamic } { bordercolor } { #1 }

```

Get the text colour.

```

\flowfram_set_tl_to_frame_tl:Nnnn
\l__flowfram_textcolor_tl
{ dynamic } { textcolor } { #1 }

```

Get the background colour.

```

\flowfram_set_tl_to_frame_tl:Nnnn
\l__flowfram_backcolor_tl
{ dynamic } { backcolor } { #1 }

```

Get the style (cname to apply to the content).

```

\flowfram_set_tl_to_frame_tl:Nnnn
\l__flowfram_style_tl
{ dynamic } { style } { #1 }

```

Get the content.

```

\flowfram_set_tl_to_frame_tl:Nnnn
\l__flowfram_contents_tl
{ dynamic } { content } { #1 }

```

Calculate the offset to take into account the border.

```

\__flowfram_dynamic_calc_offset:n {#1}%
\exp_args:Ne \rotateframe
{
  \flowfram_frame_use_tl:nnn { dynamic } { angle } { #1 }
}
{
  \flowfram_if_frame_bool:nnnTF
  { dynamic } { hasframe } { #1 }
  {
    \__flowfram_bordered_box:nnnn
    {
      \flowfram_frame_use_dim:nnn { dynamic } { width } { #1 }
    }
    {
      \flowfram_frame_use_dim:nnn { dynamic } { height } { #1 }
    }
    {
      \__flowfram_do_dynamic_content:n { #1 }
    }
    {
      \use:c { \l__flowfram_frametype_tl }
    }
  }
}
{
  \__flowfram_noborder_box:nnn
  {
    \flowfram_frame_use_dim:nnn { dynamic } { width } { #1 }
  }
  {
    \flowfram_frame_use_dim:nnn { dynamic } { height } { #1 }
  }
  {
    \__flowfram_do_dynamic_content:n { #1 }
  }
}
}
}

```

\@dodynamicbbox Now for the bounding box. Version 2.0 renamed \@dodynamicbbox

```

\cs_new:Nn \__flowfram_do_dynamic_bounding_box:n
{
  \tl_clear:N \l__flowfram_bordercolor_tl

```

Calculate the offset to take into account the border.

```

  \__flowfram_dynamic_calc_offset:n { #1 }
  \__flowfram_do_if_draft:nn
  {
    \__flowfram_noborder_box:nnn
    {

```

```

        \flowfram_frame_use_dim:nnn { dynamic } { width } { #1 }
    }
    {
        \flowfram_frame_use_dim:nnn { dynamic } { height } { #1 }
    }
    {
        \__flowfram_parbox:eeen
        {
            \dynamicframeheight { #1 }
        }
        {
            \flowfram_frame_use_tl:nnn
            { dynamic } { valign } { #1 }
        }
        {
            \dynamicframewidth { #1 }
        }
        { }
    }
}
{
    \flowfram_draft_annotate:nnn
    { D } { \int_eval:n { #1 } }
    {
        \flowfram_frame_use_tl:nnn { dynamic } { label } { #1 }
    }
}
}
}

```

\@putdynamicbox Put the dynamic frame #1 at its given position. Version 2.0 renamed

\@putdynamicbox

```

\cs_new:Nn \__flowfram_put_dynamic_box:n
{

```

Check the 'hide' and 'hidethis' attributes

```

\flowfram_if_frame_bool:nnnTF
{ dynamic } { hidethis } { #1 }
{
    \__flowfram_message:nen
    { info-dynamic-frame-hidden } { \int_eval:n { #1 } } { hidethis }
    \bool_gset_true:N \g__flowfram_not_this_frame_bool
    \flowfram_frame_set_bool_false:nnn
    { dynamic } { hidethis } { #1 }
}
{
    \flowfram_if_frame_bool:nnnTF
    { dynamic } { hide } { #1 }
    {
        \__flowfram_message:nen
        { info-dynamic-frame-hidden } { \int_eval:n { #1 } } { hide }
    }
}
}

```



```

        \bool_gset_true:N \g__flowfram_not_this_frame_bool
      }
    {
Neither 'hide' nor 'hidethis' have been set so check the page list.
      \__flowfram_dynamic_check_if_this_page:n { #1 }
    }
  }
\bool_if:NF \g__flowfram_not_this_frame_bool
{
  \@killglue
  \legacy_if:nTF { @twoside }
  {
    \int_if_odd:nTF { \c@page }
    {
      \box_move_up:nn
      {
        \flowfram_frame_use_dim:nnn { dynamic } { posy } { #1 }
      }
      {
        \hbox_to_zero:n
        {
          \kern
          \flowfram_frame_use_dim:nnn { dynamic } { posx } { #1 }
          \__flowfram_do_dynamic_box:n { #1 }
          \hss
        }
      }
    }
  }
}
{
  \box_move_up:nn
  {
    \flowfram_frame_use_dim:nnn { dynamic } { eveny } { #1 }
  }
  {
    \hbox_to_zero:n
    {
      \kern
      \flowfram_frame_use_dim:nnn { dynamic } { evenx } { #1 }
      \__flowfram_do_dynamic_box:n { #1 }
      \hss
    }
  }
}
}
{
  \box_move_up:nn
  {
    \flowfram_frame_use_dim:nnn { dynamic } { posy } { #1 }
  }
}

```

```

    {
      \hbox_to_zero:n
      {
        \kern
          \flowfram_frame_use_dim:nnn { dynamic } { posx } { #1 }
          \__flowfram_do_dynamic_box:n { #1 }
          \hss
        }
      }
    }
  \ignorespaces
}
}

```

\@putdynamicbbox Bounding box. Version 2.0 renamed \@putdynamicbbox

```

\cs_new:Nn \__flowfram_put_dynamic_bounding_box:n
{
  \flowfram_if_frame_bool:nnnTF
  { dynamic } { hidethis } { #1 }
  {
    \bool_gset_true:N \g__flowfram_not_this_frame_bool
    \flowfram_frame_set_bool_false:nnn
    { dynamic } { hidethis } { #1 }
  }
  {
    \flowfram_if_frame_bool:nnnTF
    { dynamic } { hide } { #1 }
    {
      \bool_gset_true:N \g__flowfram_not_this_frame_bool
    }
    {
      Neither 'hide' nor 'hidethis' have been set so check the page list.
      \__flowfram_dynamic_check_if_this_page:n { #1 }
    }
  }
  \bool_if:NF \g__flowfram_not_this_frame_bool
  {
    \@killglue
    \legacy_if:nTF { @twoside }
    {
      \int_if_odd:nTF { \c@page }
      {
        \box_move_up:nn
        {
          \flowfram_frame_use_dim:nnn { dynamic } { posy } { #1 }
        }
      }
      {
        \hbox_to_zero:n
        {

```

```

        \kern
        \flowfram_frame_use_dim:nnn { dynamic } { posx } { #1 }
        \__flowfram_do_dynamic_bounding_box:n { #1 }
        \hss
    }
}
{
\box_move_up:nn
{
\flowfram_frame_use_dim:nnn { dynamic } { eveny } { #1 }
}
{
\hbox_to_zero:n
{
\kern
\flowfram_frame_use_dim:nnn { dynamic } { evenx } { #1 }
\__flowfram_do_dynamic_bounding_box:n { #1 }
\hss
}
}
}
}
{
\box_move_up:nn
{
\flowfram_frame_use_dim:nnn { dynamic } { posy } { #1 }
}
{
\hbox_to_zero:n
{
\kern
\flowfram_frame_use_dim:nnn { dynamic } { posx } { #1 }
\__flowfram_do_dynamic_bounding_box:n { #1 }
\hss
}
}
}
}
\ignorespaces
}
}

```

\@@doheader Do standard header in the standard place. Version 2.0 renamed \@@doheader

```

\cs_new:Nn \__flowfram_do_standard_header:
{
\box_move_up:nn
{
\typeblockheight + \headsep
}
}

```

```

\ vbox_to_ht:nn { \ headheight }
{
\ hbox_to_zero:n
{
\ @dothehead \ hss
}
\ vss
}
}
}

```

`\@@dofooter` Do standard footer in the standard place. Version 2.0 renamed `\@@dofooter`

```

\ cs_new:Nn \__flowfram_do_standard_footer:
{
\ box_move_down:nn
{ \ footskip }
{
\ hbox_to_zero:n
{
\ @dothefoot \ hss
}
}
}
}

```

`\@s@tfr@mes` Version 2.0 renamed `\@s@tfr@mes`

```

\ box_new:N \l__all_content_box
\ dim_new:N \l__all_content_height_dim
\ cs_new:Nn \__flowfram_set_frames:n
{
\ group_begin:
\ dim_set_eq:NN \l__all_content_height_dim \typeblockheight
\ setbox
\l__all_content_box
\ hbox_to_wd:nn
{ \ typeblockwidth }
{
\ hbox:n { #1 }
\ hss
}
\ ht \l__all_content_box
\l__all_content_height_dim
\ dp \l__all_content_box \c_zero_dim
\ mbox
{
\ box_use_drop:N \l__all_content_box
}
\ group_end:
}
}

```

`\@ff@doallflowframes` Puts all the flow frames defined on the current page. Version 2.0 renamed

```

\@ff@doallflowframes
\cs_new:Nn \__flowfram_do_all_flow_frames:
{
  \int_step_function:nN { \c@maxflow }
  \__flowfram_put_flow_box:n
}

```

\@ff@doallflowframesbbox Flow frame bounding boxes. Version 2.0 renamed \@ff@doallflowframesbbox

```

\cs_new:Nn \__flowfram_do_all_flow_bounding_boxes:
{
  \int_step_function:nN { \c@maxflow }
  \__flowfram_put_flow_bounding_box:n
}

```

\@ff@doallstatics Puts all static frames defined on the current page. Version 2.0 renamed

```

\@ff@doallstatics
\cs_new:Nn \__flowfram_do_all_static_frames:
{
  \int_step_function:nN { \c@maxstatic }
  \__flowfram_put_static_box:n
  \int_zero:N \l__flowfram_current_static_int
}

```

\@ff@doallstaticsbbox Static frame bounding boxes. Version 2.0 renamed \@ff@doallstaticsbbox

```

\cs_new:Nn \__flowfram_do_all_static_bounding_boxes:
{
  \int_step_function:nN { \c@maxstatic }
  \__flowfram_put_static_bounding_box:n
}

```

\@ff@doalldynamics Puts all the dynamic frames defined on the current page. Version 2.0 renamed

```

\@ff@doalldynamics
\cs_new:Nn \__flowfram_do_all_dynamic_frames:
{
  \int_step_function:nN { \c@maxdynamic }
  \__flowfram_put_dynamic_box:n
  \int_zero:N \l__flowfram_current_dynamic_int
}

```

\@ff@doalldynamicsbbox Dynamic frame bounding boxes. Version 2.0 renamed \@ff@doalldynamicsbbox.

```

\cs_new:Nn \__flowfram_do_all_dynamic_bounding_boxes:
{
  \int_step_function:nN { \c@maxdynamic }
  \__flowfram_put_dynamic_bounding_box:n
}

```

`\@ff@dotypeblock` Draw typeblock frame if draft. Version 2.0 renamed `\@ff@dotypeblock`.

```
\cs_new:Nn \__flowfram_do_typeblock:
{
  \makebox [ \c_zero_dim ] [ l ]
  {
    \__flowfram_do_if_draft:nnn
    { \setffdrafttypeblockcolor }
    {
      \vbox_to_ht:nn { \typeblockheight }
      {
        \hbox_to_wd:nn \typeblockwidth { }
      }
    }
  }
}
```

`\ffevenoffset` Version 2.0 removed `\ffevenoffset`.

`\@ff@do@allframes` Put all frames defined on the current page. Version 2.0 renamed `\@ff@do@allframes`.

```
\cs_new:Nn \__flowfram_do_all_frames:
{
  \hbox_to_wd:nn { \typeblockwidth }
  {
    \__flowfram_set_frames:n
    {
      \__flowfram_do_all_static_frames:
      \__flowfram_do_standard_header:
      \__flowfram_do_standard_footer:
      \__flowfram_do_all_flow_frames:
      \__flowfram_do_all_dynamic_frames:
      \ifshowtypeblock
      \__flowfram_do_typeblock:
      \fi
      \ifshowframebbox
      \__flowfram_do_all_static_bounding_boxes:
      \__flowfram_do_all_flow_bounding_boxes:
      \__flowfram_do_all_dynamic_bounding_boxes:
      \fi
      \ifshowmargins
      \__flowfram_draw_margins:
      \fi
    }
  }
}
```

`\@outputdblcol` This was modified from the output routine for standard two column format. After `__flowfram_get_next_column:N` has finished, the register `\g__flowfram_current_page_int`

contains the page that the next flow frame is on. If `\g__flowfram_current_page_int` 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 T_EX can continue with the rest of the document.

```

\int_new:N \g__flowfram_next_frame_int
\renewcommand*\@outputdblcol
{
  \__flowfram_output_dbl_col:
}

\cs_new:Nn \__flowfram_output_dbl_col:
{
  \int_gset_eq:NN \g__flowfram_next_frame_int \c@thisframe

  \int_gset:Nn \g__flowfram_current_page_int
    { \g__flowfram_pagecounter_tl }
  \__flowfram_get_next_column:N \g__flowfram_next_frame_int
  \bool_if:NTF \g__flowfram_new_page_bool
    {

```

Next flow frame starts on new page.

```

\global \@firstcolumntrue
\__flowfram_set_column_box:n { \c@thisframe }
\legacy_if:nT { @specialpage }
{
  \global \@specialpagefalse
  \@nameuse { ps@ \@specialstyle }
  \relax
}
\@dodynamicthehead
\@dodynamictheft
\vbaddness=\@M
\setbox \@outputbox
  \vbox_to_ht:nn { \typeblockheight }
  {
    \__flowfram_do_all_frames:
  }
\@combinedblfloats
\@outputpage

```

Shipout pages without flow frames. Any static or dynamic frames on those pages should be included.

```

\int_gadd:Nn \g__flowfram_current_page_int
  { - \g__flowfram_pagecounter_tl }
\bool_while_do:nn
{
  \int_compare_p:nNn
    { \g__flowfram_current_page_int } > { \c_zero_int }
}
{

```

```

\int_gdecr:N \g__flowfram_current_page_int
\setbox \@outputbox
\ vbox_to_ht:nn { \typeblockheight }
{
  \__flowfram_do_all_frames:
}
\@outputpage
}
\begingroup
\@dblfloatplacement
\@startdblcolumn
\@whilesw \if@colmade \fi
{ \@outputpage \@startdblcolumn }%
\endgroup
\__flowfram_reset_statics:
\__flowfram_reset_dynamics:
}
{

```

Still on same page, save contents of box255

```
\global \@firstcolumnfalse
```

Set this frame's column box to the output box.

```

\__flowfram_set_column_box:n { \c@thisframe }
}
\int_gset_eq:NN \c@thisframe \g__flowfram_next_frame_int
\__flowfram_set_column:n { \c@thisframe }
}

```

```

\newcommand \@flowfram@update@col@count [ 1 ]
{
  \int_gset_eq:NN \col@number \c_one_int
  \int_step_inline:nnn
  { #1 + \c_one_int } { \c@maxflow }
  {
    \exp_args:Ne
    \flowfram_if_in_frame_clist:nnnF
    { \int_eval:n { \g__flowfram_pagecounter_t1 } }
    { flow } { excludelist } { ##1 }
  }
}

```

This page is in this not in frame's exclusion list so now check if in frame's page list.

```

\exp_args:Ne
\flowfram_if_frame_has_morepages:nnnT
{ \int_eval:n { \g__flowfram_pagecounter_t1 } }
{ flow } { ##1 }
{
  \int_gincr:N \col@number
}
}

```



```

    }
  \int_compare:nNnT
    { \col@number } > { \c_one_int }
  {
    \global\@twocolumntrue
  }
}

```

`\@dblfloatplacement` Version 2.0 no longer redefining `\@dblfloatplacement` (no support for column spanning floats).

1.10 Switching Frames On and Off For Subsequent Pages

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

```

\NewDocumentCommand \flowswitchonnext { s m }
{
  \IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_flow_id:n { ##1 }
      \__flowfram_flow_switch_on_next:n { \l__flowfram_id_int }
    }
  }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_flow_switch_on_next:n { ##1 }
    }
  }
}

```

`\@sflowswitchonnext` Version 2.0 removed `\@sflowswitchonnext`

`\@flowswitchonnext` The unstarred version uses IDNs. Version 2.0 removed `\@flowswitchonnext`

Switch on a given frame by its IDN.

```

\cs_new:Nn \__flowfram_flow_switch_on_next:n
{
  Is this frame already on?
  \__flowfram_check_if_this_page:nn
    { \g__flowfram_pagecounter_tl } { #1 }
  \bool_if:NTF \g__flowfram_not_this_frame_bool
  {
    \tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
    {
      \exp_not:N \flowsetpagelist
      { \int_eval:n { #1 } }
      { > \int_eval:n { \g__flowfram_pagecounter_tl } }
    }
  }
}

```

```

    }
  }
  {
    \tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
    {
      \exp_not:N \flowsetpagelist
      { \int_eval:n { #1 } }
      { \int_eval:n { \g__flowfram_pagecounter_tl } ,
        > \int_eval:n { \g__flowfram_pagecounter_tl }
      }
    }
  }
}

```

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

```

\NewDocumentCommand \flowswitchonnextodd { s m }
{
  \int_set:Nn \l__flowfram_tmpb_int
  { \g__flowfram_pagecounter_tl }
  \int_if_odd:nT { \l__flowfram_tmpb_int }
  {
    \int_incr:N \l__flowfram_tmpb_int
  }
  \IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_flow_id:n { ##1 }
      \__flowfram_flow_switch_on_next_odd:n { \l__flowfram_id_int }
    }
  }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_flow_switch_on_next_odd:n { ##1 }
    }
  }
}

\cs_new:Nn \__flowfram_flow_switch_on_next_odd:n
{
  Is this frame already on?
  \__flowfram_check_if_this_page:nn
  { \g__flowfram_pagecounter_tl } { #1 }
  \clist_clear:N \l__flowfram_pages_clist
  \bool_if:NF \g__flowfram_not_this_frame_bool
  {
    \clist_put_right:Ne \l__flowfram_pages_clist
    { \int_eval:n { \g__flowfram_pagecounter_tl } }
  }
}

```

Is this frame already switched on for the next page?

```
\__flowfram_check_if_this_page:nn { \l__flowfram_tmpb_int } { #1 }
\int_compare:nNnF
  { \l__flowfram_tmpb_int } { \g__flowfram_pagecounter_tl }
  {
    \bool_if:NF \g__flowfram_not_this_frame_bool
    {
      \clist_put_right:Ne \l__flowfram_pages_clist
        { \int_eval:n { \l__flowfram_tmpb_int } }
    }
  }
\clist_put_right:Ne \l__flowfram_pages_clist
  { > \int_eval:n { \l__flowfram_tmpb_int } }
\tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
  {
    \exp_not:N \flowsetpagelist
    { \int_eval:n { #1 } }
    { \clist_use:Nn \l__flowfram_pages_clist { , } }
  }
}
```

\@sflowswitchonnextodd Version 2.0 removed \@sflowswitchonnextodd.

\@flowswitchonnextodd Version 2.0 removed \@flowswitchonnextodd.

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

```
\NewDocumentCommand \flowswitchoffnext { s m }
{
  \IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_flow_id:n { ##1 }
      \__flowfram_flow_switch_off_next:n { \l__flowfram_id_int }
    }
  }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_flow_switch_off_next:n { ##1 }
    }
  }
}
```

```
\cs_new:Nn \__flowfram_flow_switch_off_next:n
{
```

Is this frame already off on this page?

```
\__flowfram_check_if_this_page:nn
  { \g__flowfram_pagecounter_tl } { #1 }
```

```

\bool_if:NTF \g__flowfram_not_this_frame_bool
{
  \clist_set:Nn \l__flowfram_pages_clist { none }
}
{
  \tl_set:Ne \l__flowfram_pages_clist
  {
    \int_eval:n { \g__flowfram_pagecounter_tl }
  }
}
\tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
{
  \exp_not:N \flowsetpagelist
  { \int_eval:n { #1 } }
  { \l__flowfram_pages_clist }
}
}

```

\@sflowswitchoffnext Version 2.0 removed \@sflowswitchoffnext.

\@flowswitchoffnext Version 2.0 removed \@flowswitchoffnext.

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

```

\NewDocumentCommand \flowswitchoffnextodd { s m }
{
  \int_set:Nn \l__flowfram_tmpb_int
  { \g__flowfram_pagecounter_tl }
  \int_if_odd:nT { \g__flowfram_pagecounter_tl }
  {
    \int_incr:N \l__flowfram_tmpb_int
  }
  \IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_flow_id:n { ##1 }
      \__flowfram_flow_switch_off_next_odd:n { \l__flowfram_id_int }
    }
  }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_flow_switch_off_next_odd:n { ##1 }
    }
  }
}
}

\cs_new:Nn \__flowfram_flow_switch_off_next_odd:n
{

```

Is this frame already off on this page?

```
\__flowfram_check_if_this_page:nn
{ \g__flowfram_pagecounter_t1 } { #1 }
\bool_if:NTF \g__flowfram_not_this_frame_bool
{
```

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?

```
\int_compare:nNnTF
{ \g__flowfram_pagecounter_t1 } = { \l__flowfram_tmpb_int }
{
```

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

```
\tl_set:Nn \l__flowfram_next_pages_t1 { none }
}
{
```

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

```
\__flowfram_check_if_this_page:nn
{ \l__flowfram_tmpb_int } { #1 }
\bool_if:NTF \g__flowfram_not_this_frame_bool
{
```

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

```
\tl_set:Nn \l__flowfram_next_pages_t1 { none }
}
{
```

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

```
\tl_set:Ne \l__flowfram_next_pages_t1
{ \int_use:N \l__flowfram_tmpb_int }
}
}
{
```

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?

```
\int_compare:nNnTF
{ \g__flowfram_pagecounter_t1 } = { \l__flowfram_tmpb_int }
{
```

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

```
\tl_set:Ne \l__flowfram_next_pages_t1
{ \int_eval:n { \g__flowfram_pagecounter_t1 } }
}
{
```

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

```
\__flowfram_check_if_this_page:nn
{ \l__flowfram_tmpb_int } { #1 }
\bool_if:NTF \g__flowfram_not_this_frame_bool
{
```

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

```

\l_set:Ne \l__flowfram_next_pages_tl
  { \int_eval:n { \g__flowfram_pagecounter_tl } }
}
{

```

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

```

\l_set:Ne \l__flowfram_next_pages_tl
  {
    \int_eval:n { \g__flowfram_pagecounter_tl } ,
    \int_use:N \l__flowfram_tmpb_int
  }
}
}
\l_gput_right:Ne \g__flowfram_output_adjust_frames_tl
{
  \exp_not:N \flowsetpagelist
  { \int_eval:n { #1 } }
  { \l__flowfram_next_pages_tl }
}
}

```

`\@sflowswitchoffnextodd` Version 2.0 removed `\@sflowswitchoffnextodd`.

`\@flowswitchoffnextodd` Version 2.0 removed `\sflowswitchoffnextodd`.

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

```

\NewDocumentCommand \flowswitchonnextonly { s m }
{
  \int_set:Nn \l__flowfram_tmpb_int
    { \g__flowfram_pagecounter_tl + \c_one_int }
  \IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_flow_id:n { ##1 }
      \__flowfram_flow_switch_on_next_only:n { \l__flowfram_id_int }
    }
  }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_flow_switch_on_next_only:n { ##1 }
    }
  }
}
\cs_new:Nn \__flowfram_flow_switch_on_next_only:n
{

```

Is this frame already on?

```
  \__flowfram_check_if_this_page:nn
    { \g__flowfram_pagecounter_tl } { #1 }
  \bool_if:NTF \g__flowfram_not_this_frame_bool
    {
```

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

```
  \tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
    {
      \exp_not:N \flowsetpagelist
      { \int_eval:n { #1 } }
      { \int_use:N \l__flowfram_tmpb_int }
    }
  }
  {
```

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

```
  \tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
    {
      \exp_not:N \flowsetpagelist
      { \int_eval:n { #1 } }
      {
        \int_eval:n { \g__flowfram_pagecounter_tl } ,
        \int_use:N \l__flowfram_tmpb_int
      }
    }
  }
}
```

`\@sflowswitchonnextonly` Version 2.0 removed `\@sflowswitchonnextonly`.

`\@flowswitchonnextonly` Version 2.0 removed `\@flowswitchonnextonly`. The unstarred version uses IDNs.

`\flowswitchonnextoddone` Switch on the listed flow frames for just the next odd page

```
\NewDocumentCommand \flowswitchonnextoddone { s m }
{
  \IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_flow_id:n { ##1 }
      \__flowfram_flow_switch_on_next_odd_only:n { \l__flowfram_id_int }
    }
  }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_flow_switch_on_next_odd_only:n { ##1 }
    }
  }
}
```

```

    }
  }

\cs_new:Nn \__flowfram_flow_switch_on_next_odd_only:n
{

```

Is this frame already on?

```

  \__flowfram_check_if_this_page:nn
  { \g__flowfram_pagecounter_tl } { #1 }
  \bool_if:NTF \g__flowfram_not_this_frame_bool
  {

```

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?

```

  \int_if_odd:nTF { \g__flowfram_pagecounter_tl }
  {

```

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

```

  \int_set:Nn \l__flowfram_tmpb_int
  { \g__flowfram_pagecounter_tl + \c_one_int }
  \__flowfram_check_if_this_page:nn
  { \l__flowfram_tmpb_int } { #1 }
  \bool_if:NTF \g__flowfram_not_this_frame_bool
  {

```

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.

```

  \int_incr:N \l__flowfram_tmpb_int
  \clist_set:Ne \l__flowfram_pages_clist
  { \int_use:N \l__flowfram_tmpb_int }
  }
  {

```

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.

```

  \clist_set:Ne \l__flowfram_pages_clist
  { \int_use:N \l__flowfram_tmpb_int }
  \int_incr:N \l__flowfram_tmpb_int
  \clist_put_right:Ne \l__flowfram_pages_clist
  { \int_use:N \l__flowfram_tmpb_int }
  }
  }
  {

```

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

```

  \int_set:Nn \l__flowfram_tmpb_int
  { \g__flowfram_pagecounter_tl }
  \int_incr:N \l__flowfram_tmpb_int
  \clist_set:Ne \l__flowfram_pages_clist
  { \int_use:N \l__flowfram_tmpb_int }

```



```

    }
  }
  {

```

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?

```

    \int_if_odd:nTF { \g__flowfram_pagecounter_tl }
    {

```

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

```

    \int_set:Nn \l__flowfram_tmpb_int
    {
      \g__flowfram_pagecounter_tl + \c_one_int
    }
    \__flowfram_check_if_this_page:nn
    { \l__flowfram_tmpb_int } { #1 }
    \bool_if:NTF \g__flowfram_not_this_frame_bool
    {

```

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.

```

    \int_incr:N \l__flowfram_tmpb_int
    \clist_set:Ne \l__flowfram_pages_clist
    {
      \int_eval:n { \g__flowfram_pagecounter_tl } ,
      \int_use:N \l__flowfram_tmpb_int
    }
  }
  {

```

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.

```

    \int_incr:N \l__flowfram_tmpb_int
    \clist_set:Ne \l__flowfram_pages_clist
    {
      \int_eval:n { \g__flowfram_pagecounter_tl } -
      \int_use:N \l__flowfram_tmpb_int
    }
  }
}
{

```

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.

```

    \int_set:Nn \l__flowfram_tmpb_int
    {
      \g__flowfram_pagecounter_tl + \c_one_int
    }
    \clist_set:Ne \l__flowfram_pages_clist

```

```

        {
          \int_eval:n { \g__flowfram_pagecounter_tl } ,
          \int_use:N \l__flowfram_tmpb_int
        }
      }
    }
  \tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
  {
    \exp_not:N \flowsetpagelist
    { \int_eval:n { #1 } }
    { \l__flowfram_pages_clist }
  }
}

```

\@sflowswitchonnexthodonly Version 2.0 removed \@sflowswitchonnexthodonly.

\@flowswitchonnexthodonly Version 2.0 removed \@flowswitchonnexthodonly.

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

```

\NewDocumentCommand \flowswitchoffnextonly { s m }
{
  \int_set:Nn \l__flowfram_tmpb_int
  {
    \g__flowfram_pagecounter_tl + \c_one_int
  }
  \IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_flow_id:n { ##1 }
      \__flowfram_flow_switch_off_next_only:n { \l__flowfram_id_int }
    }
  }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_flow_switch_off_next_only:n { ##1 }
    }
  }
}

\cs_new:Nn \__flowfram_flow_switch_off_next_only:n
{
  \tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
  {
    \exp_not:N \flowaddexclusion
    { \int_eval:n { #1 } }
    { \int_use:N \l__flowfram_tmpb_int }
  }
}

```

\@sflowswitchoffnextonly Version 2.0 removed \@sflowswitchoffnextonly.

\@flowswitchoffnextonly Version 2.0 removed \@flowswitchoffnextonly.

\flowswitchoffnextoddoneonly Switch off the listed flow frames for just the next odd page

```
\NewDocumentCommand \flowswitchoffnextoddoneonly { s m }
{
  \int_set:Nn \l__flowfram_tmpb_int
  {
    \g__flowfram_pagecounter_tl + \c_one_int
  }
  \int_if_odd:nF { \l__flowfram_tmpb_int }
  {
    \int_incr:N \l__flowfram_tmpb_int
  }
  \IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_flow_id:n { ##1 }
      \__flowfram_flow_switch_off_next_odd_only:n { \l__flowfram_id_int }
    }
  }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_flow_switch_off_next_odd_only:n { ##1 }
    }
  }
}

\cs_new:Nn \__flowfram_flow_switch_off_next_odd_only:n
{
  \tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
  {
    \exp_not:N \flowaddexclusion
    { \int_eval:n { #1 } }
    { \int_use:N \l__flowfram_tmpb_int }
  }
}
```

\@sflowswitchoffnextoddoneonly Version 2.0 removed \@sflowswitchoffnextoddoneonly.

\@flowswitchoffnextoddoneonly Version 2.0 removed \@flowswitchoffnextoddoneonly.

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

```
\NewDocumentCommand \dynamicswitchonnext { s m }
{
  \IfBooleanTF { #1 }
  {
```

```

\clist_map_inline:nn { #2 }
{
  \__flowfram_get_dynamic_id:n { ##1 }
  \__flowfram_dynamic_switch_on_next:n { \l__flowfram_id_int }
}
}
{
\clist_map_inline:nn { #2 }
{
  \__flowfram_dynamic_switch_on_next:n { ##1 }
}
}
}
\cs_new:Nn \__flowfram_dynamic_switch_on_next:n
{
Is this frame already on?
\__flowfram_dynamic_check_if_this_page:nn
{ \g__flowfram_pagecounter_tl } { #1 }
\bool_if:NTF \g__flowfram_not_this_frame_bool
{
\tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
{
  \exp_not:N \dynamicsetpagelist
  { \int_eval:n { #1 } }
  { > \int_eval:n { \g__flowfram_pagecounter_tl } }
}
}
{
\tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
{
  \exp_not:N \dynamicsetpagelist
  { \int_eval:n { #1 } }
  {
    \int_eval:n { \g__flowfram_pagecounter_tl } ,
    > \int_eval:n { \g__flowfram_pagecounter_tl }
  }
}
}
}
}

```

\@sdynamicswitchonnext Version 2.0 removed \@sdynamicswitchonnext.

\@dynamicswitchonnext Version 2.0 removed \@dynamicswitchonnext.

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

```

\NewDocumentCommand \dynamicswitchonnextodd { s m }
{
  \int_set:Nn \l__flowfram_tmpb_int

```

```

    { \g__flowfram_pagecounter_tl }
\int_if_odd:nT { \l__flowfram_tmpb_int }
{
  \int_incr:N \l__flowfram_tmpb_int
}
\IfBooleanTF { #1 }
{
  \clist_map_inline:nn { #2 }
  {
    \__flowfram_get_dynamic_id:n { ##1 }
    \__flowfram_dynamic_switch_on_next_odd:n { \l__flowfram_id_int }
  }
}
{
  \clist_map_inline:nn { #2 }
  {
    \__flowfram_dynamic_switch_on_next_odd:n { ##1 }
  }
}
}

\cs_new:Nn \__flowfram_dynamic_switch_on_next_odd:n
{

```

Is this frame already on?

```

  \__flowfram_dynamic_check_if_this_page:nn
  { \g__flowfram_pagecounter_tl } { #1 }
\clist_clear:N \l__flowfram_pages_clist
\bool_if:NF \g__flowfram_not_this_frame_bool
{
  \clist_set:Ne \l__flowfram_pages_clist
  { \int_use:n { \g__flowfram_pagecounter_tl } }
}

```

Is this frame already switched on for the next page?

```

  \__flowfram_dynamic_check_if_this_page:nn
  { \l__flowfram_tmpb_int } { #1 }
\int_compare:nNnF
{ \l__flowfram_tmpb_int } = { \g__flowfram_pagecounter_tl }
{
  \bool_if:NF \g__flowfram_not_this_frame_bool
  {
    \clist_put_right:Ne \l__flowfram_pages_clist
    { \int_eval:n { \l__flowfram_tmpb_int } }
  }
}
\l_gput_right:Ne \g__flowfram_output_adjust_frames_tl
{
  \exp_not:N \dynamicsetpagelist
  { \int_eval:n { #1 } }
  {

```

```

        \l__flowfram_pages_clist
        > \int_use:N \l__flowfram_tmpb_int
    }
}
}

```

\@dynamicswitchonnextodd Version 2.0 removed \@dynamicswitchonnextodd.

\@dynamicswitchonnextodd Version 2.0 removed \@dynamicswitchonnextodd.

\dynamicswitchoffnext Switch off the listed dynamic frames from the next page onwards

```

\NewDocumentCommand \dynamicswitchoffnext { s m }
{
  \IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_dynamic_id:n { ##1 }
      \__flowfram_dynamic_switch_off_next:n { \l__flowfram_id_int }
    }
  }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_dynamic_switch_off_next:n { ##1 }
    }
  }
}

\cs_new:Nn \__flowfram_dynamic_switch_off_next:n
{
  Is this frame already off on this page?
  \__flowfram_dynamic_check_if_this_page:nn
  { \g__flowfram_pagecounter_tl } { #1 }
  \bool_if:NTF \g__flowfram_not_this_frame_bool
  {
    \clist_set:Nn \l__flowfram_pages_clist { none }
  }
  {
    \clist_set:Ne \l__flowfram_pages_clist
    { \int_eval:n { \g__flowfram_pagecounter_tl } }
  }
  \tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
  {
    \exp_not:N \dynamicsetpagelist
    { \int_eval:n { #1 } }
    { \l__flowfram_pages_clist }
  }
}
}

```

\@sdynamicsswitchoffnext Version 2.0 removed \@sdynamicsswitchoffnext.

\@dynamicsswitchoffnext Version 2.0 removed \@dynamicsswitchoffnext.

\dynamicsswitchoffnextodd Switch off the listed dynamic frames from the next odd page onwards

```
\NewDocumentCommand \dynamicsswitchoffnextodd { s m }
{
  \int_set:Nn \l__flowfram_tmpb_int
    { \g__flowfram_pagecounter_tl }
  \int_if_odd:nT { \g__flowfram_pagecounter_tl }
  {
    \int_incr:N \l__flowfram_tmpb_int
  }
  \IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_dynamic_id:n { ##1 }
      \__flowfram_dynamic_switch_off_next_odd:n { \l__flowfram_id_int }
    }
  }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_dynamic_switch_off_next_odd:n { ##1 }
    }
  }
}

\cs_new:Nn \__flowfram_dynamic_switch_off_next_odd:n
{
```

Is this frame already off on this page?

```
\__flowfram_dynamic_check_if_this_page:nn
{ \g__flowfram_pagecounter_tl } { #1 }
\bool_if:NTF \g__flowfram_not_this_frame_bool
{
```

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?

```
\int_compare:nNnTF
{ \g__flowfram_pagecounter_tl } = { \l__flowfram_tmpb_int }
{
```

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

```
\tl_set:Nn \l__flowfram_next_pages_tl { none }
}
{
```

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

```
\__flowfram_dynamic_check_if_this_page:nn
```

```

    { \l__flowfram_tmpb_int } { #1 }
\bool_if:NTF \g__flowfram_not_this_frame_bool
{

```

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

```

    \tl_set:Nn \l__flowfram_next_pages_tl { none }
}
{

```

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

```

    \tl_set:Ne \l__flowfram_next_pages_tl
    { \int_use:N \l__flowfram_tmpb_int }
}
}
{

```

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?

```

\int_compare:nNnTF
{ \g__flowfram_pagecounter_tl } = { \l__flowfram_tmpb_int }
{

```

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

```

\tl_set:Ne \l__flowfram_next_pages_tl
{ \int_use:n { \g__flowfram_pagecounter_tl } }
}
{

```

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

```

\__flowfram_dynamic_check_if_this_page:nn
{ \l__flowfram_tmpb_int } { #1 }
\bool_if:NTF \g__flowfram_not_this_frame_bool
{

```

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

```

\tl_set:Ne \l__flowfram_next_pages_tl
{ \int_eval:n { \g__flowfram_pagecounter_tl } }
}
{

```

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

```

\tl_set:Ne \l__flowfram_next_pages_tl
{
  \int_eval:n { \g__flowfram_pagecounter_tl } ,
  \int_use:N \l__flowfram_tmpb_int
}
}
}
\tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
{

```



```

        \exp_not:N \dynamicsetpagelist
        { \int_eval:n { #1 } }
        { \l__flowfram_next_pages_tl }
    }
}
%
```

\@sdynamicswitchoffnextodd Version 2.0 removed \@sdynamicswitchoffnextodd.

\@dynamicswitchoffnextodd Version 2.0 removed \@dynamicswitchoffnextodd.

\dynamicswitchonnextonly Switch on the listed dynamic frames for just the next page

```

\NewDocumentCommand \dynamicswitchonnextonly { s m }
{
  \int_set:Nn \l__flowfram_tmpb_int
  { \g__flowfram_pagecounter_tl + \c_one_int }
  \IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_dynamic_id:n { ##1 }
      \__flowfram_dynamic_switch_on_next_only:n { \l__flowfram_id_int }
    }
  }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_dynamic_switch_on_next_only:n { ##1 }
    }
  }
}

\cs_new:Nn \__flowfram_dynamic_switch_on_next_only:n
{
```

Is this frame already on?

```

  \__flowfram_dynamic_check_if_this_page:nn
  { \g__flowfram_pagecounter_tl } { #1 }
  \bool_if:NTF \g__flowfram_not_this_frame_bool
  {
```

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

```

  \tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
  {
    \exp_not:N \dynamicsetpagelist
    { \int_eval:n { #1 } }
    { \int_use:N \l__flowfram_tmpb_int }
  }
}
{
```

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

```
\tl_gput_right:N\g__flowfram_output_adjust_frames_tl
{
  \exp_not:N\dynamicsetpagelist
  { \int_eval:n { #1 } }
  {
    \int_eval:n { \g__flowfram_pagecounter_tl } ,
    \int_use:N \l__flowfram_tmpb_int
  }
}
}
```

\@sdynamicsswitchonnextonly Version 2.0 removed \@sdynamicsswitchonnextonly

\@dynamicsswitchonnextonly Version 2.0 removed \@dynamicsswitchonnextonly

\dynamicsswitchonnextoddone Switch on the listed dynamic frames for just the next odd page

```
\NewDocumentCommand \dynamicsswitchonnextoddone { s m }
{
  \IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_dynamic_id:n { ##1 }
      \__flowfram_dynamic_switch_on_next_odd_only:n { \l__flowfram_id_int }
    }
  }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_dynamic_switch_on_next_odd_only:n { ##1 }
    }
  }
}

\cs_new:Nn \__flowfram_dynamic_switch_on_next_odd_only:n
{
```

Is this frame already on?

```
\__flowfram_dynamic_check_if_this_page:nn
{ \g__flowfram_pagecounter_tl } { #1 }
\bool_if:NTF \g__flowfram_not_this_frame_bool
{
```

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?

```
\int_if_odd:nTF { \g__flowfram_pagecounter_tl }
{
```

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

```

\int_set:Nn \l__flowfram_tmpb_int
  { \g__flowfram_pagecounter_tl + \c_one_int }
\__flowfram_dynamic_check_if_this_page:nn
  { \l__flowfram_tmpb_int } { #1 }
\bool_if:NTF \g__flowfram_not_this_frame_bool
  {

```

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.

```

\int_incr:N \l__flowfram_tmpb_int
\clist_set:Ne \l__flowfram_pages_clist
  { \int_use:N \l__flowfram_tmpb_int }
}
{

```

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.

```

\clist_set:Ne \l__flowfram_pages_clist
  { \int_use:N \l__flowfram_tmpb_int }
\int_inc:N \l__flowfram_tmpb_int
\clist_put_right:Ne \l__flowfram_pages_clist
  { \int_use:N \l__flowfram_tmpb_int }
}
}
{

```

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

```

\int_set:Nn \l__flowfram_tmpb_int
  { \g__flowfram_pagecounter_tl + \c_one_int }
\clist_set:Ne \l__flowfram_pages_clist
  { \int_use:N \l__flowfram_tmpb_int }
}
}
{

```

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?

```

\int_if_odd:nTF { \g__flowfram_pagecounter_tl }
  {

```

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

```

\int_set:Nn \l__flowfram_tmpb_int
  { \g__flowfram_pagecounter_tl + \c_one_int }
\__flowfram_dynamic_check_if_this_page:nn
  { \l__flowfram_tmpb_int } { #1 }
\bool_if:NTF \g__flowfram_not_this_frame_bool
  {

```

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.

```

\int_incr:N \l__flowfram_tmpb_int
\clist_set:Ne \l__flowfram_pages_clist
{
  \int_eval:n { \g__flowfram_pagecounter_tl } ,
  \int_use:N \l__flowfram_tmpb_int
}
}
{

```

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.

```

\int_incr:N \l__flowfram_tmpb_int
\clist_set:Ne \l__flowfram_pages_clist
{
  \int_eval:n { \g__flowfram_pagecounter_tl } -
  \int_use:N \l__flowfram_tmpb_int
}
}
}
{

```

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.

```

\int_set:Nn \l__flowfram_tmpb_int
{ \g__flowfram_pagecounter_tl + \c_one_int }
\clist_set:Ne \l__flowfram_pages_clist
{
  \int_eval:n { \g__flowfram_pagecounter_tl } ,
  \int_use:N \l__flowfram_tmpb_int
}
}
}
\tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
{
  \exp_not:N \dynamicsetpagelist
  { \int_eval:n { #1 } }
  { \l__flowfram_pages_clist }
}
}

```

`\sdynamicswitchonnextoddone` Version 2.0 removed `\@dynamicswitchonnextoddone`

`\@dynamicswitchonnextoddone` Version 2.0 removed `\@dynamicswitchonnextoddone`.

`\dynamicswitchoffnextonly` Switch off the listed dynamic frames for just the next page
`\NewDocumentCommand \dynamicswitchoffnextonly { s m }`

```

{
\int_set:Nn \l__flowfram_tmpb_int
  { \g__flowfram_pagecounter_tl + \c_one_int }
\IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_dynamic_id:n { ##1 }
      \__flowfram_dynamic_switch_off_next_only:n { \l__flowfram_id_int }
    }
  }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_dynamic_switch_off_next_only:n { ##1 }
    }
  }
}

\cs_new:Nn \__flowfram_dynamic_switch_off_next_only:n
{
  \tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
  {
    \exp_not:N \dynamicaddexclusion
    { \int_eval:n { #1 } }
    { \int_use:N \l__flowfram_tmpb_int }
  }
}

```

\@dynamicswitchoffnextonly Version 2.0 removed \@dynamicswitchoffnextonly.

\@dynamicswitchoffnextonly Version 2.0 removed \@dynamicswitchoffnextonly

dynamicswitchoffnextoddoneonly Switch off the listed dynamic frames for just the next odd page

```

\NewDocumentCommand \dynamicswitchoffnextoddoneonly { s m }
{
\int_set:Nn \l__flowfram_tmpb_int
  { \g__flowfram_pagecounter_tl + \c_one_int }
\int_if_odd:nF { \l__flowfram_tmpb_int }
  {
    \int_incr:N \l__flowfram_tmpb_int
  }
\IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_dynamic_id:n { ##1 }
      \__flowfram_dynamic_switch_off_next_odd_only:n { \l__flowfram_id_int }
    }
  }
}

```

```

    {
      \clist_map_inline:nn { #2 }
      {
        \__flowfram_dynamic_switch_off_next_odd_only:n { ##1 }
      }
    }
  }

\cs_new:Nn \__flowfram_dynamic_switch_off_next_odd_only:n
{
  \tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
  {
    \exp_not:N \dynamicaddexclusion
    { \int_eval:n { #1 } }
    { \int_use:N \l__flowfram_tmpb_int }
  }
}

```

\dynamicswitchoffnextoddoneonly Version 2.0 removed \@dynamicswitchoffnextoddoneonly.

\dynamicswitchoffnextoddoneonly Version 2.0 removed \@dynamicswitchoffnextoddoneonly.

\staticswitchonnext Switch on the listed static frames from the next page onwards

```

\NewDocumentCommand \staticswitchonnext { s m }
{
  \IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_static_id:n { ##1 }
      \__flowfram_static_switch_on_next:n { \l__flowfram_id_int }
    }
  }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_static_switch_on_next:n { ##1 }
    }
  }
}

\cs_new:Nn \__flowfram_static_switch_on_next:n
{
  Is this frame already on?
  \__flowfram_static_check_if_this_page:nn
  { \g__flowfram_pagecounter_tl } { #1 }
  \bool_if:NTF \g__flowfram_not_this_frame_bool
  {
    \tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
    {

```

```

        \exp_not:N \staticsetpagelist
        { \int_eval:n { #1 } }
        { > \int_eval:n { \g__flowfram_pagecounter_tl } }
    }
}
{
\tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
{
    \exp_not:N \staticsetpagelist
    { \int_eval:n { #1 } }
    {
        \int_eval:n { \g__flowfram_pagecounter_tl } ,
        > \int_eval:n { \g__flowfram_pagecounter_tl }
    }
}
}
}
}

```

\@sstaticswithchonnex Version 2.0 removed \@sstaticswithchonnex.

\@staticswithchonnex Version 2.0 removed \@staticswithchonnex.

\staticswithchonnexodd Switch on the listed static frames from the next odd page onwards

```

\NewDocumentCommand \staticswithchonnexodd { s m }
{
    \int_set:Nn \l__flowfram_tmpb_int
    { \g__flowfram_pagecounter_tl }
    \int_if_odd:nT { \l__flowfram_tmpb_int }
    {
        \int_incr:N \l__flowfram_tmpb_int
    }
    \IfBooleanTF { #1 }
    {
        \clist_map_inline:nn { #2 }
        {
            \__flowfram_get_static_id:n { ##1 }
            \__flowfram_static_switch_on_next_odd:n { \l__flowfram_id_int }
        }
    }
    {
        \clist_map_inline:nn { #2 }
        {
            \__flowfram_static_switch_on_next_odd:n { ##1 }
        }
    }
}
}
\cs_new:Nn \__flowfram_static_switch_on_next_odd:n
{

```

Is this frame already on?

```
\__flowfram_static_check_if_this_page:nn
{ \g__flowfram_pagecounter_tl } { #1 }
\clist_clear:N \l__flowfram_pages_clist
\bool_if:NF \g__flowfram_not_this_frame_bool
{
  \clist_put_right:Ne \l__flowfram_pages_clist
  { \int_eval:n { \g__flowfram_pagecounter_tl } }
}
```

Is this frame already switched on for the next page?

```
\__flowfram_static_check_if_this_page:nn
{ \l__flowfram_tmpb_int } { #1 }
\int_compare:nNnF
{ \l__flowfram_tmpb_int } = { \g__flowfram_pagecounter_tl }
{
  \bool_if:NF \g__flowfram_not_this_frame_bool
  {
    \clist_put_right:Ne \l__flowfram_pages_clist
    { \int_use:N \l__flowfram_tmpb_int }
  }
}
\tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
{
  \exp_not:N \staticsetpagelist
  { \int_eval:n { #1 } }
  {
    \l__flowfram_pages_clist ,
    > \int_use:N \l__flowfram_tmpb_int
  }
}
}
```

~~\@sstaticswitchonnextodd~~ Version 2.0 removed ~~\@sstaticswitchonnextodd~~.

~~\@staticswitchonnextodd~~ Version 2.0 removed ~~\@staticswitchonnextodd~~.

~~\staticswitchoffnext~~ Switch off the listed static frames from the next page onwards

```
\NewDocumentCommand \staticswitchoffnext { s m }
{
  \IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_static_id:n { ##1 }
      \__flowfram_static_switch_off_next:n { \l__flowfram_id_int }
    }
  }
  {
    \clist_map_inline:nn { #2 }

```



```

        {
          \_flowfram_static_switch_off_next:n { ##1 }
        }
      }
}

\cs_new:Nn \_flowfram_static_switch_off_next:n
{
  Is this frame already off on this page?
  \_flowfram_static_check_if_this_page:nn
    { \g__flowfram_pagecounter_tl } { #1 }
  \bool_if:NTF \g__flowfram_not_this_frame_bool
    {
      \clist_set:Nn \l__flowfram_pages_clist { none }
    }
    {
      \clist_set:Ne \l__flowfram_pages_clist
        { \int_eval:n { \g__flowfram_pagecounter_tl } }
    }
  \tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
  {
    \exp_not:N \staticsetpagelist
    { \int_eval:n { #1 } }
    { \l__flowfram_pages_clist }
  }
}

```

\@sstaticswitchoffnext Version 2.0 removed \@sstaticswitchoffnext.

\@staticswitchoffnext Version 2.0 removed \@staticswitchoffnext.

\staticswitchoffnextodd Switch off the listed static frames from the next odd page onwards

```

\NewDocumentCommand \staticswitchoffnextodd { s m }
{
  \int_set:Nn \l__flowfram_tmpb_int
    { \g__flowfram_pagecounter_tl }
  \int_if_odd:nT { \g__flowfram_pagecounter_tl }
  {
    \int_incr:N \l__flowfram_tmpb_int
  }
  \IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }
    {
      \_flowfram_get_static_id:n { ##1 }
      \_flowfram_static_switch_off_next_odd:n { \l__flowfram_id_int }
    }
  }
}

```

```

\clist_map_inline:nn { #2 }
{
  \__flowfram_static_switch_off_next_odd:n { ##1 }
}
}
}

\cs_new:Nn \__flowfram_static_switch_off_next_odd:n
{
Is this frame already off on this page?
  \__flowfram_static_check_if_this_page:nn
  { \g__flowfram_pagecounter_tl } { #1 }
  \bool_if:NTF \g__flowfram_not_this_frame_bool
  {
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?
  \int_compare:nNnTF
  { \g__flowfram_pagecounter_tl } = { \l__flowfram_tmpb_int }
  {
This page is even and the frame is off on this page, so set to none.
  \tl_set:Nn \l__flowfram_next_pages_tl { none }
  }
  {
This page is odd. Is the frame on or off on the next page?
  \__flowfram_static_check_if_this_page:nn
  { \l__flowfram_tmpb_int } { #1 }
  \bool_if:NTF \g__flowfram_not_this_frame_bool
  {
Off on the next page as well, so set to none.
  \tl_set:Nn \l__flowfram_next_pages_tl { none }
  }
  {
Not off on the next page, so set to next page only.
  \tl_set:Ne \l__flowfram_next_pages_tl
  { \int_use:N \l__flowfram_tmpb_int }
  }
  }
  }
  {
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?
  \int_compare:nNnTF
  { \g__flowfram_pagecounter_tl } = { \l__flowfram_tmpb_int }
  {

```

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

```
\tl_set:Ne \l__flowfram_next_pages_tl
  { \int_eval:n { \g__flowfram_pagecounter_tl } }
}
{
```

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

```
\__flowfram_static_check_if_this_page:nn
  { \l__flowfram_tmpb_int } { #1 }
\bool_if:NTF \g__flowfram_not_this_frame_bool
{
```

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

```
\tl_set:Ne \l__flowfram_next_pages_tl
  { \int_eval:n { \g__flowfram_pagecounter_tl } }
}
{
```

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

```
\tl_set:Ne \l__flowfram_next_pages_tl
  {
    \int_eval:n { \g__flowfram_pagecounter_tl } ,
    \int_use:N \l__flowfram_tmpb_int
  }
}
}
}
\tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
{
  \exp_not:N \staticsetpagelist
  { \int_eval:n { #1 } }
  { \l__flowfram_next_pages_tl }
}
}
```

~~\@sstaticswitchoffnextodd~~ Version 2.0 removed ~~\@sstaticswitchoffnextodd~~.

~~\@staticswitchoffnextodd~~ Version 2.0 removed ~~\staticswitchoffnextodd~~.

~~\staticswitchonnextonly~~ Switch on the listed static frames for just the next page

```
\NewDocumentCommand \staticswitchonnextonly { s m }
{
  \int_set:Nn \l__flowfram_tmpb_int
    { \g__flowfram_pagecounter_tl + \c_one_int }
  \IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_static_id:n { ##1 }
      \__flowfram_static_switch_on_next_only:n { \l__flowfram_id_int }
    }
  }
}
```

```

    }
    {
      \clist_map_inline:nn { #2 }
      {
        \__flowfram_static_switch_on_next_only:n { ##1 }
      }
    }
  }
}

```

```

\cs_new:Nn \__flowfram_static_switch_on_next_only:n
{

```

Is this frame already on?

```

  \__flowfram_static_check_if_this_page:nn
  { \g__flowfram_pagecounter_tl } { #1 }
  \bool_if:NTF \g__flowfram_not_this_frame_bool
  {

```

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

```

    \tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
    {
      \exp_not:N \staticsetpagelist
      { \int_eval:n { #1 } }
      { \int_use:N \l__flowfram_tmpb_int }
    }
  }
}

```

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

```

    \tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
    {
      \exp_not:N \staticsetpagelist
      { \int_eval:n { #1 } }
      {
        \int_eval:n { \g__flowfram_pagecounter_tl } ,
        \int_use:N \l__flowfram_tmpb_int
      }
    }
  }
}

```

`\@sstaticswitchonnextonly` Version 2.0 removed `\@sstaticswitchonnextonly`.

`\@staticswitchonnextonly` Version 2.0 removed `\@staticswitchonnextonly`.

`\staticswitchonnextodonly` Switch on the listed static frames for just the next odd page

```

\NewDocumentCommand \staticswitchonnextodonly { s m }
{
  \IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }

```

```

    {
      \__flowfram_get_static_id:n { ##1 }
      \__flowfram_static_switch_on_next_odd_only:n { \l__flowfram_id_int }
    }
  }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_static_switch_on_next_odd_only:n { ##1 }
    }
  }
}

\cs_new:Nn \__flowfram_static_switch_on_next_odd_only:n
{

```

Is this frame already on?

```

  \__flowfram_static_check_if_this_page:nn
  { \g__flowfram_pagecounter_tl } { #1 }
  \bool_if:NTF \g__flowfram_not_this_frame_bool
  {

```

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?

```

  \int_if_odd:nTF { \g__flowfram_pagecounter_tl }
  {

```

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

```

  \int_set:Nn \l__flowfram_tmpb_int
  { \g__flowfram_pagecounter_tl + \c_one_int }
  \__flowfram_static_check_if_this_page:nn
  { \l__flowfram_tmpb_int } { #1 }
  \bool_if:NTF \g__flowfram_not_this_frame_bool
  {

```

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.

```

  \int_incr:N \l__flowfram_tmpb_int
  \clist_set:Ne \l__flowfram_pages_clist
  { \int_use:N \l__flowfram_tmpb_int }
}
{

```

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.

```

  \clist_set:Ne \l__flowfram_pages_clist
  { \int_use:N \l__flowfram_tmpb_int }
  \int_incr:N \l__flowfram_tmpb_int
  \clist_put_right:Ne \l__flowfram_pages_clist
  {

```

```

        \int_use:N \l__flowfram_tmpb_int
      }
    }
  {

```

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

```

    \int_set:Nn \l__flowfram_tmpb_int
      { \g__flowfram_pagecounter_tl + \c_one_int }
    \clist_set:Ne \l__flowfram_pages_clist
      { \int_use:N \l__flowfram_tmpb_int }
  }
}
{

```

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?

```

\int_if_odd:nTF { \g__flowfram_pagecounter_tl }
{

```

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

```

\int_set:Nn \l__flowfram_tmpb_int
  { \g__flowfram_pagecounter_tl + \c_one_int }
\__flowfram_static_check_if_this_page:nn
  { \l__flowfram_tmpb_int } { #1 }
\bool_if:NTF \g__flowfram_not_this_frame_bool
{

```

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.

```

\int_incr:N \l__flowfram_tmpb_int
\clist_set:Ne \l__flowfram_pages_clist
{
  \int_eval:n { \g__flowfram_pagecounter_tl } ,
  \int_use:N \l__flowfram_tmpb_int
}
}
{

```

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.

```

\int_incr:N \l__flowfram_tmpb_int
\clist_set:Ne \l__flowfram_pages_clist
{
  \int_eval:n { \g__flowfram_pagecounter_tl } -
  \int_use:N \l__flowfram_tmpb_int
}
}

```

```

}
{

```

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.

```

    \int_set:Nn \l__flowfram_tmpb_int
      { \g__flowfram_pagecounter_tl + \c_one_int }
    \clist_set:Ne \l__flowfram_pages_clist
      {
        \int_eval:n { \g__flowfram_pagecounter_tl } ,
        \int_use:N \l__flowfram_tmpb_int
      }
  }
}
\tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
{
  \exp_not:N \staticsetpagelist
  { \int_eval:n { #1 } }
  { \l__flowfram_pages_clist }
}
}

```

~~@sstaticswitchonnextoddone~~ Version 2.0 removed ~~\@sstaticswitchonnextoddone~~.

~~\@staticswitchonnextoddone~~ Version 2.0 removed ~~\@staticswitchonnextoddone~~.

~~\staticswitchoffnextonly~~ Switch off the listed static frames for just the next page

```

\NewDocumentCommand \staticswitchoffnextonly { s m }
{
  \int_set:Nn \l__flowfram_tmpb_int
    { \g__flowfram_pagecounter_tl + \c_one_int }
  \IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_static_id:n { ##1 }
      \__flowfram_static_switch_off_next_only:n { \l__flowfram_id_int }
    }
  }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_static_switch_off_next_only:n { ##1 }
    }
  }
}

\cs_new:Nn \__flowfram_static_switch_off_next_only:n
{
  \tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl

```

```

    {
      \exp_not:N \staticaddexclusion
      { \int_eval:n { #1 } }
      { \int_use:N \l__flowfram_tmpb_int }
    }
  }
}

```

\@sstaticswitchoffnextonly Version 2.0 removed \@sstaticswitchoffnextonly.

\@staticswitchoffnextonly Version 2.0 removed \@staticswitchoffnextonly.

\staticswitchoffnextoddone Switch off the listed static frames for just the next odd page

```

\NewDocumentCommand \staticswitchoffnextoddone { s m }
{
  \int_set:Nn \l__flowfram_tmpb_int
  { \g__flowfram_pagecounter_tl + \c_one_int }
  \int_if_odd:nF { \l__flowfram_tmpb_int }
  {
    \int_incr:N \l__flowfram_tmpb_int
  }
  \IfBooleanTF { #1 }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_get_static_id:n { ##1 }
      \__flowfram_static_switch_off_next_odd_only:n { \l__flowfram_id_int }
    }
  }
  {
    \clist_map_inline:nn { #2 }
    {
      \__flowfram_static_switch_off_next_odd_only:n { ##1 }
    }
  }
}

\cs_new:Nn \__flowfram_static_switch_off_next_odd_only:n
{
  \tl_gput_right:Ne \g__flowfram_output_adjust_frames_tl
  {
    \exp_not:N \staticaddexclusion
    { \int_eval:n { #1 } }
    { \int_use:N \l__flowfram_tmpb_int }
  }
}

```

\@sstaticswitchoffnextoddone Version 2.0 removed \@sstaticswitchoffnextoddone.

\@staticswitchoffnextoddone Version 2.0 removed \@staticswitchoffnextoddone.

1.11 Static versions of floats

Floats can not go in saved boxes or minipages, so define static versions to go in static and dynamic frames. These just set `\@capytype` so that the `\caption` command may be used. Version 2.0 added paragraph breaks and ignore spaces.

```
statictable (env.)
  \newenvironment{statictable}
  {\par\def\@capytype{table}\ignorespaces}
  {\par\ignorespacesafterend}
```

```
staticfigure (env.)
  \newenvironment{staticfigure}
  {\def\@capytype{figure}\ignorespaces}
  {\par\ignorespacesafterend}
```

1.12 Standard Layouts

1.12.1 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 `\ffvadjustfalse` is used.

```
\newif\iffvadjust
\ffvadjustrue
```

`\onecolumn` `\onecolumn` will make a single flow frame that takes up the entire area of the typeblock (adjusted according to `\iffvadjust`.) 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 `_flowfram_one_column:n`. Syntax: `\onecolumn[<pages>]`, where *<pages>* is the page list for which the new flow frame is defined.

```
\cs_set_eq:NN \_flowfram_org_onecolumn: \onecolumn
\RenewDocumentCommand \onecolumn { 0{all} o }
{
  \_flowfram_only_preamble:Nn \onecolumn
  {
    \_flowfram_one_column:n { #1 }
    \IfValueT { #2 }
    {
      \_flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #2 }
    }
  }
}
```

`\@onecolumn` Version 2.0 replaced `\@onecolumn`

```

\cs_new:Nn \__flowfram_one_column:n
{
  \__flowfram_one_column_in_area:nnnnn
  { #1 } { \typeblockwidth } { \typeblockheight } { Opt } { Opt }
}

```

`\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 ($\langle x \rangle$, $\langle y \rangle$), relative to the typeblock, with width $\langle w \rangle$ and height $\langle h \rangle$. The only difference between `\onecolumninarea` and explicitly creating the flow frame using `\newflowframe` is the `\onecolumninarea` will adjust the vertical height to 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[ $\langle pages \rangle$ ]{ $\langle w \rangle$ }{ $\langle h \rangle$ }{ $\langle x \rangle$ }{ $\langle y \rangle$ }.
\NewDocumentCommand \onecolumninarea { 0 { all } m m m m o }
{
  \__flowfram_only_preamble:Nn \onecolumninarea
  {
    \__flowfram_one_column_in_area:nnnnn
    { #1 } { #2 } { #3 } { #4 } { #5 }
    \IfValueT { #6 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #6 }
    }
  }
}

```

`\@onecolumninarea` Version 2.0 replaced `\@onecolumninarea`

```

\cs_new:Nn \__flowfram_one_column_in_area:nnnnn
{
  \dim_set:Nn \l__flowfram_height_dim { #3 }
  \iffvadjust
  \adjustheight \l__flowfram_height_dim
  \fi
  \__flowfram_new_flow:nnnnn
  { #1 } { #2 } { \l__flowfram_height_dim } { #4 } { #5 }
}

```

If `\onecolumn` is used in the document and the column associated with that column hasn't yet been set, assume this is the one column to use.

```

\tl_if_empty:NT \g__flowfram_one_col_id_tl
{
  \tl_gset:Ne \g__flowfram_one_col_id_tl { \int_use:N \c@maxflow }
}

```

`\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 `\typeblockheight` 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]`.

```
\cs_set_eq:NN \__flowfram_org_twocolumn: \twocolumn
\RenewDocumentCommand \twocolumn { 0 { all } o }
{
  \__flowfram_only_preamble:Nn \twocolumn
  {
    \__flowfram_two_columns:n { #1 }
    \IfValueT { #2 }
    {
      \clist_set:Nc \l__flowfram_tmp_clist { #2 }
      \int_compare:nNnTF
        { \clist_count:N \l__flowfram_tmp_clist } = { 2 }
        {
          \__flowfram_set_frame_id:nnn { flow } { \c@maxflow }
          { \clist_item:Nn \l__flowfram_tmp_clist { 2 } }
          \__flowfram_set_frame_id:nnn { flow }
          { \int_eval:n { \c@maxflow - \c_one_int } }
          { \clist_item:Nn \l__flowfram_tmp_clist { 1 } }
        }
      }
    }
  }
}
```

If only one label provided just set the last frame for backward compatibility.

```
\__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #2 }
}
}
}
```

`\@twocolumn` Version 2.0 removed `\@twocolumn`

```
\cs_new:Nn \__flowfram_two_columns:n
{
  \__flowfram_two_columns_in_area:nnnnn
  { #1 } { \typeblockwidth } { \typeblockheight } { Opt } { Opt }
}
```

`\twocolumninarea` Again, `\twocolumn` is actually a special case of `\twocolumninarea`. The final optional argument is the label of the final frame. This is added for backward-compatibility where a knock-on effect of the final command being `\newflowframe` meant that a following `[` would be interpreted as an optional argument specifying its label. Syntax:

```
\twocolumninarea[pages]{w}{h}{x}{y}.
\NewDocumentCommand \twocolumninarea { 0{all} m m m m o }
{
  \__flowfram_only_preamble:Nn \twocolumninarea
  {
    \__flowfram_two_columns_in_area:nnnnn
    { #1 } { #2 } { #3 } { #4 } { #5 }
  }
}
```

```

\IfValueT { #6 }
{
  \clist_set:Nc \l__flowfram_tmp_clist { #6 }
  \int_compare:nNnTF
    { \clist_count:N \l__flowfram_tmp_clist } = { 2 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow }
      { \clist_item:Nn \l__flowfram_tmp_clist { 2 } }
      \__flowfram_set_frame_id:nnn { flow }
      { \int_eval:n { \c@maxflow - \c_one_int } }
      { \clist_item:Nn \l__flowfram_tmp_clist { 1 } }
    }
    {

```

If only one label provided just set the last frame for backward compatibility.

```

      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #6 }
    }
  }
}

```

\@twocolumninarea Version 2.0 replaced \@twocolumninarea

```

\cs_new:Nn \__flowfram_two_columns_in_area:nnnnn
{
  \dim_set:Nn \l__flowfram_height_dim { #3 }
  \iffvadjust
    \adjustheight \l__flowfram_height_dim
  \fi
  \dim_set:Nn \l__flowfram_width_dim { ( #2 - \columnsep ) / 2 }
  \dim_set:Nn \l__flowfram_x_dim
  {
    #4 + \l__flowfram_width_dim + \columnsep
  }
  \iflefttorightcolumns
    \__flowfram_new_flow:nnnnn
    { #1 } { \l__flowfram_width_dim } { \l__flowfram_height_dim } { #4 } { #5 }
    \setflowframe{\c@maxflow}{margin=left}%
  \else
    \__flowfram_new_flow:nnnnn
    { #1 } { \l__flowfram_width_dim } { \l__flowfram_height_dim }
    { \l__flowfram_x_dim } { #5 }
    \setflowframe{\c@maxflow}{margin=right}%
  \fi
  \iflefttorightcolumns
    \__flowfram_new_flow:nnnnn
    { #1 } { \l__flowfram_width_dim } { \l__flowfram_height_dim }
    { \l__flowfram_x_dim } { #5 }
    \setflowframe{\c@maxflow}{margin=right}%
  \else
    \__flowfram_new_flow:nnnnn

```

```

    { #1 } { \l__flowfram_width_dim } { \l__flowfram_height_dim }
    { #4 } { #5 }
    \setflowframe{\c@maxflow}{margin=left}%
  \fi

```

If `\twocolumn` is used in the document and `\flowfram_set_two_col:` still has its initial definition, assume this is the two column to use. (Note there's no header frame in this case.)

```

\cs_if_eq:NNT
  \flowfram_set_two_col:
  \__flowfram_set_two_col_warning:
{
  \cs_set:Ne \flowfram_set_two_col:
  {
    \exp_not:N \setallflowframes{pages=none}
    \exp_not:N \__flowfram_set_flow_by_idn:nn
      { \int_eval:n { \c@maxflow - \c_one_int } } { pages=all }
    \exp_not:N \__flowfram_set_flow_by_idn:nn
      { \int_use:N \c@maxflow } { pages=all }
  }
}

```

If `\twocolumn` is used in the document and the associated columns haven't yet been set, assume these are the columns to use.

```

\bool_lazy_and:nnT
{
  \tl_if_empty_p:N \g__flowfram_two_col_i_id_tl
}
{
  \tl_if_empty_p:N \g__flowfram_two_col_ii_id_tl
}
{
  \tl_gset:Ne \g__flowfram_two_col_i_id_tl
  {
    \int_eval:n { \c@maxflow - \c_one_int }
  }
  \tl_gset:Ne \g__flowfram_two_col_ii_id_tl { \int_use:N \c@maxflow }
}
}

```

`\Ncolumn` Again for an arbitrary number of columns ($\langle n \rangle$). Syntax: `\Ncolumn[$\langle pages \rangle$]{ $\langle n \rangle$ }`.

```

\NewDocumentCommand \Ncolumn { 0{ all } m o }
{
  \__flowfram_only_preamble:Nn \Ncolumn
  {
    \Ncolumninarea
      [ #1 ] { #2 } { \typeblockwidth } { \typeblockheight } { Opt } { Opt }
    \IfValueT { #3 }
    {

```

```

        \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #3 }
      }
    }
}

```

`\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>}`.

```

\NewDocumentCommand \Ncolumninarea { 0 { all } m m m m m o }
{
  \__flowfram_only_preamble:Nn \Ncolumninarea
  {
    \int_case:nnF { #2 }
    {
      { \c_one_int }
      {
        \__flowfram_one_column_in_area:nnnnn
        { #1 } { #3 } { #4 } { #5 } { #6 }
      }
      { 2 }
      {
        \__flowfram_two_columns_in_area:nnnnn
        { #1 } { #3 } { #4 } { #5 } { #6 }
      }
    }
    {
      \int_compare:nNnTF { #2 } < { \c_one_int }
      {
        \msg_error:nnne { flowfram } { invalid-num-frames }
        { flow } { \int_eval:n { #2 } }
      }
      {
        \__flowfram_n_columns_in_area:nnnnnn
        { #1 } { #2 } { #3 } { #4 } { #5 } { #6 }
      }
    }
  }
  \IfValueT { #7 }
  {
    \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #7 }
  }
}

```

`\@Ncolumninarea` Set up *<n>* columns in the area specified. There is a horizontal distance of `\columnsep` between them all. Version 2.0 replaced `\@Ncolumninarea`.

```

\cs_new:Nn \__flowfram_n_columns_in_area:nnnnnn
{
  \int_set:Nn \l__flowfram_col_int { #2 - \c_one_int }
  \dim_set:Nn \l__flowfram_width_dim

```

```

{
  ( #3 - \l__flowfram_col_int \columnsep ) / ( #2 )
}
\dim_set:Nn \l__flowfram_x_dim { #5 }
\iflefttorightcolumns
\else
  \dim_add:Nn \l__flowfram_x_dim { #3 - \l__flowfram_width_dim }
\fi
\dim_set:Nn \l__flowfram_height_dim { #4 }
\iffvadjust
  \adjustheight \l__flowfram_height_dim
\fi
\int_step_inline:nn { #2 }
{
  \__flowfram_new_flow:nnnnn
    { #1 } { \l__flowfram_width_dim } { \l__flowfram_height_dim }
    { \l__flowfram_x_dim } { #6 }
  \iflefttorightcolumns
    \dim_add:Nn \l__flowfram_x_dim { \l__flowfram_width_dim + \columnsep }
  \else
    \dim_add:Nn \l__flowfram_x_dim { - \l__flowfram_width_dim - \columnsep }
  \fi
}
}

```

Set up something similar but have another frame (of type $\langle type \rangle$) at the top of the other frames.

`\vcolumnsep` The vertical distance between the top frames and column flow frames when created using `\Ncolumnntop` etc is given by:

```

\newlength{\vcolumnsep}
\setlength{\vcolumnsep}{\columnsep}

```

`\onecolumnntop` `\onecolumnntop` makes one flow frame, and one $\langle type \rangle$ frame in the area specified, where the $\langle type \rangle$ frame is $\langle H \rangle$ 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:

```

\onecolumnntop[ $\langle pages \rangle$ ]{ $\langle type \rangle$ }{ $\langle H \rangle$ }
\NewDocumentCommand \onecolumnntop { 0{all} m m o }
{
  \__flowfram_only_preamble:Nn \onecolumnntop
  {
    \__flowfram_one_column_with_top_in_area:nnnnnn
      { #1 } { #2 } { #3 } { \typeblockwidth } { \typeblockheight } { Opt } { Opt }
    \IfValueT { #4 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #4 }
    }
  }
}

```

```

    }
}

```

`\onecolumnStop` Special case for static frame. Syntax: `\onecolumnStop[⟨pages⟩]{⟨H⟩}`

```

\NewDocumentCommand \onecolumnStop { 0 {all} m o }
{
  \__flowfram_only_preamble:Nn \onecolumnStop
  {
    \__flowfram_one_column_with_top_in_area:nnnnnnn
    { #1 } { static } { #2 } { \typeblockwidth } { \typeblockheight } { Opt } { Opt }
    \IfValueT { #3 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #3 }
    }
  }
}

```

`\onecolumnDtop` Special case for dynamic frame. Syntax: `\onecolumnDtop[⟨pages⟩]{⟨H⟩}`

```

\NewDocumentCommand \onecolumnDtop { 0 { all } m o }
{
  \__flowfram_only_preamble:Nn \onecolumnDtop
  {
    \__flowfram_one_column_with_top_in_area:nnnnnnn
    { #1 } { dynamic } { #2 } { \typeblockwidth } { \typeblockheight } { Opt } { Opt }
    \IfValueT { #3 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #3 }
    }
  }
}

```

`\newframe` Create a frame of given type. Syntax:

`\newframe[⟨pages⟩]{⟨type⟩}{⟨w⟩}{⟨h⟩}{⟨x⟩}{⟨y⟩}[⟨label⟩].`

```

\NewDocumentCommand \newframe { s 0{all} m m m m m o }
{
  \__flowfram_only_preamble:Nn \newframe
  {
    \cs_if_exist:cTF
    { \__flowfram_new_ #3 :nnnnnn }
    {
      \IfValueTF { #8 }
      {
        \use:c { \__flowfram_new_ #3 :nnnnnn }
        { #2 } { #4 } { #5 } { #6 } { #7 } { #8 }
      }
      {
        \use:c { \__flowfram_new_ #3 :nnnnnn }
        { #2 } { #4 } { #5 } { #6 } { #7 } { \int_use:c { c@max #3 } }
      }
    }
  }
}

```



```

        \IfBooleanT { #1 }
        {
            \flowfram_frame_set_bool_true:nnnn
            { #3 } { hasframe } { \value { max #3 } }
        }
    }
    {
        \msg_error:nne { flowfram } { invalid-frame-type } { #3 }
    }
}
}
}

```

`\onecolumntopinarea` Now for a specified area. Syntax:

```

\onecolumntopinarea[<pages>]{<type>}{<H>}{<w>}{<h>}{<x>}{<y>}.
\NewDocumentCommand \onecolumntopinarea { O{all} m m m m m m o }
{
    \__flowfram_only_preamble:Nn \onecolumntopinarea
    {
        \__flowfram_one_column_with_top_in_area:nnnnnnn
        { #1 } { #2 } { #3 } { #4 } { #5 } { #6 } { #7 }
        \IfValueT { #8 }
        {
            \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #8 }
        }
    }
}

\cs_new:Nn \__flowfram_one_column_with_top_in_area:nnnnnnn
{
    \dim_set:Nn \l__flowfram_sfdh_height_dim { #3 }
    \dim_set:Nn \l__flowfram_y_dim
    { #5 - \l__flowfram_sfdh_height_dim }
    \dim_set:Nn \l__flowfram_height_dim
    { \l__flowfram_y_dim - \vcolumnsep }
    \dim_add:Nn \l__flowfram_y_dim { #7 }
    \newframe
    [ #1 ] { #2 }
    { #4 } { \l__flowfram_sfdh_height_dim }
    { #6 } { \l__flowfram_y_dim }
    \iffvadjust
    \adjustheight \l__flowfram_height_dim
    \fi
    \__flowfram_new_flow:nnnnn
    { #1 } { #4 } { \l__flowfram_height_dim } { #6 } { #7 }
}

```

`\onecolumnStopinarea` Special case for static frame. Syntax:

```

\onecolumnStopinarea[<pages>]{<H>}{<w>}{<h>}{<x>}{<y>}.
\NewDocumentCommand \onecolumnStopinarea { O{all} m m m m m o }
{

```

```

    \__flowfram_only_preamble:Nn \onecolumnStopinarea
    {
      \__flowfram_one_column_with_top_in_area:nnnnnnn
        { #1 } { static } { #2 } { #3 } { #4 } { #5 } { #6 }
      \IfValueT { #7 }
        {
          \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #7 }
        }
      }
    }
  }
}

```

`\onecolumnDtopinarea` Special case for dynamic frame. Syntax:

```

\onecolumnDtopinarea[<pages>]{<H>}{<w>}{<h>}{<x>}{<y>}.
\NewDocumentCommand \onecolumnDtopinarea { O{all} m m m m m o }
{
  \__flowfram_only_preamble:Nn \onecolumnStopinarea
  {
    \__flowfram_one_column_with_top_in_area:nnnnnnn
      { #1 } { dynamic } { #2 } { #3 } { #4 } { #5 } { #6 }
    \IfValueT { #7 }
      {
        \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #7 }
      }
    }
  }
}

```

`\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:
\NewDocumentCommand \twocolumntop { O{all} m m o }
{
  \__flowfram_only_preamble:Nn \twocolumntop
  {
    \__flowfram_two_column_top_in_area:nnnnnnn
      { #1 } { #2 } { #3 } { \typeblockwidth } { \typeblockheight } { Opt } { Opt }
    \IfValueT { #4 }
      {
        \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #4 }
      }
    }
  }
}

```

`\twocolumnStop` Special case for static frame.

```

\NewDocumentCommand \twocolumnStop { O{all} m o }
{
  \__flowfram_only_preamble:Nn \twocolumnStop
  {
    \__flowfram_two_column_with_top_in_area:nnnnnnn
      { #1 } { static } { #2 } { \typeblockwidth } { \typeblockheight } { Opt } { Opt }
    }
  }
}

```

```

    \IfValueT { #3 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #3 }
    }
  }
}

```

`\twocolumnDtop` Special case for dynamic frame.

```

\NewDocumentCommand \twocolumnDtop { 0{all} m o }
{
  \__flowfram_only_preamble:Nn \twocolumnDtop
  {
    \__flowfram_two_column_with_top_in_area:nnnnnnn
    { #1 } { dynamic } { #2 } { \typeblockwidth } { \typeblockheight } { Opt } { Opt }
    \IfValueT { #3 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #3 }
    }
  }
}

```

Now for a general area.

`\twocolumntopinarea` Syntax:

```

\twocolumntopinarea[<pages>]{<type>}{<H>}{<w>}{<h>}{<x>}{<y>}.
\NewDocumentCommand \twocolumntopinarea { 0{all} m m m m m m o }
{
  \__flowfram_only_preamble:Nn \twocolumntopinarea
  {
    \__flowfram_two_column_with_top_in_area:nnnnnnn
    { #1 } { #2 } { #3 } { #4 } { #5 } { #6 } { #7 }
    \IfValueT { #8 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #8 }
    }
  }
}

```

`\@twocolumntopinarea` Version 2.0 replaced `\twocolumntopinarea` Syntax: `{<pages>}{<type>}{<top height>}{<area-width>}{<area-height>}{<area-x>}{<area-y>}`

```

\cs_new:Nn \__flowfram_two_column_with_top_in_area:nnnnnnn
{
  \dim_set:Nn \l__flowfram_sfdf_height_dim { #3 }

```

Work out where to put the header frame.

```

\dim_set:Nn \l__flowfram_y_dim
{ #5 - \l__flowfram_sfdf_height_dim }
\dim_set:Nn \l__flowfram_height_dim
{ \l__flowfram_y_dim }
\dim_add:Nn \l__flowfram_y_dim { #7 }

```

```

\newframe
  [ #1 ] { #2 }
  { #4 } { \l__flowfram_sdf_height_dim }
  { #6 } { \l__flowfram_y_dim }

```

If `\twocolumn` is used in the document with its optional argument and the associated frames haven't yet been set, assume this is the header frame to use.

```

\tl_if_empty:NT \g__flowfram_two_col_header_id_tl
{
  \bool_lazy_and:nnT
  {
    \tl_if_empty_p:N \g__flowfram_headed_two_col_i_id_tl
  }
  {
    \tl_if_empty_p:N \g__flowfram_headed_two_col_ii_id_tl
  }
  {
    \tl_gset:Ne \g__flowfram_two_col_header_type_tl { #2 }
    \tl_gset:Ne \g__flowfram_two_col_header_id_tl
    {
      \int_use:c { c@max #2 }
    }
  }
}

```

work out height of the flow frames

```

\dim_sub:Nn \l__flowfram_height_dim { \vcolumnsep }
\iffvadjust
  \adjustheight { \l__flowfram_height_dim }
\fi

```

work out the widths of the flow frames

```

\dim_set:Nn \l__flowfram_width_dim
  { ( #4 - \columnsep ) / 2 }

```

work out the offset of the right column

```

\dim_set:Nn \l__flowfram_x_dim
  { \l__flowfram_width_dim + \columnsep + #6 }
\iflefttorightcolumns
  \__flowfram_new_flow:nnnnn
  { #1 } { \l__flowfram_width_dim } { \l__flowfram_height_dim } { #6 } { #7 }
  \setflowframe { \c@maxflow } { margin = left }
  \__flowfram_new_flow:nnnnn
  { #1 } { \l__flowfram_width_dim } { \l__flowfram_height_dim } { \l__flowfram_x_dim } { #7 }
  \setflowframe { \c@maxflow } { margin = right }
\else
  \__flowfram_new_flow:nnnnn
  { #1 } { \l__flowfram_width_dim } { \l__flowfram_height_dim } { \l__flowfram_x_dim } { #7 }
  \setflowframe { \c@maxflow } { margin = right }
  \__flowfram_new_flow:nnnnn
  { #1 } { \l__flowfram_width_dim } { \l__flowfram_height_dim } { #6 } { #7 }

```

```

    \setflowframe { \c@maxflow } { margin = left }
\fi

```

If `\twocolumn` is used in the document with its optional argument and the associated frames haven't yet been set, assume these are the columns to use.

```

\bool_lazy_and:nnT
{
  \tl_if_empty_p:N \g__flowfram_headed_two_col_i_id_tl
}
{
  \tl_if_empty_p:N \g__flowfram_headed_two_col_ii_id_tl
}
{
  \tl_gset:Ne \g__flowfram_headed_two_col_i_id_tl
  {
    \int_eval:n { \c@maxflow - \c_one_int }
  }
  \tl_gset:Ne \g__flowfram_headed_two_col_ii_id_tl { \int_use:N \c@maxflow }
}
}

```

`\twocolumnStopinarea` Special case for static frame.

```

\NewDocumentCommand \twocolumnStopinarea { O{all} m m m m m o }
{
  \__flowfram_only_preamble:Nn \twocolumnStopinarea
  {
    \__flowfram_two_column_with_top_in_area:nnnnnnn
    { #1 } { static } { #2 } { #3 } { #4 } { #5 } { #6 }
    \IfValueT { #7 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #7 }
    }
  }
}

```

`\twocolumnDtopinarea` Special case for dynamic frame.

```

\NewDocumentCommand \twocolumnDtopinarea { O{all} m m m m m o }
{
  \__flowfram_only_preamble:Nn \twocolumnStopinarea
  {
    \__flowfram_two_column_with_top_in_area:nnnnnnn
    { #1 } { dynamic } { #2 } { #3 } { #4 } { #5 } { #6 }
    \IfValueT { #7 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #7 }
    }
  }
}

```

`\Ncolumnstop` Similarly for an arbitrary number of flow frames. Special case where the area is the typeblock.

Syntax:

```
\Ncolumnstop[{pages}]{{type}}{{n}}{{H}}
\NewDocumentCommand \Ncolumnstop { O{all} m m m o }
{
  \__flowfram_only_preamble:Nn \Ncolumnstop
  {
    \__flowfram_n_column_with_top_in_area:nnnnnnn
    { #1 } { #2 } { #3 } { #4 }
    { \typeblockwidth } { \typeblockheight } { \c_zero_dim } { \c_zero_dim }
    \IfValueT { #5 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #5 }
    }
  }
}
```

`\NcolumnStop` Special case for static frame.

```
\NewDocumentCommand \NcolumnStop { O{all} m m o }
{
  \__flowfram_only_preamble:Nn \NcolumnStop
  {
    \__flowfram_n_column_with_top_in_area:nnnnnnn
    { #1 } { static } { #2 } { #3 }
    { \typeblockwidth } { \typeblockheight } { \c_zero_dim } { \c_zero_dim }
    \IfValueT { #4 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #4 }
    }
  }
}
```

`\NcolumnDtop` Special case for dynamic frame.

```
\NewDocumentCommand \NcolumnDtop { O{all} m m o }
{
  \__flowfram_only_preamble:Nn \NcolumnDtop
  {
    \__flowfram_n_column_with_top_in_area:nnnnnnn
    { #1 } { dynamic } { #2 } { #3 }
    { \typeblockwidth } { \typeblockheight } { \c_zero_dim } { \c_zero_dim }
    \IfValueT { #4 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #4 }
    }
  }
}
```

`\Ncolumntopinarea` Again test to make sure the user requested a sensible number. `\Ncolumntopinarea`

```
[{pages}]{{type}}{{n}}{{H}}{{w}}{{h}}{{x}}{{y}}
\NewDocumentCommand \Ncolumntopinarea { O{all} m m m m m m o }
```

```

{
  \__flowfram_only_preamble:Nn \Ncolumntopinarea
  {
    \__flowfram_n_column_with_top_in_area:nnnnnnnn
    { #1 } { #2 } { #4 } { #5 } { #6 } { #7 } { #8 }
    \IfValueT { #9 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #9 }
    }
  }
}
}
{\langle pages \rangle}{\langle type \rangle}{\langle n \rangle}{\langle H \rangle}{\langle w \rangle}{\langle h \rangle}{\langle x \rangle}{\langle y \rangle}
\cs_new:Nn \__flowfram_n_column_with_top_in_area:nnnnnnnn
{
  \int_case:nnF { #3 }
  {
    { \c_one_int }
    {
      \__flowfram_one_column_with_top_in_area:nnnnnnnn
      { #1 } { #2 } { #4 } { #5 } { #6 } { #7 } { #8 }
    }
    { 2 }
    {
      \__flowfram_two_column_with_top_in_area:nnnnnnnn
      { #1 } { #2 } { #4 } { #5 } { #6 } { #7 } { #8 }
    }
  }
}
{
  \int_compare:nNnTF { #3 } < { \c_one_int }
  {
    \msg_error:nnne { flowfram } { invalid-num-frames }
    { flow } { \int_eval:n { #3 } }
  }
  {
    \__flowfram_N_column_with_top_in_area:nnnnnnnn
    { #1 } { #2 } { #3 } { #4 } { #5 } { #6 } { #7 } { #8 }
  }
}
}
}

```

\@Ncolumntopinarea Fit the frames into specified area. $\langle pages \rangle$ $\langle type \rangle$ $\langle n \rangle$ $\langle H \rangle$ $\langle w \rangle$ $\langle h \rangle$ $\langle x \rangle$ $\langle y \rangle$. Version 2.0 replaced \@Ncolumntopinarea.

```

\cs_new:Nn \__flowfram_N_column_with_top_in_area:nnnnnnnn
{
  \dim_set:Nn \l__flowfram_sfdh_height_dim { #4 }

```

Work out where to put the static frame.

```

\dim_set:Nn \l__flowfram_y_dim
{ #6 - \l__flowfram_sfdh_height_dim }

```

```

\dim_set_eq:NN \l__flowfram_height_dim \l__flowfram_y_dim
\dim_add:Nn \l__flowfram_y_dim { #8 }
\newframe
  [ #1 ] { #2 }
  { #5 } { \l__flowfram_sdf_height_dim }
  { #7 } { \l__flowfram_y_dim }
work out height of the flow frames
  \dim_sub:Nn \l__flowfram_height_dim { \vcolumnsep }
adjust the flow frame height so that it is a multiple of \baselineskip
  \iffvadjust
  \adjustheight{\l__flowfram_height_dim}%
\fi
work out the widths of the flow frames
  \int_set:Nn \l__flowfram_col_int { #3 - \c_one_int }
  \dim_set:Nn \l__flowfram_width_dim
    { ( #5 - \l__flowfram_col_int \columnsep ) / ( #3 ) }
Set the x position of the first frame
  \dim_set:Nn \l__flowfram_x_dim { #7 }
  \iflefttorightcolumns
  \else
    \dim_add:Nn \l__flowfram_x_dim { #5 - \l__flowfram_width_dim }
  \fi
  \int_step_inline:nn { #3 }
  {
    \__flowfram_new_flow:nnnnn
    { #1 } { \l__flowfram_width_dim } { \l__flowfram_height_dim }
    { \l__flowfram_x_dim } { #8 }
work out the offset for the next column
  \iflefttorightcolumns
    \dim_add:Nn \l__flowfram_x_dim { \l__flowfram_width_dim + \columnsep }
  \else
    \dim_add:Nn \l__flowfram_x_dim { - \l__flowfram_width_dim - \columnsep }
  \fi
}
}

```

`\NcolumnStopinarea` Specific case for static frame. $\{\langle pages \rangle\}\{\langle n \rangle\}\{\langle H \rangle\}\{\langle w \rangle\}\{\langle h \rangle\}\{\langle x \rangle\}\{\langle y \rangle\}$.

```

\NewDocumentCommand \NcolumnStopinarea { 0{all} m m m m m m o }
{
  \__flowfram_only_preamble:Nn \NcolumnStopinarea
  {
    \__flowfram_n_column_with_top_in_area:nnnnnnnn
    { #1 } { static } { #2 } { #3 } { #4 } { #5 } { #6 } { #7 }
    \IfValueT { #8 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #8 }
    }
  }
}

```



```

    }
}

```

`\NcolumnDtopinarea` Specific case for dynamic frame.

```

\NewDocumentCommand \NcolumnDtopinarea { 0{all} m m m m m o }
{
  \__flowfram_only_preamble:Nn \NcolumnDtopinarea
  {
    \__flowfram_n_column_with_top_in_area:nnnnnnn
    { #1 } { dynamic } { #2 } { #3 } { #4 } { #5 } { #6 } { #7 }
    \IfValueT { #8 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #8 }
    }
  }
}

```

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>}
\NewDocumentCommand \onecolumnbottom { 0{all} m m o }
{
  \__flowfram_only_preamble:Nn \onecolumnbottom
  {
    \__flowfram_one_column_with_bottom_in_area:nnnnnnn
    { #1 } { #2 } { #3 }
    { \typeblockwidth } { \typeblockheight } { Opt } { Opt }
    \IfValueT { #4 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #4 }
    }
  }
}

```

`\onecolumnSbottom` Special case for static frame.

```

\NewDocumentCommand \onecolumnSbottom { 0{all} m o }
{
  \__flowfram_only_preamble:Nn \onecolumnbottom
  {
    \__flowfram_one_column_with_bottom_in_area:nnnnnnn
    { #1 } { static } { #2 }
    { \typeblockwidth } { \typeblockheight } { Opt } { Opt }
    \IfValueT { #3 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #3 }
    }
  }
}

```

`\onecolumnDbottom` Special case for dynamic frame.

```
\NewDocumentCommand \onecolumnDbottom { 0{all} m o }
{
  \__flowfram_only_preamble:Nn \onecolumnbottom
  {
    \__flowfram_one_column_with_bottom_in_area:nnnnnn
    { #1 } { dynamic } { #2 }
    { \typeblockwidth } { \typeblockheight } { Opt } { Opt }
    \IfValueT { #3 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #3 }
    }
  }
}
```

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*.

```
\NewDocumentCommand \onecolumnbottominarea
{ 0{all} m m m m m o }
{
  \__flowfram_only_preamble:Nn \onecolumnbottominarea
  {
    \__flowfram_one_column_with_bottom_in_area:nnnnnn
    { #1 } { #2 } { #3 } { #4 } { #5 } { #6 } { #7 }
    \IfValueT { #8 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #8 }
    }
  }
}

\cs_new:Nn \__flowfram_one_column_with_bottom_in_area:nnnnnn
{
  \dim_set:Nn \l__flowfram_sdf_height_dim { #3 }
  \dim_set:Nn \l__flowfram_height_dim
  {
    #5
    - \l__flowfram_sdf_height_dim
    - \vcolumnsep
  }
  \iffvadjust
  \adjustheight { \l__flowfram_height_dim }
  \fi
  \dim_set:Nn \l__flowfram_tmpa_dim
  {
    #5 - \l__flowfram_height_dim + #7
  }
}
```

```

    }
  \newframe [ #1 ] { #2 }
    { #4 } { \l__flowfram_sfdh_height_dim }
    { #6 } { #7 }
  \__flowfram_new_flow:nnnnn
    { #1 } { #4 } { \l__flowfram_height_dim } { #6 } { \l__flowfram_tmpa_dim }
}

```

`\onecolumnSbottominarea` Special case for static frame.

```

\NewDocumentCommand \onecolumnSbottominarea { O{all} m m m m m o }
{
  \__flowfram_only_preamble:Nn \onecolumnSbottominarea
  {
    \__flowfram_one_column_with_bottom_in_area:nnnnnnn
    { #1 } { static } { #2 } { #3 } { #4 } { #5 } { #6 }
    \IfValueT { #7 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #7 }
    }
  }
}

```

`\onecolumnDbottominarea` Special case for dynamic frame.

```

\NewDocumentCommand \onecolumnDbottominarea { O{all} m m m m m o }
{
  \__flowfram_only_preamble:Nn \onecolumnDbottominarea
  {
    \__flowfram_one_column_with_bottom_in_area:nnnnnnn
    { #1 } { dynamic } { #2 } { #3 } { #4 } { #5 } { #6 }
    \IfValueT { #7 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #7 }
    }
  }
}

```

`\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>}`.

```

\NewDocumentCommand \twocolumnbottom { O{all} m m o }
{
  \__flowfram_only_preamble:Nn \twocolumnbottom
  {
    \__flowfram_two_column_with_bottom_in_area:nnnnnnn
    { #1 } { #2 } { #3 } { \typeblockwidth } { \typeblockheight }
    { \c_zero_dim } { \c_zero_dim }
    \IfValueT { #4 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #4 }
    }
  }
}

```

```

    }
  }
}

```

`\twocolumnSbottom` Special case for static frame.

```

\NewDocumentCommand \twocolumnSbottom { 0{all} m o }
{
  \__flowfram_only_preamble:Nn \twocolumnDbottom
  {
    \__flowfram_two_column_with_bottom_in_area:nnnnnnn
    { #1 } { static } { #2 }
    { \typeblockwidth } { \typeblockheight } { \c_zero_dim } { \c_zero_dim }
    \IfValueT { #3 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #3 }
    }
  }
}

```

`\twocolumnDbottom` Special case for dynamic frame.

```

\NewDocumentCommand \twocolumnDbottom { 0{all} m o }
{
  \__flowfram_only_preamble:Nn \twocolumnDbottom
  {
    \__flowfram_two_column_with_bottom_in_area:nnnnnnn
    { #1 } { dynamic } { #2 }
    { \typeblockwidth } { \typeblockheight } { \c_zero_dim } { \c_zero_dim }
    \IfValueT { #3 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #3 }
    }
  }
}

```

`\twocolumnbottominarea` Now for a general area. Syntax:

```

\twocolumnbottominarea[<pages>]{<type>}{<H>}{<w>}{<h>}{<x>}{<y>}.
\NewDocumentCommand \twocolumnbottominarea { 0{all} m m m m m o }
{
  \__flowfram_only_preamble:Nn \twocolumnbottominarea
  {
    \__flowfram_two_column_with_bottom_in_area:nnnnnnn
    { #1 } { #2 } { #3 } { #4 } { #5 } { #6 } { #7 }
    \IfValueT { #8 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #8 }
    }
  }
}
\cs_new:Nn \__flowfram_two_column_with_bottom_in_area:nnnnnnn

```

```

{
\dim_set:Nn \l__flowfram_sdf_width_dim { #4 }
\dim_set:Nn \l__flowfram_sdf_height_dim { #3 }

```

Work out height of the flow frames.

```

\dim_set:Nn \l__flowfram_height_dim
{ #5 - \l__flowfram_sdf_height_dim - \vcolumnsep }
\iffvadjust
\adjustheight { \l__flowfram_height_dim }
\fi
\newframe
[ #1 ] { #2 }
{ \l__flowfram_sdf_width_dim } { \l__flowfram_sdf_height_dim }
{ #6 } { #7 }

```

Work out the y position of the flow frames.

```

\dim_set:Nn \l__flowfram_y_dim { #5 - \l__flowfram_height_dim + #7 }

```

work out the widths of the flow frames

```

\dim_set:Nn \l__flowfram_width_dim
{
( \l__flowfram_sdf_width_dim - \columnsep ) /2
}

```

work out the x offset of the right column

```

\dim_set:Nn \l__flowfram_x_dim
{
\l__flowfram_width_dim + \columnsep + #6
}

```

Define the frames

```

\iflefttorightcolumns
\__flowfram_new_flow:nnnnn
{ #1 } { \l__flowfram_width_dim } { \l__flowfram_height_dim }
{ #6 } { \l__flowfram_y_dim }
\setflowframe{\c@maxflow}{margin=left}%
\__flowfram_new_flow:nnnnn
{ #1 } { \l__flowfram_width_dim } { \l__flowfram_height_dim }
{ \l__flowfram_x_dim } { \l__flowfram_y_dim }
\setflowframe{\c@maxflow}{margin=right}%
\else
\__flowfram_new_flow:nnnnn
{ #1 } { \l__flowfram_width_dim } { \l__flowfram_height_dim }
{ \l__flowfram_x_dim } { \l__flowfram_y_dim }
\setflowframe{\c@maxflow}{margin=right}%
\__flowfram_new_flow:nnnnn
{ #1 } { \l__flowfram_width_dim } { \l__flowfram_height_dim }
{ #6 } { \l__flowfram_y_dim }
\setflowframe{\c@maxflow}{margin=left}%
\fi
}

```

`\twocolumnSbottominarea` Special case for static frame.

```
\NewDocumentCommand \twocolumnSbottominarea { O{all} m m m m m o }
{
  \__flowfram_only_preamble:Nn \twocolumnSbottominarea
  {
    \__flowfram_two_column_with_bottom_in_area:nnnnnnn
    { #1 } { static } { #2 } { #3 } { #4 } { #5 } { #6 }
    \IfValueT { #7 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #7 }
    }
  }
}
```

`\twocolumnDbottominarea` Special case for dynamic frame.

```
\NewDocumentCommand \twocolumnDbottominarea { O{all} m m m m m o }
{
  \__flowfram_only_preamble:Nn \twocolumnDbottominarea
  {
    \__flowfram_two_column_with_bottom_in_area:nnnnnnn
    { #1 } { dynamic } { #2 } { #3 } { #4 } { #5 } { #6 }
    \IfValueT { #7 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #7 }
    }
  }
}
```

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}`.

```
\NewDocumentCommand \Ncolumnbottom { O{all} m m m o }
{
  \__flowfram_only_preamble:Nn \Ncolumnbottominarea
  {
    \__flowfram_n_column_with_bottom_in_area:nnnnnnnn
    { #1 } { #2 } { #3 } { #4 }
    { \typeblockwidth } { \typeblockheight } { \c_zero_dim } { \c_zero_dim }
    \IfValueT { #5 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #5 }
    }
  }
}
```

`\NcolumnSbottom` Special case for static frame.

```
\NewDocumentCommand \NcolumnSbottom { O{all} m m o }
```

```

{
  \__flowfram_only_preamble:Nn \NcolumnSbottom
  {
    \__flowfram_n_column_with_bottom_in_area:nnnnnnnn
    { #1 } { static } { #2 } { #3 }
    { \typeblockwidth } { \typeblockheight } { \c_zero_dim } { \c_zero_dim }
    \IfValueT { #4 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #4 }
    }
  }
}

```

\NcolumnDbottom Special case for dynamic frame.

```

\NewDocumentCommand \NcolumnDbottom { O{all} m m o }
{
  \__flowfram_only_preamble:Nn \NcolumnDbottom
  {
    \__flowfram_n_column_with_bottom_in_area:nnnnnnnn
    { #1 } { dynamic } { #2 } { #3 }
    { \typeblockwidth } { \typeblockheight } { \c_zero_dim } { \c_zero_dim }
    \IfValueT { #4 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #4 }
    }
  }
}

```

\Ncolumnbottominarea Again check the user has requested a sensible number. \Ncolumnbottominarea
 $[<pages>]{<type>}{<n>}{<H>}{<w>}{<h>}{<x>}{<y>}$

```

\NewDocumentCommand \Ncolumnbottominarea { O{all} m m m m m m o }
{
  \__flowfram_only_preamble:Nn \Ncolumnbottominarea
  {
    \__flowfram_n_column_with_bottom_in_area:nnnnnnnn
    { #1 } { #2 } { #3 } { #4 } { #5 } { #6 } { #7 } { #8 }
    \IfValueT { #9 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #9 }
    }
  }
}

```

\@Ncolumnbottominarea Version 2.0 replaced \@Ncolumnbottominarea

```

\cs_new:Nn \__flowfram_n_column_with_bottom_in_area:nnnnnnnn
{
  \int_case:nnF { #3 }
  {
    { \c_one_int }
  }
}

```

```

    {
      \_flowfram_one_column_with_bottom_in_area:nnnnnnn
      { #1 } { #2 } { #4 } { #5 } { #6 } { #7 } { #8 }
    }
  { 2 }
  {
    \_flowfram_two_column_with_bottom_in_area:nnnnnnn
    { #1 } { #2 } { #4 } { #5 } { #6 } { #7 } { #8 }
  }
}
{
\int_compare:nNnTF { #3 } > { 2 }
{
  \_flowfram_N_column_with_bottom_in_area:nnnnnnnn
  { #1 } { #2 } { #3 } { #4 } { #5 } { #6 } { #7 } { #8 }
}
{
  \msg_error:nne { flowfram } { invalid-num-frames }
  { flow } { \int_eval:n { #3 } }
}
}
}

```

Syntax: $\langle\text{pages}\rangle\langle\text{type}\rangle\langle n\rangle\langle H\rangle\langle w\rangle\langle h\rangle\langle x\rangle\langle y\rangle$

```

\cs_new:Nn \_flowfram_N_column_with_bottom_in_area:nnnnnnnn
{
  \dim_set:Nn \l_flowfram_sfdh_height_dim { #4 }

```

Work out height of the flow frames.

```

\dim_set:Nn \l_flowfram_height_dim
{ #6 - \l_flowfram_sfdh_height_dim - \vcolumnsep }

```

adjust the flow frame height so that it is a multiple of `\baselineskip`

```

\iffvadjust
\adjustheight { \l_flowfram_height_dim }
\fi

```

```

\newframe
[ #1 ] { #2 }
{ #5 } { \l_flowfram_sfdh_height_dim } { #7 } { #8 }

```

Work out the y offset of the flow frames.

```

\dim_set:Nn \l_flowfram_y_dim
{ #6 - \l_flowfram_height_dim + #8 }

```

work out the widths of the flow frames

```

\int_set:Nn \l_flowfram_col_int { #3 - \c_one_int }
\dim_set:Nn \l_flowfram_width_dim
{ ( #5 - \l_flowfram_col_int \columnsep ) / ( #3 ) }

```

Set the x offset of the first frame.

```

\dim_set:Nn \l_flowfram_x_dim { #7 }
\iflefttorightcolumns

```



```

\else
  \dim_add:Nn \l__flowfram_x_dim { #5 - \l__flowfram_width_dim }
\fi
\int_step_inline:nn { #3 }
{
  \__flowfram_new_flow:nnnnn
  { #1 } { \l__flowfram_width_dim } { \l__flowfram_height_dim }
  { \l__flowfram_x_dim } { \l__flowfram_y_dim }

```

Work out the offset for the next column.

```

\iflefttorightcolumns
  \dim_add:Nn \l__flowfram_x_dim { \l__flowfram_width_dim + \columnsep }
\else
  \dim_add:Nn \l__flowfram_x_dim { - \l__flowfram_width_dim -\columnsep }
\fi
}
}

```

`\NcolumnSbottominarea` Specific case for static frame.

```

\NewDocumentCommand \NcolumnSbottominarea { O{all} m m m m m m o }
{
  \__flowfram_only_preamble:Nn \NcolumnSbottominarea
  {
    \__flowfram_n_column_with_bottom_in_area:nnnnnnnn
    { #1 } { static } { #2 } { #3 } { #4 } { #5 } { #6 } { #7 }
    \IfValueT { #8 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #8 }
    }
  }
}

```

`\NcolumnDbottominarea` Specific case for dynamic frame.

```

\NewDocumentCommand \NcolumnDbottominarea { O{all} m m m m m m o }
{
  \__flowfram_only_preamble:Nn \NcolumnDbottominarea
  {
    \__flowfram_n_column_with_bottom_in_area:nnnnnnnn
    { #1 } { dynamic } { #2 } { #3 } { #4 } { #5 } { #6 } { #7 }
    \IfValueT { #8 }
    {
      \__flowfram_set_frame_id:nnn { flow } { \c@maxflow } { #8 }
    }
  }
}

```

`\@ff@adjh` Version 2.0 replaced `\@ff@adjh`

```

\int_new:N \l__flowfram_adjust_height_int

```

`\adjustheight` Given a height #1 (a length), adjust it so that it is a multiple of `\baselineskip`.

```

\newcommand*{\adjustheight}[1]{%

```

Convert the dimension to an integer.

```
\int_set:Nn \l__flowfram_adjust_height_int
{
  \int_div_round:nn
    { \dim_to_decimal_in_sp:n { #1 } }
    { \dim_to_decimal_in_sp:n { \baselineskip } }
}
\dim_set:Nn #1 { \l__flowfram_adjust_height_int \baselineskip }
}
```

`\adjustcolsep` Adjust the value of `\columnsep` so that the margins will fit between columns.

```
\newcommand*{\adjustcolsep}{%
  \dim_set:Nn \columnsep { 2 \columnsep + \marginparwidth }
}
```

1.12.2 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}.
```

Vertical two tone effect where the height of the static frames is equal to the paper height.

```
\NewDocumentCommand \vtwotone
{ 0{all} 0 {\c_zero_dim} m m m m m m }
{
  \__flowfram_only_preamble:Nn \vtwotone
  {
    \__flowfram_v_two_tone_bottom:nnnnnnnn
    { #1 } { #2 } { \paperheight } { #3 } { #4 } { #5 } { #6 } { #7 } { #8 }
  }
}
```

`\@vtwotone` Version 2.0 removed `\@vtwotone`.

Vertical two tone effect along the bottom of the page, of height $\langle H \rangle$. Syntax:

`[<pages>] [<xoffset>]{<H>}{<W1>}{<C1>}{<L1>}{<W2>}{<C2>}{<L2>}` where the first frame starts at $\langle xoffset \rangle$.

```
\cs_new:Nn \__flowfram_v_two_tone_bottom:nnnnnnnn
{
  \computeleftedgeodd { \l__flowfram_x_dim }
  \if@twoside
```

```

    \computeleftedgeeven { \l__flowfram_evenx_dim }
\else
    \dim_set_eq:NN \l__flowfram_evenx_dim \l__flowfram_x_dim
\fi
\computebottomedge { \l__flowfram_y_dim }
\dim_add:Nn \l__flowfram_x_dim { #2 }
\dim_add:Nn \l__flowfram_evenx_dim { #2 }
\__flowfram_next_v_band:nnnnn { #1 } { #3 } { #4 } { #5 } { #6 }
\__flowfram_next_v_band:nnnnn { #1 } { #3 } { #7 } { #8 } { #9 }
}

```

`\vtwotonebottom` Border strip along the bottom of the page

```

\NewDocumentCommand \vtwotonebottom
{ 0{all} 0 {\c_zero_dim} m m m m m m }
{
    \__flowfram_only_preamble:Nn \vtwotonebottom
    {
        \__flowfram_v_two_tone_bottom:nnnnnnnnn
        { #1 } { #2 } { #3 } { #4 } { #5 } { #6 } { #7 } { #8 } { #9 }
    }
}

```

`\@vtwotonebottom` Version 2.0 removed `\@vtwotonebottom`.

`\vtwotonetop` Border strip along the top of the page.

```

\NewDocumentCommand \vtwotonetop
{ 0{all} 0 {\c_zero_dim} m m m m m m }
{
    \__flowfram_only_preamble:Nn \vtwotonebottom
    {
        \__flowfram_v_two_tone_top:nnnnnnnnn
        { #1 } { #2 } { #3 } { #4 } { #5 } { #6 } { #7 } { #8 } { #9 }
    }
}

```

`\@vtwotonetop` Version 2.0 removed `\@vtwotonetop`

`\@@vtwotonetop` Version 2.0 replaced `\@@vtwotonetop`

```

\cs_new:Nn \__flowfram_v_two_tone_top:nnnnnnnnn
{
    \computeleftedgeodd { \l__flowfram_x_dim }
    \if@twoside
        \computeleftedgeeven { \l__flowfram_evenx_dim }
    \else
        \dim_set:Nn \l__flowfram_evenx_dim { \l__flowfram_x_dim }
    \fi
    \computetopedge { \l__flowfram_y_dim }
    \dim_sub:Nn \l__flowfram_y_dim { #3 }
    \dim_add:Nn \l__flowfram_x_dim { #2 }
}

```

```

\dim_add:Nn \l__flowfram_evenx_dim { #2 }
\__flowfram_next_v_band:nnnnn { #1 } { #3 } { #4 } { #5 } { #6 }
\__flowfram_next_v_band:nnnnn { #1 } { #3 } { #7 } { #8 } { #9 }
}

```

`\@nextvband` Make next static frame. Syntax:

```

\@nextvband{<pages>}{<height>}{<width>}{<colour specs>}{<label>}

```

x and *y* offsets are given by `\l__flowfram_x_dim` and `\l__flowfram_y_dim`.
On exit, `\l__flowfram_x_dim` is set to the right border. Version 2.0 renamed `\@nextvband`.

```

\cs_new:Nn \__flowfram_next_v_band:nnnnn
{
  \dim_set:Nn \l__flowfram_sdf_width_dim { #3 }
  \tl_if_empty:nTF { #5 }
  {
    \__flowfram_new_static:nnnnn
    { #1 } { \l__flowfram_sdf_width_dim } { #2 }
    { \l__flowfram_x_dim } { \l__flowfram_y_dim }
  }
  {
    \__flowfram_new_static:nnnnn
    { #1 } { \l__flowfram_sdf_width_dim } { #2 }
    { \l__flowfram_x_dim } { \l__flowfram_y_dim } { #5 }
  }
  \flowfram_frame_set_dim:nnnn { static } { evenx } { \c@maxstatic }
  { \l__flowfram_evenx_dim }
  \__flowfram_set_frame_color:w
  #4 \q_stop { static } { backcolor } { \c@maxstatic }
  \dim_add:Nn \l__flowfram_x_dim { \l__flowfram_sdf_width_dim }
  \dim_add:Nn \l__flowfram_evenx_dim { \l__flowfram_sdf_width_dim }
}

```

`\@thisstrip` Keep track of which strip we are doing. Version 2.0 renamed `\@thisstrip`.

```

\int_new:N \l__flowfram_this_strip_int

```

`\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*.

```

\NewDocumentCommand \vNtone { 0{all} 0 {\c_zero_dim} m }
{
  \clist_set:Nn \l__flowfram_pages_clist { #1 }
  \__flowfram_v_N_tone_bottom:nnn { #2 } { #3 } { \paperheight }
}
\@onlypreamble{\vNtone}

```

`\@vNtone` Got the first argument, now get the next. Version 2.0 removed `\@vNtone`.

`\@@vNtone` Vertical $\langle n \rangle$ tone aligned along the bottom of the page with height #3. Version 2.0 renamed `\@vNtone`.

```
\cs_new:Nn \__flowfram_v_N_tone_bottom:nnn
{
  \computeleftedgeodd { \l__flowfram_x_dim }
  \if@twoside
    \computeleftedgeeven { \l__flowfram_evenx_dim }
  \else
    \dim_set_eq:NN \l__flowfram_evenx_dim \l__flowfram_x_dim
  \fi
  \computebottomedge { \l__flowfram_y_dim }
  \dim_add:Nn \l__flowfram_x_dim { #1 }
  \dim_add:Nn \l__flowfram_evenx_dim { #1 }
  \int_set:Nn \l__flowfram_this_strip_int { #2 }
  \dim_set:Nn \l__flowfram_sfdh_height_dim { #3 }
  \__flowfram_next_v_N_band:
}
```

`\@nextvNband` Recursively do the next strip. Version 2.0 renamed `\@nextvNband`.

```
\cs_new:Nn \__flowfram_next_v_N_band:
{
  \int_compare:nNnTF { \l__flowfram_this_strip_int } > { \c_zero_int }
  {
    \let \flf@next \@nextvNband
  }
  {
    \let \flf@next \relax
  }
  \int_decr:N \l__flowfram_this_strip_int
  \flf@next
}
```

`\@@nextvNband` Do current strip, and go on to next one.

```
\newcommand*{\@@nextvNband}[3]{%
  \__flowfram_next_v_band:nnnnn
  { \l__flowfram_pages_clist } { \l__flowfram_sfdh_height_dim }
  { #1 } { #2 } { #3 }
  \__flowfram_next_v_N_band:
}
```

`\vNtonebottom` Border strip along the bottom of the page. Same as above but user specifies the height.

```
\NewDocumentCommand \vNtonebottom { 0{all} 0{\c_zero_dim} m m }
{
  \clist_set:Nn \l__flowfram_pages_clist { #1 }
  \__flowfram_v_N_tone_bottom:nnn { #2 } { #3 } { #4 }
}
\@onlypreamble{\vNtonebottom}
```

`\@vNtonebottom` Version 2.0 removed `\@vNtonebottom`.

`\vNtonetop` Border strip along the top of the page. Again two optional arguments are required. Get first optional argument.

```
\NewDocumentCommand \vNtonetop { 0{all} 0{\c_zero_dim} m m }
{
  \clist_set:Nn \l__flowfram_pages_clist { #1 }
  \__flowfram_v_N_tone_top:nnn { #2 } { #3 } { #4 }
}
\@onlypreamble{\vNtonetop}
```

`\@vNtonetop` Get next optional argument. Version 2.0 removed `\@vNtonetop`.

`\@@vNtonetop` Now get on with it. Again, it has to be done recursively. Version 2.0 renamed

```
\@@vNtonetop
\cs_new:Nn \__flowfram_v_N_tone_top:nnn
{
  \computeleftedgeodd { \l__flowfram_x_dim }
  \if@twoside
    \computeleftedgeeven { \l__flowfram_evenx_dim }
  \else
    \dim_set_eq:NN \l__flowfram_evenx_dim \l__flowfram_x_dim
  \fi
  \computetopedge { \l__flowfram_y_dim }
  \dim_sub:Nn \l__flowfram_y_dim { #3 }
  \dim_add:Nn \l__flowfram_x_dim { #1 }
  \dim_add:Nn \l__flowfram_evenx_dim { #1 }
  \int_set:Nn \l__flowfram_this_strip_int { #2 }
  \dim_set:Nn \l__flowfram_sdf_height_dim { #3 }
  \__flowfram_next_v_N_band:
}
```

`\htwotone` Now do horizontal strips. Syntax:

```
\htwotone[[pages]][[y offset]]{<H1>}{<C1>}{<L1>}{<H2>}{<C2>}{<L2>}
```

```
\NewDocumentCommand \htwotone
{ 0{all} 0 {\c_zero_dim} m m m m m m }
{
  \__flowfram_only_preamble:Nn \htwotone
  {
    \__flowfram_h_two_tone_left:nnnnnnnnn
    { #1 } { #2 } { \paperwidth } { #3 } { #4 } { #5 } { #6 } { #7 } { #8 }
  }
}
\@onlypreamble{\htwotone}
```

`\@htwotone` Version 2.0 removed `\@htwotone`.

`\@@htwotoneleft` This is all done in much the same way as the vertical strips. Version 2.0 replaced

```
\@@htwotoneleft.
```

```
\cs_new:Nn \__flowfram_h_two_tone_left:nnnnnnnnn
{
```

```

\computeleftedgeodd { \l__flowfram_x_dim }
\if@twoside
  \computeleftedgeeven { \l__flowfram_evenx_dim }
\else
  \dim_set:Nn \l__flowfram_evenx_dim { \l__flowfram_x_dim }
\fi
\computebottomedge { \l__flowfram_y_dim }
\dim_add:Nn \l__flowfram_y_dim { #2 }
\__flowfram_h_two_tone_left:nnnnnnnn { #1 } { #3 } { #4 } { #5 } { #6 }
\__flowfram_h_two_tone_left:nnnnnnnn { #1 } { #3 } { #7 } { #8 } { #9 }
}

```

`\htwotoneleft` Two tone horizontal strips along left border Syntax: `\htwotoneleft[⟨pages⟩`
`[⟨y offset⟩]{⟨width⟩}{⟨H1⟩}{⟨C1⟩}{⟨L1⟩}{⟨H2⟩}{⟨C2⟩}{⟨L2⟩}`

```

\NewDocumentCommand \htwotoneleft
  { 0{all} 0 {\c_zero_dim} m m m m m m m }
{
  \__flowfram_only_preamble:Nn \htwotoneleft
  {
    \__flowfram_h_two_tone_left:nnnnnnnn
    { #1 } { #2 } { #3 } { #4 } { #5 } { #6 } { #7 } { #8 } { #9 }
  }
}
\@onlypreamble{\htwotoneleft}

```

`\@htwotoneleft` Version 2.0 removed `\@htwotoneleft`

`\htwotoneright` Two tone horizontal strips along right border

```

\NewDocumentCommand \htwotoneright
  { 0{all} 0 {\c_zero_dim} m m m m m m m }
{
  \__flowfram_only_preamble:Nn \htwotoneright
  {
    \__flowfram_h_two_tone_right:nnnnnnnn
    { #1 } { #2 } { #3 } { #4 } { #5 } { #6 } { #7 } { #8 } { #9 }
  }
}
\@onlypreamble{\htwotoneright}

```

`\@htwotoneright` Version 2.0 removed `\@htwotoneright`.

`\@@htwotoneright` Version 2.0 replaced `\@htwotoneright`.

```

\cs_new:Nn \__flowfram_h_two_tone_right:nnnnnnnn
{
  \computerightedgeodd { \l__flowfram_x_dim }
  \if@twoside
    \computerightedgeeven { \l__flowfram_evenx_dim }
  \else
    \dim_set:Nn \l__flowfram_evenx_dim { \l__flowfram_x_dim }
  \fi
}

```

```

\fi
\computebottomedge { \l__flowfram_y_dim }
\dim_add:Nn \l__flowfram_y_dim { #2 }
\dim_sub:Nn \l__flowfram_x_dim { #3 }
\dim_sub:Nn \l__flowfram_evenx_dim { #3 }
\__flowfram_h_two_tone_left:nnnnnnnn { #1 } { #3 } { #4 } { #5 } { #6 }
\__flowfram_h_two_tone_left:nnnnnnnn { #1 } { #3 } { #7 } { #8 } { #9 }
}

```

`\hNtone` Now for $\langle N \rangle$ coloured horizontal strips

```

\NewDocumentCommand \hNtone { 0{all} 0 {\c_zero_dim} m }
{
  \clist_set:Nn \l__flowfram_pages_clist { #1 }
  \__flowfram_h_N_tone_left:nnn { #2 } { #3 } { \paperwidth }
}
\@onlypreamble{\hNtone}

```

`\@hNtone` Version 2.0 removed `\@hNtone`.

`\@@hNtone` Version 2.0 replaced `\@@hNtone`.

```

\cs_new:Nn \__flowfram_h_N_tone_left:nnn
{
  \computeleftedgeodd { \l__flowfram_x_dim }
  \if@twoside
    \computeleftedgeeven { \l__flowfram_evenx_dim }
  \else
    \dim_add:Nn \l__flowfram_evenx_dim { \l__flowfram_x_dim }
  \fi
  \computebottomedge { \l__flowfram_y_dim }
  \dim_add:Nn \l__flowfram_y_dim { #1 }
  \int_set:Nn \l__flowfram_this_strip_int { #2 }
  \dim_set:Nn \l__flowfram_static_width_dim { #3 }
  \__flowfram_next_h_N_band:
}

```

`\hNtoneleft` Now for the N tone strips along the left border

```

\NewDocumentCommand \hNtoneleft { 0{all} 0 {\c_zero_dim} m m }
{
  \clist_set:Nn \l__flowfram_pages_clist { #1 }
  \__flowfram_h_N_tone_left:nnn { #2 } { #3 } { #4 }
}
\@onlypreamble{\hNtoneleft}

```

`\@hNtoneleft` Version 2.0 removed `\@hNtoneleft`.

`\hNtoneright` Border strip along the right border

```

\NewDocumentCommand \hNtoneright { 0{all} 0 {\c_zero_dim} m m }
{
  \clist_set:Nn \l__flowfram_pages_clist { #1 }
  \__flowfram_h_N_tone_right:nnn { #2 } { #3 } { #4 }
}

```



```

}
\@onlypreamble{\hNtoneright}

```

\@hNtoneright Version 2.0 removed \@hNtoneright.

\@@hNtoneright Version 2.0 replaced \@@hNtoneright.

```

\cs_new:Nn \__flowfram_h_N_tone_right:nnn
{
  \computerightedgeodd { \l__flowfram_x_dim }
  \if@twoside
    \computerightedgeeven { \l__flowfram_evenx_dim }
  \else
    \dim_set:Nn \l__flowfram_evenx_dim { \l__flowfram_x_dim }
  \fi
  \computebottomedge { \l__flowfram_y_dim }
  \dim_add:Nn \l__flowfram_y_dim { #1 }
  \dim_sub:Nn \l__flowfram_x_dim { #3 }
  \dim_sub:Nn \l__flowfram_evenx_dim { #3 }
  \int_set:Nn \l__flowfram_this_strip_int { #2 }
  \dim_set:Nn \l__flowfram_sdf_width_dim { #3 }
  \__flowfram_next_h_N_band:
}

```

\@nextband Make next static frame. Syntax:

\@nextband{<pages>}{<width>}{<height>}{<colour specs>}{<label>}

x and *y* offsets are given by \l__flowfram_x_dim and \l__flowfram_y_dim .

On exit, \l__flowfram_y_dim is set to the top border. Version 2.0 replaced

\@nextband.

```

\cs_new:Nn \__flowfram_next_h_band:nnnnn
{
  \dim_set:Nn \l__flowfram_sdf_height_dim { #3 }
  \tl_if_empty:nTF { #5 }
  {
    \__flowfram_new_static:nnnnn
    { #1 } { #2 } { \l__flowfram_sdf_height_dim }
    { \l__flowfram_x_dim } { \l__flowfram_y_dim }
  }
  {
    \__flowfram_new_static:nnnnn
    { #1 } { #2 } { \l__flowfram_sdf_height_dim }
    { \l__flowfram_x_dim } { \l__flowfram_y_dim }
  }
  \flowfram_frame_set_dim:nnnn { static } { evenx } { \c@maxstatic }
  { \l__flowfram_evenx_dim }
  \__flowfram_set_frame_color:w
  #4 \q_stop { static } { backcolor } { \c@maxstatic }
  \dim_add:Nn \l__flowfram_y_dim { \l__flowfram_sdf_height_dim }
}

```

`\@nexthNband` Get next horizontal strip recursively. Version 2.0 renamed `\@nexthNband`.

```
\cs_new:Nn \__flowfram_next_h_N_band:
{
  \int_compare:nNnTF
    { \l__flowfram_this_strip_int } > { \c_zero_int }
  {
    \let \flf@next \@nexthNband
  }
  {
    \let \flf@next \relax
  }
  \int_decr:N \l__flowfram_this_strip_int
  \flf@next
}
```

`\@@nexthNband`

```
\newcommand*{\@@nexthNband}[3]{%
  \__flowfram_h_two_tone_left:nnnnnnnnn
  { \l__flowfram_pages_clist } { \l__flowfram_sdf_width_dim }
  { #1 } { #2 } { #3 }
  \__flowfram_next_h_N_band:
}
```

`\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}]`.

```
\NewDocumentCommand \makebackgroundframe { 0{all} o }
{
  \__flowfram_only_preamble:Nn \NcolumnDbottominarea
  {
    \int_if_zero:nF { \c@maxstatic }
    {
      \msg_warning:nn { flowfram } { background-not-first }
    }
    \computeleftedgeodd { \l__flowfram_x_dim }
    \if@twoside
      \computeleftedgeeven { \l__flowfram_evenx_dim }
    \else
      \dim_set:Nn \l__flowfram_evenx_dim { \l__flowfram_x_dim }
    \fi
    \computebottomedge { \l__flowfram_y_dim }
    \IfValueTF { #2 }
    {
      \__flowfram_new_static:nnnnnn
      { #1 } { \paperwidth } { \paperheight }
      { \l__flowfram_x_dim } { \l__flowfram_y_dim }
      { #2 }
    }
  }
}
```

```

    {
      \__flowfram_new_static:nnnnn
      { #1 } { \paperwidth } { \paperheight }
      { \l__flowfram_x_dim } { \l__flowfram_y_dim }
    }
  \flowfram_frame_set_dim:nnnn { static} { evenx } { \c@maxstatic }
  { \l__flowfram_evenx_dim }
}
}

```

1.12.3 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 $\langle y \text{ top} \rangle$ or continue below the lowest point by $\langle y \text{ bottom} \rangle$.

`\ffcolumseprule` This has changed in v1.09. Define `\ffcolumseprule` and use instead of `\columnseprule`

```

\newlength\ffcolumseprule
\setlength{\ffcolumseprule}{2pt}

```

`\ffrulerdeclarations` 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*\ffrulerdeclarations{}

```

`\insertvrule` Determine whether or not the starred version is being used. The starred version uses the label for the id. The first optional argument $\langle y\text{-top} \rangle$ extends the rule by that much above the highest point. The second optional argument $\langle y\text{-bottom} \rangle$ extends the rule by that much below the lowest point. Syntax: `\insertvrule*`

```

[⟨y-top⟩][⟨y-bottom⟩]{⟨type-1⟩}{⟨id⟩}{⟨type-2⟩}{⟨id⟩}

```

```

\NewDocumentCommand \insertvrule
  { s 0 { \c_zero_dim } 0 { \c_zero_dim } m m m m }
{
  \__flowfram_only_preamble:Nn \insertvrule
  {
    \__flowfram_get_frame_type:n { #4 }
    \__flowfram_get_frame_type:Nn
      \l__flowfram_type_ii_int
      { #6 }
    \bool_lazy_or:nnF
      {
        \int_compare_p:nNn
          { \l__flowfram_type_int } = { -\c_one_int }
      }
  }
}

```

```

    {
      \int_compare_p:nNn
        { \l__flowfram_type_ii_int } = { -\c_one_int }
    }
  {
    \IfBooleanTF { #1 }
    {
      \use:c { __flowfram_get_ #6 _id:e } { #7 }
      \int_set_eq:NN \l__flowfram_id_ii_int \l__flowfram_id_int
      \use:c { __flowfram_get_ #4 _id:e } { #5 }
      \__flowfram_insert_v_rule:nnNnNn
        { #2 } { #3 }
      \l__flowfram_type_int { \l__flowfram_id_int }
      \l__flowfram_type_ii_int { \l__flowfram_id_ii_int }
    }
    {
      \__flowfram_insert_v_rule:nnNnNn
        { #2 } { #3 }
      \l__flowfram_type_int { #5 }
      \l__flowfram_type_ii_int { #7 }
    }
  }
}
}
}

```

\@insertvrule Version 2.0 removed \@insertvrule,

Need some lengths:

\@ff@left@x Version 2.0 replaced \@ff@left@x.
 \dim_new:N \l__flowfram_left_x_dim

\@ff@left@y Version 2.0 replaced \@ff@left@y.
 \dim_new:N \l__flowfram_left_y_dim

\@ff@left@evenx Version 2.0 replaced \@ff@left@evenx.
 \dim_new:N \l__flowfram_left_evenx_dim

\@ff@left@eveny Version 2.0 replaced \@ff@left@eveny.
 \dim_new:N \l__flowfram_left_eveny_dim

\@ff@left@width Version 2.0 replaced \@ff@left@width.
 \dim_new:N \l__flowfram_left_width_dim

\@ff@left@height Version 2.0 replaced \@ff@left@height.
 \dim_new:N \l__flowfram_left_height_dim

\@@insertvrule Version 2.0 removed \@@insertvrule.

`\@@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:
 $\langle y\text{-top} \rangle \langle y\text{-bottom} \rangle \langle type\text{-1 int} \rangle \langle idn \rangle \langle type\text{-2 int} \rangle \langle idn \rangle$ Version 2.0 replaced `\@@insert@vrule`

```

\cs_new:Nn \__flowfram_insert_v_rule:nNnNn
{
  \__flowfram_get_frame_bounds_by_typeid:n { #3 } { #4 }
  \dim_set_eq:NN \l__flowfram_left_x_dim \ffareax
  \dim_set_eq:NN \l__flowfram_left_y_dim \ffareay
  \dim_set_eq:NN \l__flowfram_left_width_dim \ffareawidth
  \dim_set_eq:NN \l__flowfram_left_height_dim \ffareaheight
  \__flowfram_get_frame_bounds_by_typeid:n { #5 } { #6 }
  \dim_compare:nNnT
    { \l__flowfram_left_x_dim } > { \ffareax }
  {
    \__flowfram_swap_dim:NN \l__flowfram_left_x_dim \ffareax
    \__flowfram_swap_dim:NN \l__flowfram_left_y_dim \ffareay
    \__flowfram_swap_dim:NN \l__flowfram_left_evenx_dim \ffareaevenx
    \__flowfram_swap_dim:NN \l__flowfram_left_eveny_dim \ffareaeveny
    \__flowfram_swap_dim:NN \l__flowfram_left_width_dim \ffareawidth
    \__flowfram_swap_dim:NN \l__flowfram_left_height_dim \ffareaheight
  }
  \dim_set:Nn \l__flowfram_x_dim
  {
    \l__flowfram_left_x_dim
    + \l__flowfram_left_width_dim
  }
  \dim_set:Nn \l__flowfram_sdf_width_dim
  {
    \ffareax - \l__flowfram_x_dim
  }
  \dim_set:Nn \l__flowfram_sdf_height_dim
  {
    \l__flowfram_left_y_dim
    + \l__flowfram_left_height_dim
  }
  \dim_set:Nn \l__flowfram_y_dim
  {
    \ffareay + \ffareaheight
  }
  \dim_compare:nNnT
    { \l__flowfram_y_dim } > { \l__flowfram_sdf_height_dim }
  {
    \dim_set_eq:NN \l__flowfram_sdf_height_dim \l__flowfram_y_dim
  }
  \dim_compare:nNnTF
    { \l__flowfram_left_y_dim } < { \ffareay }
  {
    \dim_set_eq:NN \l__flowfram_y_dim \l__flowfram_left_y_dim
  }
}

```

```

}
{
  \dim_set_eq:NN \l__flowfram_y_dim \ffareay
}
\dim_add:Nn \l__flowfram_sdf_height_dim
{
  - \l__flowfram_y_dim
}
\__flowfram_new_static:nnnn
{ all }
{ \l__flowfram_sdf_width_dim }
{ \l__flowfram_sdf_height_dim }
{ \l__flowfram_x_dim }
{ \l__flowfram_y_dim }
\dim_add:Nn \l__flowfram_sdf_height_dim
{
  #1 + #2
}
\setstaticcontents { \c@maxstatic }
{
  \ffruleddeclarations
  \ffvrule { #2 } { \ffcolumseprule }
  { \l__flowfram_sdf_height_dim }
}
\int_case:nn { #3 }
{
  { \c_flowfram_frame_type_flow_int }
  {
    \flowfram_set_clist_to_frame_clist:Nnnn
    \l__flowfram_pages_clist
    { flow } { pagelist } { #4 }
  }
  { \c_flowfram_frame_type_static_int }
  {
    \flowfram_set_clist_to_frame_clist:Nnnn
    \l__flowfram_pages_clist
    { static } { pagelist } { #4 }
  }
  { \c_flowfram_frame_type_dynamic_int }
  {
    \flowfram_set_clist_to_frame_clist:Nnnn
    \l__flowfram_pages_clist
    { dynamic } { pagelist } { #4 }
  }
}
\setstaticframe { \c@maxstatic }
{ pages = \l__flowfram_pages_clist }

```

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.)

```
\dim_add:Nn \l__flowfram_x_dim
{
  \l__flowfram_left_evenx_dim
  - \l__flowfram_left_x_dim
}
\dim_add:Nn \l__flowfram_y_dim
{
  \l__flowfram_left_eveny_dim
  - \l__flowfram_left_y_dim
}
\setstaticframe { \c@maxstatic }
{
  evenx=\l__flowfram_x_dim ,
  eveny=\l__flowfram_y_dim
}
}
```

`\ffvrule` `\ffvrule{<offset>}{<width>}{<height>}`

Draws the rule for `\insertvrule`

```
\newcommand*{\ffvrule}[3]{%
  \hfill
  \rule [ - #1 ] { #2 } { #3 }
  \hfill
  \mbox { }
}
```

`\@sinsertvrule` Version 2.0 removed `\@sinsertvrule`.

`\@@sinsertvrule` Version 2.0 removed `\@@sinsertvrule`.

`\inserthrule` Now for a horizontal rule. Syntax similar to `\insertvrule`. Syntax:

`\inserthrule* [<x-left>] [<x-right>] { <type-1> } { <id> } { <type-2> } { <id> }`

```
\NewDocumentCommand \inserthrule
{ s 0 { \c_zero_dim } 0 { \c_zero_dim } m m m m }
{
  \__flowfram_only_preamble:Nn \inserthrule
  {
    \__flowfram_get_frame_type:n { #4 }
    \__flowfram_get_frame_type:Nn
      \l__flowfram_type_ii_int
      { #6 }
    \bool_lazy_or:nnF
      {
        \int_compare_p:nNn
          { \l__flowfram_type_int } = { -\c_one_int }
      }
      {
        \int_compare_p:nNn

```



```

}
\dim_set:Nn \l__flowfram_sdf_height_dim
{
  \ffareay - \l__flowfram_y_dim
}
\dim_set:Nn \l__flowfram_sdf_width_dim
{
  \l__flowfram_left_x_dim
  + \l__flowfram_left_width_dim
}
\dim_set:Nn \l__flowfram_x_dim
{
  \ffareax + \ffareawidth
}
\dim_compare:nNnT
  { \l__flowfram_x_dim } > { \l__flowfram_sdf_width_dim }
{
  \dim_set:Nn \l__flowfram_sdf_width_dim { \l__flowfram_x_dim }
}
\dim_compare:nNnTF
  { \l__flowfram_left_x_dim } < { \ffareax }
{
  \dim_set:Nn \l__flowfram_x_dim { \l__flowfram_left_x_dim }
}
{
  \dim_set:Nn \l__flowfram_x_dim { \ffareax }
}
\dim_add:Nn \l__flowfram_sdf_width_dim
{
  - \l__flowfram_x_dim
}
\__flowfram_new_static:nnnnn
{ all }
{ \l__flowfram_sdf_width_dim }
{ \l__flowfram_sdf_height_dim }
{ \l__flowfram_x_dim }
{ \l__flowfram_y_dim }
\dim_add:Nn \l__flowfram_sdf_width_dim
{ #1 + #2 }
\setstaticcontents { \c@maxstatic }
{
  \ffruleddeclarations
  \ffhrule { #1 } { \l__flowfram_sdf_width_dim } { \ffcolumseprule }
}
\int_case:nn { #3 }
{
  { \c_flowfram_frame_type_flow_int }
  {
    \flowfram_set_clist_to_frame_clist:Nnnn
    \l__flowfram_pages_clist
  }
}

```

```

        { flow } { pagelist } { #4 }
    }
{ \c_flowfram_frame_type_static_int }
{
    \flowfram_set_clist_to_frame_clist:Nnnn
    \l__flowfram_pages_clist
    { static } { pagelist } { #4 }
}
{ \c_flowfram_frame_type_dynamic_int }
{
    \flowfram_set_clist_to_frame_clist:Nnnn
    \l__flowfram_pages_clist
    { dynamic } { pagelist } { #4 }
}
}
\setstaticframe { \c@maxstatic }
{ pages = \clist_use:N \l__flowfram_pages_clist }
\dim_add:Nn \l__flowfram_x_dim
{
    \l__flowfram_left_evenx_dim
    - \l__flowfram_left_x_dim
}
\dim_add:Nn \l__flowfram_y_dim
{
    \l__flowfram_left_eveny_dim
    - \l__flowfram_left_y_dim
}
\setstaticframe {\c@maxstatic}
{
    evenx = \l__flowfram_x_dim ,
    eveny = \l__flowfram_y_dim
}
}

\ffhrule \ffhrule{<offset>}{<width>}{<height>}
    Draws the rule for \inserthrule
\newcommand*{\ffhrule}[3]{%
    \hspace* { -#1 } \rule { #2 } { #3 }
}

```

\@sinserthrule Version 2.0 removed \@sinserthrule.

\@@sinserthrule Version 2.0 removed \@@sinserthrule.

1.13 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.

This is implemented by redefining `\@makechapterhead` and `\@makeschapterhead` so that they place their argument in the applicable dynamic frame. Unfortunately this causes a problem if a class doesn't use these commands. As from v2.0, an alternative approach is available but the original method is retained.

```
\NewDocumentCommand \dfchaphead { s m }
{
  \cs_if_exist:NTF \chapter
  {
    \IfBooleanTF { #1 }
    {
      \__flowfram_get_dynamic_id:n { #2 }
    }
    {
      \int_set:Nn \l__flowfram_id_int { #2 }
    }
  }
}
```

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{%
  \exp_not:N
  \__flowfram_set_dynamic_contents:nn
  { \int_use:N \l__flowfram_id_int }
  {
    \noexpand\DFchapterstyle { ##1 }
  }
}
```

Now the starred version:

```
\xdef\@makeschapterhead##1{%
  \exp_not:N \__flowfram_set_dynamic_contents:nn
  { \int_use:N \l__flowfram_id_int }
  {
    \noexpand \DFschapterstyle { ##1 }
  }
}
}
{
  \msg_error:nn { flowfram } { no-chapters }
}
```

```

    }
}

```

Define style for the chapter heading. These commands are only used with `\dfchaphead`.

`\DFchapterstyle`

```
\newcommand{\DFchapterstyle}[1]{#1}
```

`\DFschapterstyle`

```
\newcommand{\DFschapterstyle}[1]{#1}
```

```
\int_new:N \g__flowfram_chaphead_frame_id
```

`\ChapterInDynamic` An alternative to `\dfchaphead` that doesn't alter `\@makechapterhead` and `\@makeschapterhead`.

```

\NewDocumentCommand \ChapterInDynamic { s m }
{
  \cs_if_exist:NTF \chapter
  {
    \IfBooleanTF { #1 }
    {
      \__flowfram_get_dynamic_id:e { #2 }
      \int_gset_eq:NN
        \g__flowfram_chaphead_frame_id
        \l__flowfram_id_int
    }
    {
      \int_gset:Nn \g__flowfram_chaphead_frame_id { #2 }
    }
    \__flowfram_make_chapter_head:n { \g__flowfram_chaphead_frame_id }
  }
  {
    \msg_error:nn { flowfram } { no-chapters }
  }
}

```

With the alternative approach, `\chapter` command is placed in the dynamic frame content instead of redefining the internal commands to put their argument in the dynamic frame. There are a number of problems with this: the initial clear page (or clear double page) can't go in the `\parbox` used by the dynamic frame. If there are any sections that follow the chapter heading on the same page their numbering will be out of sync as the dynamic contents aren't typeset until the page is output. So this approach first typesets the chapter heading in a box (having locally disabled the problematic commands) and then places the box in the dynamic contents. This rather goes against the idea that dynamic frames don't typeset their content until the page is output, but it's the simplest way around it that avoids messing around with internals that may break certain

classes. Extra content can still be appended to the dynamic frame (which can't be done in a static frame).

Replicate report and book class `\@makechapterhead` and `\@makeschapterhead` to test if those definitions currently apply. This means they can be redefined to remove the hard-coded top vertical space and allow more flexibility. Other classes, such as memoir and the KOMA Script classes have their own hooks so don't worry about them.

```
\ExplSyntaxOff
\newcommand*\@flowfram@standard@makechapterhead[1]{%
\vspace *{50\p@ }%
{\parindent \z@ \raggedright \normalfont
\ifnum \c@secnumdepth >\m@ne
\if@mainmatter \huge \bfseries \@chapapp \space \thechapter
\par
\nobreak \vskip 20\p@
\fi
\fi
\interlinepenalty \@M
\Huge \bfseries #1\par
\nobreak
\vskip 40\p@
}}
\ExplSyntaxOn

\cs_if_exist:NTF \chapter
{
```

Redefine `\@makechapterhead` and `\@makeschapterhead` if they have the book or report definition to allow for greater flexibility. Only testing the definition of `\@makechapterhead`. This assumes that if that definition matches the standard classes then it's likely that `\@makeschapterhead` will also be the standard definition.

```
\cs_if_eq:NNT \@makechapterhead \@flowfram@standard@makechapterhead
{
```

`\ffchapterpreheadskip`

```
\newcommand \ffchapterpreheadskip
{
\vspace *{50\p@ }
}
```

`\ffchapterpostheadskip`

```
\newcommand \ffchapterpostheadskip
{
\vspace *{40\p@ }
}
```

`\ffchapterheadstyle`

```
\newcommand \ffchapterheadstyle
```

```

{
  \parindent \z@ \raggedright
}

\ffchapternamenumfont
\newcommand \ffchapternamenumfont [ 1 ] { { \huge #1 } }

\ffchapternamenum
\newcommand \ffchapternamenum [ 2 ]
{
  #1 \space #2
}

\ffchapterpostnamenum
\newcommand \ffchapterpostnamenum
{
  \par
  \nobreak
  \vspace *{20\p@ }
}

\ffchaptertitlefont
\newcommand \ffchaptertitlefont [ 1 ] { { \Huge #1 } }

\ffchapterdefaultfont
\newcommand{\ffchapterdefaultfont}{\normalfont \bfseries}

\@makechapterhead
\renewcommand*\@makechapterhead[1]
{
  \ffchapterpreheadskip
  {
    \ffchapterheadstyle
    \ffchapterdefaultfont
    \ifnum \c@secnumdepth >\m@ne
    \if@mainmatter
    \ffchapternamenumfont
    {
      \ffchapternamenum { \@chapapp } { \thechapter }
      \ffchapterpostnamenum
    }
    \fi
  \fi
  \interlinepenalty \@M
  \ffchaptertitlefont { #1 \par }
  \nobreak
  \ffchapterpostheadskip
}
}

```

`\@makeschapterhead`

```
\renewcommand*\@makeschapterhead[1]
{
  \ffchapterpreheadskip
  {
    \ffchapterheadstyle
    \ffchapterdefaultfont
    \interlinepenalty \@M
    \ffchaptertitlefont { #1 \par }
    \nobreak
    \ffchapterpostheadskip
  }
}
```

`\dfchapterclearpage`

```
\cs_if_exist:NTF \if@openright
{
  \newcommand \dfchapterclearpage
  {
    \legacy_if:nTF { @openright }
    {
      \cleardoublepage
    }
    {
      \clearpage
    }
  }
}
{
  \newcommand \dfchapterclearpage
  {
    \clearpage
  }
}
```

Box used to store the chapter heading:

```
\newsavebox \flowfram@dfchap@sbox
```

Hooks for use within the box:

```
\NewMirroredHookPair
{ flowfram / chaphead / before }
{ flowfram / chaphead / after }
\NewMirroredHookPair
{ flowfram / chaphead / box / before }
{ flowfram / chaphead / box / after }
```

If `\ffchapterpreheadskip` is defined, redefine to do nothing at the start of the box:

```

\cs_if_exist:NT \ffchapterpreheadskip
{
  \AddToHook
  { flowfram / chaphead / box / before }
  {
    \let \ffchapterpreheadskip \relax
  }
}

```

Gather toc lines so they can be expanded straight away:

```

\tl_new:N \g__flowfram_df_chapter_lines_tl
\cs_new:Nn \__flowfram_df_addcontentsline:nnn
{
  \tl_gput_right:Nn \g__flowfram_df_chapter_lines_tl
  {
    \addcontentsline { #1 } { #2 } { #3 }
  }
}
\cs_new:Nn \__flowfram_df_markright:n
{
  \tl_gput_right:Nn \g__flowfram_df_chapter_lines_tl
  {
    \markright { #1 }
  }
}
\cs_new:Nn \__flowfram_df_markboth:nn
{
  \tl_gput_right:Nn \g__flowfram_df_chapter_lines_tl
  {
    \markboth { #1 } { #2 }
  }
}

```

The box needs to have the same width as the dynamic frame.

```

\dim_new:N \l__flowfram_df_chapbox_head_width_dim

```

The argument is the actual chapter command (with its arguments):

```

\cs_new:Nn \__flowfram_df_chapter:n
{
  \dfchapterclearpage
  \tl_gclear:N \g__flowfram_df_chapter_lines_tl
  \flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_df_chapbox_head_width_dim
  { dynamic } { width } { \g__flowfram_chaphead_id_int }
  \UseHook { flowfram / chaphead / before }
  \sbox \flowfram@dfchap@sbox
  {

```

Set the frame's text colour:

```

\flowfram_set_tl_to_frame_tl:Nnnn
\l__flowfram_textcolor_tl

```



```

    { dynamic } { textcolor } { \g__flowfram_chaphead_id_int }
  \__flowfram_set_text_color:

```

Typeset the heading in a parbox:

```

\parbox
{ \l__flowfram_df_chapbox_head_width_dim }
{
  \let \clearpage \relax
  \let \cleardoublepage \relax
  \let \onecolumn \relax
  \@twocolumnfalse
  \let \addcontentsline \__flowfram_df_addcontentsline:nnn
  \let \markright \__flowfram_df_markright:n
  \let \markboth \__flowfram_df_markboth:nn
  \UseHook { flowfram / chaphead / box / before }
  #1
}

```

The scoping introduced by the `\parbox` interferes with KOMA's local redefinition of `\@currenttocentry`, so if this command is defined, ensure that its definition is still available outside this box.

```

  \tl_if_exist:NT \@currenttocentry
  {
    \tl_gset:Ne \@currenttocentry { \@currenttocentry }
  }
  \UseHook { flowfram / chaphead / box / after }
}
}
\__flowfram_set_dynamic_contents:nn
{ \g__flowfram_chaphead_id_int }
{
  \box_use_drop:N \flowfram@dfchap@sbox
  \@afterheading
}
\g__flowfram_df_chapter_lines_tl
\tl_gclear:N \g__flowfram_df_chapter_lines_tl
\UseHook { flowfram / chaphead / after }
}

```

Unstarred chapter with one optional argument.

```

\cs_new:Nn \__flowfram_df_chapter:nn
{
  \__flowfram_df_chapter:n
  {
    \__flowfram_org_chapter: [ #1 ] { #2 }
  }
}

```

Unstarred chapter with two optional arguments.

```

\cs_new:Nn \__flowfram_df_chapter:nnn
{
  \__flowfram_df_chapter:n

```

```

    {
      \__flowfram_org_chapter: [ #1 ] [ #2 ] { #3 }
    }
  }

```

Starred chapter.

```

\cs_new:Nn \__flowfram_df_chapter_star:nn
{
  \__flowfram_df_chapter:n
  {
    \flowfram_nonum_section:Nnn
    \__flowfram_org_chapter: { #1 } { #2 }
  }
}

```

Return true if the argument is `chapter` and a dynamic frame has been identified for chapter headings.

```

\prg_new_conditional:Nnn \__flowfram_if_df_chapter:n
{ T, F, TF }
{
  \int_if_zero:nTF { \g__flowfram_chaphead_id_int }
  { \prg_return_false: }
  {
    \tl_if_eq:nnTF { #1 } { chapter }
    { \prg_return_true: }
    { \prg_return_false: }
  }
}
}
{

```

Always returns false as there are no chapters.

```

\prg_new_conditional:Nnn \__flowfram_if_df_chapter:n
{ T, F, TF }
{
  \prg_return_false:
}
}

```

`\@dynamicchap` Version 2.0 replaced `\@dynamicchap`.

```

\cs_new:Nn \__flowfram_make_chapter_head:n
{
  \int_set:Nn \g__flowfram_chaphead_id_int { #1 }
}
\int_new:N \g__flowfram_chaphead_id_int

```

`\@sdynamicchap` Version 2.0 removed `\@sdynamicchap`.

`\NoChapterInDynamic` Revert back.

```

\NewDocumentCommand \NoChapterInDynamic { }
{

```

```

\int_zero:N \g__flowfram_chaphead_id_int
}

```

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?

1.14 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.

```

\cs_if_exist:NTF \chapter
{
  \newcommand*{\defaultthumbtabtype}{chapter}
}
{
  \newcommand*{\defaultthumbtabtype}{section}
}

```

`\@ttb@type` Section type to assign to thumbtabs. Version 2.0 renamed `\@ttb@type`

```

\tl_new:N \g__flowfram_thumbtab_type_tl
\tl_gset:Nn \g__flowfram_thumbtab_type_tl
{ \defaultthumbtabtype }

```

The table of contents isn't read until the start of the document, but the thumbtabs are created in the preamble, so the toc information can't be used as it will be too late by the time it's read. This means a separate file is required.

```

\cs_new:Nn \__flowfram_ttb_num:n
{
  \addtocontents {ttb}
  {
    \protect \thumbtab
    { \theFramePageCounter }
    { \arabic { \g__flowfram_thumbtab_type_tl } }
    { #1 }
    { \@currentHref }
  }
}

```

Provide a counter for unnumbered tabs.

```

\newcounter{thumbtabnonum}

```

```

\cs_new:Nn \__flowfram_ttb_nonum:n
{
  \bool_if:NT \g__flowfram_save_nonum_thumbtabs_bool
  {
    \refstepcounter { thumbtabnonum }
    \addtocontents { ttb }
    {
      \protect \thumbtab
      { \theFramePageCounter }
      { }
      { #1 }
      { thumbtabnonum . \thethumbtabnonum }
    }
  }
}

```

`\makethumbtabs` Make the thumbtabs. Read in information from .ttb file, and open it for output.

Syntax:

`\makethumbtabs[<y offset>】{<height>}[<sec type>]`.

```

\NewDocumentCommand \makethumbtabs
{ 0 { \c_zero_dim } m 0 { \defaultthumbtabtype } }
{
  \exp_args:Nnne
  \__flowfram_make_thumbtabs:nnn { #1 } { #2 } { #3 }
  \cs_set:Nn \__flowfram_make_thumbtabs:nnn
  {
    \msg_note:nn { flowfram } { ignoring-multiple-makethumbtabs }
  }
}

```

`@makethumbtabs` Now all arguments are known, first redefine the appropriate sectioning command, then input the ttb file, and create the thumbtabs. Version 2.0 replaced `\makethumbtabs`.

```

\cs_new:Nn \__flowfram_make_thumbtabs:nnn
{
  \cs_if_exist:cTF { #3 }
  {
    \tl_gset:Ne \g__flowfram_thumbtab_type_tl { #3 }
    \@starttoc { ttb }
    \__flowfram_create_thumbtabs:nn { #1 } { #2 }
    \cs_set:Nn \__flowfram_write_backmatter_ttb:
    {
      \addtocontents {ttb}
      {
        \protect \thumbtab@backmatter
        { \theFramePageCounter }
      }
    }
  }
  \cs_set:Nn \__flowfram_no_thumbtabs_warn:

```

```

    {
      \msg_warning:nn { flowfram } { no-thumbtabs-rerun }
    }
  }
  {
    \msg_error:nnn { flowfram } { section-unit-not-defined } { #3 }
  }
}

```

Warning to produce if no thumbtabs:

```

\cs_new:Nn \__flowfram_no_thumbtabs_warn:
{
  \msg_warning:nn { flowfram } { no-thumbtabs-missing-makethumbtabs }
}

```

Write the page number to the ttb file when `\backmatter` is used to get the end page range if not thumbtabs in backmatter.

```

\cs_new:Nn \__flowfram_write_backmatter_ttb:
{
}

```

`\@makethumbchapter` Version 2.0 removed `\@makethumbchapter`.

`\@makethumbpart` Version 2.0 removed `\@makethumbpart`.

`\@makethumbsection` Version 2.0 removed `\@makethumbsection`

`\thumbtab` The thumbtab file, `.ttb`, will have a series of `\thumbtab` commands, when this file is read in, just store the information for now. The table of contents needs to be read first to pick up any unnumbered sections that should be included. Syntax: `{<page>}{<unit num>}{<title>}{<target>}`

```

\newcommand{\thumbtab}[4]{%
  \stepcounter { maxthumbtabs }
  \__flowfram_message:nnnnn { info-found-thumbtab }
  { #1 } { #2 } { #3 } { #4 }
  \tl_new:c { g__thumbtab_ \romannumeral \c@maxthumbtabs _pages_tl }
  \int_new:c { g__thumbtab_ \romannumeral \c@maxthumbtabs _start_int }
  \int_gset:cn { g__thumbtab_ \romannumeral \c@maxthumbtabs _start_int }
  { #1 }
  \tl_new:c { g__thumbtab_ \romannumeral \c@maxthumbtabs _unitnum_tl }
  \tl_gset:cn
  { g__thumbtab_ \romannumeral \c@maxthumbtabs _unitnum_tl }
  { #2 }
  \tl_new:c { g__thumbtab_ \romannumeral \c@maxthumbtabs _title_tl }
  \tl_gset:cn { g__thumbtab_ \romannumeral \c@maxthumbtabs _title_tl }
  { #3 }
  \tl_new:c { g__thumbtab_ \romannumeral \c@maxthumbtabs _link_tl }
  \tl_gset:cn { g__thumbtab_ \romannumeral \c@maxthumbtabs _link_tl }
  { #4 }
  \tl_gclear:N \g__flowfram_ttb_endpage_tl
}

```

`\thumbtab@backmatter`

```
\tl_new:N \g__flowfram_ttb_endpage_tl
\newcommand{\thumbtab@backmatter}[1]{%
  \tl_gset:Ne \g__flowfram_ttb_endpage_tl { \int_eval:n { #1 - \c_one_int } }
}
```

Use the thumbtab information.

```
\cs_new:Nn \__flowfram_thumbtab_start:n
{
  \int_use:c { g__thumbtab_ \romannumeral #1 _start_int }
}
\cs_new:Nn \__flowfram_thumbtab_unitnum:n
{
  \tl_use:c { g__thumbtab_ \romannumeral #1 _unitnum_tl }
}
\cs_new:Nn \__flowfram_thumbtab_pages:n
{
  \tl_use:c { g__thumbtab_ \romannumeral #1 _pages_tl }
}
\cs_new:Nn \__flowfram_thumbtab_set_pages:nn
{
  \tl_gset:cn { g__thumbtab_ \romannumeral #1 _pages_tl }
  { #2 }
}
\cs_generate_variant:Nn \__flowfram_thumbtab_set_pages:nn { ne }
\cs_new:Nn \__flowfram_thumbtab_title:n
{
  \tl_use:c { g__thumbtab_ \romannumeral #1 _title_tl }
}
```

`\SetThumbTabTitle` The title for individual thumbtabs may be changed.

```
\NewDocumentCommand \SetThumbTabTitle { m m }
{
  \tl_if_exist:cTF { g__thumbtab_ \romannumeral #1 _title_tl }
  {
    \tl_gset:cn { g__thumbtab_ \romannumeral #1 _title_tl } { #2 }
  }
  {
    \msg_warning:nnn { flowfram } { cant-find-thumbtab } { #1 }
  }
}

\cs_new:Nn \__flowfram_thumbtab_link:n
{
  \tl_use:c { g__thumbtab_ \romannumeral #1 _link_tl }
}
```

`\@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 variable 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 should 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: `{(y offset)}{(height)}`. Version 2.0 replaced `\@dothumbtabs`.

```
\cs_new:Nn \__flowfram_create_thumbtabs:nn
{
  \dim_set:Nn \l__flowfram_y_dim
  {
    \typeblockheight - ( #2 ) - ( #1 )
  }
  \computerightedgeodd { \l__flowfram_x_dim }
  \dim_sub:Nn \l__flowfram_x_dim { \thumbtabwidth }
  \computeleftedgeeven { \l__flowfram_evenx_dim }
  \int_step_inline:nn { \c@maxthumbtabs }
  {
```

First set the data obtained from the ttb file:

```
\int_set:Nn \l__flowfram_range_start_int
{
  \__flowfram_thumbtab_start:n { ##1 }
}
\int_compare:nNnTF { ##1 } = { \c@maxthumbtabs }
{
  \tl_if_empty:NTF \g__flowfram_ttb_endpage_tl
  {
    \__flowfram_thumbtab_set_pages:ne { ##1 }
    {
      \int_use:N \l__flowfram_range_start_int ,
      > \int_use:N \l__flowfram_range_start_int
    }
  }
  {
    \__flowfram_thumbtab_set_pages:ne { ##1 }
    {
      \int_use:N \l__flowfram_range_start_int
      - \g__flowfram_ttb_endpage_tl
    }
  }
}
}
```

Last page of this thumbtab is likely to be one less than the start of the next thumbtab.

```
\int_set:Nn \l__flowfram_range_end_int
{
```

```

        \_flowfram_thumbtab_start:n
        { \int_eval:n { ##1 + \c_one_int } }
    }
\int_decr:N \l_flowfram_range_end_int
\int_compare:nNnTF
  { \l_flowfram_range_end_int }
  >
  { \l_flowfram_range_start_int }
  {
    \_flowfram_thumbtab_set_pages:ne { ##1 }
    {
      \int_use:N \l_flowfram_range_start_int
      - \int_use:N \l_flowfram_range_end_int
    }
  }
  {
    \_flowfram_thumbtab_set_pages:ne { ##1 }
    {
      \int_use:N \l_flowfram_range_start_int
    }
  }
}
}

```

Now create the dynamic frames associated with this thumbtab:

```

\_flowfram_create_thumbtab:nn { ##1 } { #2 }
}

```

Now that all thumbtabs have been defined, set the page lists:

```

\bool_if:NTF \g_flowfram_has_bespoke_thumbtabs_bool
{
  \int_step_inline:nn { \c_maxthumbtabs }
  {
    \exp_args:Nnee
    \_flowfram_set_thumbtab_pagelist:nnn { ##1 }
    {
      \seq_item:Nn \g_flowfram_thumbtab_seq { ##1 }
    }
    {
      \seq_item:Nn \g_flowfram_even_thumbtab_seq { ##1 }
    }
  }
  \flowfram_frame_set_bool_true:nnn
  { dynamic } { hide }
  {
    \seq_item:Nn \g_flowfram_thumbtab_seq { ##1 }
  }
  \flowfram_frame_set_bool_true:nnn
  { dynamic } { hide }
  {
    \seq_item:Nn \g_flowfram_even_thumbtab_seq { ##1 }
  }
}
}

```



```

}
{
  \seq_map_indexed_inline:Nn \g__flowfram_thumbtab_seq
  {
    \clist_set:Ne \l__flowfram_pages_clist
    {
      \__flowfram_thumbtab_pages:n
      { ##1 }
    }
    \flowfram_frame_set_clist:nnnV
    { dynamic } { pagelist } { ##2 }
    \l__flowfram_pages_clist
    \flowfram_frame_set_bool_true:nnn
    { dynamic } { hide } { ##2 }
  }
}
}

\cs_new:Nn \__flowfram_set_thumbtab_pagelist:nnn
{
  \clist_set:Ne \l__flowfram_pages_clist
  {
    \__flowfram_thumbtab_pages:n { #1 }
  }
  \__flowfram_separate_odd_even:
  \__flowfram_set_thumbtab_pages:nn { #2 } { #3 }
}

```

`\thumbtabwidth` Default thumbtab width.

```

\newlength{\thumbtabwidth}
\setlength{\thumbtabwidth}{1cm}

```

`\thumbtabnumtitle` Used to format number and title if both shown.

```

\NewDocumentCommand \thumbtabnumtitle { m m }
{
  #1 \c_space_tl #2
}

```

`\thumbtabformat` Individual thumbtab format. Syntax: `\thumbtabformat{<text>}{<height>}`

```

\newcommand{\thumbtabformat}[2]{%
  \__flowfram_thumbtab_fmt:nn { #1 } { #2 }
}

```

`\@flf@subsp` Version 2.0 removed `\@flf@subsp`.

`\@ttb@stack` Stack letters vertically. Version 2.0 replaced `\@ttb@stack`.

```

\cs_new:Nn \flowfram_thumbtab_stack:n
{
  \tl_clear:N \l__flowfram_tmpa_tl
  \exp_args:Nc \text_map_inline:nn { \text_purify:n { #1 } }
}

```

```

{
  \tl_if_empty:NF \l__flowfram_tmpa_tl
  {
    \tl_put_right:Nn \l__flowfram_tmpa_tl { \ }
  }
  \tl_put_right:Nn \l__flowfram_tmpa_tl { ##1 }
}
\tl_put_left:Nn \l__flowfram_tmpa_tl { \begin { tabular } { 1 } }
\tl_put_right:Nn \l__flowfram_tmpa_tl { \end { tabular } }
\l__flowfram_tmpa_tl
}

```

\@ttb@stack Version 2.0 removed \@ttb@stack.

\@greyscale Count register to compute the grey scale. Version 2.0 replaced \@greyscale

```
\fp_new:N \l__flowfram_greyscale_fp
```

Keep track of IDNs of thumbtabs.

```
\seq_new:N \g__flowfram_thumbtab_seq
\seq_new:N \g__flowfram_thumbtab_index_seq
```

Version 2.0 allows bespoke thumbtabs. The dynamic frames should be labelled `thumbtab1`, `thumbtab2`, etc. With corresponding dynamic frames for the thumbtab index labelled `thumbtabindex1`, `thumbtabindex2`, etc. The index thumbtabs should have the same height as the thumbtabs. Allow for bespoke even thumbtabs, which should be labelled `eventhumbtab1`, `eventhumbtab2`, etc, and `eventhumbtabindex1`, `eventhumbtabindex2`, etc.

```
\seq_new:N \g__flowfram_even_thumbtab_seq
\seq_new:N \g__flowfram_even_thumbtab_index_seq
\bool_new:N \g__flowfram_has_bespoke_thumbtabs_bool
\bool_gset_false:N \g__flowfram_has_bespoke_thumbtabs_bool
```

\@dothumbtabs 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. If the frame is already defined (for example via `flowframtk`) then omit defining it but set the contents as appropriate. Syntax: `{<index>}{<height>}`

```
\cs_new:Nn \__flowfram_create_thumbtab:nn
{
  \fp_set:Nn \l__flowfram_greyscale_fp
  {
    0.01 * ( ( 60 * ( #1 ) / \c@maxthumbtabs ) + 25 )
  }
}
```

Create dynamic frame for thumbtab, unless it already exists.

```
\__flowfram_if_frame_label_exists:nnNTF
{ dynamic }
```

```

    { thumbtab \int_eval:n { #1 } }
    \l__flowfram_id_int
  }

```

Bespoke thumbtab. Change the offset in case we run out and need to create extra.

```

    \bool_set_true:N \g__flowfram_has_bespoke_thumbtabs_bool
    \flowfram_set_dim_to_frame_dim:Nnnn
      \l__flowfram_y_dim
      { dynamic } { posy } { \l__flowfram_id_int }
  }
  {
    \__flowfram_new_dynamic:nnnnnn
      { none } { \thumbtabwidth } { #2 }
      { \l__flowfram_x_dim } { \l__flowfram_y_dim }
      { thumbtab \int_eval:n { #1 } }
      \int_set_eq:NN \l__flowfram_id_int \c@maxdynamic

```

Set the position for even pages.

```

    \flowfram_frame_set_dim:nnnn { dynamic } { evenx }
      { \l__flowfram_id_int }
      { \l__flowfram_evenx_dim }

```

Set the background colour.

```

    \flowfram_frame_set_tl:nnne
      { dynamic } { backcolor } { \l__flowfram_id_int }
      { [gray] { \fp_to_decimal:N \l__flowfram_greyscale_fp } }
  }
  \__flowfram_message:neee { info-thumbtab }
  { \int_eval:n { #1 } }
  { thumbtab \int_eval:n { #1 } }
  { \int_use:N \l__flowfram_id_int }
  \seq_put_right:NV \g__flowfram_thumbtab_seq
  \l__flowfram_id_int

```

Set the contents of the thumbtab dynamic frame. The title isn't expanded at this point to allow individual thumbtab titles to be changed with `\SetThumbTabTitle`.

```

  \__flowfram_set_thumbtab_content:NVneen
  \thumbtabregularformat
  \l__flowfram_id_int
  { \__flowfram_thumbtab_link:n { #1 } }
  { \__flowfram_thumbtab_unitnum:n { #1 } }
  { \exp_not:N \__flowfram_thumbtab_title:n { #1 } }
  { #2 }

```

Has an even frame been defined?

```

  \__flowfram_if_frame_label_exists:nnNTF
  { dynamic }
  { eventthumbtab \int_eval:n { #1 } }
  \l__flowfram_id_ii_int
  {
    \bool_set_true:N \g__flowfram_has_bespoke_thumbtabs_bool

```

```

\seq_put_right:NV \g__flowfram_even_thumbtab_seq
\l__flowfram_id_ii_int

```

Set the frame content.

```

\__flowfram_set_thumbtab_content:NWneen
\thumbtabregularformat
\l__flowfram_id_ii_int
{ \__flowfram_thumbtab_link:n { #1 } }
{ \__flowfram_thumbtab_unitnum:n { #1 } }
{ \exp_not:N \__flowfram_thumbtab_title:n { #1 } }
{ #2 }
}
{
\seq_put_right:NV \g__flowfram_even_thumbtab_seq
\l__flowfram_id_int
}

```

Create dynamic frame for thumbtab index, unless it already exists.

```

\__flowfram_if_frame_label_exists:nnNTF
{ dynamic }
{ thumbtabindex \int_eval:n { #1 } }
\l__flowfram_id_int
{
\bool_set_true:N \g__flowfram_has_bespoke_thumbtabs_bool
}
{
\__flowfram_new_dynamic:nnnnnn
{ none } { \thumbtabwidth } { #2 }
{ \l__flowfram_x_dim } { \l__flowfram_y_dim }
{ thumbtabindex \int_eval:n { #1 } }
\int_set_eq:NN \l__flowfram_id_int \c@maxdynamic
}

```

Set the position for even pages.

```

\flowfram_frame_set_dim:nnnn { dynamic } { evenx }
{ \l__flowfram_id_int }
{ \l__flowfram_evenx_dim }

```

Set the background colour.

```

\flowfram_frame_set_tl:nnne
{ dynamic } { backcolor } { \l__flowfram_id_int }
{ [gray] { \fp_to_decimal:N \l__flowfram_greyscale_fp } }
}
\__flowfram_message:nnee {info-thumbtab-index}
{ \int_eval:n { #1 } }
{ thumbtabindex \int_eval:n { #1 } }
{ \int_use:N \l__flowfram_id_int }
\seq_put_right:NV \g__flowfram_thumbtab_index_seq
\l__flowfram_id_int

```

Set the contents of the dynamic frame.

```

\__flowfram_set_thumbtab_content:NVeeen
\thumbtabindexformat

```

```

\l__flowfram_id_int
{ \__flowfram_thumbtab_link:n { #1 } }
{ \__flowfram_thumbtab_unitnum:n { #1 } }
{ \exp_not:N \__flowfram_thumbtab_title:n { #1 } }
{ #2 }

```

Has an even index frame been defined?

```

\__flowfram_if_frame_label_exists:nnNTF
{ dynamic }
{ eventthumbtabindex \int_eval:n { #1 } }
\l__flowfram_id_ii_int
{
\bool_set_true:N \g__flowfram_has_bespoke_thumbtabs_bool
\seq_put_right:NV \g__flowfram_even_thumbtab_index_seq
\l__flowfram_id_ii_int

```

Set the frame content.

```

\__flowfram_set_thumbtab_content:NVneen
\thumbtabindexformat
\l__flowfram_id_ii_int
{ \__flowfram_thumbtab_link:n { #1 } }
{ \__flowfram_thumbtab_unitnum:n { #1 } }
{ \exp_not:N \__flowfram_thumbtab_title:n { #1 } }
{ #2 }
}
{
\seq_put_right:NV \g__flowfram_even_thumbtab_index_seq
\l__flowfram_id_int
}
\dim_sub:Nn \l__flowfram_y_dim { #2 }
}

```

Set the frame content taking the package options into account.

```

\cs_new:Nn \__flowfram_set_thumbtab_content:Nnnnnn
{
\tl_clear:N \l__flowfram_contents_tl
\tl_if_empty:NTF { #4 }
{

```

No number (starred sectioning command) so just use title.

```

\tl_set:Ne \l__flowfram_contents_tl
{
\exp_not:N #1
{ #3 }
{ \exp_not:n { #5 } }
{ \dim_eval:n { #6 } }
}
}
{
\bool_if:NTF \l__flowfram_ttb_show_number_bool
{

```

```

\bool_if:NTF \l__flowfram_ttb_show_title_bool
{
  \tl_set:Ne \l__flowfram_contents_tl
  {
    \exp_not:N #1
    { #3 }
    {
      \exp_not:N \thumbtabnumtitle
      { #4 } { \exp_not:n { #5 } }
    }
    { \dim_eval:n { #6 } }
  }
}
{
  \tl_set:Ne \l__flowfram_contents_tl
  {
    \exp_not:N #1
    { #3 }
    { #4 } { \dim_eval:n { #6 } }
  }
}
}
{
  \tl_set:Ne \l__flowfram_contents_tl
  {
    \exp_not:N #1
    { #3 }
    { \exp_not:n { #5 } }
    { \dim_eval:n { #6 } }
  }
}
}
\exp_args:NnV \__flowfram_set_dynamic_contents:nn
{ #2 }
\l__flowfram_contents_tl
}
\cs_generate_variant:Nn \__flowfram_set_thumbtab_content:Nnnnnn
{ NVneen , NVeeen }

```

`\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.)

```

\NewDocumentCommand \enablethumbtabs { }
{

```

Ensure that `\FlowFramSectionUnit` writes to the ttb file:

```

\cs_set_eq:NN
  \__flowfram_add_thumbtab:n
  \__flowfram_actual_add_thumbtab:n
\int_if_zero:nTF

```

```

{ \seq_count:N \g__flowfram_thumbtab_seq }
{
  \__flowfram_no_thumbtabs_warn:
}
{
  \seq_map_inline:Nn \g__flowfram_thumbtab_seq
  {
    \flowfram_frame_set_bool_false:nnn
    { dynamic } { hide } { ##1 }
  }
  \bool_if:NT \g__flowfram_has_bespoke_thumbtabs_bool
  {
    \seq_map_inline:Nn \g__flowfram_even_thumbtab_seq
    {
      \flowfram_frame_set_bool_false:nnn
      { dynamic } { hide } { ##1 }
    }
  }
}
}
}

```

`\disablethumbtabs` Disable all thumbtabs.

```

\NewDocumentCommand \disablethumbtabs { }
{
  Prevent \FlowFramSectionUnit from writing to the ttb file:
  \cs_set_eq:NN
  \__flowfram_add_thumbtab:n
  \use_none:n
  \seq_map_inline:Nn \g__flowfram_thumbtab_seq
  {
    \flowfram_frame_set_bool_true:nnn
    { dynamic } { hide } { ##1 }
  }
  \seq_map_inline:Nn \g__flowfram_thumbtab_index_seq
  {
    \flowfram_frame_set_clist:nnnn
    { dynamic } { pagelist } { ##1 } { none }
  }
  \bool_if:NT \g__flowfram_has_bespoke_thumbtabs_bool
  {
    \seq_map_inline:Nn \g__flowfram_even_thumbtab_seq
    {
      \flowfram_frame_set_clist:nnnn
      { dynamic } { hide } { ##1 } \c_true_bool
    }
    \seq_map_inline:Nn \g__flowfram_even_thumbtab_index_seq
    {
      \flowfram_frame_set_clist:nnnn
      { dynamic } { pagelist } { ##1 } { none }
    }
  }
}

```

```

    }
  }
}

```

`\thumbtabindex` Show thumbtab indexes on current page (if no optional argument) or for the given page list. Version 2.0 added optional argument.

```

\NewDocumentCommand \thumbtabindex { o }
{
  \IfValueTF { #1 }
  {

```

```

    \clist_set:Nc \l__flowfram_pages_clist { #1 }

```

If bespoke thumbtabs, need to separate the odd and even pages.

```

    \bool_if:NT \g__flowfram_has_bespoke_thumbtabs_bool
    {
      \__flowfram_separate_odd_even:
    }
  }
  {
    \clist_set:Nc \l__flowfram_pages_clist
    {
      \int_eval:n { \g__flowfram_pagecounter_tl }
    }
    \bool_if:NT \g__flowfram_has_bespoke_thumbtabs_bool
    {
      \int_if_odd:nTF { \g__flowfram_pagecounter_tl }
      {
        \clist_set_eq:NN \l__flowfram_odd_pages_clist
          \l__flowfram_pages_clist
        \clist_set:Nn \l__flowfram_even_pages_clist { none }
      }
      {
        \clist_set_eq:NN \l__flowfram_even_pages_clist
          \l__flowfram_pages_clist
        \clist_set:Nn \l__flowfram_odd_pages_clist { none }
      }
    }
  }
}
\bool_if:NTF \g__flowfram_has_bespoke_thumbtabs_bool
{
  \seq_map_pairwise_function:NNN
  \g__flowfram_thumbtab_index_seq
  \g__flowfram_even_thumbtab_index_seq
  \__flowfram_set_thumbtab_pages:nn
}
{
  \seq_map_inline:Nn \g__flowfram_thumbtab_index_seq
  {
    \flowfram_frame_set_clist:nnnV
    { dynamic } { pagelist } { ##1 }
  }
}

```



```

        \l__flowfram_pages_clist
    }
}
\cs_new:Nn \__flowfram_separate_odd_even:
{
  \clist_if_empty:NTF \l__flowfram_pages_clist
  {
    \clist_set:Nn \l__flowfram_pages_clist { none }
    \clist_set:Nn \l__flowfram_even_pages_clist { none }
    \clist_set:Nn \l__flowfram_odd_pages_clist { none}
  }
  {
    \exp_args:Ne \__flowfram_get_range:n
      { \clist_item:Nn \l__flowfram_pages_clist { \c_one_int } }
    \clist_clear:N \l__flowfram_even_pages_clist
    \clist_clear:N \l__flowfram_odd_pages_clist
    \int_compare:nNnTF
      { \l__flowfram_range_end_int } < { \c_flowfram_max_page_int }
      {
        \int_step_inline:nnn
          { \l__flowfram_range_start_int } { \l__flowfram_range_end_int }
          {
            \int_if_odd:nTF { ##1 }
            {
              \clist_put_right:Nn
                \l__flowfram_odd_pages_clist { ##1 }
            }
            {
              \clist_put_right:Nn
                \l__flowfram_even_pages_clist { ##1 }
            }
          }
        \clist_if_empty:NT \l__flowfram_odd_pages_clist
        {
          \clist_set:Nn \l__flowfram_odd_pages_clist { none }
        }
        \clist_if_empty:NT \l__flowfram_even_pages_clist
        {
          \clist_set:Nn \l__flowfram_even_pages_clist { none }
        }
      }
  }
  {
    \clist_set:Nn \l__flowfram_even_pages_clist { even }
    \clist_set:Nn \l__flowfram_odd_pages_clist { odd }
  }
}
\cs_new:Nn \__flowfram_set_thumbtab_pages:nn

```

```

{
  \int_compare:nNnTF { #1 } = { #2 }
  {
    \flowfram_frame_set_clist:nnnV
      { dynamic } { pagelist } { #1 }
    \l__flowfram_pages_clist
  }
  {
    \flowfram_frame_set_clist:nnnV
      { dynamic } { pagelist } { #1 }
    \l__flowfram_odd_pages_clist
    \flowfram_frame_set_clist:nnnV
      { dynamic } { pagelist } { #2 }
    \l__flowfram_even_pages_clist
  }
}
}

```

`\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.

```

\NewDocumentCommand \setthumbtab { m m }
{
  \tl_if_eq:nnTF { #1 } { all }
  {
    \int_step_inline:nn { \c@maxthumbtabs }
    {
      \__flowfram_set_thumbtab:nn { ##1 } { #2 }
    }
  }
  {
    \clist_map_inline:nn { #1 }
    {
      \int_compare:nNnTF { ##1 } < { \c_one_int }
      {
        \msg_error:nnn { flowfram }
          { invalid-thumbtab-index }
          { ##1 }
      }
      {
        \int_compare:nNnTF
          { ##1 } >
          { \seq_count:N \g__flowfram_thumbtab_seq }
          {
            \msg_warning:nnn {flowfram} {cant-find-thumbtab} { ##1 }
          }
      }
    }
  }
}

```

```

        {
            \__flowfram_set_thumbtab:nn { ##1 } { #2 }
        }
    }
}
}
}
}

```

\@setthumbtab Set individual thumbtab and its index tab. Version 2.0 replaced \@setthumbtab.

```

\cs_new:Nn \__flowfram_set_thumbtab:nn
{
Find the ID for the index thumbtab:
\int_set:Nn \l__flowfram_id_int
{
    \seq_item:Nn
        \g__flowfram_thumbtab_index_seq
        { #1 }
}
\__flowfram_set_dynamic_by_idn:nn
{
    \l__flowfram_id_int
}
{ #2 }
\bool_if:NT \g__flowfram_has_bespoke_thumbtabs_bool
{
If bespoke thumbtabs, find the ID for the even index thumbtab:
\int_set:Nn \l__flowfram_id_ii_int
{
    \seq_item:Nn
        \g__flowfram_even_thumbtab_index_seq
        { #1 }
}
\int_compare:nNnF
{
    \l__flowfram_id_int
}
=
{
    \l__flowfram_id_ii_int
}
{
    \__flowfram_set_dynamic_by_idn:nn
        {
            \l__flowfram_id_ii_int
        }
        { #2 }
}
}
}
}

```

Find the ID for the thumbtab:

```

\int_set:Nn \l__flowfram_id_int
{
  \seq_item:Nn
    \g__flowfram_thumbtab_seq
    { #1 }
}
\__flowfram_set_dynamic_by_idn:nn
{
  \l__flowfram_id_int
}
{ #2 }
\bool_if:NT \g__flowfram_has_bespoke_thumbtabs_bool
{

```

If bespoke thumbtabs, find the ID for the even thumbtab:

```

\int_set:Nn \l__flowfram_id_ii_int
{
  \seq_item:Nn
    \g__flowfram_even_thumbtab_seq
    { #1 }
}
\int_compare:nNnF
{
  \l__flowfram_id_int
}
=
{
  \l__flowfram_id_ii_int
}
{
  \__flowfram_set_dynamic_by_idn:nn
  {
    \l__flowfram_id_ii_int
  }
  { #2 }
}
}
}

```

`\setthumbtabindex` Only change settings for the thumbtab index. This can take a comma-separated number list.

```

\NewDocumentCommand \setthumbtabindex { m m }
{
  \tl_if_eq:nnTF { #1 } { all }
  {
    \int_step_inline:nn { \c@maxthumbtabs }
    {
      \__flowfram_set_thumbtab_index:nn { ##1 } { #2 }
    }
  }
}

```

```

}
{
  \clist_map_inline:nn { #1 }
  {
    \int_compare:nNnTF { ##1 } < { \c_one_int }
    {
      \msg_error:nnn { flowfram }
      { invalid-thumbtab-index }
      { ##1 }
    }
    {
      \int_compare:nNnTF
      { ##1 } >
      { \seq_count:N \g__flowfram_thumbtab_seq }
      {
        \msg_warning:nnn {flowfram} {cant-find-thumbtab} { ##1 }
      }
      {
        \__flowfram_set_thumbtab_index:nn { ##1 } { #2 }
      }
    }
  }
}
}
}
}

```

`\@setthumbtabindex` Change setting for individual thumbtab index entry. Version 2.0 replaced `\@setthumbtabindex`.

```

\cs_new:Nn \__flowfram_set_thumbtab_index:nn
{

```

Find the ID for the index thumbtab:

```

  \int_set:Nn \l__flowfram_id_int
  {
    \seq_item:Nn
    \g__flowfram_thumbtab_index_seq
    { #1 }
  }
  \__flowfram_set_dynamic_by_idn:nn
  {
    \l__flowfram_id_int
  }
  { #2 }
  \bool_if:NT \g__flowfram_has_bespoke_thumbtabs_bool
  {

```

If bespoke thumbtabs, find the ID for the even index thumbtab:

```

    \int_set:Nn \l__flowfram_id_ii_int
    {
      \seq_item:Nn
      \g__flowfram_even_thumbtab_index_seq

```

```

        { #1 }
    }
\int_compare:nNnF
{
    \l__flowfram_id_int
}
=
{
    \l__flowfram_id_ii_int
}
{
    \__flowfram_set_dynamic_by_idn:nn
    {
        \l__flowfram_id_ii_int
    }
    { #2 }
}
}
}

```

`\tocandhumbtabindex` Do both the table of contents and the thumbtab index. This is intended for single paged table of contents.

```

\NewDocumentCommand \tocandthumbtabindex { }
{
    \aligntoctrue
    \bool_set_true:N \l__flowfram_thumbtabs_in_toc_bool
    \tableofcontents
    \aligntofalse
}

```

1.15 Minitocs

`\@ttb@minitoc` Sectioning type for the minitoc, by default it is the same as the thumbtabs. Version 2.0 replaced `\@ttb@minitoc`.

```

\tl_new:N \g__flowfram_minitoc_type_tl

```

`\if@storetoc` Version 2.0 removed `\if@storetoc`.

`\@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. A boolean is used to determine whether to store the contents, or act as normal. `\@starttoc` will be set to the following:

```

\cs_new:Nn \__flowfram_starttoc:n
{
    \__flowfram_store_toc:n { #1 }
}

```

`\@ttb@storetoc` Store the contents of the file with the given extension. Version 2.0 replaced `\@ttb@storetoc`.

```
\cs_new:Nn \__flowfram_store_toc:n
{
  \begingroup
  \makeatletter
  \__flowfram_parse_toc:n { \jobname . #1 }
```

The rest is the same as for the kernel definition of `\@starttoc` but includes the extra check used by `rerunfilecheck`.

```
\if@filesw
  \RerunFileCheck { \jobname . #1 }
  {
    \@ifundefined{tf@#1}{}
    {
      \immediate\closeout\csname tf@#1 \endcsname
    }
  }
  { }
  \@ifundefined{tf@#1}
  {
    \expandafter\newwrite\csname tf@#1\endcsname
  }
  { }
  \immediate\openout\csname tf@#1\endcsname\jobname.#1\relax
\fi
\@nobreakfalse
\endgroup
}
```

`\ior_new:N \g__flowfram_toc_ior`

`\@ttb@minitoclevel` Version 2.0 replaced `\@ttb@minitoclevel`.

```
\int_new:N \g__flowfram_minitoc_level_int
```

`\@storefileconts` Store the contents of named file, if it exists. Version 2.0 replaced `\@storefileconts`.

```
\cs_new:Nn \__flowfram_parse_toc:n
{
  \ior_open:NnTF \g__flowfram_toc_ior { #1 }
  {
    \int_if_exist:cTF
    {
      c_flowfram_level_
      \g__flowfram_minitoc_type_tl
      _int
    }
    {
      \int_gset_eq:Nc \g__flowfram_minitoc_level_int
      {
```

```

        c_flowfram_level_
        \g__flowfram_minitoc_type_tl
        _int
    }
}
{
  \int_gset:Nn \g__flowfram_minitoc_level_int { 6 }
}
\int_gzero:N \c@maxtocunits
\int_gzero:N \c@maxminitoc
\int_gzero:N \g__flowfram_toc_level_int
\ior_map_inline:Nn \g__flowfram_toc_ior
{
  \__flowfram_add_to_toc_list:Nn
  \c@maxtocunits { ##1 }
}
\ior_close:N \g__flowfram_toc_ior
}
{
  \msg_info:nnn { flowfram } { info-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` Version 2.0 removed `\@@storefileconts`.

`\if@contsline` Version 2.0 removed `\if@contsline`.

`\@ttb@level` Version 2.0 removed `\@ttb@level`.

`\@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 deeper than `\g__flowfram_minitoc_type_tl`. The number of minitoc lists is given by `\c@maxminitoc`. Version 2.0 replaced `\@addtotoclist`

Keep track of current level:
`\int_new:N \g__flowfram_toc_level_int`

Token list that will contain the whole toc:

```
\tl_new:N \g__flowfram_table_of_contents_tl
```

Add a line from the toc to the applicable list. Syntax: $\langle int-var \rangle \{ \langle toc line \rangle \}$ The variable keeps track of the number of toc sub lists.

```
\cs_new:Nn \__flowfram_add_to_toc_list:Nn  
{
```

First add to the whole list:

```
\tl_gput_right:Nn \g__flowfram_table_of_contents_tl { #2 }
```

Now check if the line starts with `\contentsline`.

```
\tl_if_head_eq_meaning:nNT  
{ #2 } \contentsline  
{
```

Get the content of the first argument that follows `\contentsline`. First discard `\contentsline`.

```
\tl_set:Ne \l__flowfram_remainder_tl  
{  
  \tl_tail:n { #2 }  
}
```

Now get the first argument:

```
\tl_set:Ne \l__flowfram_contents_tl  
{  
  \tl_head:N \l__flowfram_remainder_tl  
}
```

If this section type is recognised, get the numeric identifier:

```
\int_if_exist:cTF  
{  
  c_flowfram_level_  
  \l__flowfram_contents_tl  
  _int  
}  
{  
  \int_gset_eq:Nc \g__flowfram_toc_level_int  
  {  
    c_flowfram_level_  
    \l__flowfram_contents_tl  
    _int  
  }  
}  
{
```

Not recognised, assume it comes after the known lowest level.

```
\int_gset:Nn \g__flowfram_toc_level_int  
{ \c_flowfram_level_subparagraph_int + \c_one_int }  
}
```

If this level is the designated thumbtab level, increment the counter that keeps track of the number of toc units.

```
\tl_if_eq:eeT
  { \g__flowfram_thumbtab_type_tl }
  { \l__flowfram_contents_tl }
  {
    \int_gincr:N #1
  }
}
```

If a token list is not yet available for this toc sub list, create it.

```
\tl_if_exist:cF
  { g__flowfram_toc_level_ \romannumeral #1 _tl }
  {
    \tl_new:c
      { g__flowfram_toc_level_ \romannumeral #1 _tl }
  }
}
```

Append the toc line to this toc sub list.

```
\tl_gput_right:cn
  { g__flowfram_toc_level_ \romannumeral #1 _tl }
  { #2 }
}
```

Now do minitoc stuff if applicable.

```
\bool_if:NT \g__flowfram_minitocs_enabled_bool
  {
```

If the current toc level matches the designated minitoc level, increment the counter that stores the maximum number of minitocs and create the associated token list.

```
\int_compare:nNnTF
  { \g__flowfram_toc_level_int }
  =
  { \g__flowfram_minitoc_level_int }
  {
    \int_gincr:N \c@maxminitoc
```

Create the associated token list but don't add this toc line to it.

```
\tl_clear_new:c { g__flowfram_minitoc_ \romannumeral \c@maxminitoc _tl }
}
{
  \int_compare:nNnT
    { \g__flowfram_toc_level_int }
    >
    { \g__flowfram_minitoc_level_int }
  {
```

If the current section level is deeper than the designated minitoc level, add this toc line to the list.

```
\tl_if_exist:cF { g__flowfram_minitoc_ \romannumeral \c@maxminitoc _tl }
  {
```

```

        \tl_new:c { g__flowfram_minitoc_ \romannumeral \c@maxminitoc _tl }
      }
    \tl_gput_right:cn
      { g__flowfram_minitoc_ \romannumeral \c@maxminitoc _tl }
      { #2 }
    }
  }
}

```

Is there already a way of determining the sectioning level from its name?

```

\@ttb@part@level Version 2.0 replaced \@ttb@part@level
  \int_const:Nn \c_flowfram_level_part_int { -1 }

```

```

\@ttb@chapter@level Version 2.0 replaced \@ttb@chapter@level
  \int_const:Nn \c_flowfram_level_chapter_int { 0 }

```

```

\@ttb@section@level Version 2.0 replaced \@ttb@section@level
  \int_const:Nn \c_flowfram_level_section_int { 1 }

```

```

\@ttb@subsection@level Version 2.0 replaced \@ttb@subsection@level
  \int_const:Nn \c_flowfram_level_subsection_int { 2 }

```

```

\@ttb@subsubsection@level Version 2.0 replaced \@ttb@subsubsection@level
  \int_const:Nn \c_flowfram_level_subsubsection_int { 3 }

```

```

\@ttb@paragraph@level Version 2.0 replaced \@ttb@paragraph@level
  \int_const:Nn \c_flowfram_level_paragraph_int { 4 }

```

```

\@ttb@subparagraph@level Version 2.0 replaced \@ttb@subparagraph@level
  \int_const:Nn \c_flowfram_level_subparagraph_int { 5 }

```

```

\@checkcontentsline Version 2.0 removed \@checkcontentsline.

```

```

\@gettype Version 2.0 removed \@gettype

```

`\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. This is the default code used for `\tableofcontents`:

```

\cs_new:Nn \__flowfram_tableofcontents:
{
  \cs_set_eq:NN \@starttoc \__flowfram_starttoc:n
  \flowfram_toc:
  \bool_lazy_and:nnTF

```

```

{
  \bool_not_p:n
  {
    \tl_if_empty_p:N \g__flowfram_table_of_contents_tl
  }
}
{
  \bool_if_p:N \l__flowfram_thumbtabs_in_toc_bool
}
{
  \bool_if:NTF \l__flowfram_thumbtabs_span_toc_bool
  {
    \thumbtabindex[all]
  }
  {
    \thumbtabindex
  }
  \legacy_if:nTF { aligntoc }
  {
    \__flowfram_print_aligned_toc:
  }
  {
    \__flowfram_print_toc:
  }
  \bool_if:NT \l__flowfram_thumbtabs_span_toc_bool
  {
    \afterpage
    {
      \thumbtabindex[none]
    }
  }
}
{
  \__flowfram_print_toc:
}
\cs_set_eq:NN \@starttoc \__flowfram_org_starttoc:
\int_gzero:N \c@minitoc
}

```

`\beforeminitocskip` Vertical space to add before minitoc.

```

\newlength\beforeminitocskip
\setlength{\beforeminitocskip}{Opt}

```

`\afterminitocskip` Vertical space to add after minitoc.

```

\newlength\afterminitocskip
\setlength{\afterminitocskip}{\baselineskip}

```

`\dominitoc` Do the minitoc for unit #1. (Version 2.0 removed check that minitocs have been enabled since this command won't be used if they haven't been enabled). The argument is the minitoc index.

```

\NewDocumentCommand \dominitoc { m }
{
  \flowfram_do_minitoc:n { #1 }
}

\@dominitoc This command is redefined by \appenddfminitoc. Version 2.0 renamed
\@dominitoc.
\cs_new:Nn \flowfram_do_minitoc:n
{
  \flowfram_use_minitoc:n { #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. Version 2.0 renamed \@@dominitoc.
\cs_new:Nn \flowfram_use_minitoc:n
{
  \bool_lazy_and:nnT
  {
    \tl_if_exist_p:c { g__flowfram_minitoc_ \romannumeral #1 _tl }
  }
  {
    \bool_not_p:n
    { \tl_if_empty_p:c { g__flowfram_minitoc_ \romannumeral #1 _tl } }
  }
  {
    \vskip \beforeminitocskip
    \group_begin:
    \minitocstyle
    {
      \tl_use:c { g__flowfram_minitoc_ \romannumeral #1 _tl }
    }
    \group_end:
    \vskip \afterminitocskip
  }
}

```

We don't want to repeatedly add the minitoc to the dynamic frame but `\ChapterInDynamic` automatically resets the contents which means the minitoc will need to be appended at the start of each chapter if they are both using the same frame. Also, if the clear setting is on then the minitoc will need to be appended again when needed.

```

\cs_new:Nn \flowfram_clear_do_minitoc:
{
  \int_compare:nNnF

```

```

    { \g__flowfram_minitoc_frame_id }
    =
    { \g__flowfram_chaphead_frame_id }
  {
    \flowfram_if_frame_bool:nnnF
      { dynamic } { clear } { \g__flowfram_minitoc_frame_id }
      {
        \cs_gset_eq:NN \flowfram_do_minitoc:n \use_none:n
      }
    }
  }
}
\int_new:N \g__flowfram_minitoc_frame_id

```

`\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 `\appendminitocdynamicframe` but decided it was too long, for I've opted instead for a slightly more cryptic name.

```

\NewDocumentCommand \appenddfminitoc { s m }
{
  \renewcommand{\beforeminitocskip}{\baselineskip}%
  \IfBooleanTF { #1 }
  {
    \__flowfram_get_dynamic_id:e { #2 }
    \int_gset_eq:NN
      \g__flowfram_minitoc_frame_id
      \l__flowfram_id_int
  }
  {
    \int_gset:Nn \g__flowfram_minitoc_frame_id { #2 }
  }
  \cs_set:Nn \flowfram_do_minitoc:n
  {
    \__flowfram_append_dynamic_contents_idn:nn
    { \g__flowfram_minitoc_frame_id }
    {
      \flowfram_use_minitoc:n { ##1 }
    }
    \flowfram_clear_do_minitoc:
  }
}

```

`\@sappendminitocdf` Version 2.0 removed `\@sappendminitocdf`

`\@appendminitocdf` Version 2.0 removed `\@appendminitocdf`

`\@printtoc` Do the table of contents, which has been gathered from the toc file. Version 2.0 replaced `\@printtoc`.

```

\cs_new:Nn \__flowfram_print_toc_units:
{

```

```

\int_step_inline:nn { \c@maxtocunits }
{
  \tl_use:c { g__flowfram_toc_level_ \romannumeral ##1 _tl }
}
}

```

Print all the content found in the toc.

```

\cs_new:Nn \__flowfram_print_toc:
{
  \g__flowfram_table_of_contents_tl
}

```

`\@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. Version 2.0 replaced `\@printalignedtoc`

```

\cs_new:Nn \__flowfram_print_aligned_toc:
{
  \tl_if_exist:NT \g__flowfram_toc_level__tl
  {
    \g__flowfram_toc_level__tl
    \par
    \noindent
    \hrulefill
  }
  \int_step_inline:nn { \c@maxtocunits }
  {
    \int_compare:nNnTF { ##1 } > { \c@maxthumbtabs }
    {
      \tl_use:c { g__flowfram_toc_level_ \romannumeral ##1 _tl }
    }
    {
      \__flowfram_get_dynamic_id:n
      { thumbtabindex ##1 }
      \flowfram_set_dim_to_frame_dim:Nnnn
      \l__flowfram_x_dim
      { dynamic } { width } { \l__flowfram_id_int }
      \flowfram_set_dim_to_frame_dim:Nnnn
      \l__flowfram_y_dim
      { dynamic } { height } { \l__flowfram_id_int }
      \vbox_to_ht:nn { \l__flowfram_y_dim }
      {
        \noindent
        \parbox { \linewidth }
        {
          \tl_use:c { g__flowfram_toc_level_ \romannumeral ##1 _tl }
        }
        \vfill
        \par
        \noindent
      }
    }
  }
}

```

```

        \hrulefill
      }
    }
  }
}

```

`\if@minitoc` Version 2.0 replaced `\if@minitoc`.

```

\bool_new:N \g__flowfram_minitocs_enabled_bool
\bool_gset_false:N \g__flowfram_minitocs_enabled_bool

```

Counter to keep track of the current minitoc.

```

\newcounter{minitoc}

```

`\enableminitoc` Make mini tocs appear at the start of given sectional unit.

```

\NewDocumentCommand \enableminitoc
  { 0 { \g__flowfram_thumbtab_type_tl } }
{
  \bool_gset_true:N \g__flowfram_minitocs_enabled_bool
  \int_gzero:N \c@minitoc
  \cs_if_exist:cTF { c_flowfram_level_ #1 _int }
  {
    \tl_gset:Ne \g__flowfram_minitoc_type_tl { #1 }
    \cs_set:Nn \__flowfram_minitoc_post_section_unit:n
    {
      \tl_if_eq:NnT \g__flowfram_minitoc_type_tl { ##1 }
      {
        \stepcounter { minitoc }
        \dominitoc { \c@minitoc }
      }
    }
  }
  {
    \msg_error:nne { flowfram } { section-unit-not-defined } { #1 }
  }
}

```

This command should only appear in the preamble. (This ensures that it is used before `\tableofcontents`.)

```

\@onlypreamble{\enableminitoc}

```

`\@makeminitocchapter` Version 2.0 removed `\@makeminitocchapter`.

`\@makeminitocpart` Version 2.0 removed `\@makeminitocpart`.

`\@makeminitocsection` Version 2.0 removed `\@makeminitocsection`.

```

\ExplSyntaxOff

```


2 Support for FlowframTk

This package (new in v2.0) is provided to support commands written by FlowframTk when exporting to a LaTeX class, package, document file or image file. This package doesn't automatically load `flowfram` as the exported file could simply just contain a `pgfpicture` environment, in which case `flowfram` isn't required.

A FlowframTk image consists of the actual image itself, which is defined by a set of objects, that may be paths, text (no line breaks), bitmaps, composite shapes (defined by an underlying shape with replicas or a reflection or text decoration), or groups (a subset of objects). Each object may have `flowfram` data assigned to it. Additionally, the image may include early preamble code, mid preamble code and late preamble code. These may be inserted when exporting to a class, package or complete document.

FlowframTk has two types of export: create an image using the `pgfpicture` environment, provided by the `pgf` package, or export the `flowfram` data assigned to objects, which will write the applicable `\newflowframe`, `\newstaticframe` or `\newdynamicframe`. Objects that don't have `flowfram` data assigned will be omitted from the export. The object's graphical representation (shape, text, bitmap, etc) may be used as the border for the frame, or may simply be used to identify the location and dimension of the frame.

```
\NeedsTeXFormat{LaTeX2e}
```

Declare package:

```
\ProvidesPackage{flowframtkutils}[2025/11/24 v2.0 (NLCT)]
```

Switch on L3 syntax.

```
\ExplSyntaxOn
```

Provide outline support if true:

```
\bool_new:N \l__flowframtk_outline_bool  
\bool_set_false:N \l__flowframtk_outline_bool
```

Provide text-path support if true:

```
\bool_new:N \l__flowframtk_textpath_bool  
\bool_set_false:N \l__flowframtk_textpath_bool
```

Provide package options:

```
\keys_define:nn { flowframtkutils }  
{  
  outline .bool_set:N = \l__flowframtk_outline_bool ,  
  outline .usage:n = { load },  
  textpath .bool_set:N = \l__flowframtk_textpath_bool ,  
  textpath .usage:n = { load },  
}
```

```
\ExplSyntaxOff
```

Process options:

```
\ProcessKeyOptions[flowframtkutils]
```

Load pgf and graphicx:

```
\RequirePackage{pgf}
\RequirePackage{graphicx}
\ExplSyntaxOn
```

The following is for use when FlowframTk exports to cls or sty. This allows all flowfram options to be passed in the created class or package option list. This means that FlowframTk doesn't have to keep track of all flowfram['s] options and, in the case of an exported class file, the remaining options can be passed to the underlying class. Some of these options are not applicable with FlowframTk but they are provided for completeness.

```
\clist_new:N \g__flowframtk_options_clist
```

`\FlowFramTkSetKey` Check if flowfram has already been loaded. Bear in mind that some options can only be specified when flowfram is loading. FlowframTk will write the require package line for flowfram after that for flowframtkutils.

```
\IfPackageLoadedTF { flowfram }
{
  \newcommand \FlowFramTkSetKey [ 1 ]
  {
    \keys_set:nn { flowfram } { #1 }
  }
}
{
  \newcommand \FlowFramTkSetKey [ 1 ]
  {
    \clist_gput_right:Nn \g__flowframtk_options_clist { #1 }
  }
}
```

`\FlowFramTkDeclareKeys` Either set the option (if flowfram has already been loaded) or add to the list of options to pass to flowfram.

```
\newcommand \FlowFramTkDeclareKeys [ 1 ]
{
  \keys_define:nn { #1 }
  {
    adjust-toc .code:n =
    {
      \FlowFramTkSetKey { adjust-toc = { ##1 } }
    } ,
    backmatter-sections .code:n =
    {
      \FlowFramTkSetKey { backmatter-sections = { ##1 } }
    } ,
    color .code:n =
    {
      \FlowFramTkSetKey { color = { ##1 } }
    } ,
  } ,
}
```

```

color .default:n = { true } ,
column-changes .code:n =
  {
    \FlowFramTkSetKey { column-changes = { ##1 } }
  } ,
draft .code:n =
  {
    \FlowFramTkSetKey { draft }
  } ,
dynamic-empty-page-style .code:n =
  {
    \FlowFramTkSetKey { dynamic-empty-page-style = { ##1 } }
  } ,
dynamic-header-case .code:n =
  {
    \FlowFramTkSetKey { dynamic-header-case = { ##1 } }
  } ,
dynamic-page-style .code:n =
  {
    \FlowFramTkSetKey { dynamic-page-style = { ##1 } }
  } ,
dynamic-page-style-header-font .code:n =
  {
    \FlowFramTkSetKey { dynamic-page-style-header-font = { ##1 } }
  } ,
dynamic-page-style-subheader-font .code:n =
  {
    \FlowFramTkSetKey { dynamic-page-style-subheader-font = { ##1 } }
  } ,
dynamic-subheader-case .code:n =
  {
    \FlowFramTkSetKey { dynamic-subheader-case = { ##1 } }
  } ,
final .code:n =
  {
    \FlowFramTkSetKey { final }
  } ,
LR .code:n =
  {
    \FlowFramTkSetKey { LR = ##1 }
  } ,
LR .default:n = { true } ,
matter-thumbtabs .code:n =
  {
    \FlowFramTkSetKey { matter-thumbtabs = { ##1 } }
  } ,
nocolor .code:n =
  {
    \FlowFramTkSetKey { nocolor }
  } ,

```

```

norotate .code:n =
{
  \FlowFramTkSetKey { norotate }
},
pages .code:n =
{
  \FlowFramTkSetKey { pages = ##1 }
},
prohibit-frame-rotation .code:n =
{
  \FlowFramTkSetKey { prohibit-frame-rotation = { ##1 } }
},
prohibit-frame-rotation .default:n = { true } ,
RL .code:n =
{
  \FlowFramTkSetKey { RL = ##1 }
},
RL .default:n = { true } ,
rotate .code:n =
{
  \FlowFramTkSetKey { rotate = { ##1 } }
},
rotate .default:n = { true } ,
thumbtabs .code:n =
{
  \FlowFramTkSetKey { thumbtabs = ##1 }
},
thumbtabs .default:n = { title } ,
thumbtab-links .code:n =
{
  \FlowFramTkSetKey { thumbtab-links = { ##1 } }
},
thumbtab-text .code:n =
{
  \FlowFramTkSetKey { thumbtab-text = { ##1 } }
},
toc-thumbtabs .code:n =
{
  \FlowFramTkSetKey { toc-thumbtabs = { ##1 } }
},
toc-thumbtabs .default:n = { true } ,
ttbnotitle .code:n =
{
  \FlowFramTkSetKey { ttbnotitle }
},
ttbnonum .code:n =
{
  \FlowFramTkSetKey { ttbnonum }
},
ttbnum .code:n =

```

```

    {
      \FlowFramTkSetKey { ttbnum }
    } ,
    ttbtitle .code:n =
    {
      \FlowFramTkSetKey { ttbtitle }
    } ,
    sections-extra-option .code:n =
    {
      \FlowFramTkSetKey { sections-extra-option = { ##1 } }
    } ,
    verbose .code:n =
    {
      \FlowFramTkSetKey { verbose = { ##1 } }
    } ,
    verbose .default:n = { true } ,
    unstarred-thumbtabs .code:n =
    {
      \FlowFramTkSetKey { unstarred-thumbtabs = { ##1 } }
    } ,
    unstarred-thumbtabs .default:n = { true } ,
  }
}

```

`\jdrimagebox` For encapsulated images when FlowframTk creates a L^AT_EX document file that simply contains the image in a `pgfpicture` environment. This command is just in case numerical rounding errors cause the image to be marginally larger than the typeblock, which can cause a spurious blank page.

```

\NewDocumentCommand \jdrimagebox { m }
{
  \noindent
  \vbox_to_ht:nn { \textheight } { \hbox_to_wd:nn { \textwidth } { #1 } }
}

```

Provide support for outline text, if applicable. This depends on the output format. First the code if LuaL^AT_EX:

```

\newcommand \@flowframtk@def@outlinelua
{

```

The `pdf-trans` file uses pdfL^AT_EX primitives so these need to be defined and the simplest way to do that is to load `luatex85`.

```

  \RequirePackage{luatex85}

```

The rest is the same as for pdfL^AT_EX:

```

  \@flowframtk@def@outlinepdf
}

```

The code if pdfL^AT_EX. This is much the same as the above, except `luatex85` isn't required.

```

\newcommand \@flowframtk@def@outlinepdf
{

```

pdf-trans defines `\mod` which is likely to conflict with other packages so save it so that it can be restored afterwards.

```
\cs_set_eq:NN \__flowframtk_mod: \mod
```

Now input pdf-trans:

```
\input{pdf-trans}
```

Make pdf-trans's `\mod` available under a different name if required:

```
\let \flowframtkpdftransmod \mod
```

Restore the original `\mod`:

```
\cs_set_eq:NN \mod \__flowframtk_mod:
```

Now define `\jdroutine` so that it uses the `\boxgs` command provided by pdf-trans:

```
\NewDocumentCommand \jdroutine { m m m }
{
  \setbox\@tempboxa\hbox{##3}
  \boxgs{##1}{-}\copy\@tempboxa
}
}
```

Otherwise fallback on using PSTricks:

```
\newcommand \@flowframtk@def@outlinepst
{
  \RequirePackage{pst-char}
  \NewDocumentCommand \jdroutine { m m m }
  {
    \begin{pspicture}(0,0)
    \pscharpath[##2]{##3}
    \end{pspicture}
  }
}

\msg_new:nnn { flowframtk }
{ no-outline }
{
  outline ~ support ~ not ~ enabled
}
```

`\@flowframtk@outline` This command will be set to the applicable above command according to the engine, but only if outline support was requested with the `outline` package option. This will be set later after `exp3` syntax has been switched off in case it interferes with the required file or package.

```
\newcommand \@flowframtk@outline
{
  \NewDocumentCommand \jdroutine { m m m }
  {
    \msg_warning:nn { flowframtk } { no-outline }
    ##3
  }
}
```

Provide outline support if requested:

```
\bool_if:NT \l__flowframtk_outline_bool
{
  \sys_if_engine luatex:TF
  {
    \let \@flowframtk@outline \@flowframtk@def@outlinelua
  }
  {
    \sys_if_engine pdftex:TF
    {
      \let \@flowframtk@outline \@flowframtk@def@outlinepdf
    }
    {
      \let \@flowframtk@outline \@flowframtk@def@outlinepst
    }
  }
}
```

Provide text-path support if requested. This is simply a matter of loading the decorations.text pgf library:

```
\bool_if:NTF \l__flowframtk_textpath_bool
{
  \newcommand \@flowframtk@provide@textpath
  {
    \usepgflibrary{decorations.text}
  }
}
{
  \newcommand \@flowframtk@provide@textpath { }
}

\ExplSyntaxOff
\@flowframtk@outline
\@flowframtk@provide@textpath
\ExplSyntaxOn
```

Scratch variable:

```
\tl_new:N \l__flowframtk_tmp_tl
```

Object markup is an optional setting in FlowframTk's export dialog (v0.8.8+). Each object in the image can either be encapsulated or bookended. Each object has a unique value of $\langle n \rangle$, starting with 0 for the entire image (which is an implicit group).

```
\flowframtkencapobject{\langle n \rangle}{\langle class name \rangle}{\langle description \rangle}
{\langle tag \rangle}{\langle pgf
point \rangle}{\langle width \rangle}{\langle height \rangle}{\langle object \rangle}
```

\flowframtkencapobject

Encapsulate the image object.

```
\newcommand\flowframtkencapobject[8]{#8}
```

Alternatively, bookend with start and matching end pairs. Each pair must have the same $\langle n \rangle$ that identifies the object they are marking.

```
\flowframtkstartobject{\langle n \rangle}{\langle class name \rangle}{\langle description \rangle}
{\langle tag \rangle}{\langle pgf point \rangle}{\langle width \rangle}{\langle height \rangle}
```

`\flowframtkstartobject`

Markup just before object $\langle n \rangle$ starts.

```
\newcommand\flowframtkstartobject[7]{}
```

`\flowframtkendobject` Markup when object $\langle n \rangle$ ends. Syntax as for `\flowframtkstartobject`.

```
\newcommand\flowframtkendobject[7]{}
```

Provide a definition of `\flowframtkencapobject` that can be used to uncover parts of the image in `beamer` or similar. First the command that does the uncovering. This may be redefined as applicable. Individual uncover:

```
\cs_new:Nn \flowframtkutils_uncover:nn
{
  \uncover < #1 > { #2 }
}
Uncover from:
\cs_new:Nn \flowframtkutils_uncover_from:nn
{
  \uncover < #1 - > { #2 }
}
\cs_new:Nn \__flowframtkutils_uncover:nn
{
  \flowframtkutils_uncover_from:nn { #1 } { #2 }
}
\cs_new:Nn \__flowframtkutils_current_encap:nn { #2 }
\int_new:N \g_flowframtkutils_overlay_int
\tl_new:N \l__flowframtkutils_overlay_tl
\bool_new:N \l__flowframtkutils_auto_overlay_bool
\bool_set_true:N \l__flowframtkutils_auto_overlay_bool
\tl_new:N \l__flowframtkutils_propval_tl
Tag to text mapping (for annotations):
\prop_new:N \l_flowframtkutils_tag_text_prop
Tag to overlay number:
\prop_new:N \l_flowframtkutils_tag_overlay_prop
Object type to overlay number:
\prop_new:N \l_flowframtkutils_object_type_overlay_prop
```


Note that groups will have nested objects with a single all-encompassing group. Unexpected results may occur if an overlay is applied to a group.

Annotations.

```
\bool_new:N \l_flowframkutils_annotate_bool
\bool_set_false:N \l_flowframkutils_annotate_bool
```

Include arrows:

```
\bool_new:N \l_flowframkutils_annotate_arrow_bool
\bool_set_true:N \l_flowframkutils_annotate_arrow_bool

\dim_new:N \l_flowframkutils_annotate_offsetx_dim
\dim_set:Nn \l_flowframkutils_annotate_offsetx_dim { 1cm }
\dim_new:N \l_flowframkutils_annotate_offsety_dim
\dim_set:Nn \l_flowframkutils_annotate_offsety_dim { 1cm }

\int_new:N \l_flowframkutils_annotate_location_int
\int_new:N \l_flowframkutils_current_annotate_location_int
\int_const:Nn \c_flowframkutils_annotate_location_auto_int { 0 }
\int_const:Nn \c_flowframkutils_annotate_location_north_int { 1 }
\int_const:Nn \c_flowframkutils_annotate_location_northeast_int { 2 }
\int_const:Nn \c_flowframkutils_annotate_location_east_int { 3 }
\int_const:Nn \c_flowframkutils_annotate_location_southeast_int { 4 }
\int_const:Nn \c_flowframkutils_annotate_location_south_int { 5 }
\int_const:Nn \c_flowframkutils_annotate_location_southwest_int { 6 }
\int_const:Nn \c_flowframkutils_annotate_location_west_int { 7 }
\int_const:Nn \c_flowframkutils_annotate_location_northwest_int { 8 }
```

```
\cs_new:Nn \_flowframkutils_annotate_uncover:nn
{
  \flowframkutils_uncover:nn { #1 } { #2 }
}
```

Syntax: $\langle tl\ var \rangle \{ \langle description \rangle \} \{ \langle tag \rangle \}$

```
\cs_new:Nn \_flowframkutils_get_annotation_text:Nnn
{
  \tl_if_empty:nF { #2 }
  {
    \tl_set:Nn #1 { #2 }
  }
}
```

```
\bool_new:N \l_flowframkutils_default_auto_bool
\bool_set_false:N \l_flowframkutils_default_auto_bool
\tl_new:N \l_flowframkutils_default_overlay_tl
```

Get default overlay if auto overlay setting not on. Syntax: $\langle tl\ var \rangle \{ \langle object\ index \rangle \} \{ \langle object \rangle \}$

```
\cs_new:Nn \flowframkutils_get_default_overlay:Nnn
{
  \bool_if:NT \l_flowframkutils_default_auto_bool
  {
    \tl_if_eq:nnF { #3 } { JDRGroup }
  }
}
```

```

    {
      \tl_if_empty:NTF \l_flowframtkutils_default_overlay_tl
      {
        \int_gincr:N \g_flowframtkutils_overlay_int
      }
      {
        \int_gset:Nn \g_flowframtkutils_overlay_int
        { \l_flowframtkutils_default_overlay_tl }
      }
      \tl_set:NV
      #1
      \g_flowframtkutils_overlay_int
    }
  }
}

\cs_new:Nn \flowframtkutils_get_default_annotation_overlay:Nnn
{
  \tl_set:Nn #1 { 1 }
}

\tl_new:N \l__flowframtkutils_annotation_tl

\keys_define:nn { flowframtkutils / overlay }
{
  uncover .choice: ,
  uncover / single .code:n =
  {
    \cs_set:Nn \__flowframtkutils_uncover:nn
    {
      \flowframtkutils_uncover:nn { ##1 } { ##2 }
    }
  } ,
  uncover / from .code:n =
  {
    \cs_set:Nn \__flowframtkutils_uncover:nn
    {
      \flowframtkutils_uncover_from:nn { ##1 } { ##2 }
    }
  } ,
  uncover / false .code:n =
  {
    \cs_set:Nn \__flowframtkutils_uncover:nn { ##2 }
  } ,
  annotate .choice: ,
  annotate / match .code:n =
  {
    \cs_new:Nn \__flowframtkutils_annotate_uncover:nn
    {
      \__flowframtkutils_uncover:nn { ##1 } { ##2 }
    }
    \bool_set_true:N \l_flowframtkutils_annotate_bool
  }
}

```

```

    } ,
  annotate / single .code:n =
  {
    \cs_new:Nn \__flowframtkutils_annotate_uncover:nn
    {
      \flowframtkutils_uncover:nn { ##1 } { ##2 }
    }
    \bool_set_true:N \l_flowframtkutils_annotate_bool
  } ,
  annotate / from .code:n =
  {
    \cs_new:Nn \__flowframtkutils_annotate_uncover:nn
    {
      \flowframtkutils_uncover_from:nn { ##1 } { ##2 }
    }
    \bool_set_true:N \l_flowframtkutils_annotate_bool
  } ,
  annotate / true .code:n =
  {
    \bool_set_true:N \l_flowframtkutils_annotate_bool
  } ,
  annotate / false .code:n =
  {
    \bool_set_false:N \l_flowframtkutils_annotate_bool
  } ,
  annotate .default:n = true ,
  annotate-position .choices:nn =
  {
    auto , north, northeast , east, southeast , south ,
    southwest, west , northwest
  }
  {
    \int_set:Nn
    \l_flowframtkutils_annotate_location_int
    { \l_keys_choice_int - \c_one_int }
  } ,
  annotate-arrow .code:n =
  {
    \keys_set:nn { flowframtkutils / overlay / arrow } { #1 }
  } ,
  annotate-arrow .default:n = show ,
  annotate-text .choice: ,
  annotate-text / hide .code:n =
  {
    \cs_set:Nn \__flowframtkutils_get_annotation_text:Nnn { }
  } ,
  annotate-text / description .code:n =
  {
    \cs_set:Nn \__flowframtkutils_get_annotation_text:Nnn
    {

```

```

        \tl_if_empty:nF { ##2 }
        {
            \tl_set:Nn ##1 { ##2 }
        }
    }
},
annotate-text / tag .code:n =
{
    \cs_set:Nn \__flowframtkutils_get_annotation_text:Nnn
    {
        \tl_if_empty:nF { ##3 }
        {
            \tl_set:Nn ##1 { ##3 }
        }
    }
},
annotate-text / description-or-tag .code:n =
{
    \cs_set:Nn \__flowframtkutils_get_annotation_text:Nnn
    {
        \tl_if_empty:nTF { ##2 }
        {
            \tl_if_empty:nF { ##3 }
            {
                \tl_set:Nn ##1 { ##3 }
            }
        }
        {
            \tl_set:Nn ##1 { ##2 }
        }
    }
},
annotate-text / tag-map .code:n =
{
    \cs_set:Nn \__flowframtkutils_get_annotation_text:Nnn
    {
        \tl_if_empty:nF { ##3 }
        {
            \prop_get:NnN \l_flowframtkutils_tag_text_prop
            { ##3 } ##1
        }
    }
},
tag-text .code:n =
{
    \prop_set_from_keyval:Nn \l_flowframtkutils_tag_text_prop { #1 }
    \prop_if_empty:NF \l_flowframtkutils_tag_text_prop
    {
        \cs_set:Nn \__flowframtkutils_get_annotation_text:Nnn
        {

```

```

        \tl_if_empty:nF { ##3 }
        {
            \prop_get:NnN \l_flowframtkutils_tag_text_prop
            { ##3 } ##1
        }
    }
} ,
tag-overlay .code:n =
{
    \prop_set_from_keyval:Nn \l_flowframtkutils_tag_overlay_prop { #1 }
    \prop_if_empty:NF \l_flowframtkutils_tag_overlay_prop
    {
        \bool_set_false:N \l__flowframtkutils_auto_overlay_bool
    }
} ,
object-overlay .code:n =
{
    \prop_clear:N \l_flowframtkutils_object_type_overlay_prop
    \keys_set:nn { flowframtkutils / overlay / objects } { #1 }
    \prop_if_empty:NF \l_flowframtkutils_object_type_overlay_prop
    {
        \bool_set_false:N \l__flowframtkutils_auto_overlay_bool
    }
} ,
auto-overlay .bool_set:N = \l__flowframtkutils_auto_overlay_bool ,
fallback-overlay .code:n =
{
    \keys_set:nn { flowframtkutils / overlay / fallback } { #1 }
} ,
fallback-overlay .default:n = { enable } ,
}
\keys_define:nn { flowframtkutils / overlay / fallback }
{
    enable .bool_set:N = \l_flowframtkutils_default_auto_bool ,
    fixed .tl_set:N = \l_flowframtkutils_default_overlay_tl ,
    increment .code:n =
    {
        \tl_clear:N \l_flowframtkutils_default_overlay_tl
    } ,
    increment .value_forbidden:n = true ,
}
\keys_define:nn { flowframtkutils / overlay / objects }
{
    group .code:n =
    {
        \prop_put:Nne
        \l_flowframtkutils_object_type_overlay_prop
        { JDRGroup } { #1 }
    }
}

```

```

    },
    path .code:n =
    {
        \prop_put:Nne
        \l_flowframtkutils_object_type_overlay_prop
        { JDRPath } { #1 }
    },
    text .code:n =
    {
        \prop_put:Nne
        \l_flowframtkutils_object_type_overlay_prop
        { JDRText } { #1 }
    },
    text-path .code:n =
    {
        \prop_put:Nne
        \l_flowframtkutils_object_type_overlay_prop
        { JDRTextPath } { #1 }
    },
    symmetric-path .code:n =
    {
        \prop_put:Nne
        \l_flowframtkutils_object_type_overlay_prop
        { JDRSymmetricPath } { #1 }
    },
    bitmap .code:n =
    {
        \prop_put:Nne
        \l_flowframtkutils_object_type_overlay_prop
        { JDRBitmap } { #1 }
    },
    scaled-pattern .code:n =
    {
        \prop_put:Nne
        \l_flowframtkutils_object_type_overlay_prop
        { JDRScaledPattern } { #1 }
    },
    spiral-pattern .code:n =
    {
        \prop_put:Nne
        \l_flowframtkutils_object_type_overlay_prop
        { JDRSpiralPattern } { #1 }
    },
    rotational-pattern .code:n =
    {
        \prop_put:Nne
        \l_flowframtkutils_object_type_overlay_prop
        { JDRRotationalPattern } { #1 }
    },
}

```

```

\keys_define:nn { flowframtkutils / overlay / arrow }
{
  show .bool_set:N = \l_flowframtkutils_annotate_arrow_bool ,
  hide .bool_set_inverse:N = \l_flowframtkutils_annotate_arrow_bool ,
  dx .dim_set:N = \l_flowframtkutils_annotate_offsetx_dim ,
  dy .dim_set:N = \l_flowframtkutils_annotate_offsety_dim ,
  color .code:n =
  {
    \tl_if_empty:nTF { #1 }
    {
      \tl_clear:N \l_flowframtkutils_annotation_stroke_color_tl
    }
    {
      \tl_set:Nn \l_flowframtkutils_annotation_stroke_color_tl
      { \pgfsetstrokecolor { #1 } }
    }
  } ,
  start .code:n =
  {
    \tl_if_empty:nTF { #1 }
    {
      \tl_clear:N \l_flowframtkutils_annotation_start_arrow_tl
    }
    {
      \tl_set:Nn \l_flowframtkutils_annotation_start_arrow_tl
      {
        \pgfsetarrowsstart { #1 }
      }
    }
  } ,
  end .code:n =
  {
    \tl_if_empty:nTF { #1 }
    {
      \tl_clear:N \l_flowframtkutils_annotation_end_arrow_tl
    }
    {
      \tl_set:Nn \l_flowframtkutils_annotation_end_arrow_tl
      {
        \pgfsetarrowsend { #1 }
      }
    }
  } ,
}

\dim_new:N \l__flowframtkutils_image_minx_dim
\dim_new:N \l__flowframtkutils_image_miny_dim
\dim_new:N \l__flowframtkutils_image_maxx_dim
\dim_new:N \l__flowframtkutils_image_maxy_dim
\dim_new:N \l__flowframtkutils_image_cx_dim

```

```

\dim_new:N \l__flowframkutils_image_cy_dim
\dim_new:N \l__flowframkutils_image_width_dim
\dim_new:N \l__flowframkutils_image_height_dim

```

Object boundary:

```

\dim_new:N \l__flowframkutils_object_minx_dim
\dim_new:N \l__flowframkutils_object_miny_dim
\dim_new:N \l__flowframkutils_object_maxx_dim
\dim_new:N \l__flowframkutils_object_maxy_dim
\dim_new:N \l__flowframkutils_object_cx_dim
\dim_new:N \l__flowframkutils_object_cy_dim
\dim_new:N \l__flowframkutils_object_width_dim
\dim_new:N \l__flowframkutils_object_height_dim

\dim_new:N \l__flowframkutils_x_dim
\dim_new:N \l__flowframkutils_y_dim

```

Keep track of current group:

```

\int_new:N \l_flowframkutils_group_level_int
\cs_new:Nn \flowframkutils_uncover_encap:nnnnnnnn
{

```

Default to no overlay:

```

\cs_set_eq:NN \__flowframkutils_current_encap:nn \use_ii:nn
\int_if_zero:nTF { #1 }
{

```

Start of the image (0, JDRGroup, no tag, image description).

```

\int_gzero:N \g_flowframkutils_overlay_int
\tl_clear:N \l__flowframkutils_overlay_tl

```

This will be incremented later. So the group level for the entire image will be 1. Actual top-level groups will be level 2, sub-groups level 3, etc.

```

\int_zero:N \l_flowframkutils_group_level_int

```

The bottom left position will be in the form $\pgfpoint\{x\}\{y\}$.

```

\pgfextractx \l__flowframkutils_image_minx_dim { #5 }
\pgfextracty \l__flowframkutils_image_miny_dim { #5 }

```

Width and height:

```

\dim_set:Nn \l__flowframkutils_image_width_dim { #6 }
\dim_set:Nn \l__flowframkutils_image_height_dim { #7 }

```

Calculate maximum.

```

\dim_set:Nn \l__flowframkutils_image_maxx_dim
{
  \l__flowframkutils_image_minx_dim
  + \l__flowframkutils_image_width_dim
}
\dim_set:Nn \l__flowframkutils_image_maxy_dim
{
  \l__flowframkutils_image_miny_dim
  + \l__flowframkutils_image_height_dim
}

```


Calculate centre point.

```
\dim_set:Nn \l__flowframtkutils_image_cx_dim
{
  \l__flowframtkutils_image_minx_dim
  + 0.5 \l__flowframtkutils_image_width_dim
}
\dim_set:Nn \l__flowframtkutils_image_cy_dim
{
  \l__flowframtkutils_image_miny_dim
  + 0.5 \l__flowframtkutils_image_height_dim
}
```

If annotate setting is on, set the current object bounds to the image bounds.

```
\bool_if:NT \l_flowframtkutils_annotate_bool
{
  \dim_set_eq:NN
    \l__flowframtkutils_object_minx_dim
    \l__flowframtkutils_image_minx_dim
  \dim_set_eq:NN
    \l__flowframtkutils_object_miny_dim
    \l__flowframtkutils_image_miny_dim
  \dim_set_eq:NN
    \l__flowframtkutils_object_maxx_dim
    \l__flowframtkutils_image_maxx_dim
  \dim_set_eq:NN
    \l__flowframtkutils_object_maxy_dim
    \l__flowframtkutils_image_maxy_dim
  \dim_set_eq:NN
    \l__flowframtkutils_object_cx_dim
    \l__flowframtkutils_image_cx_dim
  \dim_set_eq:NN
    \l__flowframtkutils_object_cy_dim
    \l__flowframtkutils_image_cy_dim
  \dim_set_eq:NN
    \l__flowframtkutils_object_width_dim
    \l__flowframtkutils_image_width_dim
  \dim_set_eq:NN
    \l__flowframtkutils_object_height_dim
    \l__flowframtkutils_image_height_dim
}
}
```

Obtain the overlay, if set.

```
\tl_set:Nn \l__flowframtkutils_propval_tl { \q_no_value }
\bool_if:NTF \l__flowframtkutils_auto_overlay_bool
{
```

Auto setting is on. Overlay is incremented but skip groups to prevent nesting.

```
\tl_if_eq:nnF { #2 } { JDRGroup }
{
```

```

\int_gincr:N \g_flowframkutils_overlay_int
\tl_set:NV
  \l_flowframkutils_propval_tl
  \g_flowframkutils_overlay_int
}
}
{

```

Tag setting takes precedence.

```

\tl_if_empty:nF { #4 }
{
  \prop_get:NnN \l_flowframkutils_tag_overlay_prop { #4 }
  \l_flowframkutils_propval_tl
}
\quark_if_no_value:NT
  \l_flowframkutils_propval_tl
{

```

If no tag setting, use the object setting. The second argument is the object type and will always be non-empty.

```

\prop_get:NnN \l_flowframkutils_object_type_overlay_prop { #2 }
\l_flowframkutils_propval_tl

```

If no object setting either, get the default:

```

\quark_if_no_value:NT
  \l_flowframkutils_propval_tl
{
  \flowframkutils_get_default_overlay:Nnn
  \l_flowframkutils_propval_tl
  { #1 } { #2 }
}
}
}

```

Set the overlay and uncover functions as appropriate.

```

\quark_if_no_value:NTF
  \l_flowframkutils_propval_tl
{
  \tl_clear:N \l_flowframkutils_overlay_tl
}
{
  \tl_if_empty:NF \l_flowframkutils_propval_tl
  {
    \cs_set_eq:NN
      \__flowframkutils_current_encap:nn
      \__flowframkutils_uncover:nn
  }
  \tl_set_eq:NN
    \l_flowframkutils_overlay_tl
    \l_flowframkutils_propval_tl
}
}

```

```

\bool_if:NT \l_flowframkutils_annotate_bool
{

```

Annotation on. Store the current object's bounding box dimensions.

```

\pgfextractx \l__flowframkutils_object_minx_dim { #5 }
\pgfextracty \l__flowframkutils_object_miny_dim { #5 }
\dim_set:Nn \l__flowframkutils_object_width_dim { #6 }
\dim_set:Nn \l__flowframkutils_object_height_dim { #7 }

```

Calculate maximum point.

```

\dim_set:Nn \l__flowframkutils_object_maxx_dim
{
  \l__flowframkutils_object_minx_dim
  + \l__flowframkutils_object_width_dim
}
\dim_set:Nn \l__flowframkutils_object_maxy_dim
{
  \l__flowframkutils_object_miny_dim
  + \l__flowframkutils_object_height_dim
}

```

Calculate centre point.

```

\dim_set:Nn \l__flowframkutils_object_cx_dim
{
  \l__flowframkutils_object_minx_dim
  + 0.5 \l__flowframkutils_object_width_dim
}
\dim_set:Nn \l__flowframkutils_object_cy_dim
{
  \l__flowframkutils_object_miny_dim
  + 0.5 \l__flowframkutils_object_height_dim
}
}
}
\tl_if_eq:nnT { #2 } { JDRGroup }
{

```

Increment group level. This isn't scoped so it will need decrementing afterwards.

```

\int_incr:N \l_flowframkutils_group_level_int
}

```

Encapsulate object.

```

\exp_args:NV
\__flowframkutils_current_encap:nn
\l__flowframkutils_overlay_tl { #8 }
\bool_if:NT \l_flowframkutils_annotate_bool
{

```

Add annotation (arrow, if applicable, and text).

```

\tl_clear:N \l_flowframkutils_annotation_tl
\tl_set:Nn \l__flowframkutils_propval_tl { \q_no_value }
\__flowframkutils_get_annotation_text:Nnn

```

```

\l__flowframkutils_propval_tl
{ #3 } { #4 }

```

If no text value, skip.

```

\quark_if_no_value:NF
  \l__flowframkutils_propval_tl
  {
    \int_if_zero:nTF
      { \l_flowframkutils_annotate_location_int }
      {

```

Position set to auto so calculate suitable position id.

```

  \__flowframkutils_auto_set_annotate_location:
  }
  {

```

Specific position, so just store position id.

```

  \int_set_eq:NN
    \l_flowframkutils_current_annotate_location_int
    \l_flowframkutils_annotate_location_int
  }

```

Set the annotation content.

```

  \exp_args:NV
  \__flowframkutils_set_annotate_content:n
  \l__flowframkutils_propval_tl
}

```

If an annotation has been set, draw it with the applicable overlay.

```

\tl_if_empty:NF \l__flowframkutils_annotation_tl
{
  \tl_if_empty:NT \l__flowframkutils_overlay_tl
  {

```

No overlay for the current object so use the default annotation overlay.

```

  \flowframkutils_get_default_annotation_overlay:Nnn
  \l__flowframkutils_overlay_tl { #1 } { #2 }
  }

```

Draw the annotation:

```

  \exp_args:NVV
  \__flowframkutils_annotate_uncover:nn
  \l__flowframkutils_overlay_tl
  \l__flowframkutils_annotation_tl
  }
}

```

Decrement the group level counter.

```

\tl_if_eq:nnT { #2 } { JDRGroup }
{
  \int_decr:N \l_flowframkutils_group_level_int
}
}

```

Used when the annotation location is set to auto:

```
\cs_new:Nn \__flowframkutils_auto_set_annotate_location:
{
  \dim_compare:nNnTF
    { \l__flowframkutils_object_cx_dim }
    <
    { \l__flowframkutils_image_cx_dim }
  {
```

The centre of the object's bounding box is to the left of the centre of the image's bounding box (West).

```
    \int_set_eq:NN
      \l_flowframkutils_current_annotate_location_int
      \c_flowframkutils_annotate_location_west_int
    \dim_compare:nNnTF
      { \l__flowframkutils_object_miny_dim }
      >
      { \l__flowframkutils_image_cy_dim }
    {
```

The lowest point of the object's bounding box is higher than the middle of the image's bounding box.

```
    \int_set_eq:NN
      \l_flowframkutils_current_annotate_location_int
      \c_flowframkutils_annotate_location_northwest_int
    }
  {
    \dim_compare:nNnT
      { \l__flowframkutils_object_maxy_dim }
      <
      { \l__flowframkutils_image_cy_dim }
    {
```

The highest point of the object's bounding box is lower than the middle of the image's bounding box.

```
    \int_set_eq:NN
      \l_flowframkutils_current_annotate_location_int
      \c_flowframkutils_annotate_location_southwest_int
    }
  }
}
```

The centre of the object's bounding box is not to the left of the centre of the image's bounding box (East).

```
\int_set_eq:NN
  \l_flowframkutils_current_annotate_location_int
  \c_flowframkutils_annotate_location_east_int
\dim_compare:nNnTF
  { \l__flowframkutils_object_miny_dim }
  >
```

```

    { \l__flowframkutils_image_cy_dim }
  {

```

The lowest point of the object's bounding box is higher than the middle of the image's bounding box.

```

    \int_set_eq:NN
      \l_flowframkutils_current_annotate_location_int
      \c_flowframkutils_annotate_location_northeast_int
  }
  {
    \dim_compare:nNnT
      { \l__flowframkutils_object_maxy_dim }
      <
      { \l__flowframkutils_image_cy_dim }
  {

```

The highest point of the object's bounding box is lower than the middle of the image's bounding box.

```

    \int_set_eq:NN
      \l_flowframkutils_current_annotate_location_int
      \c_flowframkutils_annotate_location_southeast_int
  }
}
}
}

```

Default arrow colour is black:

```

\tl_new:N \l_flowframkutils_annotation_stroke_color_tl
\tl_set:Nn \l_flowframkutils_annotation_stroke_color_tl
{
  \pgfsetstrokecolor { black }
}

```

Default start arrow head (at the bounding box edge):

```

\tl_new:N \l_flowframkutils_annotation_start_arrow_tl
\tl_set:Nn \l_flowframkutils_annotation_start_arrow_tl
{
  \pgfsetarrowsstart { < }
}

```

No end arrow head by default (the point where the text is placed).

```

\tl_new:N \l_flowframkutils_annotation_end_arrow_tl
\cs_new:Nn \__flowframkutils_set_annotate_content:n
{

```

The code to draw the arrow and text is stored in a token list variable.

```

\tl_set:Nn \l__flowframkutils_annotation_tl
{
  \begin { pgfscope }
  \l_flowframkutils_annotation_stroke_color_tl
  \l_flowframkutils_annotation_start_arrow_tl

```

```

\l_flowframkutils_annotation_end_arrow_tl
}

```

The position of the text in relation to the object's bounding box is indicated by a numeric id.

```

\int_case:nn { \l_flowframkutils_current_annotate_location_int }
{
  { \c_flowframkutils_annotate_location_north_int }
  {

```

North (vertical arrow pointing downwards with text on top). Calculate the co-ordinates of the text (the end of the arrow).

```

\dim_set_eq:NN
\l_flowframkutils_x_dim
\l_flowframkutils_object_cx_dim
\dim_set:Nn \l_flowframkutils_y_dim
{
  \l_flowframkutils_object_maxy_dim
  + \l_flowframkutils_annotate_offsety_dim
}

```

Add the code to draw the arrow if applicable.

```

\bool_if:NT \l_flowframkutils_annotate_arrow_bool
{
  \tl_put_right:Ne \l_flowframkutils_annotation_tl
  {
    \exp_not:N \pgfpathmoveto
    {
      \exp_not:N \pgfpoint
      { \dim_use:N \l_flowframkutils_object_cx_dim }
      { \dim_use:N \l_flowframkutils_object_maxy_dim }
    }
    \exp_not:N \pgfpathlineto
    {
      \exp_not:N \pgfpoint
      {
        \dim_use:N \l_flowframkutils_x_dim
      }
      {
        \dim_use:N \l_flowframkutils_y_dim
      }
    }
    \exp_not:N \pgfusepath { stroke }
  }
}

```

Add the code to position text.

```

\tl_put_right:Ne \l_flowframkutils_annotation_tl
{
  \exp_not:N \pgftext
  [

```

```

bottom ,
at = {
  \exp_not:N \pgfpoint
  {
    \dim_use:N \l__flowframkutils_x_dim
  }
  {
    \dim_use:N \l__flowframkutils_y_dim
  }
} ]
{ #1 }
}
}
{ \c_flowframkutils_annotate_location_northeast_int }
{

```

North-East (slanted arrow pointing down left with text to the upper right).
Calculate the co-ordinates of the text (the end of the arrow).

```

\dim_set:Nn \l__flowframkutils_x_dim
{
  \l__flowframkutils_object_maxx_dim
  + \l_flowframkutils_annotate_offsetx_dim
}
\dim_set:Nn \l__flowframkutils_y_dim
{
  \l__flowframkutils_object_maxy_dim
  + \l_flowframkutils_annotate_offsety_dim
}

```

Add the code to draw the arrow if applicable.

```

\bool_if:NT \l_flowframkutils_annotate_arrow_bool
{
  \tl_put_right:Ne \l__flowframkutils_annotation_tl
  {
    \exp_not:N \pgfpathmoveto
    {
      \exp_not:N \pgfpoint
      { \dim_use:N \l__flowframkutils_object_maxx_dim }
      { \dim_use:N \l__flowframkutils_object_maxy_dim }
    }
  }
  \exp_not:N \pgfpathlineto
  {
    \exp_not:N \pgfpoint
    {
      \dim_use:N \l__flowframkutils_x_dim
    }
    {
      \dim_use:N \l__flowframkutils_y_dim
    }
  }
}
\exp_not:N \pgfusepath { stroke }

```



```

    }
}

```

Add the code to position text.

```

\tl_put_right:Ne \l__flowframtkutils_annotation_tl
{
  \exp_not:N \pgftext
  [
    left , bottom ,
    at = {
      \exp_not:N \pgfpoint
      {
        \dim_use:N \l__flowframtkutils_x_dim
      }
      {
        \dim_use:N \l__flowframtkutils_y_dim
      }
    } ]
  { #1 }
}
}
{ \c_flowframtkutils_annotate_location_east_int }
{

```

East (horizontal arrow pointing leftwards with text to the right). Calculate the co-ordinates of the text (the end of the arrow).

```

\dim_set:Nn \l__flowframtkutils_x_dim
{
  \l__flowframtkutils_object_maxx_dim
  + \l__flowframtkutils_annotate_offsetx_dim
}
\dim_set_eq:NN
\l__flowframtkutils_y_dim
\l__flowframtkutils_object_cy_dim

```

Add the code to draw the arrow if applicable.

```

\bool_if:NT \l_flowframtkutils_annotate_arrow_bool
{
  \tl_put_right:Ne \l__flowframtkutils_annotation_tl
  {
    \exp_not:N \pgfpathmoveto
    {
      \exp_not:N \pgfpoint
      { \dim_use:N \l__flowframtkutils_object_maxx_dim }
      { \dim_use:N \l__flowframtkutils_object_cy_dim }
    }
    \exp_not:N \pgfpathlineto
    {
      \exp_not:N \pgfpoint
      {
        \dim_use:N \l__flowframtkutils_x_dim

```

```

    }
    {
      \dim_use:N \l__flowframtkutils_y_dim
    }
  }
  \exp_not:N \pgfusepath { stroke }
}

```

Add the code to position text.

```

\tl_put_right:Ne \l__flowframtkutils_annotation_tl
{
  \exp_not:N \pgftext
  [
    left ,
    at = {
      \exp_not:N \pgfpoint
      {
        \dim_use:N \l__flowframtkutils_x_dim
      }
      {
        \dim_use:N \l__flowframtkutils_y_dim
      }
    } ]
  { #1 }
}
}
{ \c_flowframtkutils_annotate_location_southeast_int }
{

```

South-East (slanted arrow pointing up left with text to the lower right). Calculate the co-ordinates of the text (the end of the arrow).

```

\dim_set:Nn \l__flowframtkutils_x_dim
{
  \l__flowframtkutils_object_maxx_dim
  + \l__flowframtkutils_annotate_offsetx_dim
}
\dim_set:Nn \l__flowframtkutils_y_dim
{
  \l__flowframtkutils_object_miny_dim
  - \l__flowframtkutils_annotate_offsety_dim
}

```

Add the code to draw the arrow if applicable.

```

\bool_if:NT \l__flowframtkutils_annotate_arrow_bool
{
  \tl_put_right:Ne \l__flowframtkutils_annotation_tl
  {
    \exp_not:N \pgfpathmoveto
    {
      \exp_not:N \pgfpoint

```

```

        { \dim_use:N \l__flowframtkutils_object_maxx_dim }
        { \dim_use:N \l__flowframtkutils_object_miny_dim }
    }
\exp_not:N \pgfpathlineto
{
    \exp_not:N \pgfpoint
    {
        \dim_use:N \l__flowframtkutils_x_dim
    }
    {
        \dim_use:N \l__flowframtkutils_y_dim
    }
}
\exp_not:N \pgfusepath { stroke }
}
}

```

Add the code to position text.

```

\tl_put_right:Ne \l__flowframtkutils_annotation_tl
{
    \exp_not:N \pgftext
    [
        top, left ,
        at = {
            \exp_not:N \pgfpoint
            {
                \dim_use:N \l__flowframtkutils_x_dim
            }
            {
                \dim_use:N \l__flowframtkutils_y_dim
            }
        } ]
    { #1 }
}
}
{ \c_flowframtkutils_annotate_location_south_int }
{

```

South (vertical arrow pointing upwards with text below). Calculate the coordinates of the text (the end of the arrow).

```

\dim_set_eq:NN
    \l__flowframtkutils_x_dim
    \l__flowframtkutils_object_cx_dim
\dim_set:Nn \l__flowframtkutils_y_dim
{
    \l__flowframtkutils_object_miny_dim
    - \l__flowframtkutils_annotate_offsety_dim
}

```

Add the code to draw the arrow if applicable.

```

\bool_if:NT \l_flowframtkutils_annotate_arrow_bool

```

```

{
  \tl_put_right:Ne \l__flowframkutils_annotation_tl
  {
    \exp_not:N \pgfpathmoveto
    {
      \exp_not:N \pgfpoint
      { \dim_use:N \l__flowframkutils_object_cx_dim }
      { \dim_use:N \l__flowframkutils_object_miny_dim }
    }
    \exp_not:N \pgfpathlineto
    {
      \exp_not:N \pgfpoint
      {
        \dim_use:N \l__flowframkutils_x_dim
      }
      {
        \dim_use:N \l__flowframkutils_y_dim
      }
    }
    \exp_not:N \pgfusepath { stroke }
  }
}

```

Add the code to position text.

```

\tl_put_right:Ne \l__flowframkutils_annotation_tl
{
  \exp_not:N \pgftext
  [
    top,
    at = {
      \exp_not:N \pgfpoint
      {
        \dim_use:N \l__flowframkutils_x_dim
      }
      {
        \dim_use:N \l__flowframkutils_y_dim
      }
    } ]
  { #1 }
}
}
{ \c_flowframkutils_annotate_location_southwest_int }
{

```

South-West (slanted arrow pointing up right with text to the lower left). Calculate the co-ordinates of the text (the end of the arrow).

```

\dim_set:Nn \l__flowframkutils_x_dim
{
  \l__flowframkutils_object_minx_dim
  - \l__flowframkutils_annotate_offsetx_dim
}

```

```

\dim_set:Nn \l__flowframkutils_y_dim
{
  \l__flowframkutils_object_miny_dim
  - \l_flowframkutils_annotate_offsety_dim
}

```

Add the code to draw the arrow if applicable.

```

\bool_if:NT \l_flowframkutils_annotate_arrow_bool
{
  \tl_put_right:Ne \l__flowframkutils_annotation_tl
  {
    \exp_not:N \pgfpathmoveto
    {
      \exp_not:N \pgfpoint
      { \dim_use:N \l__flowframkutils_object_minx_dim }
      { \dim_use:N \l__flowframkutils_object_miny_dim }
    }
    \exp_not:N \pgfpathlineto
    {
      \exp_not:N \pgfpoint
      {
        \dim_use:N \l__flowframkutils_x_dim
      }
      {
        \dim_use:N \l__flowframkutils_y_dim
      }
    }
    \exp_not:N \pgfusepath { stroke }
  }
}

```

Add the code to position text.

```

\tl_put_right:Ne \l__flowframkutils_annotation_tl
{
  \exp_not:N \pgftext
  [
    top , right,
    at = {
      \exp_not:N \pgfpoint
      {
        \dim_use:N \l__flowframkutils_x_dim
      }
      {
        \dim_use:N \l__flowframkutils_y_dim
      }
    } ]
  { #1 }
}
}
{ \c_flowframkutils_annotate_location_west_int }
{

```

West (horizontal arrow pointing rightwards with text to the left). Calculate the co-ordinates of the text (the end of the arrow).

```

\dim_set:Nn \l__flowframkutils_x_dim
{
  \l__flowframkutils_object_minx_dim
  - \l__flowframkutils_annotate_offsetx_dim
}
\dim_set_eq:NN
\l__flowframkutils_y_dim
\l__flowframkutils_object_cy_dim

```

Add the code to draw the arrow if applicable.

```

\bool_if:NT \l__flowframkutils_annotate_arrow_bool
{
  \tl_put_right:Ne \l__flowframkutils_annotation_tl
  {
    \exp_not:N \pgfpathmoveto
    {
      \exp_not:N \pgfpoint
      { \dim_use:N \l__flowframkutils_object_minx_dim }
      { \dim_use:N \l__flowframkutils_object_cy_dim }
    }
    \exp_not:N \pgfpathlineto
    {
      \exp_not:N \pgfpoint
      {
        \dim_use:N \l__flowframkutils_x_dim
      }
      {
        \dim_use:N \l__flowframkutils_y_dim
      }
    }
    \exp_not:N \pgfusepath { stroke }
  }
}

```

Add the code to position text.

```

\tl_put_right:Ne \l__flowframkutils_annotation_tl
{
  \exp_not:N \pgftext
  [
    right,
    at = {
      \exp_not:N \pgfpoint
      {
        \dim_use:N \l__flowframkutils_x_dim
      }
      {
        \dim_use:N \l__flowframkutils_y_dim
      }
    }
  ]
}

```

```

    } ]
    { #1 }
  }
}
{ \c_flowframtkutils_annotate_location_northwest_int }
{

```

North-West (slanted arrow pointing down right with text to the upper left).
Calculate the co-ordinates of the text (the end of the arrow).

```

\dim_set:Nn \l__flowframtkutils_x_dim
{
  \l__flowframtkutils_object_minx_dim
  - \l_flowframtkutils_annotate_offsetx_dim
}
\dim_set:Nn \l__flowframtkutils_y_dim
{
  \l__flowframtkutils_object_maxy_dim
  + \l_flowframtkutils_annotate_offsety_dim
}

```

Add the code to draw the arrow if applicable.

```

\bool_if:NT \l_flowframtkutils_annotate_arrow_bool
{
  \tl_put_right:Ne \l__flowframtkutils_annotation_tl
  {
    \exp_not:N \pgfpathmoveto
    {
      \exp_not:N \pgfpoint
      { \dim_use:N \l__flowframtkutils_object_minx_dim }
      { \dim_use:N \l__flowframtkutils_object_maxy_dim }
    }
    \exp_not:N \pgfpathlineto
    {
      \exp_not:N \pgfpoint
      {
        \dim_use:N \l__flowframtkutils_x_dim
      }
      {
        \dim_use:N \l__flowframtkutils_y_dim
      }
    }
    \exp_not:N \pgfusepath { stroke }
  }
}

```

Add the code to position text.

```

\tl_put_right:Ne \l__flowframtkutils_annotation_tl
{
  \exp_not:N \pgftext
  [
    bottom, right,

```

```

        at = {
          \exp_not:N \pgfpoint
          {
            \dim_use:N \l__flowframtkutils_x_dim
          }
          {
            \dim_use:N \l__flowframtkutils_y_dim
          }
        } ]
      { #1 }
    }
  }
}
\l_put_right:Nn \l__flowframtkutils_annotation_tl
{
  \end { pgfscope }
}
}

```

FlowFramTkUtilsSetOverlayEncap

```

\NewDocumentCommand \FlowFramTkUtilsSetOverlayEncap { o }
{
  \IfValueT { #1 }
  {
    \keys_set:nn { flowframtkutils / overlay } { #1 }
  }
  \renewcommand \flowframtkencapobject
  {
    \flowframtkutils_uncover_encap:nnnnnnnn
  }
}

```

FlowFramTkUtilsOverlayEncapSetup

```

\NewDocumentCommand \FlowFramTkUtilsOverlayEncapSetup { m }
{
  \keys_set:nn { flowframtkutils / overlay } { #1 }
}

```

`\includeteximage` FlowframTk's export to `pgfpicture` option just creates a `.tex` file that contains the `pgfpicture` environment. This file may simply be `\input` but provide a command similar to `\includegraphics` that can be used with graphics options to resize the image. This is just provided as there are other packages (e.g. `jmlrutils`) that define this command.

NB although it's possible to change `pgf` co-ordinate scaling, the image may include text or bitmaps that may need to be scaled too.

```

\ProvideDocumentCommand \includeteximage { o m }
{
  \group_begin:
  \let\input@path\Ginput@path

```



```

\IfValueTF { #1 }
{
  \hbox_set:Nn \l__flowframtk_img_box
    { \file_input:n { #2 } }
  \keys_set:nn { flowframtk / img } { #1 }
  \__flowframtk_img_apply_resize:
  \mbox { \box_use:N \l__flowframtk_img_box }
}
{
  \file_input:n { #2 }
}
\group_end:
}

```

Variables for use in the above.

```

\box_new:N \l__flowframtk_img_box
\tl_new:N \l__flowframtk_img_width_tl
\tl_new:N \l__flowframtk_img_height_tl

```

Apply resizing options:

```

\cs_new:Nn \__flowframtk_img_apply_resize:
{
  \tl_if_empty:NTF \l__flowframtk_img_width_tl
  {
    \tl_if_empty:NF \l__flowframtk_img_height_tl
    {
      \box_resize_to_ht:Nn
        \l__flowframtk_img_box { \l__flowframtk_img_height_tl }
    }
    \tl_clear:N \l__flowframtk_img_height_tl
  }
  {
    \tl_if_empty:NTF \l__flowframtk_img_height_tl
    {
      \box_resize_to_wd:Nn
        \l__flowframtk_img_box { \l__flowframtk_img_width_tl }
    }
    {
      \box_resize_to_wd_and_ht:Nnn
        \l__flowframtk_img_box
        { \l__flowframtk_img_width_tl }
        { \l__flowframtk_img_height_tl }
      \tl_clear:N \l__flowframtk_img_height_tl
    }
    \tl_clear:N \l__flowframtk_img_width_tl
  }
}
\keys_define:nn { flowframtk / img }
{
  scale .code:n =

```

```

{
  \__flowframtk_img_apply_resize:
  \box_scale:Nnn \l__flowframtk_img_box { #1 } { #1 }
},
angle .code:n =
{
  \__flowframtk_img_apply_resize:
  \box_rotate:Nn \l__flowframtk_img_box { #1 }
},
width .tl_set:N = \l__flowframtk_img_width_tl ,
height .tl_set:N = \l__flowframtk_img_height_tl ,
alt .code:n = { }
}

```

`\flowframtkimgtitlechar{original-char}{PDF-char}`

`\flowframtkimgtitlechar`

In case it's necessary to use `\pdfinfo`, FlowframTk will detect and encapsulate problematic characters in the title with this command. The first argument is the original character and the second is the escaped character suitable for use in a PDF string. For example, if the first argument is (or) then the second will be \ (or \). NB this must be able to expand.

```
\newcommand \flowframtkimgtitlechar [ 2 ] { #1 }
```

The above will be locally changed to the following when the title needs to be written to the /Title PDF string.

```

\cs_new:Nn \__flowframtk_imgtitlechar:nn
{
  \tl_to_str:n { #2 }
}

```

Provide a command to use if `hyperref` hasn't been loaded. This will use `\pdfinfo` or `\pdfextension`, if available.

```

\cs_if_exist:NTF \pdfinfo
{
  \cs_new:Nn \__flowframtk_pdfinfo:n
  {
    \pdfinfo { #1 }
  }
}
{
  \cs_if_exist:NTF \pdfextension
  {
    \cs_new:Nn \__flowframtk_pdfinfo:n
    {
      \pdfextension info { #1 }
    }
  }
}

```

```

        \cs_new:Nn \__flowframtk_pdfinfo:n { }
    }
}

```

Token list variable to hold the title:

```
\tl_new:N \l_flowframtk_image_title_tl
```

\flowframtkSetTitle This command is provided to allow the image's early preamble to provide an alternative definition.

```

\ProvideDocumentCommand \flowframtkSetTitle { m }
{
  \tl_set:Ne \l_flowframtk_image_title_tl { #1 }
  \cs_if_exist:NTF \hypersetup
  {
    \hypersetup { pdftitle = { \l_flowframtk_image_title_tl } }
  }
  {
    \cs_set_eq:NN \flowframtkimgtitlechar \__flowframtk_imgtitlechar:nn
    \__flowframtk_pdfinfo:n { /Title ( #1 ) }
    \cs_set_eq:NN \flowframtkimgtitlechar \use_i:nn
  }
  \cs_if_exist:NT \title
  {
    \exp_args:NV \title \l_flowframtk_image_title_tl
  }
}

```

Token list variable to hold the creation date:

```
\tl_new:N \l_flowframtk_image_creationdate_tl
```

\flowframtkSetCreationDate Again, this command is provided to allow the image's early preamble to provide an alternative definition.

```

\ProvideDocumentCommand \flowframtkSetCreationDate { m }
{
  \tl_set:Ne \l_flowframtk_image_creationdate_tl
  {
    \tl_to_str:n { #1 }
  }
  \cs_if_exist:NTF \hypersetup
  {
    \exp_args:Ne
    \hypersetup
    { pdfcreationdate = { \l_flowframtk_image_creationdate_tl } }
  }
  {
    \exp_args:Ne
    \__flowframtk_pdfinfo:n
    { /CreationDate ( \l_flowframtk_image_creationdate_tl ) }
  }
}

```

```

\keys_define:nn { flowframtk / imageinfo }
{
  title .code:n = \flowframtkSetTitle { #1 } ,
  creationdate .code:n = \flowframtkSetCreationDate { #1 } ,
}

```

`\flowframtkimageinfo` This is the command that FlowframTk will actually write to the exported file, if the export settings request PDF data to be added:

```

\NewDocumentCommand \flowframtkimageinfo { m }
{
  \keys_set:nn { flowframtk / imageinfo } { #1 }
}

```

The following commands are specifically for export options that define flow, static and dynamic frames. The image objects that have `flowfram` data assigned to them may either use the associated object as a border or simply use the object's bounding box to determine the frame's location and size. Where the object should be used as a border, a command needs to be defined that can be used as the border frame command.

```

\flowframtkNewFrameBorder{<label>}{<pgfpicture-code>}

```

`\flowframtkNewFrameBorder`

Defines an internal command that takes a single argument, which will be the content of the applicable flow, static or dynamic frame. This command will then be used as the frame's border command (instead of `\fbox`). The second argument is the `pgf` code needed to replicate the object (path, bitmap etc) from the image and position the frame content within the picture according to the margins setup in FlowframTk.

```

\NewDocumentCommand \flowframtkNewFrameBorder { m m }
{
  \cs_set:cpn { @flf@border@ #1 } ##1 { #2 }
}

```

`\flowframtkUseFrameBorderCsName` Expands to the control sequence name of the custom border frame command that was defined by `\flowframtkNewFrameBorder`.

```

\newcommand \flowframtkUseFrameBorderCsName [ 1 ]
{
  @flf@border@ #1
}

```

`\flowframtkUseFrameBorderCsName` Use the custom border frame command that was defined by `\flowframtkNewFrameBorder`.

```

\newcommand \flowframtkUseFrameBorderCs [ 1 ]
{
  \use:n { @flf@border@ #1 }
}

```

`\flowframtkNewDynamicStyle` Create a style command for a dynamic frame

```

\NewDocumentCommand \flowframtkNewDynamicStyle { m m }
{
  \cs_set:cpn { @flf@dfstyle@ #1 } { #2 }
}

```

`\flowframtkUseDynamicStyleCsName` Expands to the control sequence name of a style command for a dynamic frame

```

\NewDocumentCommand \flowframtkUseDynamicStyleCsName { m }
{
  @flf@dfstyle@ #1
}

```

`\flowframtkUseDynamicStyle` Expands to the control sequence name of a style command for a dynamic frame

```

\NewDocumentCommand \flowframtkUseDynamicStyle { m }
{
  \use:n { @flf@dfstyle@ #1 }
}

```

These commands are used instead of `\makedfheaderfooter` for a bespoke dynamic headers and footers.

`\flowframtkSetDynamicOddHead` The same header frame is used for both odd and even pages or it's a single-sided document.

```

\NewDocumentCommand \flowframtkSetDynamicOddHead { m }
{
  \__flowfram_get_dynamic_id:n { #1 }
  \int_set_eq:NN \g__flowfram_dynamic_header_int \l__flowfram_id_int
  \renewcommand \@dothehead { }
  \renewcommand \@dodynamicthehead
  {
    \__flowfram_set_dynamic_contents:nn
    { \g__flowfram_dynamic_header_int }
    {
      \flowfram_dynamic_header:n
      { \g__flowfram_dynamic_header_int }
    }
  }
}

```

`\flowframtkSetDynamicOddEvenHead` There are two dynamic header frames, one for odd pages and one for even pages.

```

\NewDocumentCommand \flowframtkSetDynamicOddEvenHead { m m }
{
  \__flowfram_get_dynamic_id:n { #1 }
  \int_set_eq:NN \g__flowfram_dynamic_header_int \l__flowfram_id_int
  \__flowfram_get_dynamic_id:n { #2 }
  \int_set_eq:NN \g__flowfram_dynamic_even_header_int \l__flowfram_id_int
  \renewcommand \@dothehead { }
  \renewcommand \@dodynamicthehead
  {

```

odd head

```
  \__flowfram_set_dynamic_contents:nn
  { \g__flowfram_dynamic_header_int }
  {
    \flowfram_dynamic_odd_header:n
    { \g__flowfram_dynamic_header_int }
  }
}
```

even head

```
  \__flowfram_set_dynamic_contents:nn
  { \g__flowfram_dynamic_even_header_int }
  {
    \flowfram_dynamic_even_header:n
    { \g__flowfram_dynamic_even_header_int }
  }
}
```

Ensure twoside option is on:

```
  \@twosidetrue
}
```

`\flowframtkSetDynamicEvenHead` Only the header frame for even pages has been defined. NB the document must have the twoside setting on.

```
\NewDocumentCommand \flowframtkSetDynamicEvenHead { m }
{
  \__flowfram_get_dynamic_id:n { #1 }
  \int_set_eq:NN \g__flowfram_dynamic_even_header_int \l__flowfram_id_int
  \renewcommand \@dothehead { }
  \cs_set:Ne \@dodynamicthehead
  {
    \__flowfram_set_dynamic_contents:nn
    { \g__flowfram_dynamic_even_header_int }
    {
      \flowfram_dynamic_even_header:n
      { \g__flowfram_dynamic_even_header_int }
    }
  }
}
```

Ensure twoside option is on:

```
  \@twosidetrue
}
```

`\flowframtkSetDynamicOddFoot` The same footer frame is used for both odd and even pages or it's a single-sided document.

```
\NewDocumentCommand \flowframtkSetDynamicOddFoot { m }
{
  \__flowfram_get_dynamic_id:n { #1 }
  \int_set_eq:NN \g__flowfram_dynamic_footer_int \l__flowfram_id_int
  \renewcommand \@dothefoot { }
  \renewcommand \@dodynamicthefoot
}
```

```

    {
      \__flowfram_set_dynamic_contents:nn
      { \g__flowfram_dynamic_footer_int }
      {
        \flowfram_dynamic_footer:n
        { \g__flowfram_dynamic_footer_int }
      }
    }
  }
}

```

`\flowframtkSetDynamicOddEvenFoot` There are two dynamic footer frames, one for odd pages and one for even pages.

```

\NewDocumentCommand \flowframtkSetDynamicOddEvenFoot { m m }
{
  \__flowfram_get_dynamic_id:n { #1 }
  \int_set_eq:NN \g__flowfram_dynamic_footer_int \l__flowfram_id_int
  \__flowfram_get_dynamic_id:n { #2 }
  \int_set_eq:NN \g__flowfram_dynamic_even_footer_int \l__flowfram_id_int
  \renewcommand \dothefoot { }
  \renewcommand \dodynamicthefoot
  {
    odd foot
      \__flowfram_set_dynamic_contents:nn
      { \g__flowfram_dynamic_footer_int }
      {
        \flowfram_dynamic_odd_footer:n
        { \g__flowfram_dynamic_footer_int }
      }
    even foot
      \__flowfram_set_dynamic_contents:nn
      { \g__flowfram_dynamic_even_footer_int }
      {
        \flowfram_dynamic_even_footer:n
        { \g__flowfram_dynamic_even_footer_int }
      }
  }
}

```

Ensure twoside option is on:

```

\@twosidetrue
}

```

`\flowframtkSetDynamicEvenFoot` Only the footer frame for even pages has been defined. NB the document must have the twoside setting on.

```

\NewDocumentCommand \flowframtkSetDynamicEvenFoot { m }
{
  \__flowfram_get_dynamic_id:n { #1 }
  \int_set_eq:NN \g__flowfram_dynamic_even_footer_int \l__flowfram_id_int
  \renewcommand \dothefoot { }
  \renewcommand \dodynamicthefoot
  {

```

```

    \__flowfram_set_dynamic_contents:nn
    { \g__flowfram_dynamic_even_footer_int }
    {
      \flowfram_dynamic_even_footer:n
      { \g__flowfram_dynamic_even_footer_int }
    }
  }
}

```

Ensure twoside option is on:

```

  \@twosidetrue
}

```

`\flowframtkMakeDFHeaderFooter` End part of `\makedfheaderfooter`:

```

\NewDocumentCommand \flowframtkMakeDFHeaderFooter { }
{
  \tl_clear:N \g__flowfram_headercolor_tl
  \tl_clear:N \g__flowfram_footercolor_tl
  \__flowfram_adjust_page_styles:n
  {
    \let \ps@myheadings \ps@ffmyheadings
    \let \ps@headings \ps@ffheadings
    \let \ps@plain \ps@ffplain
    \let \ps@empty \ps@ffempty
  }
  \pagestyle { ffheadings }
  \cs_set:Nn \__flowfram_only_pre_makedfheaderfooter:nn
  {
    \msg_error:nnn { flowfram } { option-too-late } { ##1 }
    { \flowframtkMakeDFHeaderFooter }
  }
}

\ExplSyntaxOff

```

3 Rollback v1.17 (flowfram-2014-09-30)

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{flowfram}[2014/09/30 v1.17 (NLCT)]
\RequirePackage{ifthen}
\RequirePackage{xkeyval}
\RequirePackage{graphics}
\RequirePackage{afterpage}
\RequirePackage{xfor}
\RequirePackage{etoolbox}
\@ifundefined{@ldc@l@r}{\RequirePackage{color}}{}
\newcommand{\setffdraftcolor}{\color[gray]{0.8}}
\newcommand{\setffdrafttypeblockcolor}{\color[gray]{0.9}}
\newlength\fflabelsep
\fflabelsep=1pt

```



```

\newcommand*\fflabelfont{\small\sffamily}
\newif\ifshowtypeblock
\newif\ifshowmargins
\newif\ifshowframebbox
\newcommand*\@ffdraft{%
  \showtypeblocktrue
  \showmarginstrue
  \showframebboxtrue
}
\newcommand*\@ffnodraft{%
  \showtypeblockfalse
  \showmarginsfalse
  \showframebboxfalse
}
\newcommand*\@fr@meifdraft}[3] [\setffdraftcolor]{%
  \def\ff@backcol{\none}}%
  \ifundefined{color}{\frame{#2}}{#1\frame{#2}}%
  \ifthenelse{\equal{#3}{}}{}%
  {%
    \makebox[Opt][l]{\hskip\fflabelsep\fflabelfont{[#3]}}%
  }%
}%
\newcommand*\@s@tffcol{}
\newcommand*\@s@tffttextcol{}
\newcommand*\@ffbackground}[1]{#1}
\DeclareOptionX{draft}{\@ffdraft}
\DeclareOptionX{final}{\@ffnodraft}
\@ffnodraft
\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
  }
\newcommand*\flf@message}[1]{%
  \flf@doifverbose
  {%
    \PackageInfo{flowfram}{##1}%
  }%
}
\newcommand*\flf@doifverbose}[1]{}
\define@boolkey{flowfram.sty}[@ttb@]{rotate}[true]{}
\@ttb@rotatetrue
\DeclareOptionX{norotate}{\@ttb@rotatefalse}

```

```

\newcommand{\rotateframe}[2]{%
  \if@ttb@rotate
    \rotatebox{#1}{#2}%
  \else
    #2%
  \fi
}
\newif\if@ttb@num
\@ttb@numfalse
\newif\if@ttb@title
\@ttb@titletrue
\define@choicekey{flowfram.sty}%
{thumbtabs}[\val\nr]%
{title,number,both,none}[title]%
{%
  \ifcase\nr\relax
    \@ttb@numfalse
    \@ttb@titletrue
  \or
    \@ttb@numtrue
    \@ttb@titlefalse
  \or
    \@ttb@numtrue
    \@ttb@titletrue
  \or
    \@ttb@numfalse
    \@ttb@titlefalse
  \fi
}
\DeclareOptionX{ttbtitle}{\@ttb@titletrue}
\DeclareOptionX{ttbnotitle}{\@ttb@titlefalse}
\DeclareOptionX{ttbnum}{\@ttb@numtrue}
\DeclareOptionX{ttbnonum}{\@ttb@numfalse}
\define@choicekey{flowfram.sty}{pages}[\val\nr]%
{relative,absolute}%
{%
  \ifcase\nr\relax
    \renewcommand*{\@ff@pages@countreg}{\c@page}%
  \or
    \renewcommand*{\@ff@pages@countreg}{\c@absolutepage}%
  \fi
}
\newcommand*{\@ff@pages@countreg}{\c@page}
\newcounter{absolutepage}
\define@choicekey{flowfram.sty}{color}[\val\nr]{true,false}[true]{%
  \ifcase\nr\relax
    \@ff@enablecolor
  \or
    \@ff@disablecolor
  \fi
}

```

```

}
\DeclareOptionX{nocolor}{%
  \@ff@disablecolor
}
\newcommand*{\@ff@enablecolor}{%
  \def\flowframecol{{black}}%
  \def\flowframetextcol{{black}}%
  \renewcommand*\@s@tffcol{%
    \ifthenelse{\equal{\ff@col}{}}%
    {}%
    {%
      \expandafter\color\ff@col}%
    }%
  \renewcommand*\@s@tffttextcol{%
    \ifthenelse{\equal{\ff@txtcol}{}}%
    {}%
    {%
      \expandafter\color\ff@txtcol
    }%
  }%
  \renewcommand*{\@ff@background}[1]{%
    \ifthenelse{\equal{\ff@backcol}{{none}}}%
    {}%
    {
      ##1%
    }%
    {%
      {\fboxsep=0pt\expandafter\colorbox\ff@backcol{##1}}%
    }%
  }%
}
\newcommand*{\@ff@disablecolor}{%
  \def\flowframetextcol{}%
  \def\flowframecol{}%
  \renewcommand*\@s@tffcol{}\renewcommand*\@s@tffttextcol{}%
  \renewcommand*\@ff@background[1]{##1}%
}
\newif\iflefttorightcolumns
\lefttorightcolumnstrue
\DeclareOptionX{LR}{\lefttorightcolumnstrue}
\DeclareOptionX{RL}{\lefttorightcolumnstrue}
\ifx\normalcolor\relax
  \@ff@disablecolor
\else
  \@ff@enablecolor
\fi
\ProcessOptionsX
\ifx\normalcolor\relax
  \ifthenelse{\equal{\flowframetextcol}{}}%
  {}%
  {%

```

```

\RequirePackage{color}%
}
\fi
\@ifundefined{chapter}{}%
{%
\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
}%
\newcommand*\ffprechapterhook{}
\let\@ff@OLD@chapter\chapter
\renewcommand*\chapter{%
\ffprechapterhook
\@ff@OLD@chapter
}
}
\newcounter{maxflow}
\c@maxflow=0\relax
\newcounter{thisframe}
\c@thisframe=0\relax
\@ifpackageloaded{hyperref}
{%
\def\theHthisframe{\thepage.\arabic{thisframe}}%
}%
{}
\newcommand*\labelflowidn[1]{%
\def\@currentlabel{\thethisframe}%
\label{#1}%
}%
}
\newcounter{displayedframe}
\c@displayedframe=0
\@ifpackageloaded{hyperref}%
{%
\def\theHdisplayedframe{\thepage.\arabic{displayedframe}}%
}%
{}
\newcommand*\labelflow[1]{%
\def\@currentlabel{\thedisplaysframe}%
\label{#1}%
}%
}

```

```

}
\newcounter{maxstatic}
\c@maxstatic=0\relax
\newcounter{maxdynamic}
\c@maxdynamic=0\relax
\newcount\@colN
\newcount\@ff@tmpN
\newcount\ff@id
\newlength\@ff@offset
\newlength\@ff@tmp@x
\newlength\@ff@tmp@x@even
\newlength\@ff@tmp@y
\newlength\sdf@parindent
\newlength\flowframesep
\flowframesep=\fboxsep
\newlength\flowframerule
\flowframerule=\fboxrule
\newcommand*{\flowframeshowlayout}{%
  \finishthispage
  {%
    \@ffdraft\mbox{ }\finishthispage\clearpage
  }%
}
\newif\ifusedframebreak
\newcommand{\framebreak}[1][4]{%
  \global\usedframebreaktrue
  {%
    \parfillskip=0pt\pagebreak[#1]\parskip=0pt\par\noindent
  }%
}
\newcommand{\finishthispage}{%
  \ifvmode
    \@colN=\c@thisframe\relax
    \count@=\c@absolute@page\relax
    \ifdim \pagetotal<\topskip
      \hbox{ }%
    \fi
    \newpage \write \m@ne {} \vbox {} \penalty -\@Mi
    \ifnum\count@=\c@absolute@page\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
}
\def\cleardoublepage{%
\clearpage
\if@twoside
\ifodd\c@page
\else
\hbox{}%
\clearpage
\fi
\fi
}
\preto\newpage{\global\usedframebreaktrue}
\@twocolumnfalse
\@mparswitchfalse
\newcommand{\globalreversemargin}{%
\global\@mparbottom\z@
\global\@reversemargintrue
}
\newcommand{\globalnormalmargin}{%
\global\@mparbottom\z@\global
\@reversemarginfalse
}
\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}%
}%
}
\newcommand{\setmargin}{%
\@getmarginpos
\csname @ff@margin@\romannumeral\c@thisframe\endcsname

```

```

}%
\ifthenelse{\equal{\ff@margin}{left}}%
{\globalreversemargin}%
{\globalnormalmargin}%
}
\newcommand{\newflowframe}{\@n@wflowframe}
\@onlypreamble{\newflowframe}
\newcommand{\@n@wflowframe}{%
  \global\advance\c@maxflow by 1\relax
  \expandafter\global\expandafter
  \newif\csname ifcolumnframe\romannumeral\c@maxflow\endcsname
  \@ifstar\@snewflowframe\@newflowframe
}
\newcommand{\@snewflowframe}{%
  \expandafter\global\expandafter
  \let\csname ifcolumnframe\romannumeral\c@maxflow\endcsname\iftrue
  \@@newflowframe
}
\newcommand{\@newflowframe}{%
  \expandafter\global\expandafter
  \let\csname ifcolumnframe\romannumeral\c@maxflow\endcsname\iffalse
  \@@newflowframe
}
\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}%
\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]%
    }%
  }
\def\@s@tflowframeid#1[#2]{%
  \edef\ff@label{#2}%
  \@ff@checkuniqueidl{#1}{\ff@label}%
  \expandafter
  \xdef\csname @col@id@\romannumeral#1\endcsname{\ff@label}%
}
\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
}%
}
\newcommand*\@getflowlabel}[1]{%
  \csname @col@id@\romannumeral#1\endcsname
}
\newcommand*\@getflowid}[2]{%
  \@flowframeid{#2}%
  \edef#1{\number\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
  \@colN=\c@maxflow
}%
{}%
}%
\ifnum\ff@id=0\relax
  \PackageError{flowfram}{Can't find flow frame id '#1'}{}%
\fi
}
\define@key{flowframe}{width}%
{
  \ifthenelse{\equal{#1}{}%
}%
  \PackageError{flowfram}{Missing value for 'width' key}{}%
}%
{}%
\def\ff@width{#1}%
}
\define@key{flowframe}{height}%
{
  \ifthenelse{\equal{#1}{}%
}%
  \PackageError{flowfram}{Missing value for 'height' key}{}%
}%
{}%
\def\ff@height{#1}%
}
\define@key{flowframe}{x}%
{
  \ifthenelse{\equal{#1}{}%
}%
  \PackageError{flowfram}{Missing value for 'x' key}{}%
}%
{}%
\def\ff@x{#1}%
}
\define@key{flowframe}{y}%
{
  \ifthenelse{\equal{#1}{}%
}%
  \PackageError{flowfram}{Missing value for 'y' key}{}%
}%
{}%
\def\ff@y{#1}%

```

```

}
\define@key{flowframe}{evenx}%
{%
  \ifthenelse{\equal{#1}{}}%
  {%
    \PackageError{flowfram}{Missing value for 'evenx' key}{}%
  }%
  {}%
  \def\ff@evenx{#1}%
}
\define@key{flowframe}{eveny}%
{%
  \ifthenelse{\equal{#1}{}}%
  {%
    \PackageError{flowfram}{Missing value for 'eveny' key}{}%
  }%
  {}%
  \def\ff@eveny{#1}%
}
\define@key{flowframe}{oddx}%
{%
  \ifthenelse{\equal{#1}{}}%
  {%
    \PackageError{flowfram}{Missing value for 'oddx' key}{}%
  }%
  {}%
  \def\ff@oddx{#1}%
}
\define@key{flowframe}{oddy}%
{%
  \ifthenelse{\equal{#1}{}}%
  {%
    \PackageError{flowfram}{Missing value for 'oddy' key}{}%
  }%
  {}%
  \def\ff@oddy{#1}%
}
\define@key{flowframe}{label}%
{%
  \ifthenelse{\equal{#1}{}}%
  {%
    \PackageError{flowfram}{Missing value for 'label' key}{}%
  }%
  {}%
  \def\ff@label{#1}%
}
\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}%
  }%
}%
}
\define@key{flowframe}{bordercolor}%
{%
  \ifthenelse{\equal{#1}{}}%
  {%
    \PackageError{flowfram}{Missing value for 'bordercolor' key}{}%
  }%
  {}%
  \def\ff@col{#1}%
}
\define@key{flowframe}{textcolor}%
{%
  \ifthenelse{\equal{#1}{}}%
  {%
    \PackageError{flowfram}{Missing value for 'textcolor' key}{}%
  }%
  {}%
  \def\ff@txtcol{#1}%
}
\define@key{flowframe}{backcolor}%
{%
  \ifthenelse{\equal{#1}{}}%
  {%
    \PackageError{flowfram}{Missing value for 'backcolor' key}{}%
  }%
  {}%
  \def\ff@backcol{#1}%
}

```

```

\define@key{flowframe}{pages}%
{%
  \ifthenelse{\equal{#1}{}}%
  {%
    \PackageError{flowfram}{Missing value for 'pages' key}{}%
  }%
  {}%
  \def\ff@pages{#1}%
}
\define@key{flowframe}{excludepages}%
{%
  \def\ff@xpages{#1}%
}
\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%
    }%
  }%
  {}%
}
\define@key{flowframe}{angle}{\def\ff@angle{#1}%
}
\define@choicekey{flowframe}{margin}{left,right,inner,outer}%
{%
  \def\ff@margin{#1}%
}
\define@choicekey{flowframe}{clear}{true,false}[true]{%
  \def\ff@clear{#1}%
}
\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}%
    }%
}
\define@key{flowframe}{shape}%
{%
    \def\ff@shape{#1}%
}
\define@choicekey{flowframe}{valign}{c,t,b}%
{%
    \def\ff@valign{#1}%
}
\define@choicekey{flowframe}{hide}{true,false}[true]{%
    \def\ff@hide{#1}%
}
\define@choicekey{flowframe}{hidethis}{true,false}[true]{%
    \def\ff@hidethis{#1}%
}
}
\newcommand*\setallflowframes[1]{%
    \@colN=0\relax
    \whiledo{\@colN<\c@maxflow}%
    {%
        \advance\@colN by 1\relax
        \@@setflowframe{\@colN}{#1}%
    }%
}
}
\newcommand*\setflowframe{\@ifstar\@ssetflowframe\@setflowframe}
\newcommand{\@ssetflowframe}[2]{%
    \@for\ff@id:=#1\do{%
        \@flowframeid{\ff@id}%
        \@@setflowframe{\ff@id}{#2}%
    }%
}
}
\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}%
        }%
    }%
}

```



```

        {\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
\edef\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
\edef\csname @ff@offset@\romannumeral#1\endcsname{%
\ff@offset}%
}%
\ifdefempty{\ff@angle}{}%
{%
\expandafter
\edef\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}{}%
}%
}
}
\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"}%
}
\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"}%
}
\newcommand*{\flowaddexclusion}[2]{%
  \ifcsempy{@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"}%
}
}

```

```

\newcommand*{\@fflowframeswapcoords}[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}%
}
\newcommand*{\ffswapoddeven}{%
  \@ifstar\@sflowframeswapcoords\@fflowframeswapcoords
}
\newcommand*{\@sflowframeswapcoords}[1]{%
  \@for\@ff@id:=#1\do
  {%
    \@fflowframeid{\@ff@id}%
    \@fflowframeswapcoords{\@ff@id}%
  }%
}
\newcommand*{\@fflowframeswapcoords}[1]{%
  \ifthenelse{\equal{#1}{all}}{%
    {%
      \ff@id=0\relax
      \whiledo{\ff@id<\c@maxflow}{%
        {%
          \advance\ff@id by 1\relax
          \@fflowframeswapcoords{\@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}{%
        {%
          \@fflowframeswapcoords{\@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 \TEor \@colN=\@ff@numend}%
{%
\@flowframeswapcoords{\@colN}%
\advance\@colN by 1\relax
}%
}%
}%
}
\newcommand*\flowframex}[1]{%
\csname col@romannumeral#1@posx\endcsname
}
\newcommand*\flowframey}[1]{%
\csname col@romannumeral#1@posy\endcsname
}
\newcommand*\flowframeevenx}[1]{%
\csname col@romannumeral#1@evenx\endcsname
}
\newcommand*\flowframeeveny}[1]{%
\csname col@romannumeral#1@eveny\endcsname
}
\newcommand*\flowframewidth}[1]{%
\csname colwidth\romannumeral#1\endcsname
}
\newcommand*\flowframeheight}[1]{%
\csname colheight\romannumeral#1\endcsname
}
\def\@setframecol{\@ifnextchar[\@setframecol\@setfr@mecol}
\def\@setframecol[#1]#2\end#3#4#5{%
\expandafter\edef\csname @#5@#4@romannumeral#3\endcsname{%
[#1]#2}}%
}
\def\@setfr@mecol#1\end#2#3#4{%
\expandafter\edef\csname @#4@#3@romannumeral#2\endcsname{{#1}}%
}
\newcommand*\newstaticframe{\@n@wstaticframe}
\newcommand*\@n@wstaticframe{%
\global\advance\c@maxstatic by 1\relax
\newboolean{staticframe\romannumeral\c@maxstatic}%
\@ifstar\@snewstaticframe\@newstaticframe
}
\newcommand*\@snewstaticframe{%
\setboolean{staticframe\romannumeral\c@maxstatic}{true}%
}

```

```

    \@@newstaticframe
  }
\newcommand{\@newstaticframe}{%
  \setboolean{staticframe\romannumeral\c@maxstatic}{false}%
  \@@newstaticframe
}
\newcommand*{\@newstaticframe}[5][all]{%
  \expandafter
    \newbox\c@name @staticframe\romannumeral\c@maxstatic\endcsname
  \expandafter
    \newlength\c@name @sf\romannumeral\c@maxstatic @posx\endcsname
  \expandafter
    \newlength\c@name @sf\romannumeral\c@maxstatic @posy\endcsname
  \expandafter\setlength
    \c@name @sf\romannumeral\c@maxstatic @posx\endcsname{#4}%
  \expandafter\setlength
    \c@name @sf\romannumeral\c@maxstatic @posy\endcsname{#5}%
  \expandafter\newlength
    \c@name @sf\romannumeral\c@maxstatic @evenx\endcsname
  \expandafter\newlength
    \c@name @sf\romannumeral\c@maxstatic @eveny\endcsname
  \expandafter\setlength
    \c@name @sf\romannumeral\c@maxstatic @evenx\endcsname{#4}%
  \expandafter\setlength
    \c@name @sf\romannumeral\c@maxstatic @eveny\endcsname{#5}%
  {\@ff@tmp@x=#2\relax
  \@ff@tmp@y=#3\relax
  \expandafter
    \xdef\c@name @sf@dim@\romannumeral\c@maxstatic\endcsname{%
      [c][\the\@ff@tmp@y][c]{\the\@ff@tmp@x}}%
  \expandafter
    \def\c@name @sf@col@\romannumeral\c@maxstatic\endcsname{%
      \flowframecol}%
  \expandafter
    \def\c@name @sf@txtcol@\romannumeral\c@maxstatic\endcsname{%
      \flowframetextcol}%
  \expandafter
    \def\c@name @sf@backcol@\romannumeral\c@maxstatic\endcsname{%
      {none}}%
  \expandafter
    \xdef\c@name @sf@pages@\romannumeral\c@maxstatic\endcsname{#1}%
  \expandafter
    \gdef\c@name @sf@xpages@\romannumeral\c@maxflow\endcsname{}%
  \expandafter
    \gdef\c@name @sf@offset@\romannumeral\c@maxstatic\endcsname{%
      compute}%
  \expandafter
    \gdef\c@name @sf@angle@\romannumeral\c@maxstatic\endcsname{0}%
  \expandafter
    \gdef\c@name @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]}%
}
\def\@s@tstaticframeid#1[#2]{%
  \edef\ff@label{#2}%
  \@sf@checkuniqueidl{#1}{\ff@label}%
  \expandafter
    \xdef\csname @sf@id@romannumeral#1\endcsname{\ff@label}%
}
\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
}%
}
\newcommand*{\getstaticlabel}[1]{%
  \csname @sf@id@romannumeral#1\endcsname
}
\newcommand*{\getstaticid}[2]{%
  \@staticframeid{#2}\edef#1{\number\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
    \@colN=\c@maxstatic
  }%
  {}%
}%
\ifnum\ff@id=0\relax
  \PackageError{flowfram}%
  {Can't find static frame id '#1'}{}%
\fi
}
\newcommand*\staticframex}[1]{%
  \csname @sf@\romannumeral#1@posx\endcsname
}
\newcommand*\staticframey}[1]{%
  \csname @sf@\romannumeral#1@posy\endcsname
}
\newcommand*\staticframeevenx}[1]{%
  \csname @sf@\romannumeral#1@evenx\endcsname
}
\newcommand*\staticframeeveny}[1]{%
  \csname @sf@\romannumeral#1@eveny\endcsname
}
\newcommand*\setallstaticframes}[1]{%
  \@colN=0\relax
  \whiledo{\@colN<\c@maxstatic}%
  {%
    \advance\@colN by 1\relax
    \@setstaticframe{\@colN}{#1}%
  }%
}
\newcommand*\setstaticframe{%
  \@ifstar\@ssetstaticframe\setstaticframe
}
\newcommand*\@ssetstaticframe}[2]{%
  \@for\ff@id:=#1\do
  {%
    \@staticframeid{\ff@id}%
    \@setstaticframe{\ff@id}{#2}%
  }%
}
\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
    }%
  }%
}%
}%
}
\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
\undef\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}{}%
}%
}
%newcommand*\simpar{\hfil\vadjust{\vskip\parskip}\break\indent}
\newcommand*\simpar{\hfill\\ \indent\mbox{}}
\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
}
\def\@ff@parshape{\parshape=0}
\newcommand*{\@ff@sectionhead}[1]{%
  \def\ff@sehead{#1}%
  \ffpshpar
  \@ifstar{\@s@ff@heading}{\@dblarg\@ff@heading}%
}
\def\@s@ff@heading#1{%
  \@ifundefined{\ff@old\ff@sehead}%
  {%
    \PackageError{flowfram}%
    {Unknown heading command '\ff@sehead'}{}%
  }%
  {%
    \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@sehead\endcsname*{%
      \@ff@parshape\flf@next #1}%
    \xdef\flf@next{%
      \@ff@parshape

```

```

        \prevgraf=\the\prevgraf}%
    \endgroup
}%
\mbox{} \flf@next
\let\flf@next\undefined
}
\def\@ff@heading[#1]#2{%
  \@ifundefined{\@ff@old\ff@sehead}%
  {%
    \PackageError{flowfram}%
    {Unknown heading command '\ff@sehead'}{}}%
  }%
  {%
    \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@sehead\endcsname[#1]{%
      \@ff@parshape\flf@next #2}%
    \xdef\flf@next{\@ff@parshape
      \prevgraf=\the\prevgraf}%
    \endgroup
  }%
  \mbox{} \flf@next
  \let\flf@next\undefined
}
\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}}%
}
\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
  }%
}%
}
\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-}%
  }%
}
\newbox\staticframe
\newenvironment{staticcontents}[1]{%
  \let\continueonframe=\@staticcontinueonframe
  \@beginstaticcontents{#1}%
}%
{%
  \@endstaticcontents
  \ignorespaces
}
\newenvironment{staticcontents*}[1]{%
  \@staticframeid{#1}%

```

```

\let\continueonframe=\@staticscontinueonframe
\@beginstaticcontents{\ff@id}%
}%
{%
\@endstaticcontents
\ignorespaces
}
\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
\edef\@sf@mpg{%
\noexpand
\begin{minipage}\csname @sf@dim@\romannumeral#1\endcsname
\noexpand\begin{group}
\noexpand\let\@ff@parshape=\@ff@parshape\relax
}%
\or
\edef\@sf@mpg{%
\noexpand
\begin{minipage}\csname @sf@dim@\romannumeral#1\endcsname
\@ff@parshape
\noexpand\begin{group}
\noexpand\let\@ff@parshape=\@ff@parshape\relax
\noexpand\let\@ff@setsecthead=\@ff@setsecthead
}%
\or
\edef\@sf@mpg{%
\noexpand
\begin{minipage}\csname @sf@dim@\romannumeral#1\endcsname
\noexpand\begin{group}
\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@tff@textcol\@ff@txtcol
\@sf@mpg
\setlength\@ff@parindent\@sf@parindent
}
\newcommand*{\@endstaticcontents}{%
\ifnum\ff@shape=2\relax

```

```

    \par
  \else
    \FLForgpar
  \fi
\endgroup
\end{minipage}%
\end{lrbox}%
\expandafter\global\expandafter
  \sbox\@sf@thisframe{\usebox\staticframe}%
}
\newcommand{\setstaticcontents}{%
  \ifstar\@sstaticconts\@staticconts
}
\newcommand{\@sstaticconts}[2]{%
  \begin{staticcontents*}{#1}%
    #2%
  \end{staticcontents*}%
}
\newcommand{\@staticconts}[2]{%
  \begin{staticcontents}{#1}%
    #2%
  \end{staticcontents}%
}
\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"}%
}
\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"}%
}
\newcommand*{\staticaddexclusion}[2]{%
  \ifcsemt{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"}%
}

```

```

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

```



```

{%
  \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}%
\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]}%
}
\def\@s@tdynamicframeid#1[#2]{%
  \edef\ff@label{#2}%
  \@df@checkuniqueidl{#1}{\ff@label}%
  \expandafter
    \xdef\csname @df@id@\romannumeral#1\endcsname{\ff@label}%
}
\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
}%
}
\newcommand*\getdynamiclabel}[1]{%
  \csname @df@id@\romannumeral#1\endcsname
}
\newcommand*\getdynamicid}[2]{%
  \@dynamicframeid{#2}\edef#1{\number\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
      \@colN=\c@maxdynamic
    }%
  }%
  \ifnum\ff@id=0\relax
  \PackageError{flowfram}%
  {Can't find dynamic frame id '#1'}{}%
  \fi
}

```

```

\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}%
  }%
}
\newcommand*{\dynamicframex}[1]{%
  \csname @df@romannumeral#1@posx\endcsname
}
\newcommand*{\dynamicframey}[1]{%
  \csname @df@romannumeral#1@posy\endcsname
}
\newcommand*{\dynamicframeevenx}[1]{%
  \csname @df@romannumeral#1@evenx\endcsname
}
\newcommand*{\dynamicframeeveny}[1]{%
  \csname @df@romannumeral#1@eveny\endcsname
}
\newcommand*{\setalldynamicframes}[1]{%
  \@colN=0\relax
  \whiledo{\@colN<\c@maxdynamic}%
  {%
    \advance\@colN by 1\relax
    \@setdynamicframe{\@colN}{#1}%
  }%
}
\newcommand*{\setdynamicframe}{%
  \@ifstar\@ssetdynamicframe\@setdynamicframe
}
\newcommand*{\@ssetdynamicframe}[2]{%
  \@for\@ff@id:=#1\do{%
    \@dynamicframeid{\@ff@id}%
    \@setdynamicframe{\@ff@id}{#2}%
  }%
}
\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
      }%
    }%
  }%
}
\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}%
}%
}
\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"}%
}
\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"}%
}
\newcommand*{\dynamicaddexclusion}[2]{%
    \ifcsempy{@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"}%
}
\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}%
}
\newcommand*{\dfswapoddeven}{%
\@ifstar\@sdynamicframeswapcoords\@dynamicframeswapcoords}
\newcommand*{\@sdynamicframeswapcoords}[1]{%
\@for\@ff@id:=#1\do{%
\@dynamicframeid{\@ff@id}%
\@@dynamicframeswapcoords{\@ff@id}}%
}
\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
    }%
  }%
}%
}%
}
\newenvironment{dynamiccontents}[1]{%
  \def\@flf@{dynamiccontents}%
  \xdynamiccontents{#1}%
  \endxdynamiccontents
}
\newtoks\@dynamicctok
\def\xdynamiccontents#1{%
  \def\@flf@idn{#1}%
  \@dynamicctok{}\@flf@get@body
}
\long\def\@flf@get@body#1\end{%
  \@flf@checkcontinued#1\continueonframe\@nil
  \ifdfcontinued
    \expandafter\flf@ta\expandafter{\@flf@tmpa}%
    \edef\@flf@tmp{\the\@dynamicctok\the\flf@ta}%
    \@dynamicctok\expandafter{\@flf@tmp}%
  \else
    \@dynamicctok\expandafter{\the\@dynamicctok#1}%
  \fi
  \@flf@find@end
}
\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\def\@lempty{}
\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}%
}
\def\@flf@find@end#1{%
  \def\@tempa{#1}%
  \global\let\flf@next=\relax
  \ifdfcontinued
    \@dynamictok\expandafter
      {\the\@dynamictok\ffcontinuedtextlayout}%
    \protected@edef\@tempa{\the\@dynamictok{\@ff@text}}%
    \@dynamictok\expandafter{\@tempa}%
    \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
}
\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}%
  \fi
}

```

```

\expandafter
\fi
\expandafter\end\expandafter{\@flf@}%
}
\newenvironment{dynamiccontents*}[1]{%
\def\@flf@{dynamiccontents*}%
\@dynamicframeid{#1}%
\xdynamiccontents{\ff@id}}{%
\enddynamiccontents
}
\newcommand{\setdynamiccontents}{%
\@ifstar\@ssetdynamiccontents\@setdynamiccontents
}
\newcommand{\@ssetdynamiccontents}[2]{%
\@dynamicframeid{#1}\@setdynamiccontents{\ff@id}{#2}%
}
\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
}
\newcommand{\appenddynamiccontents}{%
\@ifstar\@sappenddynamic\@appenddynamic
}
\newcommand{\@sappenddynamic}[2]{%
\@dynamicframeid{#1}\@appenddynamic{\ff@id}{#2}%
}
\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
}
\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}%
}
\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)%
}%
}
\newcommand*{\@staticscontinueonframe}[2][1]{%
\ffcontinuedtextlayout{#1}%
\end{staticcontents*}%
\begin{staticcontents*}{#2}\par\noindent\ignorespaces
}
\newcommand*{\@staticcontinueonframe}[2][1]{%
\ffcontinuedtextlayout{#1}%
\end{staticcontents*}%
\begin{staticcontents*}{#2}\par\noindent\ignorespaces
}
\newcommand{\ffcontinuedtextlayout}[1]{%
\parfillskip=0pt\par\hfill
\ffcontinuedtextfont{#1}%
}
\newcommand*{\ffcontinuedtextfont}[1]{\emph{\small #1}}
\newcommand*{\computeleftedgeodd}[1]{%
\setlength{#1}{-1in}%
\addtolength{#1}{-\hoffset}%
\addtolength{#1}{-\oddsidemargin}%
}
\newcommand*{\computeleftedgeeven}[1]{%
\setlength{#1}{-1in}%
\addtolength{#1}{-\hoffset}%
\addtolength{#1}{-\evensidemargin}%
}
\newcommand*{\computetopedge}[1]{%
\setlength{#1}{\textheight}%
\addtolength{#1}{\headheight}%
\addtolength{#1}{\headsep}%
\addtolength{#1}{1in}%
\addtolength{#1}{\voffset}%
\addtolength{#1}{\topmargin}%
}
\newcommand*{\computebottomedge}[1]{%

```

```

\computetopedge{#1}%
\addtolength{#1}{-\paperheight}%
}
\newcommand*\computerightedgeodd}[1]{%
\computeleftedgeodd{#1}%
\addtolength{#1}{\paperwidth}%
}
\newcommand*\computerightedgeeven}[1]{%
\computeleftedgeeven{#1}%
\addtolength{#1}{\paperwidth}%
}
\newlength\ffareawidth
\newlength\ffareaheight
\newlength\ffareax
\newlength\ffareay
\newlength\ffareaevenx
\newlength\ffareaeveny
\newcommand*\computeflowframearea){%
@ifstar\@scomputeffarea\@computeffarea
}
\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}%
\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}%
}
\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}%
}
\newcommand*{\@ff@swaplen}[2]{%
  \setlength{\@ff@tmp@x}{#1}%
  \setlength{#1}{#2}%
  \setlength{#2}{\@ff@tmp@x}%
}
\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

```

```

\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{\ffareaawidth}{\flowframewidth{#2}}%
  \setlength{\ffareaaheight}{\flowframeheight{#2}}%
\fi
\or
\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{\ffareaawidth}{\@ff@tmp@x}%
  \setlength{\ffareaaheight}{\@ff@tmp@y}%
\fi
\or
\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{\ffareaawidth}{\@ff@tmp@x}%
  \setlength{\ffareaaheight}{\@ff@tmp@y}%
\fi
\else
  \PackageError{flowfram}%
  {Unknown frame ID type '#1'}%

```



```

    {%
      Frame ID types are: 1 (flow), 2 (static) and 3 (dynamic)%
    }%
  \fi
}
\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
    \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
    \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
    \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
}
\newcommand*\getstaticbounds{%
\@ifstar\@sgetstaticbounds\@getstaticbounds
}
\newcommand*\@sgetstaticbounds[1]{%
\@staticframeid{#1}\@getstaticbounds{\ff@id}%
}
\newcommand*\@getstaticbounds[1]{\@ff@getdim{2}{#1}}
\newcommand*\getstaticevenbounds{%
\@ifstar\@sgetstaticevenbounds\@getstaticevenbounds
}
\newcommand*\@sgetstaticevenbounds[1]{%
\@staticframeid{#1}\@getstaticevenbounds{\ff@id}%
}
\newcommand*\@getstaticevenbounds[1]{\@ff@getevendim{2}{#1}}
\newcommand*\getflowbounds{%
\@ifstar\@sgetflowbounds\@getflowbounds
}
\newcommand*\@sgetflowbounds[1]{%
\@flowframeid{#1}\@getflowbounds{\ff@id}%
}
\newcommand*\@getflowbounds[1]{\@ff@getdim{1}{#1}}
\newcommand*\getflowevenbounds{%
\@ifstar\@sgetflowevenbounds\@getflowevenbounds
}
\newcommand*\@sgetflowevenbounds[1]{%
\@flowframeid{#1}\@getflowevenbounds{\ff@id}%
}
\newcommand*\@getflowevenbounds[1]{\@ff@getevendim{1}{#1}}

```

```

\newcommand*\getdynamicbounds}{%
  \@ifstar\@sgetdynamicbounds\@getdynamicbounds
}
\newcommand*\@sgetdynamicbounds}[1]{%
  \@dynamicframeid{#1}\@getdynamicbounds{\ff@id}%
}
\newcommand*\@getdynamicbounds}[1]{\@ff@getdim{3}{#1}}
\newcommand*\getdynamicsevenbounds}{%
  \@ifstar\@sgetdynamicsevenbounds\@getdynamicsevenbounds
}
\newcommand*\@sgetdynamicsevenbounds}[1]{%
  \@dynamicframeid{#1}\@getdynamicsevenbounds{\ff@id}%
}
\newcommand*\@getdynamicsevenbounds}[1]{\@ff@getevendim{3}{#1}}
\newif\ifFLFAbove
\newif\ifFLFbelow
\newif\ifFLFleft
\newif\ifFLFright
\newcommand*\checkifframeabove}{%
  \@ifstar\@scheckifframeabove\@checkifframeabove
}
\newcommand*\@scheckifframeabove}[4]{%
  \ifodd\c@page
    \@soddcheckifframeabove{#1}{#2}{#3}{#4}%
  \else
    \@sevencheckifframeabove{#1}{#2}{#3}{#4}%
  \fi
}
\newcommand*\@checkifframeabove}[4]{%
  \ifodd\c@page
    \@oddcheckifframeabove{#1}{#2}{#3}{#4}%
  \else
    \@evencheckifframeabove{#1}{#2}{#3}{#4}%
  \fi
}
\newcommand*\@oddcheckifframeabove}{%
  \@ifstar\@soddcheckifframeabove\@oddcheckifframeabove
}
\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
  \FLFabove>true
\else
  \FLFabove>false
\fi
}
\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
    \FLFabove>true
  \else
    \FLFabove>false
  \fi
}
\newcommand*{\checkifframebelow}{%
  \@ifstar\@scheckifframebelow\@checkifframebelow

```

```

}
\newcommand*{\@scheckifframebelow}[4]{%
  \ifodd\c@page
    \@soddcheckifframebelow{#1}{#2}{#3}{#4}%
  \else
    \@sevencheckifframebelow{#1}{#2}{#3}{#4}%
  \fi
}
\newcommand*{\@checkifframebelow}[4]{%
  \ifodd\c@page
    \@oddcheckifframebelow{#1}{#2}{#3}{#4}%
  \else
    \@evencheckifframebelow{#1}{#2}{#3}{#4}%
  \fi
}
\newcommand*{\oddcheckifframebelow}{%
  \@ifstar\@soddcheckifframebelow\@oddcheckifframebelow
}
\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
}
\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
}
}
\newcommand*{\checkifframeleft}{%
  \@ifstar\@scheckifframeleft\@checkifframeleft
}
\newcommand*{\@scheckifframeleft}[4]{%
  \ifodd\c@page
    \@soddcheckifframeleft{#1}{#2}{#3}{#4}%
  \else
    \@sevencheckifframeleft{#1}{#2}{#3}{#4}%
  \fi
}
\newcommand*{\@checkifframeleft}[4]{%
  \ifodd\c@page
    \@oddcheckifframeleft{#1}{#2}{#3}{#4}%
  \else
    \@evencheckifframeleft{#1}{#2}{#3}{#4}%
  \fi
}
\newcommand*{\oddcheckifframeleft}{%
  \@ifstar\@soddcheckifframeleft\@oddcheckifframeleft
}

```

```

\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
    \FLlefttrue
  \else
    \FLleftfalse
  \fi
}
\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
    \FLlefttrue
\else
    \FLleftfalse
\fi
}
\newcommand*\checkifframeright}{%
    \@ifstar\@scheckifframeright\@checkifframeright
}
\newcommand*\@scheckifframeright}[4]{%
    \ifodd\c@page
        \@soddcheckifframeright{#1}{#2}{#3}{#4}%
    \else
        \@sevencheckifframeright{#1}{#2}{#3}{#4}%
    \fi
}
\newcommand*\@checkifframeright}[4]{%
    \ifodd\c@page
        \@oddcheckifframeright{#1}{#2}{#3}{#4}%
    \else
        \@evencheckifframeright{#1}{#2}{#3}{#4}%
    \fi
}
\newcommand*\oddcheckifframeright}{%
    \@ifstar\@soddcheckifframeright\@oddcheckifframeright
}
\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
\FLFrightright
\else
\FLFrightrightfalse
\fi
}
\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
\FLFrightright
\else
\FLFrightrightfalse
\fi
}
\newcommand*{\@evencheckifframeabove}{%
\@ifstar\@sevencheckifframeabove\@evencheckifframeabove
}
\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
  \FLFabovevtrue
\else
  \FLFabovevfalse
\fi
}
\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
  \FLFabove>true
\else
  \FLFabove>false
\fi
}
\newcommand*\@evencheckifframebelow{%
  \@ifstar\@sevencheckifframebelow\@evencheckifframebelow
}
\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
  \FLFbelow>true
\else
  \FLFbelow>false
\fi
}
\newcommand*\@sevencheckifframebelow[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
}
\newcommand*\@evencheckifframeleft{%
  \@ifstar\@sevencheckifframeleft\@evencheckifframeleft
}
\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#3sevenbounds\endcsname{#4}%
\expandafter\ifdim\@ff@check<\ffareax
  \FLlefttrue
\else
  \FLleftfalse
\fi
}
\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#3sevenbounds}%
{%
  \PackageError{flowfram}%
  {Unknown frame type '#3'}%
  {%
    Frame types may only be one of: static, dynamic or
    flow%
  }%
}%
}%
\csname @get#3sevenbounds\endcsname{#4}%
\expandafter\ifdim\@ff@check<\ffareax
  \FLlefttrue
\else
  \FLleftfalse
\fi
}
\newcommand*\@evencheckifframeright}{%
  \@ifstar\@sevencheckifframeright\@evencheckifframeright
}
\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
  \FLFrightright
\else
  \FLFrightrightfalse
\fi
}
\newcommand*{\@evencheckiframeright}[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
  \FLFrightright
\else

```

```

        \FLFrighfalse
    \fi
}
\newcommand*\FFaboveleft{above left}
\newcommand*\FFaboveright{above right}
\newcommand*\FFbelowleft{below left}
\newcommand*\FFbelowright{below right}
\newcommand*\FFleft{on the left}
\newcommand*\FFright{on the right}
\newcommand*\FFabove{above}
\newcommand*\FFbelow{below}
\newcommand*\FFoverlap{overlap}
\DeclareRobustCommand*\relativeframelocation{%
    \@ifstar\@srelativeframelocation\@relativeframelocation
}
\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
            \ifFLFrigh
                \FFaboveright
            \else
                \FFabove
            \fi
        \fi
    \else
        \ifFLFbelow
            \ifFLFleft
                \FFbelowleft
            \else
                \ifFLFrigh
                    \FFbelowright
                \else
                    \FFbelow
                \fi
            \fi
        \else
            \ifFLFleft
                \FFleft
            \else
                \ifFLFrigh
                    \FFright
                \else
                    \FFoverlap
                \fi
            \fi
        \fi
    \fi
}

```

```

        \fi
    \fi
\fi
}
\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
}
\DeclareRobustCommand*\reldynamicloc{%
  \ifstar\@sreldynamicloc\@reldynamicloc
}
\newcommand*{\@sreldynamicloc}[2]{%
  \@srelativeframelocation{dynamic}{#1}{dynamic}{#2}%
}
\newcommand*{\@reldynamicloc}[2]{%

```



```

    \@relativeframelocation{dynamic}{#1}{dynamic}{#2}%
  }
\DeclareRobustCommand*\relstaticloc{%
  \ifstar\@srelstaticloc\@relstaticloc
}
\newcommand*\@srelstaticloc[2]{%
  \@srelativeframelocation{static}{#1}{static}{#2}%
}
\newcommand*\@relstaticloc[2]{%
  \@relativeframelocation{static}{#1}{static}{#2}%
}
\DeclareRobustCommand*\relflowloc{%
  \ifstar\@srelflowloc\@relflowloc
}
\newcommand*\@srelflowloc[2]{%
  \@srelativeframelocation{flow}{#1}{flow}{#2}%
}
\newcommand*\@relflowloc[2]{%
  \@relativeframelocation{flow}{#1}{flow}{#2}%
}
\newcommand*\setinitialframe[1]{%
  \c@thisframe=#1%
  \global\usedframebreaktrue
  \global\setlength{\hsize}
  {%
    \csname colwidth\romannumeral\c@thisframe\endcsname
  }%
}
\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}%
    \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@mestru
    \edef\ff@txtcol{%
    \csname @ff@txtcol@\romannumeral\c@thisframe\endcsname}%
    \@s@tfftextcol
}
\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>Opt\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
}
\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>Opt\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
}
\AtBeginDocument{%
  \c@absolutepage=1\relax
  \ifnum\c@maxflow=0\relax
    \PackageWarning{flowfram}{No flow frames, adding one}%
    \@onecolumn
  \fi
}

```

```

\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%
}%
}
\newlength\fftolerance
\setlength\fftolerance{2pt}
\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\vsname colheight\romannumeral#1\endcsname
    \global\@colht\vsname
    \global\@colroom\@colht
  \fi
}

```

```

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

```

```

\fi
\@makefcolumn\@deferlist
\@whilesw \if@colmade \fi
{%
  \@opcol
  \@makefcolumn\@deferlist
}%
\if@firstcolumn
\edef\@dbldeferlist{\@dbltoplist\@dbldeferlist}%
\global\let\@dbltoplist\@empty
\global\@colht\vsiz
\begingroup
\@dblfloatplacement
\@makefcolumn\@dbldeferlist
\@whilesw \if@colmade \fi
{%
  \@outputpage
  \@makefcolumn\@dbldeferlist
}%
\endgroup
\else
\ vbox{}%
\clearpage
\fi
\else
\setbox\@cclv\ vbox{\ box\@cclv\ vfil}%
\@makecol\@opcol
\clearpage
\fi
}
\newcommand{\@dothehead}{%
\ vbox to \headheight
{%
\color@hbox\normalcolor\hbox to \textwidth{\@thehead}%
\color@endbox
}%
}
\newcommand{\@dothefoot}{%
\color@hbox\normalcolor\hbox to \textwidth{\@thefoot}%
\color@endbox
}
\newcommand{\@dodynamicthead}{-}
\newcommand{\@dodynamictfoot}{-}
\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}%
\stepcounter{absolute page}%
\setcounter{displayed frame}{0}%
\let\firstmark\botmark
}
\newcommand*\makedfheaderfooter{%
  \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
  }%
}%
}
\@onlypreamble{\makedfheaderfooter}

```

```

\newcommand{\footnotecolor}{%
  \@ifundefined{ff@txtcol@romannumeral@c@thisframe}%
  {%
    \normalcolor
  }%
  {%
    \edef\ff@txtcol{%
      \csname @ff@txtcol@romannumeral@c@thisframe\endcsname
    }%
    \@stfftextcol
  }%
}
\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
}
\def\@opcol{%
  \@outputdblcol
  \global\@mparbottom\z@
  \global\@textfloatsheight\z@
  \@floatplacement
}
\newif\if@ff@moreframes
\newcommand*{\@ff@checkifmoreframes}{%

```

```

\@ff@moreframesfalse
\@colN=\c@thisframe
\whiledo{\@colN<\c@maxflow}%
{%
  \advance\@colN by 1\relax
  \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
\else
  \edef\ff@pages{\csname @ff@pages@\romannumeral\@colN\endcsname}%
  \@ff@checkpages{\ff@pages}%
  \if@ff@moreframes
    \@colN=\c@maxflow\relax
  \fi
\fi
}%
\if@ff@moreframes
\else
  \@ff@tmpN=\@ff@pages@countreg
  \count@=0\relax
  \loop
  \advance\@ff@tmpN by 1\relax
  \@colN=0\relax
  \whiledo{\@colN<\c@maxflow}%
  {%
    \advance\@colN by 1\relax
    \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
  \else
    \edef\ff@pages{\csname @ff@pages@\romannumeral\@colN\endcsname}%
    \@ff@checkpages[\@ff@tmpN]{\ff@pages}%
    \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
  }
\newcommand*\@ff@checkpages}[2][\@ff@pages@countreg]{%
  \@for\@ff@pp:=#2\do{%
    \@ff@checkthispage{#1}{\@ff@pp}%
  }%
}
\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}%
    }%
  }%
}
}
\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
}
\newcount\c@ffrangenum
\newcommand*\@ff@getrange#[1]{%
  \expandafter\@ff@@getrange#1-\relax\end
}
\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
}

```

```

}
\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}%
}
\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}%
}
\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
}
\def\@@ff@getrange#1-\relax\end{%
  \def\@ff@numend{#1}%
}
\newcommand*\@ff@output@adjustframes{}
\newcommand*\@ff@flowswitchonnext{%
  \@ifstar\@sfflowswitchonnext\@ff@flowswitchonnext
}
\newcommand{\@sfflowswitchonnext}[1]{%
  \@for\@ff@id:=#1\do{%
    \@ff@flowframeid{\@ff@id}%
    \@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
  }%
}
\newcommand{\@ff@flowswitchonnext}[1]{%
  \@for\@ff@id:=#1\do{%

```

```

\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
}%
}
\newcommand*{\flowswitchonnextodd}{%
  \@ifstar\@sflowswitchonnextodd\flowswitchonnextodd
}
\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}%
    \ff@chckifthispg{\ff@pages@countreg}{\ff@id}%
    \def\ff@prepages{}%
    \if@notthiscol
      \else
        \def\ff@prepages{\number\ff@pages@countreg,}%
      \fi
      \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@}%
          }%
        }%
      }%
    }
\newcommand{\@flowswitchonnextodd}[1]{%
  \count@=\ff@pages@countreg\relax

```

```

\ifodd\count@\relax
  \advance\count@ by 1\relax
\fi
\@for\@ff@id:=#1\do{%
  \@ff@chckifthispg{\@ff@pages@countreg}{\@ff@id}%
  \def\@ff@prepages{}%
  \if@notthiscol
  \else
    \def\@ff@prepages{\number\@ff@pages@countreg,}%
  \fi
  \@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@}%
  }%
}%
}
\newcommand*\@flowswitchoffnext{%
  \@ifstar\@sflowswitchoffnext\@flowswitchoffnext
}
\newcommand{\@sflowswitchoffnext}[1]{%
  \@for\@ff@id:=#1\do{%
    \@flowframeid{\@ff@id}%
    \@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}%
    }%
  }%
}%
}
\newcommand{\@flowswitchoffnext}[1]{%
  \@for\@ff@id:=#1\do{%
    \@ff@chckifthispg{\@ff@pages@countreg}{\@ff@id}%
    \if@notthiscol
      \def\@ff@pages{none}%
    \fi
  }%
}

```

```

\else
  \def\@ff@pages{\number\@ff@pages@countreg}%
\fi
\expandafter\toks@\expandafter{\@ff@output@adjustframes}%
\edef\@ff@output@adjustframes{%
  \the\toks@
  \noexpand\flowsetpagelist{\number\@ff@id}{\@ff@pages}%
}%
}%
}
\newcommand*\@flowswitchoffnextodd{%
  \@ifstar\@sflowswitchoffnextodd\@flowswitchoffnextodd
}
\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}%
    \@ff@chckifthispg{\@ff@pages@countreg}{\ff@id}%
    \if@notthiscol
      \ifnum\@ff@pages@countreg=\count@\relax
        \def\@ff@nextpages{none}%
      \else
        \@ff@chckifthispg{\count@}{\ff@id}%
        \if@notthiscol
          \def\@ff@nextpages{none}%
        \else
          \def\@ff@nextpages{\number\count@}%
        \fi
      \fi
    \else
      \ifnum\@ff@pages@countreg=\count@\relax
        \def\@ff@nextpages{\number\@ff@pages@countreg}%
      \else
        \@ff@chckifthispg{\count@}{\ff@id}%
        \if@notthiscol
          \def\@ff@nextpages{\number\@ff@pages@countreg}%
        \else
          \def\@ff@nextpages{\number\@ff@pages@countreg,\number\count@}%
        \fi
      \fi
    \fi
  \fi
  \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
  \edef\@ff@output@adjustframes{%
    \the\toks@
    \noexpand\flowsetpagelist{\number\ff@id}{\@ff@nextpages}%
  }%
}%
}

```

```

}
\newcommand{\@flowswitchoffnextodd}[1]{%
  \count@=\@ff@pages@countreg\relax
  \ifodd\@ff@pages@countreg\relax
    \advance\count@ by 1\relax
  \fi
  \@for\@ff@id:=#1\do{%
    \@ff@chckifthispg{\@ff@pages@countreg}{\@ff@id}%
    \ifnotthiscol
      \ifnum\@ff@pages@countreg=\count@\relax
        \def\@ff@nextpages{none}%
      \else
        \@ff@chckifthispg{\count@}{\@ff@id}%
        \ifnotthiscol
          \def\@ff@nextpages{none}%
        \else
          \def\@ff@nextpages{\number\count@}%
        \fi
      \fi
    \else
      \ifnum\@ff@pages@countreg=\count@\relax
        \def\@ff@nextpages{\number\@ff@pages@countreg}%
      \else
        \@ff@chckifthispg{\count@}{\@ff@id}%
        \ifnotthiscol
          \def\@ff@nextpages{\number\@ff@pages@countreg}%
        \else
          \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}%
  }%
}%
}
\newcommand*{\flowswitchonnextonly}{%
  \@ifstar\@sflowswitchonnextonly\@flowswitchonnextonly
}
\newcommand{\@sflowswitchonnextonly}[1]{%
  \count@=\@ff@pages@countreg\relax
  \advance\count@ by 1\relax
  \@for\@ff@id:=#1\do{%
    \@flowframeid{\@ff@id}%
    \@ff@chckifthispg{\@ff@pages@countreg}{\@ff@id}%
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \ifnotthiscol
      \xdef\@ff@output@adjustframes{%

```

```

        \the\toks@
        \noexpand\flowsetpagelist{\number\ff@id}{\number\count@}%
    }%
\else
    \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\flowsetpagelist{\number\ff@id}%
        {\number\@ff@pages@countreg,\number\count@}%
    }%
\fi
}%
}
\newcommand{\@flowswitchonnextonly}[1]{%
    \count@=\@ff@pages@countreg\relax
    \advance\count@ by 1\relax
    \@for\@ff@id:=#1\do{%
        \@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\count@}%
            }%
        \else
            \xdef\@ff@output@adjustframes{%
                \the\toks@
                \noexpand\flowsetpagelist{\number\@ff@id}%
                {\number\@ff@pages@countreg,\number\count@}%
            }%
        \fi
    }%
}
\newcommand*\@flowswitchonnextoddonly{%
    \@ifstar\@sflowswitchonnextoddonly\@flowswitchonnextoddonly
}
\newcommand{\@sflowswitchonnextoddonly}[1]{%
    \@for\@ff@id:=#1\do{%
        \@flowframeid{\@ff@id}%
        \@ff@chckifthispg{\@ff@pages@countreg}{\@ff@id}%
        \if@notthiscol
            \ifodd\@ff@pages@countreg
                \count@=\@ff@pages@countreg\relax
                \advance\count@ by 1\relax
                \@ff@chckifthispg{\count@}{\@ff@id}%
            \if@notthiscol
                \advance\count@ by 1\relax
            \edef\@ff@pages{\number\count@}%
        \else
            \edef\@ff@pages{\number\count@}%
            \advance\count@ by 1\relax
        \fi
    }%
}

```

```

        \edef\@ff@pages{\@ff@pages, \number\count@}%
        \fi
    \else
        \count@=\@ff@pages@countreg\relax
        \advance\count@ by 1\relax
        \edef\@ff@pages{\number\count@}%
        \fi
    \else
        \ifodd\@ff@pages@countreg
            \count@=\@ff@pages@countreg\relax
            \advance\count@ by 1\relax
            \@ff@chckifthispg{\count@}{\ff@id}%
            \if@notthiscol
                \advance\count@ by 1\relax
                \edef\@ff@pages{\number\@ff@pages@countreg, \number\count@}%
            \else
                \advance\count@ by 1\relax
                \edef\@ff@pages{\number\@ff@pages@countreg-\number\count@}%
            \fi
        \else
            \count@=\@ff@pages@countreg\relax
            \advance\count@ by 1\relax
            \edef\@ff@pages{\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@pages}%
    }%
}
}
\newcommand{\@flowswitchonnextodonly}[1]{%
    \@for\@ff@id=#1\do{%
        \@ff@chckifthispg{\@ff@pages@countreg}{\@ff@id}%
        \if@notthiscol
            \ifodd\@ff@pages@countreg
                \count@=\@ff@pages@countreg\relax
                \advance\count@ by 1\relax
                \@ff@chckifthispg{\count@}{\@ff@id}%
                \if@notthiscol
                    \advance\count@ by 1\relax
                    \edef\@ff@pages{\number\count@}%
                \else
                    \edef\@ff@pages{\number\count@}%
                    \advance\count@ by 1\relax
                    \edef\@ff@pages{\@ff@pages, \number\count@}%
                \fi
            \else
                \count@=\@ff@pages@countreg\relax

```



```

        \advance\count@ by 1\relax
        \edef\@ff@pages{\number\count@}%
    \fi
\else
\ifodd\@ff@pages@countreg
    \count@=\@ff@pages@countreg\relax
    \advance\count@ by 1\relax
    \@ff@chckifthispg{\count@}{\@ff@id}%
    \if@notthiscol
        \advance\count@ by 1\relax
        \edef\@ff@pages{\number\@ff@pages@countreg,\number\count@}%
    \else
        \advance\count@ by 1\relax
        \edef\@ff@pages{\number\@ff@pages@countreg-\number\count@}%
    \fi
\else
    \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}%
}%
}%
}
\newcommand*{\flowswitchoffnextonly}{%
    \ifstar\@sflowswitchoffnextonly\@flowswitchoffnextonly
}
\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\flowaddeclusion{\number\@ff@id}{\number\count@}%
        }%
    }%
}
}
\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\flowaddeclusion{\number\ff@id}{\number\count@}%
    }%
}
}
\newcommand*\flowswitchoffnextodonly}{%
  \ifstar\@sflowswitchoffnextodonly\@flowswitchoffnextodonly
}
\newcommand{\@sflowswitchoffnextodonly}[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\flowaddeclusion{\number\ff@id}{\number\count@}%
    }%
  }%
}
\newcommand{\@flowswitchoffnextodonly}[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\flowaddeclusion{\number\ff@id}{\number\count@}%
    }%
  }%
}
}
\newcommand*\@dynamicswitchonnext}{%
  \ifstar\@sdynamicswitchonnext\@dynamicswitchonnext
}
\newcommand{\@sdynamicswitchonnext}[1]{%
  \@for\ff@id:=#1\do{%
    \@dynamicframeid{\ff@id}%
    \@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}%
      }%
    \fi
  }%
}

```

```

    }%
  \else
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\dynamicsetpagelist{\number\ff@id}%
      {\number\@ff@pages@countreg,>\number\@ff@pages@countreg}%
    }%
  \fi
}
}
\newcommand{\@dynamicswitchonnext}[1]{%
  \@for\@ff@id:=#1\do{%
    \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
  }%
}
\newcommand*{\dynamicswitchonnextodd}{%
  \@ifstar\@sdynamicswitchonnextodd\@dynamicswitchonnextodd
}
\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}%
    \df@chckifthispg[\@ff@pages@countreg]{\ff@id}%
    \def\@ff@prepages{}%
    \if@notthiscol
      \else
        \def\@ff@prepages{\number\@ff@pages@countreg,}%
      \fi
    \df@chckifthispg[\count@]{\ff@id}%
    \ifnum\count@=\@ff@pages@countreg\relax
      \else
        \if@notthiscol
          \else
            \edef\@ff@prepages{\@ff@prepages\number\count@,}%
          \fi
        \fi
      \fi
    \fi
  }%
}

```

```

    \fi
  \fi
  \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
  \xdef\@ff@output@adjustframes{%
    \the\toks@
    \noexpand\dynamicsetpagelist{\number\ff@id}%
    {\@ff@prepages>\number\count@}%
  }%
}%
}
}
\newcommand{\@dynamicswitchonnextodd}[1]{%
  \count@=\@ff@pages@countreg\relax
  \ifodd\count@\relax
    \advance\count@ by 1\relax
  \fi
  \@for\ff@id:=#1\do{%
    \@df@chckifthispg[\@ff@pages@countreg]{\ff@id}%
    \def\@ff@prepages{}%
    \if@notthiscol
      \else
        \def\@ff@prepages{\number\@ff@pages@countreg,}%
      \fi
    \@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@}%
      }%
    }%
  }%
}
}
\newcommand*\@dynamicswitchoffnext}{%
  \ifstar\@dynamicswitchoffnext\@dynamicswitchoffnext
}
}
\newcommand{\@dynamicswitchoffnext}[1]{%
  \@for\ff@id:=#1\do{%
    \@dynamicframeid{\ff@id}%
    \@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}%
}%
}%
}
\newcommand{\@dynamicsswitchoffnext}[1]{%
  \@for\@ff@id:=#1\do{%
    \@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}%
    }%
  }%
}
\newcommand*\@dynamicsswitchoffnextodd{%
  \@ifstar\@sdynamicsswitchoffnextodd\@dynamicsswitchoffnextodd
}
\newcommand{\@sdynamicsswitchoffnextodd}[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}%
    \@df@chckifthispg[\@ff@pages@countreg]{\ff@id}%
    \if@notthiscol
      \ifnum\@ff@pages@countreg=\count@\relax
        \def\@ff@nextpages{none}%
      \else
        \@df@chckifthispg[\count@]{\ff@id}%
        \if@notthiscol
          \def\@ff@nextpages{none}%
        \else
          \def\@ff@nextpages{\number\count@}%
        \fi
      \fi
    \else
      \ifnum\@ff@pages@countreg=\count@\relax
        \def\@ff@nextpages{\number\@ff@pages@countreg}%
      \else
        \@df@chckifthispg[\count@]{\ff@id}%
        \if@notthiscol

```

```

        \def\@ff@nextpages{\number\@ff@pages@countreg}%
        \else
        \def\@ff@nextpages{\number\@ff@pages@countreg,\number\count@}%
        \fi
    \fi
\fi
\expandafter\toks@\expandafter{\@ff@output@adjustframes}%
\edef\@ff@output@adjustframes{%
    \the\toks@
    \noexpand\dynamicsetpagelist{\number\ff@id}{\@ff@nextpages}%
}%
}%
}
\newcommand{\@dynamicswitchoffnextodd}[1]{%
    \count@=\@ff@pages@countreg\relax
    \ifodd\@ff@pages@countreg\relax
        \advance\count@ by 1\relax
    \fi
    \@for\@ff@id:=#1\do{%
        \df@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
        \ifnotthiscol
            \ifnum\@ff@pages@countreg=\count@\relax
                \def\@ff@nextpages{none}%
            \else
                \df@chckifthispg[\count@]{\@ff@id}%
                \ifnotthiscol
                    \def\@ff@nextpages{none}%
                \else
                    \def\@ff@nextpages{\number\count@}%
                \fi
            \fi
        \else
            \ifnum\@ff@pages@countreg=\count@\relax
                \def\@ff@nextpages{\number\@ff@pages@countreg}%
            \else
                \df@chckifthispg[\count@]{\@ff@id}%
                \ifnotthiscol
                    \def\@ff@nextpages{\number\@ff@pages@countreg}%
                \else
                    \def\@ff@nextpages{\number\@ff@pages@countreg,\number\count@}%
                \fi
            \fi
        \fi
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \edef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\dynamicsetpagelist{\number\@ff@id}{\@ff@nextpages}%
    }%
}%
}

```

```

\newcommand*\dynamicswitchonnexonly}{%
  \ifstar\@dynamicswitchonnexonly\@dynamicswitchonnexonly
}
\newcommand{\@dynamicswitchonnexonly}[1]{%
  \count@=\@ff@pages@countreg\relax
  \advance\count@ by 1\relax
  \@for\@ff@id:=#1\do{%
    \@dynamicframeid{\@ff@id}%
    \@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\count@}%
      }%
    \else
      \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\dynamicsetpagelist{\number\@ff@id}%
        {\number\@ff@pages@countreg,\number\count@}%
      }%
    \fi
  }%
}
\newcommand{\@dynamicswitchonnexonly}[1]{%
  \count@=\@ff@pages@countreg\relax
  \advance\count@ by 1\relax
  \@for\@ff@id:=#1\do{%
    \@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\count@}%
      }%
    \else
      \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\dynamicsetpagelist{\number\@ff@id}%
        {\number\@ff@pages@countreg,\number\count@}%
      }%
    \fi
  }%
}
\newcommand*\dynamicswitchonnexoddonly}{%
  \ifstar\@dynamicswitchonnexoddonly\@dynamicswitchonnexoddonly
}
\newcommand{\@dynamicswitchonnexoddonly}[1]{%
  \@for\@ff@id:=#1\do{%
    \@dynamicframeid{\@ff@id}%

```

```

\df@chckifthispg[\@ff@pages@countreg]{\ff@id}%
\if@notthiscol
  \ifodd\@ff@pages@countreg
    \count@=\@ff@pages@countreg\relax
    \advance\count@ by 1\relax
    \df@chckifthispg[\count@]{\ff@id}%
    \if@notthiscol
      \advance\count@ by 1\relax
      \edef\@ff@pages{\number\count@}%
    \else
      \edef\@ff@pages{\number\count@}%
      \advance\count@ by 1\relax
      \edef\@ff@pages{\@ff@pages, \number\count@}%
    \fi
  \else
    \count@=\@ff@pages@countreg\relax
    \advance\count@ by 1\relax
    \edef\@ff@pages{\number\count@}%
  \fi
\else
  \ifodd\@ff@pages@countreg
    \count@=\@ff@pages@countreg\relax
    \advance\count@ by 1\relax
    \df@chckifthispg[\count@]{\ff@id}%
    \if@notthiscol
      \advance\count@ by 1\relax
      \edef\@ff@pages{\number\@ff@pages@countreg, \number\count@}%
    \else
      \advance\count@ by 1\relax
      \edef\@ff@pages{\number\@ff@pages@countreg- \number\count@}%
    \fi
  \else
    \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}%
\edef\@ff@output@adjustframes{%
  \the\toks@
  \noexpand\dynamicsetpagelist{\number\ff@id}{\@ff@pages}%
}%
}%
}
\newcommand{\@dynamicswitchonnextoddoneonly}[1]{%
  \@for\@ff@id:=#1\do{%
    \df@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
    \if@notthiscol
      \ifodd\@ff@pages@countreg
        \count@=\@ff@pages@countreg\relax

```



```

\advance\count@ by 1\relax
\@df@chckifthispg[\count@]{\@ff@id}%
\if@notthiscol
\advance\count@ by 1\relax
\edef\@ff@pages{\number\count@}%
\else
\edef\@ff@pages{\number\count@}%
\advance\count@ by 1\relax
\edef\@ff@pages{\@ff@pages,\number\count@}%
\fi
\else
\count@=\@ff@pages@countreg\relax
\advance\count@ by 1\relax
\edef\@ff@pages{\number\count@}%
\fi
\else
\ifodd\@ff@pages@countreg
\count@=\@ff@pages@countreg\relax
\advance\count@ by 1\relax
\@df@chckifthispg[\count@]{\@ff@id}%
\if@notthiscol
\advance\count@ by 1\relax
\edef\@ff@pages{\number\@ff@pages@countreg,\number\count@}%
\else
\advance\count@ by 1\relax
\edef\@ff@pages{\number\@ff@pages@countreg-\number\count@}%
\fi
\else
\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}%
}%
}%
}
\newcommand*\@dynamicswitchoffnextonly}{%
\ifstar\@sdynamicswitchoffnextonly\@dynamicswitchoffnextonly
}
\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@}%
    }%
}
}
\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@}%
        }%
    }%
}
\newcommand*{\dynamicswitchoffnextoddonly}{%
    \ifstar\@sdynamicswitchoffnextoddonly\@dynamicswitchoffnextoddonly
}
\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@}%
        }%
    }%
}
\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@}%
        }%
    }%
}
}

```

```

\newcommand*\staticswitchonnext}{%
  \ifstar\@sstaticswitchonnext\@staticswitchonnext
}
\newcommand{\@sstaticswitchonnext}[1]{%
  \for\@ff@id:=#1\do{%
    \@staticframeid{\@ff@id}%
    \@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
  }%
}
\newcommand{\@staticswitchonnext}[1]{%
  \for\@ff@id:=#1\do{%
    \@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
  }%
}
\newcommand*\staticswitchonnextodd}{%
  \ifstar\@sstaticswitchonnextodd\@staticswitchonnextodd
}
\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}%

```

```

\sf@chckifthispg[\@ff@pages@countreg]{\ff@id}%
\def\@ff@prepages{}%
\if@notthiscol
\else
\def\@ff@prepages{\number\@ff@pages@countreg,}%
\fi
\@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@}%
}%
}%
}
\newcommand{\@staticswitchonnextodd}[1]{%
\count@=\@ff@pages@countreg\relax
\ifodd\count@\relax
\advance\count@ by 1\relax
\fi
\@for\@ff@id:=#1\do{%
\@sf@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
\def\@ff@prepages{}%
\if@notthiscol
\else
\def\@ff@prepages{\number\@ff@pages@countreg,}%
\fi
\@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@}%
}%
}%
}

```

```

\newcommand*{\staticswitchoffnext}{%
  \@ifstar\@sstaticswitchoffnext\@staticswitchoffnext
}
\newcommand{\@sstaticswitchoffnext}[1]{%
  \@for\@ff@id:=#1\do{%
    \@staticframeid{\@ff@id}%
    \@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}%
    }%
  }%
}
\newcommand{\@staticswitchoffnext}[1]{%
  \@for\@ff@id:=#1\do{%
    \@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}%
    }%
  }%
}
\newcommand*{\staticswitchoffnextodd}{%
  \@ifstar\@sstaticswitchoffnextodd\@staticswitchoffnextodd
}
\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}%
    \@sf@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
    \if@notthiscol
      \ifnum\@ff@pages@countreg=\count@\relax
        \def\@ff@nextpages{none}%
      \else
        \@sf@chckifthispg[\count@]{\@ff@id}%
      \fi
    \fi
  }%
}

```

```

        \if@notthiscol
        \def\@ff@nextpages{none}%
        \else
        \def\@ff@nextpages{\number\count@}%
        \fi
    \fi
\else
\ifnum\@ff@pages@countreg=\count@\relax
\def\@ff@nextpages{\number\@ff@pages@countreg}%
\else
\@sf@chckifthispg[\count@]{\ff@id}%
\if@notthiscol
\def\@ff@nextpages{\number\@ff@pages@countreg}%
\else
\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}%
}%
}%
}
\newcommand{\@staticswitchoffnextodd}[1]{%
\count@=\@ff@pages@countreg\relax
\ifodd\@ff@pages@countreg\relax
\advance\count@ by 1\relax
\fi
\@for\@ff@id:=#1\do{%
\@sf@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
\if@notthiscol
\ifnum\@ff@pages@countreg=\count@\relax
\def\@ff@nextpages{none}%
\else
\@sf@chckifthispg[\count@]{\@ff@id}%
\if@notthiscol
\def\@ff@nextpages{none}%
\else
\def\@ff@nextpages{\number\count@}%
\fi
\fi
\fi
\else
\ifnum\@ff@pages@countreg=\count@\relax
\def\@ff@nextpages{\number\@ff@pages@countreg}%
\else
\@sf@chckifthispg[\count@]{\@ff@id}%
\if@notthiscol
\def\@ff@nextpages{\number\@ff@pages@countreg}%

```

```

        \else
        \def\@ff@nextpages{\number\@ff@pages@countreg,\number\count@}%
        \fi
    \fi
\expandafter\toks@\expandafter{\@ff@output@adjustframes}%
\edef\@ff@output@adjustframes{%
    \the\toks@
    \noexpand\staticsetpagelist{\number\@ff@id}{\@ff@nextpages}%
}%
}%
}
\newcommand*\@staticswitchonnextonly{%
    \@ifstar\@sstaticswitchonnextonly\@staticswitchonnextonly
}
\newcommand{\@sstaticswitchonnextonly}[1]{%
    \count@=\@ff@pages@countreg\relax
    \advance\count@ by 1\relax
    \@for\@ff@id:=#1\do{%
        \@staticframeid{\@ff@id}%
        \@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\count@}%
            }%
        \else
            \xdef\@ff@output@adjustframes{%
                \the\toks@
                \noexpand\staticsetpagelist{\number\@ff@id}%
                {\number\@ff@pages@countreg,\number\count@}%
            }%
        \fi
    }%
}
\newcommand{\@staticswitchonnextonly}[1]{%
    \count@=\@ff@pages@countreg\relax
    \advance\count@ by 1\relax
    \@for\@ff@id:=#1\do{%
        \@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\count@}%
            }%
        \else
            \xdef\@ff@output@adjustframes{%
                \the\toks@

```



```

\edef\@ff@output@adjustframes{%
  \the\toks@
  \noexpand\staticsetpagelist{\number\ff@id}{\@ff@pages}%
}%
}%
}
\newcommand{\@staticswitchonnextoddone}[1]{%
\@for\@ff@id:=#1\do{%
  \@sf@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
  \if@notthiscol
    \ifodd\@ff@pages@countreg
      \count@=\@ff@pages@countreg\relax
      \advance\count@ by 1\relax
      \@sf@chckifthispg[\count@]{\@ff@id}%
      \if@notthiscol
        \advance\count@ by 1\relax
        \edef\@ff@pages{\number\count@}%
      \else
        \edef\@ff@pages{\number\count@}%
        \advance\count@ by 1\relax
        \edef\@ff@pages{\@ff@pages,\number\count@}%
      \fi
    \else
      \count@=\@ff@pages@countreg\relax
      \advance\count@ by 1\relax
      \edef\@ff@pages{\number\count@}%
    \fi
  \else
    \ifodd\@ff@pages@countreg
      \count@=\@ff@pages@countreg\relax
      \advance\count@ by 1\relax
      \@sf@chckifthispg[\count@]{\@ff@id}%
      \if@notthiscol
        \advance\count@ by 1\relax
        \edef\@ff@pages{\number\@ff@pages@countreg,\number\count@}%
      \else
        \advance\count@ by 1\relax
        \edef\@ff@pages{\number\@ff@pages@countreg-\number\count@}%
      \fi
    \else
      \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}%
\edef\@ff@output@adjustframes{%
  \the\toks@
  \noexpand\staticsetpagelist{\number\@ff@id}{\@ff@pages}%
}%
}

```

```

    }%
  }
  \newcommand*{\staticswitchoffnextonly}{%
    \ifstar\@sstaticswitchoffnextonly\@staticswitchoffnextonly
  }
  \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@}%
      }%
    }%
  }%
}
\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@}%
    }%
  }%
}
}
\newcommand*{\staticswitchoffnextoddonly}{%
  \ifstar\@sstaticswitchoffnextoddonly\@staticswitchoffnextoddonly
}
\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@}%
    }%
  }%
}
}
\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@}%
  }%
}%
}
\newcommand*\ffaddtoadjustframeshook}[1]{%
  \@ff@addtolist\@ff@output@adjustframes\entry{#1}%
}
\newif\if@notthiscol
\newif\if@ff@nwpg
\newcount\c@curpg
\newcommand*\@g@tnextcol}[1]{%
  \@ff@output@adjustframes
  \global\let\@ff@output@adjustframes\@empty
  \@ff@checkifmoreframes
  \if@ff@moreframes
  \else
    \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
  \@colN=1\relax
  \@ff@nwpgtrue
  \advance\c@curpg by 1\relax
\else
  \advance\@colN by 1\relax
\fi
\@ff@chckifthispg{\c@curpg}{\@colN}%
\if@notthiscol
\repeat
#1=\@colN\relax
}
\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
}
\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
        }%
        {%
          \@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
        }%
        }%
        }%
        }%
        }%
        }%
    }
    \newcommand*{\@sf@chckifthispg}[2][\@ff@pages@countreg]{%
        \@notthiscoltrue
        \edef\ff@pages{\csname @sf@pages@\romannumeral#2\endcsname}%
        \@ff@chckifthispg{#1}%
    }
    \newcommand*{\@df@chckifthispg}[2][\@ff@pages@countreg]{%
        \@notthiscoltrue
        \edef\ff@pages{\csname @df@pages@\romannumeral#2\endcsname}%
        \@ff@chckifthispg{#1}%
    }
    \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
    }
    \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}%
        \edef\ff@col{\csname @ff@col@\romannumeral#1\endcsname}%
        \edef\ff@txtcol{\csname @ff@txtcol@\romannumeral#1\endcsname}%
        \edef\ff@backcol{\csname @ff@backcol@\romannumeral#1\endcsname}%
        \@ff@setoffset{#1}%
        \rotateframe{\csname @ff@angle@\romannumeral#1\endcsname}%
        {%
        \ifthenelse{\boolean{columnframe\romannumeral#1}}%
        {%
        \@ff@fbox
            {\csname colwidth\romannumeral#1\endcsname}%
            {\csname colheight\romannumeral#1\endcsname}%
            {%
            \expandafter\box\csname column\romannumeral#1\endcsname
            }%
            {%
            \csname\ff@frametype\endcsname
            }%
        }%
    }

```

```

}%
{%
  \@ff@box
    {\csname colwidth\romannumeral#1\endcsname}%
    {\csname colheight\romannumeral#1\endcsname}%
    {%
      \expandafter\box\csname column\romannumeral#1\endcsname
    }%
  }%
}%
}
\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}%
}
\newcommand{\@ff@fbox}[4]{%
  {%
    \fboxsep=\flowframesep
    \fboxrule=\flowframerule
    \@s@tffcol
    \kern\@ff@offset
    #4{\@ff@box{#1}{#2}{#3}}%
  }%
}
\newcommand{\@ff@box}[3]{%
  {%
    \@ffbackground
    {%
      \vbox to#2 {\hb@xt@ #1{\hss{\@s@tffttextcol #3}\hss}\vss\kern\z@}%
    }%
  }%
}
\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".
    }
  \fi
}

```

```

        Exclusion list: "\csname @ff@pages@\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
}
\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
                }%
            \fi
        \fi
    \fi
}

```

```

\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
}
\newcommand*\@ff@s@t@doubleboxoffset}{%
  \setlength{\@ff@offset}{-\flowframesep}%
  \addtolength{\@ff@offset}{-3.75\flowframerule}%
  \addtolength{\@ff@offset}{-.5pt}%
}
\newcommand*\@ff@s@t@ovalboxoffset}{%
  \@ff@offset=-\fontdimen 8\tenln\relax
  \advance\@ff@offset by -\flowframesep\relax
}
\newcommand*\@ff@s@t@ovalboxoffset}{%
  \@ff@offset=-\fontdimen 8\tenlnw\relax
  \advance\@ff@offset by -\flowframesep\relax
}
\newcommand*\@ff@s@t@defaultoffset}{%
  \@ff@offset=-\flowframesep\relax
  \addtolength{\@ff@offset}{-\flowframerule}%
}
\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}%
      {0ovalbox}%
  }%
  {%
    \ff@s@t@0ovalboxoffset
  }%
  {%
    \ff@s@t@defaultoffset
  }%
}%
}%
}%
{}%
}%
{%
  \setlength{\@ff@offset}%
    {\csname @ff@offset@romannumeral#1\endcsname}%
}%
}
\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
        {%

```



```

        {ovalbox}%
    }%
    {%
    \ff@s@t@ovalboxoffset
    }%
    {%
    \ifthenelse
    {%
    \equal{\csname @df@frametype@romannumeral#1\endcsname}%
    {0valbox}%
    }%
    {%
    \ff@s@t@0valboxoffset
    }%
    {%
    \ff@s@t@defaultoffset
    }%
    }%
    }%
    }%
    {}%
    }%
    {%
    \setlength{\ff@offset}%
    {\csname @df@offset@romannumeral#1\endcsname}%
    }%
}
\newcommand*{\putmarginbox}[1]{%
\ff@chkifthispg{\ff@pages@countreg}{#1}%
\ifnotthiscol
\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
\@f@meifdraft{\@ff@box{\marginparwidth}%
{\csname colheight\romannumeral#1\endcsname}{}}%
{M:\number#1}\hss
}%
\fi
\ignorespaces
}
\newcommand*\@ff@drawmargins{%
\@colN=0\relax
\whiledo{\@colN<\c@maxflow}%
{%
\advance\@colN by 1\relax
\makebox[Opt][l]{\@putmarginbox{\@colN}}%
}%
}
\def\@ff@getstaticpos[#1][#2][#3]#4{%
\@ff@tmp@x=#4\relax
\@ff@tmp@y=#2\relax
\def\ff@valign{#3}%
}
\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
      }%
    }%
  }%
}
\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}%
}
\newcommand*\@putstaticbox}[1]{%
  \ifthenelse{\boolean{@sf@hidethis@romannumeral#1}}{%
    {%
      \@notthiscoltrue
      \global\csletcs@if@sf@hidethis@romannumeral#1}{iffalse}%
    }%
    {%
      \ifthenelse{\boolean{@sf@hide@romannumeral#1}}{%
        {%
          \@notthiscoltrue
        }%
      }%
    }%
    \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
}
\newcommand*{\@putstaticbbox}[1]{%
\@sf@chckifthispg{#1}%
\ifnotthiscol
\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
}
\newcommand*{\@resetst@tics}{%
    \@colN=0\relax
    \whiledo{\@colN<\c@maxstatic}%
    {%
        \advance\@colN by 1\relax
        \ifthenelse{\boolean{@sf@clear@romannumeral\@colN}}{%
            {%
                \global\sbox
                {%
                    \csname @staticframe@romannumeral\@colN\endcsname
                }%
            }%
        }%
    }%
}
}
\newcommand*{\@resetydyn@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{}%
            }%
        }%
    }%
}
}
\newcommand*{\@dodfparbox}[1]{%
    \expandafter\let\expandafter
    \@ff@parshape\csname @df@shape@romannumeral#1\endcsname
    \expandafter\@ff@getshape\@ff@parshape\relax
    \ifcase\@ff@shape
        \expandafter\expandafter\expandafter
        \parbox\csname @df@dim@romannumeral#1\endcsname
        {%
            \setlength\parindent\sdfparindent
            \csname\@ff@style\endcsname{\@ff@contents}%
        }%
    \or
        \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
\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
}
\newcommand*{\@dynamicbox}[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}}%
      }%
    }%
  }%
}

```



```

}%
}
\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}%
}
\newcommand*{\@putdynamicbox}[1]{%
  \ifthenelse{\boolean{@df@hidethis@\romannumeral#1}}{%
    {%
      \@notthiscoltrue
      \global\csletcs@if@df@hidethis@\romannumeral#1}{iffalse}%
    }%
    {%
      \ifthenelse{\boolean{@df@hide@\romannumeral#1}}{%
        {%
          \@notthiscoltrue
        }%
        {%
          \@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
}
\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
}
\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}}%
}
\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}}%
}
\newcommand{\@s@tfr@mes}[1]{%
{%
\@picht\textheight
\setbox\@picbox\hb@xt@ \textwidth
\bggroup
\hbox
\bggroup
#1\relax
\egroup
\hss
\egroup
\ht\@picbox\@picht
\dp\@picbox\z@
\mbox{\box\@picbox}%
}%
}
\newcommand*\@ff@doallflowframes}{%
\@colN=0\relax
\whiledo{\@colN<\c@maxflow}%
{%
\advance\@colN by 1\relax
\@putcolbox{\@colN}%
}%
}
\newcommand*\@ff@doallflowframesbbox}{%
\@colN=0\relax
\whiledo{\@colN<\c@maxflow}%
{%
\advance\@colN by 1\relax
\@putcolbbox{\@colN}%
}%
}
\newcommand*\@ff@doallstatics}{%
\@colN=0\relax
\whiledo{\@colN<\c@maxstatic}%
{%

```

```

        \advance\@colN by 1\relax
        \@putstaticbox{\@colN}%
    }%
}
\newcommand*\@ff@doallstaticsbbbox}{%
    \@colN=0\relax
    \whiledo{\@colN<\c@maxstatic}%
    {%
        \advance\@colN by 1\relax
        \@putstaticbbox{\@colN}%
    }%
}
\newcommand*\@ff@doalldynamics}{%
    \@colN=0\relax
    \whiledo{\@colN<\c@maxdynamic}%
    {%
        \advance\@colN by 1\relax
        \@putdynamicbox{\@colN}%
    }%
}
\newcommand*\@ff@doalldynamicsbbbox}{%
    \@colN=0\relax
    \whiledo{\@colN<\c@maxdynamic}%
    {%
        \advance\@colN by 1\relax
        \@putdynamicbbox{\@colN}%
    }%
}
\newcommand*\@ff@dotypeblock}{%
    \makebox[0pt][l]{%
        {%
            \@fr@meifdraft[\setffdrafttypeblockcolor]%
            {%
                \vbox to \textheight{\hbox to \textwidth{}}%
            }%
        }%
    }%
}
\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
  }%
}%
}
\newcount\@nxtcol
\def\@outputdblcol{%
  \@nxtcol=\c@thisframe
  \c@curpg=\ff@pages@countreg
  \@g@tnextcol{\@nxtcol}%
  \ifff@nwp@
    \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
  \@dynamicthehead\@dynamicthefoot

```

```

\badness=\@M
\setbox\@outputbox\vbox{\hbox to \textwidth{\@ff@do@allframes}}%
\@combinedblfloats
\@outputpage
\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
\global\@firstcolumnfalse
\@setcolbox\c@thisframe
\fi
\global\c@thisframe=\@nxtcol
\@setcol{\c@thisframe}\relax
\global\@colht\vsiz
}
\def\@dblfloatplacement{%
\global\@dbltopnum\c@dbltopnumber
\global\@dbltoproom\dbltopfraction\@colht\@textmin
\@colht\advance\@textmin -\@dbltoproom
\@fpmin\dblfloatpagefraction\vsiz
\@fptop \@dblfpsep \@fpsep \@dblfpsep \@fpbot \@dblfpbot
}
\newenvironment{statictable}{\def\@captype{table}}{}
\newenvironment{staticfigure}{\def\@captype{figure}}{}
\newif\iffvadjust
\ffvadjusttrue
\renewcommand*\@onecolumn{\@onecolumn}
\newcommand*\@onecolumn[1][all]{%
  \@onecolumninarea[#1]{\textwidth}{\textheight}{0pt}{0pt}%
}
\newlength\columnheight
\newcommand*\@onecolumninarea{\@onecolumninarea}
\@onlypreamble{\@onecolumninarea}
\newcommand*\@onecolumninarea[5][all]{%
  \setlength\columnheight{#3}%
  \iffvadjust
  \adjustheight{\columnheight}%
}
\fi

```

```

\@n@wflowframe[#1]{#2}{\columnheight}{#4}{#5}%
}
\renewcommand*\@twocolumn{\@twocolumn}
\newcommand*\@twocolumn[1][all]{%
\@twocolumninarea[#1]{\textwidth}{\textheight}{0pt}{0pt}%
}
\newcommand*\@twocolumninarea{\@twocolumninarea}
\@onlypreamble{\@twocolumninarea}
\newcommand*\@twocolumninarea[5][all]{%
\setlength{\columnheight}{#3}%
\iffvadjust
\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
}
\newcommand*\Ncolumn[2][all]{%
\Ncolumninarea[#1]{#2}{\textwidth}{\textheight}{0pt}{0pt}%
}
\@onlypreamble{\Ncolumn}
\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}
\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}%
\iffvadjust\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
}
\newlength{\vcolumnsep}
\setlength{\vcolumnsep}{\columnsep}
\newcommand*{\onecolumnntop}[3][all]{%
\onecolumnntopinarea[#1]{#2}{#3}{\textwidth}{\textheight}{Opt}{Opt}%

```



```

}
\onlypreamble{\onecolumnstop}
\newcommand*\onecolumnstop[2][all]{%
  \onecolumnstopinarea[#1]{static}{#2}{\textwidth}{\textheight}{Opt}{Opt}%
}
\newcommand*\onecolumnstop[2][all]{%
  \onecolumnstopinarea[#1]{dynamic}{#2}{\textwidth}{\textheight}{Opt}{Opt}%
}
\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'%
                }%
            }%
          }%
        }%
      }%
    }%
  }
}
\newlength\@ff@staticH

\newcommand*\onecolumnstopinarea[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{\onecolumnstopinarea}

```

```

\newcommand*\onecolumnStopinarea}[6][all]{%
  \onecolumntopinarea[#1]{static}{#2}{#3}{#4}{#5}{#6}%
}
\newcommand*\onecolumnDtopinarea}[6][all]{%
  \onecolumntopinarea[#1]{dynamic}{#2}{#3}{#4}{#5}{#6}%
}
\newcommand*\twocolumntop}[3][all]{%
  \twocolumntopinarea[#1]{#2}{#3}{\textwidth}{\textheight}{Opt}{Opt}%
}
\@onlypreamble{\twocolumntop}
\newcommand*\twocolumnStop}[2][all]{%
  \twocolumntopinarea[#1]{static}{#2}{\textwidth}{\textheight}{Opt}{Opt}%
}
\newcommand*\twocolumnDtop}[2][all]{%
  \twocolumntop[#1]{dynamic}{#2}%
}
\newcommand*\twocolumntopinarea{\@twocolumntopinarea}
\newcommand*\@twocolumntopinarea}[7][all]{%
  \setlength{\@ff@staticH}{#3}%
  \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}%
  \addtolength{\columnheight}{-\vcolumnsep}%
  \iffvadjust\adjustheight{\columnheight}\fi
  \setlength{\columnwidth}{#4}%
  \addtolength{\columnwidth}{-\columnsep}%
  \divide\columnwidth by 2\relax
  \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}
\newcommand*\twocolumnStopinarea}[6][all]{%
  \twocolumntopinarea[#1]{static}{#2}{#3}{#4}{#5}{#6}%
}

```

```

}
\newcommand*\twocolumnDtopinarea}[6][all]{%
  \twocolumntopinarea[#1]{dynamic}{#2}{#3}{#4}{#5}{#6}%
}
\newcommand*\Ncolumnstop}[4][all]{%
  \Ncolumntopinarea[#1]{#2}{#3}{#4}{\textwidth}{\textheight}{Opt}{Opt}%
}
\@onlypreamble{\Ncolumnstop}
\newcommand*\Ncolumnstop}[3][all]{%
  \Ncolumntop[#1]{static}{#2}{#3}%
}
\newcommand*\NcolumnDtop}[3][all]{%
  \Ncolumntop[#1]{dynamic}{#2}{#3}%
}
\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}
\newcommand*\@Ncolumntopinarea}[8][all]{%
  \setlength{\@ff@staticH}{#4}%
  \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}%
}

```

```

\addtolength{\columnheight}{-\vcolumnsep}%
\iffvadjust
  \adjustheight{\columnheight}%
\fi
\@colN=#3\relax
\advance\@colN by -1\relax
\setlength{\columnwidth}{#5}%
\addtolength{\columnwidth}{-\@colN\columnsep}%
\divide\columnwidth by #3\relax
\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}%
  \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
}
\newcommand*\NcolumnStopinarea[7][all]{%
  \Ncolumnstopinarea[#1]{static}{#2}{#3}{#4}{#5}{#6}{#7}%
}
\newcommand*\NcolumnDtopinarea[7][all]{%
  \NcolumnDtopinarea[#1]{dynamic}{#2}{#3}{#4}{#5}{#6}{#7}%
}
\newcommand*\onecolumnbottom[3][all]{%
  \onecolumnbottominarea[#1]{#2}{#3}{\textwidth}{\textheight}{Opt}{Opt}%
}
\@onlypreamble{\onecolumnbottom}
\newcommand*\onecolumnSbottom[2][all]{%
  \onecolumnbottom[#1]{static}{#2}%
}
\newcommand*\onecolumnDbottom[2][all]{%
  \onecolumnbottom[#1]{dynamic}{#2}%
}
\newcommand*\onecolumnbottominarea[7][all]{%
  \setlength{\@ff@staticH}{#3}%
  \setlength{\columnheight}{#5}%
  \addtolength{\columnheight}{-\@ff@staticH}%
  \addtolength{\columnheight}{-\vcolumnsep}%
}

```

```

\iffvadjust
  \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}%
}
\@onlypreamble{\onecolumnbottominarea}
\newcommand*{\onecolumnSbottominarea}[6][all]{%
  \onecolumnbottominarea[#1]{static}{#2}{#3}{#4}{#5}{#6}%
}
\newcommand*{\onecolumnDbottominarea}[6][all]{%
  \onecolumnbottominarea[#1]{dynamic}{#2}{#3}{#4}{#5}{#6}%
}
\newcommand*{\twocolumnbottom}[3][all]{%
  \twocolumnSbottominarea[#1]{#2}{#3}{\textwidth}{\textheight}{Opt}{Opt}%
}
\@onlypreamble{\twocolumnbottom}
\newcommand*{\twocolumnSbottom}[2][all]{%
  \twocolumnbottom[#1]{static}{#2}%
}
\newcommand*{\twocolumnDbottom}[2][all]{%
  \twocolumnbottom[#1]{dynamic}{#2}%
}
\newcommand*{\twocolumnbottominarea}[7][all]{%
  \setlength{\@ff@staticW}{#4}%
  \setlength{\@ff@staticH}{#3}%
  \setlength{\columnheight}{#5}%
  \addtolength{\columnheight}{-\@ff@staticH}%
  \addtolength{\columnheight}{-\vcolumnsep}%
  \iffvadjust\adjustheight{\columnheight}\fi%
  \newframe[#1]{#2}{\@ff@staticW}{\@ff@staticH}{#6}{#7}%
  \setlength{\@ff@tmp@y}{#5}%
  \addtolength{\@ff@tmp@y}{-\columnheight}%
  \addtolength{\@ff@tmp@y}{#7}%
  \setlength{\columnwidth}{\@ff@staticW}%
  \addtolength{\columnwidth}{-\columnsep}%
  \divide\columnwidth by 2\relax
  \setlength{\@ff@tmp@x}{\columnwidth}%
  \addtolength{\@ff@tmp@x}{\columnsep}%
  \addtolength{\@ff@tmp@x}{#6}%
  \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
}

```

```

\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}
\newcommand*{\twocolumnSbottominarea}[6][all]{%
  \twocolumnbottominarea[#1]{static}{#2}{#3}{#4}{#5}{#6}%
}
\newcommand*{\twocolumnDbottominarea}[6][all]{%
  \twocolumnbottominarea[#1]{dynamic}{#2}{#3}{#4}{#5}{#6}%
}
\newcommand*{\Ncolumnbottom}[4][all]{%
  \Ncolumnbottominarea[#1]{#2}{#3}{#4}{\textwidth}{\textheight}{Opt}{Opt}%
}
\@onlypreamble{\Ncolumnbottom}
\newcommand*{\NcolumnSbottom}[3][all]{%
  \Ncolumnbottom[#1]{static}{#2}{#3}%
}
\newcommand*{\NcolumnDbottom}[3][all]{%
  \Ncolumnbottom[#1]{dynamic}{#2}{#3}%
}
\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}
\newcommand*{\@NcolumnSbottominarea}[8][all]{%
  \setlength{\@ff@staticH}{#4}%
  \setlength{\columnheight}{#6}%
  \addtolength{\columnheight}{-\@ff@staticH}%
  \addtolength{\columnheight}{-\vcolumnsep}%
  \iffvadjust
  \adjustheight{\columnheight}%
  \fi
  \newframe[#1]{#2}{#5}{\@ff@staticH}{#7}{#8}%
  \setlength{\@ff@tmp@y}{#6}%
  \addtolength{\@ff@tmp@y}{-\columnheight}%
  \addtolength{\@ff@tmp@y}{#8}%
  \@colN=#3\relax
  \advance\@colN by -1\relax
  \setlength{\columnwidth}{#5}%
  \addtolength{\columnwidth}{-\@colN\columnsep}%
  \divide\columnwidth by #3\relax
  \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}%
    \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
}
\newcommand*{\NcolumnSbottominarea}[1][all]{%
  \Ncolumnbottominarea[#1]{static}%
}
\newcommand*{\NcolumnDbottominarea}[1][all]{%
  \Ncolumnbottominarea[#1]{dynamic}%
}
\newcount\@ff@adjh
\newcommand*{\adjustheight}[1]{%
  \@ff@adjh=#1\relax
  \divide\@ff@adjh by \baselineskip\relax
}

```

```

#1=\baselineskip\relax
\multiply#1 by \@ff@adjh\relax
}
\newcommand*\adjustcolsep{%
\multiply\columnsep by 2\relax
\addtolength{\columnsep}{\marginparwidth}%
}
\newlength\@ff@staticW
\newcommand*\vtwotone}[1][all]{%
\def\ff@pages{#1}%
\@vtwotone
}

\newcommand*\@vtwotone}[1][Opt]{\@@vtwotonebottom{#1}{\paperheight}}
\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}
\newcommand*\vtwotonebottom}[1][all]{%
\def\ff@pages{#1}%
\@vtwotonebottom
}

\@onlypreamble{\vtwotonebottom}

\newcommand*\@vtwotonebottom}[2][Opt]{\@@vtwotonebottom{#1}{#2}}
\newcommand*\vtwotonetop}[1][all]{%
\def\ff@pages{#1}%
\@vtwotonetop
}

\newcommand*\@vtwotonetop}[2][Opt]{\@@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}%
}
\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}%
}
\newcount\@thisstrip
\newcommand*{\vNtone}[1][all]{%
\def\ff@pages{#1}%
\@vNtone
}
\newcommand*{\@vNtone}[2][Opt]{%
\@vNtone{#1}{#2}{\paperheight}%
}
\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
}
\newcommand*{\@nextvNband}{%
\ifnum\@thisstrip>0\relax
\let\flf@next\@nextvNband
\else

```

```

\let\flf@next\relax
\fi
\advance\@thisstrip by -1\relax
\flf@next
}
\newcommand*{\@nextvNband}[3]{%
\@nextvband{\ff@pages}{\ff@staticH}{#1}{#2}{#3}%
\@nextvNband
}

\@onlypreamble{\vNtone}
\newcommand*{\vNtonebottom}[1][all]{%
\def\ff@pages{#1}%
\@vNtonebottom
}
\@onlypreamble{\vNtonebottom}
\newcommand*{\vNtonebottom}[3][Opt]{%
\@vNtone{#1}{#2}{#3}%
}
\newcommand*{\vNtonetop}[1][all]{%
\def\ff@pages{#1}%
\@vNtonetop
}
\@onlypreamble{\vNtonetop}
\newcommand*{\vNtonetop}[3][Opt]{%
\@vNtonetop{#1}{#2}{#3}%
}
\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%
}
\newcommand*{\htwotone}[1][all]{%
\def\ff@pages{#1}%
\@htwotone
}
\newcommand*{\@htwotone}[1][Opt]{\@htwotoneleft{#1}{\paperwidth}}
\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}%
    \@nextband{\ff@pages}{#2}{#3}{#4}{#5}%
    \@nextband{\ff@pages}{#2}{#6}{#7}{#8}%
}

\@onlypreamble{\htwotone}
\newcommand*{\htwotoneleft}[1][all]{%
    \def\ff@pages{#1}%
    \@htwotoneleft
}
\@onlypreamble{\htwotoneleft}
\newcommand*{\htwotoneleft}[2][Opt]{\@@htwotoneleft{#1}{#2}}
\newcommand*{\htwotoneright}[1][all]{%
    \def\ff@pages{#1}%
    \@htwotoneright
}
\@onlypreamble{\htwotoneright}
\newcommand*{\htwotoneright}[2][Opt]{\@@htwotoneright{#1}{#2}}
\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}%
    \@nextband{\ff@pages}{#2}{#3}{#4}{#5}%
    \@nextband{\ff@pages}{#2}{#6}{#7}{#8}%
}
\newcommand*{\hNtone}[1][all]{%
    \def\ff@pages{#1}%
    \@hNtone
}
\@onlypreamble{\hNtone}
\newcommand*{\@hNtone}[2][Opt]{%
    \@@hNtone{#1}{#2}{\paperwidth}%
}
\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
}
\newcommand*\@hNtoneleft}[1][all]{%
    \def\@ff@pages{#1}%
    \@hNtoneleft
}
\@onlypreamble{\@hNtoneleft}
\newcommand*\@hNtoneleft}[3][Opt]{%
    \@@hNtone{#1}{#2}{#3}%
}
\newcommand*\@hNtoneright}[1][all]{%
    \def\@ff@pages{#1}%
    \@hNtoneright
}
\@onlypreamble{\@hNtoneright}
\newcommand*\@hNtoneright}[3][Opt]{%
    \@@hNtoneright{#1}{#2}{#3}%
}
\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
}
\newcommand*\@nextband}[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}%
}
\newcommand*{\@nexthNband}{%
    \ifnum\@thisstrip>0\relax
        \let\flf@next\@nexthNband
    \else
        \let\flf@next\relax
    \fi
    \advance\@thisstrip by -1\relax
    \flf@next
}
\newcommand*{\@@nexthNband}[3]{%
    \@nexthband{\ff@pages}{\ff@staticW}{#1}{#2}{#3}%
    \@nexthNband
}
\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}%
}
\newlength\ffcolumseprule
\setlength{\ffcolumseprule}{2pt}
\newcommand*{\ffruledeclarations}{}
\newcommand*{\insertvrule}{\@ifstar\@insertvrule\@insertvrule}
\newcommand*{\@insertvrule}[1][Opt]{%
    \@ifnextchar[{\@insertvrule[#1]}{\@insertvrule[#1][Opt]}}%
}
\newlength\@ff@left@x
\newlength\@ff@left@y
\newlength\@ff@left@evenx
\newlength\@ff@left@eveny

```

```

\newlength\@ff@left@width
\newlength\@ff@left@height
\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}%
}
\newcommand*{\@@insert@vrule}[6]{%
\@@ff@getdim{#3}{#4}%
\setlength{\@@ff@left@x}{\@@ff@fareax}%
\setlength{\@@ff@left@y}{\@@ff@fareay}%
\setlength{\@@ff@left@width}{\@@ff@fareawidth}%
\setlength{\@@ff@left@height}{\@@ff@fareaheight}%
\@@ff@getdim{#5}{#6}%
\ifnum\@@ff@left@x>\@@ff@fareax\relax
\@@ff@swaplen{\@@ff@left@x}{\@@ff@fareax}%
\@@ff@swaplen{\@@ff@left@y}{\@@ff@fareax}%
\@@ff@swaplen{\@@ff@left@evenx}{\@@ff@fareaevenx}%
\@@ff@swaplen{\@@ff@left@eveny}{\@@ff@fareaeveny}%
\@@ff@swaplen{\@@ff@left@width}{\@@ff@fareawidth}%
\@@ff@swaplen{\@@ff@left@height}{\@@ff@fareaheight}%
\fi
\setlength{\@@ff@tmp@x}{\@@ff@left@x}
\addtolength{\@@ff@tmp@x}{\@@ff@left@width}%
\setlength{\@@ff@staticW}{\@@ff@fareax}%
\addtolength{\@@ff@staticW}{-\@@ff@tmp@x}%
\setlength{\@@ff@staticH}{\@@ff@left@y}%
\addtolength{\@@ff@staticH}{\@@ff@left@height}%
\setlength{\@@ff@tmp@y}{\@@ff@fareay}%
\addtolength{\@@ff@tmp@y}{\@@ff@fareaheight}%
\ifnum\@@ff@tmp@y>\@@ff@staticH
\setlength{\@@ff@staticH}{\@@ff@tmp@y}%
\fi
\ifnum\@@ff@left@y<\@@ff@fareay\relax
\setlength{\@@ff@tmp@y}{\@@ff@left@y}%
\else
\setlength{\@@ff@tmp@y}{\@@ff@fareay}%
\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{\c@name \@@ff@pages@\romannumeral#4\endcsname}%
\or \edef\@@ff@pages{\c@name \sf@pages@\romannumeral#4\endcsname}%
\or \edef\@@ff@pages{\c@name \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}%
}
\newcommand*\ffvrule}[3]{%
\hfill \rule[-#1]{#2}{#3}\hfill\mbox{}}%
}
\newcommand*\@sinsertvrule}[1][Opt]{%
\@ifnextchar[{\@sinsertvrule[#1]}{\@sinsertvrule[#1][Opt]}%
}
\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}%
}
\newcommand*{\insert@rule}{\@ifstar\@sinsert@rule\@insert@rule}
\newcommand*{\@insert@rule}[1][Opt]{%
  \@ifnextchar[{\@insert@rule[#1]}{\@insert@rule[#1][Opt]}%
}
\def\@insert@rule[#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'%
      }%
    }%
  }%
}
\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
}

```

```

\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}%
}
\newcommand*{\ffhrule}[3]{%
  \hspace*{-#1}\rule{#2}{#3}%
}
\newcommand*{\@sinserthrule}[1][0pt]{%
  \@ifnextchar[{\@sinserthrule[#1]}{\@sinserthrule[#1][0pt]}%
}
\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@ii}{\@ff@tmpN}%
{\@ff@type@ii}{\ff@id}%
}
\newcommand*{\dfchaphead}{%
  \ifstar\@sdynamicchap\@dynamicchap
}
\newcommand{\DFchapterstyle}[1]{#1}
\newcommand{\DFschapterstyle}[1]{#1}
\newcommand{\@dynamicchap}[1]{%
  \@ifundefined{chapter}%
  {%
    \PackageError{flowfram}%
    {Chapters aren't defined}%
  }
}

```

```

    {%
      The document class you are using does not
      define chapters%
    }%
  }%
  {%
    \let\@ff@OLDmakechapterhead\@makechapterhead
    \let\@ff@OLDmakeschapterhead\@makeschapterhead
    \renewcommand{\DFchapterstyle}[1]{\@ff@OLDmakechapterhead{##1}}%
    \renewcommand{\DFschapterstyle}[1]{\@ff@OLDmakeschapterhead{##1}}%
    \xdef\@makechapterhead##1{%
      \noexpand\@setdynamiccontents{\number##1}%
      {%
        \noexpand\DFchapterstyle{##1}%
      }%
    }%
    \xdef\@makeschapterhead##1{%
      \noexpand\@setdynamiccontents{\number##1}%
      {%
        \noexpand\DFschapterstyle{##1}%
      }%
    }%
  }%
}
\newcommand{\@sdynamicchap}[1]{%
  \@dynamicframeid{##1}%
  \@dynamicchap{\ff{id}}%
}
\newcounter{maxthumbtabs}
\@ifundefined{chapter}%
{%
  \newcommand*\@defaultthumbtabtype{section}%
}%
{%
  \newcommand*\@defaultthumbtabtype{chapter}%
}
\newcommand*\@ttb@type{\@defaultthumbtabtype}
\newcommand*\makethumbtabs[2][Opt]{%
  \@ifnextchar[%
    {\@makethumbtabs[##1]{##2}}%
    {%
      \@makethumbtabs[##1]{##2}[\@defaultthumbtabtype]%
    }%
}
\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}%
\@dothumtbs{#1}{#2}%
}
\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
  }%
}
\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
}
}%
}
}
\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
}%
{}%
}%
}
\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}%
}
\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
\edef\csname thumbtab@pages@\romannumeral\@colN\endcsname{%
\ff@pages-\number\ff@tmpN}%
\fi
\fi
}%
\@@dothumbtabs{#1}{#2}%
}
\newlength{\thumbtabwidth}
\setlength{\thumbtabwidth}{1cm}
\@ifundefined{hyperlink}{%
{%
\newcommand{\thumbtabindexformat}[3]{%
\thumbtabformat{#2}{#3}%
}%
}%
{%
\newcommand{\thumbtabindexformat}[3]{%
\hyperlink{#1}{\thumbtabformat{#2}{#3}}%
}%
}
\newcommand{\thumbtabformat}[2]{%
\if@ttb@rotate
\rotatebox{-90}{%
{%
\parbox[c][\thumbtabwidth]{#2}{%
\centering#1%
}%
}%
\else
\parbox[c][#2]{\thumbtabwidth}{%
\centering\@ttb@stack{#1}%
}%
\fi
\fi
}
\def\@flf@subsp#1 #2{%
\expandafter\flf@ta\expandafter{\@flf@substext}%
\flf@tb{#1}%
\edef\@flf@substext{\the\flf@ta\the\flf@tb}%
\def\@flf@tmp{#2}%
\ifx\@flf@tmp\@nnil
\let\@flf@donextsubsp=\@gobble
\else
\expandafter\flf@ta\expandafter{\@flf@substext}%
\edef\@flf@substext{\the\flf@ta\noexpand\space}%
\let\@flf@donextsubsp=\@flf@subsp
\fi
\@flf@donextsubsp{#2}%

```



```

}
\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}%
  }
}
\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}%
}
\newcount\@greyscale
\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}%
    \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}%
  \ifthenelse{\boolean{@ttb@title}\and\boolean{@ttb@num}}{%
    \expandafter
      \xdef\csname @dynamicframe@romannumeral\c@maxdynamic\endcsname{%
        \noexpandthumbtabformat

```

```

    {%
      \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}}
\newdynamicframe[none]{\thumbtabwidth}{#2}%
  {\@ff@tmpx}{\@ff@tmpy}{thumbtabindex\number\@ff@tmpN}%
\expandafter\global\expandafter
  \setlength\csname @df@\romannumeral\c@maxdynamic @evenx\endcsname
    {\@ff@tmpx@even}%
\expandafter
\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}%
}%
}%
\newcommand*\enablethumbtabs{%
\ifnum\c@maxthumbtabs>0\relax
\@ff@tmpN=0\relax
\@dynamicframeid{thumbtab1}%
\whiledo{\@ff@tmpN<\c@maxthumbtabs}%
{%
\advance\@ff@tmpN by 1\relax
\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
}
\newcommand*\disablethumbtabs>{%
\ifnum\c@maxthumbtabs>0\relax
\@ff@tmpN=0\relax
\@dynamicframeid{thumbtab1}%
\whiledo{\@ff@tmpN<\c@maxthumbtabs}%
{%
\advance\@ff@tmpN by 1\relax
\expandafter\xdef\csname @df@pages@\romannumeral\ff@id\endcsname
{none}%
\advance\ff@id by 1\relax
\expandafter\xdef\csname @df@pages@\romannumeral\ff@id\endcsname
{none}%
\advance\ff@id by 1\relax
}%
\fi
}
\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
}
\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}}%
}

```

```

    }%
  }
  \newcommand{\@setthumtab}[2]{%
    \ifthenelse{\(<\c@maxthumtaps<#1\> \or \(<#1<1\>)}%
    {%
      \PackageWarning{flowfram}%
      {%
        Can't find thumtab number '#1', ttb file may not be
        up-to-date%
      }%
    }%
  }%
  {%
    \@dynamicframeid{thumtab\number#1}%
    \@setdynamicframe{\ff@id}{#2}%
    \@dynamicframeid{thumtabindex\number#1}%
    \@setdynamicframe{\ff@id}{#2}%
  }%
}
\newcommand{\setthumtabindex}[2]{%
  \ifthenelse{\equal{#1}{all}}%
  {%
    \ff@tmpN=0\relax
    \whiledo{\ff@tmpN<\c@maxthumtaps}%
    {%
      \advance\ff@tmpN by 1\relax
      \setthumtabindex{\ff@tmpN}{#2}%
    }%
  }%
  {%
    \for\@ttb@id:=#1\do{\setthumtabindex{\@ttb@id}{#2}}%
  }%
}
\newcommand{\@setthumtabindex}[2]{%
  \ifthenelse{\(<\c@maxthumtaps<#1\> \or \(<#1<1\>)}%
  {%
    \PackageWarning{flowfram}%
    {%
      Can't find thumtab number '\number#1',
      ttb file may not be up-to-date%
    }%
  }%
  {%
    \@dynamicframeid{thumtabindex\number#1}%
    \@setdynamicframe{\ff@id}{#2}%
  }%
}
\newcommand*\tocandthumtabindex{%
  \alignoctrue
  \tableofcontents
  \thumtabindex
}

```

```

\aligntocfalse
}
\newcommand*{\@ttb@minitoc}{\@ttb@type}
\let\@ttb@old@starttoc\@starttoc
\newif\if@storetoc
\@storetocfalse
\renewcommand*{\@starttoc}[1]{%
\if@storetoc
\@ttb@storetoc{#1}%
\else
\@ttb@old@starttoc{#1}%
\fi
}
\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
}
\newcommand*{\@storefileconts}[1]{%
\IfFileExists{#1}%
{%
\@storefileconts\@filef@und
}%
{%
\PackageInfo{flowfram}{No file #1.}%
}%
}
\newcount\c@maxtocunits
\newcount\c@maxminitoc
\newcommand{\@storefileconts}[1]{%
\@ifundefined{\@ttb@minitoc}{%
{%
\@ttb@minitoclevel=6\relax
}%
{%
\expandafter\@ttb@minitoclevel\expandafter
=\csname @ttb@\@ttb@minitoc @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
}
\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}%
}%
\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
}
\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}
\long\def\@checkcontentsline#1#2\end{%
  \ifx#1\contentsline
    \@contsline>true
  \else
    \@contsline>false
  \fi
}
\def\@gettype\contentsline#1#2\end{%
  \def\@ttb@contstype{#1}%
}
\newif\ifaligntoc
\aligntocfalse
\let\@ttb@old@tableofcontents\tableofcontents
\renewcommand{\tableofcontents}{%
  \@storetoctrue
  \@ttb@old@tableofcontents
  \ifaligntoc
    \@printalignedtoc
  \else
    \@printtoc
  \fi
  \@storetocfalse
  \global\c@minitoc=0\relax
}
\newlength\beforeminitocskip
\setlength{\beforeminitocskip}{0pt}
\newlength\afterminitocskip
\setlength{\afterminitocskip}{\baselineskip}
\newcommand*{\dominitoc}[1]{%
  \if@minitoc
    \@dominitoc{#1}%
  \fi
}
\newcommand*{\@dominitoc}[1]{\@dominitoc{#1}}
\newcommand{\minitocstyle}[1]{%
  \normalfont\normalsize\normalcolor
  #1%
}

```



```

}
\newcommand*{\@dominitoc}[1]{%
  {%
    \minitocstyle
    {%
      \vskip\beforeminitocskip
      \csname @minitoc@romannumeral#1\endcsname
    }%
  }%
  \vskip\afterminitocskip
}
\newcommand*\appenddfminitoc{%
  \renewcommand{\beforeminitocskip}{\baselineskip}%
  \@ifstar\@sappendminitocdf\@appendminitocdf
}
\newcommand*\@sappendminitocdf}[1]{%
  \renewcommand{\@dominitoc}[1]{%
    \@sappenddynamic{#1}{\@dominitoc{##1}}%
  }%
}
\newcommand*\@appendminitocdf}[1]{%
  \renewcommand{\@dominitoc}[1]{%
    \@appenddynamic{#1}{\@dominitoc{##1}}%
  }%
}
\newcommand*\@printtoc{%
  \@colN=0\relax
  \csname @toc@romannumeral\@colN\endcsname
  \whiledo{\@colN<\c@mactocunits}%
  {%
    \advance\@colN by 1\relax
    \csname @toc@romannumeral\@colN\endcsname
  }%
}
\newcommand*\@printalignedtoc{%
  \@ff@tmpN=0\relax
  \@ifundefined{toc@romannumeral\@ff@tmpN}%
  {}%
  {%
    \csname @toc@romannumeral\@ff@tmpN\endcsname
    \par\noindent\hrulefill
  }%
  \whiledo{\@ff@tmpN<\c@mactocunits}%
  {%
    \advance\@ff@tmpN by 1\relax
    \ifnum\@ff@tmpN>\c@maxthumbtabs
      \csname @toc@romannumeral\@ff@tmpN\endcsname
    \else
      \dynamicframeid{thumbtabindex\@ff@tmpN}%
      \expandafter\expandafter\expandafter
    \fi
  }%
}

```

```

        \off@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
}%
}
\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@minitoc@type}{#1}%
        \ifthenelse{\equal{#1}{chapter}}{%
            {%
                \@makeminitocchapter
            }%
            {%
                \ifthenelse{\equal{#1}{part}}%
                {\@makeminitocpart}%
                {%
                    \@makeminitocsection{#1}%
                }%
            }%
        }%
    }%
}
\@onlypreamble{\enableminitoc}
\newcommand*\@makeminitocchapter{%
    \let\@mtoc@old@chapter\@chapter
    \def\@chapter[##1]##2{%
        \@mtoc@old@chapter[##1]{##2}%
        \stepcounter{minitoc}%
        \dominitoc{\c@minitoc}%
        \@afterheading
    }%
}
\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
}%
}
}
\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
      }%
    }%
  }%
}
}

```

4 Rollback v1.18 (flowfram-2025-08-23)

Declare package, and identify it as a L^AT_EX 2_ε package.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback release:

```
\DeclareRelease{v1.17}{2014-09-30}{flowfram-2014-09-30.sty}
\DeclareCurrentRelease{v1.18}{2025-08-23}
```

Declare package:

```
\ProvidesPackage{flowfram}[2025/08/23 v1.18 (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.

```

\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@tffc@col{}
\newcommand*\@s@tfft@text@col{}
```

`\@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}[\@flf@val\@flf@nr]%
{true,false}[true]%
{%
  \ifcase\@flf@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@rotatetrue}
```

`\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}[\@flf@val\@flf@nr]%
{title,number,both,none}[title]%
{%
\ifcase\@flf@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}[\@flf@val\@flf@nr]%
{relative,absolute}%
{%
\ifcase\@flf@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 `[true]color` 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@tffc`

and `\@s@tffttextcol` 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}{[\@flf@val\@flf@nr]{true,false}[true]{%
\ifcase\@flf@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@tffttextcol{%
\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@tffttextcol}{}%
\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 `[true]color` option as the default, otherwise implement the `[false]color` 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 [true]color 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@chapter\chapter
\renewcommand{\chapter}{%
  \ffprechapterhook
  \@ff@OLD@chapter
}

```

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 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{\thedisplaysdisplayedframe}%
    \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
```

4.1 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.

```
\newcommand*\flowframeshowlayout}{%
  \finishthispage
  {%
    \offdraft\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@absolute page\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@absolute page\relax
  \whiledo{\@colN<\c@maxflow \OR \@colN=\c@maxflow}%
  {%
    \ff@chckifthispg{\ff@pages@countreg}{\@colN}%
    \if@notthiscol
      \else
        \@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, and make the output routine use the internal command

```

\@onlypreamble{\newflowframe}

\@newflowframe
\newcommand{\@newflowframe}{%
  \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{\hspace}{#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{\hspace}{#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 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 identify 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
\edef\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
\edef\csname @ff@offset@\romannumeral#1\endcsname{%
\ff@offset}%
}%
\ifdefempty{\ff@angle}{}%
{%
\expandafter
\edef\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]{%
  \ifcsemtyp{\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}}%
        }

```

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

Page exclusion list:

```
\expandafter
\gdef\c@sf@xpages\romannumeral\c@maxflow\endcsname{}%
\expandafter
\gdef\c@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{<idl>}` 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 \TEor \@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
\edef\csname @sf@offset@\romannumeral#1\endcsname{\ff@offset}%
}%
\ifdefempty{\ff@angle}{}%
{%
\expandafter
\edef\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@sehead{#1}%
  \ffpshpar
  \@ifstar{\@s@ff@heading}{\@dblarg\@ff@heading}%
}

```

\@s@ff@heading

```

\def\@s@ff@heading#1{%
  \@ifundefined{\ff@old\ff@sehead}%
  {%
    \PackageError{flowfram}%
      {Unknown heading command '\ff@sehead'}{}%
  }%
  {%
    \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@sehead\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@sehead}%
  {%
    \PackageError{flowfram}%
    {Unknown heading command '\ff@sehead'}{}}%
  }%
  {%
    \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@sehead\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 (*env.*) 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* (*env.*) 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\begin{group}
  \noexpand\let\noexpand\FLForgpar=\noexpand\par
  }%
\or

```

```

\parshape:
  \edef\@sf@mpg{%
    \noexpand
    \begin{minipage}\csname @sf@dim@\romannumeral#1\endcsname
      \@ff@parshape
      \noexpand\begin{group}
        \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\begin{group}
        \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}%
    \@stffftextcol\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
\edef\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
\edef\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]{%
\ifcsemtyp@{sf@xpages@\romannumeral#1}
{%
\expandafter
\edef\csname @sf@xpages@\romannumeral#1\endcsname{#2}%
}%
{%
\expandafter
\edef\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} \TEor \equal{#1}{even}}{%
    {%
      \ifthenelse{\equal{#1}{odd}}{\@colN=1}{\@colN=2}%
      \whiledo{\@colN<\c@maxflow\TEor\@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
  }%
}%
}%
}%
}

```

4.3 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{<idl>} Gets the IDL for the dynamic frame identified by its
IDL.
  \newcommand*{\getdynamiclabel}[1]{%
    \csname @df@id@\romannumeral#1\endcsname
  }

\getdynamicid \getdynamicid{<cmd>}{<idl>} Gets the IDL 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
IDL 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
\edef\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]{%
\ifcsemtyp{\@df@xpages@\romannumeral#1}
{%
\expandafter
\edef\csname @df@xpages@\romannumeral#1\endcsname{#2}%
}%
{%
\expandafter
\edef\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\@dynamicframeswapcoords\@dynamicframeswapcoords}

```

\@dynamicframeswapcoords Starred form

```

\newcommand*\@dynamicframeswapcoords}[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} \TEor \equal{#1}{even}}%
{%
\ifthenelse{\equal{#1}{odd}}%
{\@colN=1}%
{\@colN=2}%
\whiledo{\@colN<\c@maxflow\TEor\@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 \TEor \@colN=\@ff@numend}%
{%
\@dynamicframeswapcoords{\@colN}%
\advance\@colN by 1\relax
}%
}

```



```

    }%
  }%
}

```

Set the contents of a dynamic frame.

`dynamiccontents` (*env.*) 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\@empty
    \dfcontinuedfalse
  \else
    \dfcontinuedtrue
    \flf@getcontargs#2\@ff@text\@ff@nextid\@ff@rest
  \fi
}

```

Long equivalent of \@empty:

```
\long\def\@empty{}
```

Get the first optional argument and store in the fourth 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\@tempa{\the\@dynamictok{\@ff@text}}%
    \@dynamictok\expandafter{\@tempa}%
    \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*` (*env.*) 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}}

```

4.4 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 indicating 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 indicating 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}}`

`\getdynamicsevenbounds` Even pages:
`\newcommand*\getdynamicsevenbounds}{%`
`\@ifstar\@sgetdynamicsevenbounds\@getdynamicsevenbounds`
`}`

`\@sgetdynamicsevenbounds` Starred version (specify by IDL):
`\newcommand*\@sgetdynamicsevenbounds}[1]{%`
`\@dynamicframeid{#1}\@getdynamicsevenbounds{\ff@id}%`
`}`

`\@getdynamicsevenbounds` Unstarred version (specify by IDN):
`\newcommand*\@getdynamicsevenbounds}[1]{\@ff@getevendim{3}{#1}}`

4.5 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.

```
\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
  \FLFabove>true
\else
  \FLFabove>false
\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
  \FLFabove>true
  \else
  \FLFabove=false
  \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
  \FLlefttrue
  \else
  \FLleftfalse
  \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
    \FLlefttrue
  \else
    \FLleftfalse
  \fi
}

```

`\checkiframeright` `\checkiframeright{<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 `\oddcheckiframeright` or `\evencheckiframeright` 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*{\checkiframeright}{%
  \ifstar\@scheckiframeright\@checkiframeright
}

```

Starred version:

```

\newcommand*{\@scheckiframeright}[4]{%
  \ifodd\c@page
    \@soddcheckiframeright{#1}{#2}{#3}{#4}%
  \else
    \@sevencheckiframeright{#1}{#2}{#3}{#4}%
  \fi
}

```

Unstarred version:

```

\newcommand*\@checkiframeright}[4]{%
  \ifodd\c@page
    \@oddcheckiframeright{#1}{#2}{#3}{#4}%
  \else
    \@evencheckiframeright{#1}{#2}{#3}{#4}%
  \fi
}

```

```

\oddcheckiframeright \oddcheckiframeright{<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*\oddcheckiframeright*{%
  \@ifstar\@soddcheckiframeright\@oddcheckiframeright
}

```

The starred version

```

\newcommand*\@soddcheckiframeright*[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
    \FLFabove>true
\else
    \FLFabove>false
\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#3sevenbounds\endcsname{#4}%
\advance\ffareay by \ffareaheight\relax
\expandafter\ifdim\@ff@check>\ffareay
\FLFabove>true
\else
\FLFabove>false
\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#3sevenbounds}%
{%
\PackageError{flowfram}%
{Unknown frame type '#3'}%
{%
Frame types may only be one of: static, dynamic or
flow%
}%
}%
}%
\csname @sget#3sevenbounds\endcsname{#4}%
\expandafter\ifdim\@ff@check<\ffareay
\FLFbelow>true
\else
\FLFbelow>false
\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
    \FLlefttrue
\else
    \FLleftfalse
\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
}

```

`\evencheckiframeright` `\evencheckiframeright{<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*{\evencheckiframeright}{%
  \@ifstar\@sevencheckiframeright\@evencheckiframeright
}

```

The starred version

```

\newcommand*{\@sevencheckiframeright}[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
    \FLFrightrue
  \else
    \FLFrighfalse
  \fi
}

```

The unstarred version

```
\newcommand*{\@evencheckiframeright}[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 IDN 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
        \ifFLFrigh
            \FFright
        \else
            \FFoverlap
        \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
      \ifFLFrigh
        \FFaboveleft
      \else
        \FFabove
      \fi
    \fi
  \else
    \ifFLFbelow
      \ifFLFleft
        \FFbelowleft
      \else
        \ifFLFrigh
          \FFbelowright
        \else
          \FFbelow
        \fi
      \fi
    \else
      \ifFLFleft
        \FFleft
      \else
        \ifFLFrigh
          \FFright
        \else
          \FFoverlap
        \fi
      \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}%
}

```

4.6 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@setframes
\@setframesfalse
\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
\@setframestrue
\edef\ff@txtcol{%
\csname @ff@txtcol@\romannumeral\c@thisframe\endcsname}%
\@s@tff@textcol
}
```

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

```
\newcommand{\emulatetwocolumn}[1][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@absolute@page=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%
}%
}

```

4.7 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\size\csname colheight\romannumeral#1\endcsname
\global\@colht\vsizel
\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\vsizel\colroom
    \fi
\else
    \global\vsizel\maxdimen
\fi
}

```

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

```

\def\doclearpage{%
\ifvoid\footins
\setbox\@tempboxa\vsplit\@cclv to\z@
\unvbox\@tempboxa
\setbox\@tempboxa\box\@cclv
\edef\@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
\edef\@dbldeferlist{\@dbltoplist\@dbldeferlist}%
\global\let\@dbltoplist\@empty
\global\@colht\vsizel
\begingroup
\@dblfloatplacement
\@makefcolumn\@dbldeferlist
\@whilesw \if@fcolmade \fi
{%
\@outputpage
\@makefcolumn\@dbldeferlist
}%
\endgroup
\else
\ vbox{}%
\clearpage
\fi
\else
\setbox\@cclv\ vbox{\box\@cclv\vfll}%

```

```

    \@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 absolute page counter.

```

\stepcounter{absolute page}%
\setcounter{displayed frame}{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.

```
\ifff@moreframes  
  \@colN=\c@maxflow\relax  
\fi  
\fi  
}%  
\ifff@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{10000}%
  \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
}

```

`\@sflswitchonnext` The starred version uses IDLs.

```
\newcommand{\@sflswitchonnext}[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
}%
}
```

`\@flswitchonnext` The unstarred version uses IDNs.

```
\newcommand{\@flswitchonnext}[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
  }%
}
```

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

```
\newcommand*{\flswitchonnextodd}{%
  \@ifstar\@sflswitchonnextodd\@flswitchonnextodd
}
```

`\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{%
    \@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}%
    }%
}
}
}

```

`\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@  
\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  
}%  
}
```

`\flowswitchonnextoddoneonly` Switch on the listed flow frames for just the next odd page

```

\newcommand*{\flowswitchonnexthodonly}{%
  \@ifstar\@sflowswitchonnexthodonly\@flowswitchonnexthodonly
}

```

\@sflowswitchonnexthodonly The starred version uses IDLs.

```

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

```

Is this frame already on?

```

    \@ff@chckifthispg{\@ff@pages@countreg}{\@ff@id}%
    \@ifnotthiscol

```

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}%
    \@ifnotthiscol

```

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}%
}%
}

```

```

}

```

`\@flowswitchonnextodonly` The unstarred version uses IDNs.

```

\newcommand{\@flowswitchonnextodonly}[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@chkifthispg{\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\flowaddeclusion{\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\flowaddeclusion{\number\@ff@id}{\number\count@}%
    }%
  }%
}

```

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

```

\newcommand*\flowswitchoffnextoddoneonly}{%
  \ifstar\@sflowswitchoffnextoddoneonly\flowswitchoffnextoddoneonly
}

```

`\@sflownexttodonly` The starred version uses IDLs.

```
\newcommand{\@sflownexttodonly}[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\flowaddeclusion{\number\@ff@id}{\number\count@}%
    }%
  }%
}
```

`\@flownexttodonly` The unstarred version uses IDNs.

```
\newcommand{\@flownexttodonly}[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\flowaddeclusion{\number\@ff@id}{\number\count@}%
    }%
  }%
}
```

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

```
\newcommand*\@dynamicsonnext{%
  \ifstar\@sdynamicsonnext\@dynamicsonnext
}
```

`\@sdynamicsonnext` The starred version uses IDLs.

```
\newcommand{\@sdynamicsonnext}[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 IDNs.

```

\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@}%
}%
}%
}

```

\@dynamicswitchonnexodd The unstarred version uses IDNs.

```

\newcommand{\@dynamicswitchonnexodd}[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}%

```



```

\edef\@ff@output@adjustframes{%
  \the\toks@
  \noexpand\dynamicsetpagelist{\number\@ff@id}%
  {\@ff@prepages>\number\count@}%
}%
}%
}

```

`\dynamicsswitchoffnext` Switch off the listed dynamic frames from the next page onwards

```

\newcommand*\dynamicsswitchoffnext}{%
  \@ifstar\@sdynamicsswitchoffnext\dynamicsswitchoffnext
}

```

`\@sdynamicsswitchoffnext` The starred version uses IDLs.

```

\newcommand{\@sdynamicsswitchoffnext}[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}%
    }%
  }%
}

```

`\@dynamicsswitchoffnext` The unstarred version uses IDNs.

```

\newcommand{\@dynamicsswitchoffnext}[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}%
    }%
  }%
}

```

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

```
\newcommand*{\dynamicsswitchoffnextodd}{%
  \ifstar\@dynamicsswitchoffnextodd\dynamicsswitchoffnextodd
}
```

`\@dynamicsswitchoffnextodd` The starred version uses IDLs.

```
\newcommand{\@dynamicsswitchoffnextodd}[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}%  
}%  
}%  
}
```

\@dynamicsswitchoffnextodd The unstarred version uses IDNs.

```
\newcommand{\@dynamicsswitchoffnextodd}[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}%
}%
}%
```

```
}
```

`\dynamicswitchonnexonly` Switch on the listed dynamic frames for just the next page

```
\newcommand*\dynamicswitchonnexonly{%
\ifstar\@sdynamicswitchonnexonly\@dynamicswitchonnexonly
}
```

`\@sdynamicswitchonnexonly` The starred version uses IDLs.

```
\newcommand{\@sdynamicswitchonnexonly}[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
}%
}
```

\@dynamicsswitchonnexonly The unstarred version uses IDNs.

```
\newcommand{\@dynamicsswitchonnexonly}[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
}%
}
```

\dynamicsswitchonnexoddonly Switch on the listed dynamic frames for just the next odd page

```
\newcommand*\dynamicsswitchonnexoddonly{%
  \ifstar\@sdynamicsswitchonnexoddonly\@dynamicsswitchonnexoddonly
}
```

\@sdynamicsswitchonnexoddonly The starred version uses IDNs.

```
\newcommand{\@sdynamicsswitchonnexoddonly}[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
\fi
\expandafter\toks\expandafter{\@ff@output@adjustframes}%
\xdef\@ff@output@adjustframes{%
\the\toks@
\noexpand\dynamicsetpagelist{\number\ff@id}{\@ff@pages}%
}%
}%
}

```

@dynamicswitchonnexthodonly The unstarred version uses IDNs.

```

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

```

Is this frame already on?

```

\df@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
\ifnotthiscol

```

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}%
\ifnotthiscol

```

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}%
}%
}%

```

```

}

```



```

\dynamicswitchoffnextonly Switch off the listed dynamic frames for just the next page
    \newcommand*{\dynamicswitchoffnextonly}{%
      \ifstar\@sdynamicswitchoffnextonly\dynamicswitchoffnextonly
    }

\@sdynamicswitchoffnextonly 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@}%
        }%
      }%
    }

\@dynamicswitchoffnextonly 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@}%
        }%
      }%
    }

\dynamicswitchoffnextoddonly Switch off the listed dynamic frames for just the next odd page
    \newcommand*{\dynamicswitchoffnextoddonly}{%
      \ifstar\@sdynamicswitchoffnextoddonly\dynamicswitchoffnextoddonly
    }

\dynamicswitchoffnextoddonly 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@}%
    }%
}
}

```

`\dynamicsswitchoffnextodonly` The unstarred version uses IDNs.

```

\newcommand{\@dynamicsswitchoffnextodonly}[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@}%
    }%
  }%
}

```

`\staticsswitchonnext` Switch on the listed static frames from the next page onwards

```

\newcommand*\staticsswitchonnext{%
  \ifstar\@sstaticsswitchonnext\@staticsswitchonnext
}

```

`\@sstaticsswitchonnext` The starred version uses IDLs.

```

\newcommand{\@sstaticsswitchonnext}[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}%
\edef\@ff@output@adjustframes{%
    \the\toks@
    \noexpand\staticsetpagelist{\number\ff@id}%
        {\@ff@prepages>\number\count@}%
    }%
}%
}
}

```

\@staticswitchonnexodd The unstarred version uses IDNs.

```

\newcommand{\@staticswitchonnexodd}[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}%
\edef\@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}%
    \ifnotthiscol

```

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}%
    \ifnotthiscol

```

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}%
    \ifnotthiscol

```

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}%
\edef\@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
}%
}

```

`\staticswitchonnexoddonly` Switch on the listed static frames for just the next odd page

```

\newcommand*{\staticswitchonnexoddonly}{%
\@ifstar\@sstaticswitchonnexoddonly\@staticswitchonnexoddonly
}

```

`@sstaticswitchonnexoddonly` The starred version uses IDLs.

```

\newcommand{\@sstaticswitchonnexoddonly}[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}%
    }%
}
}
}

```

\@staticswitchonnexoddonly The unstarred version uses IDNs.

```

\newcommand{\@staticswitchonnexoddonly}[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
}
```

`\@sstaticswitchoffnextonly` 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@}%
        }%
    }%
}

```

`\staticswitchoffnextoddone` Switch off the listed static frames for just the next odd page

```

\newcommand*\staticswitchoffnextoddone{%
    \ifstar\@sstaticswitchoffnextoddone\@staticswitchoffnextoddone
}

```

`\@sstaticswitchoffnextoddone` The starred version uses IDNs.

```

\newcommand{\@sstaticswitchoffnextoddone}[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@}%
        }%
    }%
}

```

`\@staticswitchoffnextoddone` The unstarred version uses IDNs.

```

\newcommand{\@staticswitchoffnextoddone}[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.
    }%
  \loop
}

```

```

        \ifnum\count@<\c@maxflow
        \repeat
        \PackageInfo{flowfram}{\flf@messinfo@gobbletwo}%
    }%
    \@onecolumn
    #1=\c@maxflow
\fi
\@notthiscoltrue
\@ff@nwpfalse
\@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@nwptrue
        \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}%
\ifnotthiscol
\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
    }%
\ifnotthiscol
\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}}%
                {%
        Put the required border around the frame
            \@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 T_EX 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 T_EX 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}%
  \ifnotthiscol
  \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@0ovalboxoffset` Compute offset for `\ovalbox`:

```

\newcommand*\@ff@s@t@0ovalboxoffset}{%
  \@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}%
        {0ovalbox}%
    }%
  {%
    \@ff@s@t@0ovalboxoffset
  }%
}

```

```

        \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}%
            {0ovalbox}%
          }%
          {%
            \ff@s@t@0ovalboxoffset
          }%
          {%

```



```

        {%
        \@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}%
    \ifnotthiscol
    \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[Opt][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}%
    }
  }
}

```



```

\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
\bggroup
\hbox
\bggroup
#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
    \@putdynamicbbox{\@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 T_EX 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\@oddfont
  \else
    \let\@thehead\@evenhead
    \let\@thefoot\@evenfoot
  \fi
\else
  \let\@thehead\@oddhead
  \let\@thefoot\@oddfont
\fi
\@begindvi
\@dodynamicthehead\@dodynamicthefoot
\vbaddness=\@M
\setbox\@outputbox\vbbox{\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\vbbox{\hbox to \textwidth{\@ff@do@allframes}}%
  \@outputpage
}
\beginngroup
  \@dblfloatplacement
  \@startdblcolumn
  \@whilesw \if@fcolmade \fi
    {\@outputpage \@startdblcolumn }%
\endngroup
\@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\vsiz
}

```

```

\@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 \@dblftop \@fpsep \@dblfpsep \@fpbot \@dblfpbot
}

```

4.8 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 \@capttype so that the \caption command may be used.

```

statictable (env.)
  \newenvironment{statictable}{\def\@capttype{table}}{}

staticfigure (env.)
  \newenvironment{staticfigure}{\def\@capttype{figure}}{}

```

4.9 Standard Layouts

4.9.1 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 \ffvadjustfalse is used.

```

\newif\iffvadjust
\ffvadjusttrue

```

\onecolumn \onecolumn will make a single flow frame that takes up the entire area of the typeblock (adjusted according to \iffvadjust.) 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}{Opt}{Opt}%
}

```

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 ($\langle x \rangle$, $\langle y \rangle$), relative to the typeblock, with width $\langle w \rangle$ and height $\langle h \rangle$. The only difference between `\onecolumninarea` and explicitly creating the flow frame using `\newflowframe` is the `\onecolumninarea` will adjust the vertical height to 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[ $\langle pages \rangle$ ]{ $\langle w \rangle$ }{ $\langle h \rangle$ }{ $\langle x \rangle$ }{ $\langle y \rangle$ }.  
\newcommand*{\onecolumninarea}{\@onecolumninarea}  
\@onlypreamble{\onecolumninarea}
```

`\@onecolumninarea`

```
\newcommand*{\@onecolumninarea}[5][all]{%  
  \setlength{\columnheight}{#3}%  
  \iffvadjust  
    \adjustheight{\columnheight}%  
  \fi  
  \@nwfrowframe[#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` [$\langle pages \rangle$].

```
\renewcommand*{\twocolumn}{\@twocolumn}
```

`\@twocolumn`

```
\newcommand*{\@twocolumn}[1][all]{%  
  \@twocolumninarea[#1]{\textwidth}{\textheight}{Opt}{Opt}%  
}
```

`\twocolumninarea` Again, `\twocolumn` is actually a special case of `\twocolumninarea`. Syntax:

```
\twocolumninarea[ $\langle pages \rangle$ ]{ $\langle w \rangle$ }{ $\langle h \rangle$ }{ $\langle x \rangle$ }{ $\langle y \rangle$ }.  
\newcommand*{\twocolumninarea}{\@twocolumninarea}  
\@onlypreamble{\twocolumninarea}
```

`\@twocolumninarea`

```
\newcommand*{\@twocolumninarea}[5][all]{%  
  \setlength{\columnheight}{#3}%  
  \iffvadjust  
    \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@w@flow@frame[#1]{\columnwidth}{\columnheight}{#4}{#5}%
  \setflowframe{\c@maxflow}{margin=left}%
\else
  \@n@w@flow@frame[#1]{\columnwidth}{\columnheight}{\@ff@tmp@x}{#5}%
  \setflowframe{\c@maxflow}{margin=right}%
\fi
\iflefttorightcolumns
  \@n@w@flow@frame[#1]{\columnwidth}{\columnheight}{\@ff@tmp@x}{#5}%
  \setflowframe{\c@maxflow}{margin=right}%
\else
  \@n@w@flow@frame[#1]{\columnwidth}{\columnheight}{#4}{#5}%
  \setflowframe{\c@maxflow}{margin=left}%
\fi
}

```

`\Ncolumn` Again for an arbitrary number of columns ($\langle n \rangle$). Syntax: `\Ncolumn[$\langle pages \rangle$]{ $\langle n \rangle$ }`.

```

\newcommand*{\Ncolumn}[2][all]{%
  \Ncolumninarea[#1]{#2}{\textwidth}{\textheight}{Opt}{Opt}%
}
\@onlypreamble{\Ncolumn}

```

`\Ncolumninarea` Check the number of flow frames requested, and do one of the special cases if available. Syntax:

`\Ncolumninarea[$\langle pages \rangle$]{ $\langle n \rangle$ }{ $\langle w \rangle$ }{ $\langle h \rangle$ }{ $\langle x \rangle$ }{ $\langle y \rangle$ }`.

```

\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!%
        }%
    \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 $\langle n \rangle$ 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}%
  \iffvadjust\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 $\langle type \rangle$) at the top of the other frames.

`\vcolumnsep` The vertical distance between the top frames and column flow frames when

created using `\Ncolumnstop` etc is given by:

```
\newlength{\vcolumnsep}
\setlength{\vcolumnsep}{\columnsep}
```

`\onecolumnstop` `\onecolumnstop` 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:

```
\onecolumnstop[(pages)]{(type)}{(H)}
\newcommand*{\onecolumnstop}[3][all]{%
  \onecolumnstopinarea[#1]{#2}{#3}{\textwidth}{\textheight}{Opt}{Opt}%
}
\@onlypreamble{\onecolumnstop}
```

`\onecolumnstop` Special case for static frame. Syntax: `\onecolumnstop[(pages)]{(H)}`

```
\newcommand*{\onecolumnstop}[2][all]{%
  \onecolumnstopinarea[#1]{static}{#2}{\textwidth}{\textheight}{Opt}{Opt}%
}
```

`\onecolumnstop` Special case for dynamic frame. Syntax: `\onecolumnstop[(pages)]{(H)}`

```
\newcommand*{\onecolumnstop}[2][all]{%
  \onecolumnstopinarea[#1]{dynamic}{#2}{\textwidth}{\textheight}{Opt}{Opt}%
}
```

`\newframe` Create a frame of given type. Syntax:

```
\newframe[(pages)]{(type)}{(w)}{(h)}{(x)}{(y)}.
```

```
\newcommand*{\newframe}[6][all]{%
  \ifthenelse{\equal{#2}{flow}}{%
    {%
      \@nwfrowframe[#1]{#3}{#4}{#5}{#6}%
    }%
    {%
      \ifthenelse{\equal{#2}{dynamic}}{%
        {%
          \@nwdynamicframe[#1]{#3}{#4}{#5}{#6}%
        }%
      }%
      \ifthenelse{\equal{#2}{static}}{%
        {%
          \@nwstaticframe[#1]{#3}{#4}{#5}{#6}%
        }%
      }%
      {%
        \PackageError{flowfram}%
        {Unknown frame type '#2'}%
      }%
      Available frame types are: 'flow', 'static' and 'dynamic'%
    }%
  }%
}
```

```

    }%
  }%
}

```

`\onecolumnstopinarea` Now for a specified area. Syntax:

```

\onecolumnstopinarea[pages]{type}{H}{w}{h}{x}{y}.
\newlength\@ff@staticH

\newcommand*\onecolumnstopinarea[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}%
  \iffvadjust
  \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{\onecolumnstopinarea}

```

`\onecolumnstopinarea` Special case for static frame. Syntax:

```

\onecolumnstopinarea[pages]{H}{w}{h}{x}{y}.
\newcommand*\onecolumnstopinarea[6][all]{%
  \onecolumnstopinarea[#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]{%
  \onecolumnstopinarea[#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}{Opt}{Opt}%
}
\@onlypreamble{\twocolumntop}

```

`\twocolumnstop` Special case for static frame.

```

\newcommand*\twocolumnstop[2][all]{%
  \@twocolumntopinarea[#1]{static}{#2}{\textwidth}{\textheight}{Opt}{Opt}%
}

```

`\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}%
\iffvadjust\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}%
}

```

`\Ncolumnstop` Similarly for an arbitrary number of flow frames. Special case where the area is the typeblock.

Syntax:

```

\Ncolumnstop[<pages>][<type>][<n>][<H>]
\newcommand*\Ncolumnstop}[4][all]{%
  \Ncolumnstopinarea[#1]{#2}{#3}{#4}{\textwidth}{\textheight}{Opt}{Opt}%
}
\@onlypreamble{\Ncolumnstop}

```

`\Ncolumnstop` Special case for static frame.

```

\newcommand*\Ncolumnstop}[3][all]{%
  \Ncolumnstop[#1]{static}{#2}{#3}%
}

```

`\NcolumnDstop` Special case for dynamic frame.

```

\newcommand*\NcolumnDstop}[3][all]{%
  \Ncolumnstop[#1]{dynamic}{#2}{#3}%
}

```

`\Ncolumnstopinarea` Again test to make sure the user requested a sensible number.

```

\newcommand*\Ncolumnstopinarea}[8][all]{%
  \ifnum#3>2\relax
    \@Ncolumnstopinarea[#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
    \onecolumnstopinarea[#1]{#2}{#4}{#5}{#6}{#7}{#8}%
  \or
    \twocolumnstopinarea[#1]{#2}{#4}{#5}{#6}{#7}{#8}%
  \else
    \PackageError{flowfram}%
    {%
      Can't create a negative number of flow frames!%
    }%
  \fi
}

```

```

    }%
    {%
        You have asked for \number#3 \space flow frames
        which really doesn't make sense%
    }%
    \fi
  \fi
}
\@onlypreamble{\Ncolumnpinarea}

```

`\@Ncolumnpinarea` Fit the frames into specified area. Syntax:

```
\Ncolumnpinarea[<pages>]{<type>}{<n>}{<H>}{<w>}{<h>}{<x>}{<y>}.
```

```
\newcommand*{\@Ncolumnpinarea}[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`

```

\iffvadjust
  \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]{%
  \Ncolumnstopinarea[#1]{static}{#2}{#3}{#4}{#5}{#6}{#7}%
}

```

`\NcolumnDtopinarea` Specific case for dynamic frame.

```

\newcommand*\NcolumnDtopinarea[7][all]{%
  \Ncolumnstopinarea[#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}%
\iffvadjust
  \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}{Opt}{Opt}%
}
\@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}%
\iffvadjust\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}{Opt}{Opt}%
}
\@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
}
```

`\@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
  \iffvadjust
  \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}%
}

```

4.9.2 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 $\langle W1 \rangle$ with background colour $\langle C1 \rangle$ and label $\langle L1 \rangle$. The second frame has width $\langle W2 \rangle$ with background colour $\langle C2 \rangle$ and label $\langle L2 \rangle$. 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][Opt]{\@@vtwotonebottom{#1}{\paperheight}}

```

`\vtwotonebottom` Vertical two tone effect along the bottom of the page, of height $\langle H \rangle$. Syntax:
`\vtwotonebottom`[$\langle pages \rangle$][$\langle xoffset \rangle$]{ $\langle H \rangle$ }{ $\langle W1 \rangle$ }{ $\langle C1 \rangle$ }{ $\langle L1 \rangle$ }{ $\langle W2 \rangle$ }{ $\langle C2 \rangle$ }{ $\langle L2 \rangle$ } where the first frame starts at $\langle xoffset \rangle$.

```

\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][Opt]{\@@vtwotonebottom{#1}{#2}}

```

`\vtwotonetop` Border strip along the top of the page

```

\newcommand*\vtwotonetop}[1][all]{%
  \def\ff@pages{#1}%
  \@vtwotonetop
}

```

```

\newcommand*\@vtwotonetop}[2][Opt]{\@@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>}\dots{<Wn>}{<Cn>}{<Ln>}

```

where the first frame has width $\langle W1 \rangle$ with background colour $\langle C1 \rangle$ and label $\langle L1 \rangle$ all the way up to the $\langle n \rangle$ th frame which has width $\langle Wn \rangle$, background colour $\langle Cn \rangle$ and IDL $\langle Ln \rangle$.

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][Opt]{%
  \@vNtone{#1}{#2}{\paperheight}%
}

```

`\@@vNtone` Vertical $\langle n \rangle$ 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][Opt]{%
  \@@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][Opt]{%
  \@@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][Opt]{\@@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][Opt]{\@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][Opt]{\@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}%
  \@nextband{\ff@pages}{#2}{#3}{#4}{#5}%
  \@nextband{\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][Opt]{%
  \@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}%
  \@nextNband
}

```

\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][Opt]{%
  \@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][Opt]{%
  \@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\c@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\c@sf@romannumeral\c@maxstatic @even\endcsname
  {\@ff@tmp@x@even}%
}

```

4.9.3 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 $\langle y \text{ top} \rangle$ or continue below the lowest point by $\langle y \text{ bottom} \rangle$.

`\ffcolumseprule` This has changed in v1.09. Define `\ffcolumseprule` and use instead of `\columnseprule`

```

\newlength\ffcolumseprule
\setlength{\ffcolumseprule}{2pt}

```

`\ffrulerdeclarations` 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*\ffrulerdeclarations{}

```

`\insertvrule` Determine whether or not the starred version is being used.

```

\newcommand*\insertvrule{\@ifstar\@sinsertvrule\@insertvrule}

```

`\@insertvrule` Two optional arguments required.

```

\newcommand*\@insertvrule[1][Opt]{%
  \@ifnextchar[{\@@insertvrule[#1]}{\@@insertvrule[#1][Opt]}%
}

```

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` 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}{\@@ff@areax}%
  \setlength{\@@ff@left@y}{\@@ff@areay}%
  \setlength{\@@ff@left@width}{\@@ff@areawidth}%
  \setlength{\@@ff@left@height}{\@@ff@areahheight}%
  \@@ff@getdim{#5}{#6}%
  \ifnum\@@ff@left@x>\@@ff@areax\relax
    \@@ff@swaplen{\@@ff@left@x}{\@@ff@areax}%
    \@@ff@swaplen{\@@ff@left@y}{\@@ff@areax}%
    \@@ff@swaplen{\@@ff@left@evenx}{\@@ff@areaevenx}%
    \@@ff@swaplen{\@@ff@left@eveny}{\@@ff@areaevenx}%
    \@@ff@swaplen{\@@ff@left@width}{\@@ff@areawidth}%
    \@@ff@swaplen{\@@ff@left@height}{\@@ff@areahheight}%
  \fi
  \setlength{\@@ff@tmp@x}{\@@ff@left@x}
  \addtolength{\@@ff@tmp@x}{\@@ff@left@width}%
  \setlength{\@@ff@staticW}{\@@ff@areax}%
  \addtolength{\@@ff@staticW}{-\@@ff@tmp@x}%
  \setlength{\@@ff@staticH}{\@@ff@left@y}%
  \addtolength{\@@ff@staticH}{\@@ff@left@height}%
  \setlength{\@@ff@tmp@y}{\@@ff@areay}%
  \addtolength{\@@ff@tmp@y}{\@@ff@areahheight}%
  \ifnum\@@ff@tmp@y>\@@ff@staticH
    \setlength{\@@ff@staticH}{\@@ff@tmp@y}%
  \fi
  \ifnum\@@ff@left@y<\@@ff@areay\relax
    \setlength{\@@ff@tmp@y}{\@@ff@left@y}%
  \else
    \setlength{\@@ff@tmp@y}{\@@ff@areay}%
  \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][Opt]{%
  \@ifnextchar[{\@sinsertvrule[#1]}{\@sinsertvrule[#1][Opt]}%
}

```

```

\@@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][Opt]{%
    \@ifnextchar[{\@inserthrule[#1]}{\@inserthrule[#1][Opt]}}
```

```
}
```

`\@@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}%
}

\@sinerthrule Starred version. Two optional arguments required.
\newcommand*\@sinerthrule[1][Opt]{%
\@ifnextchar[{\@sinerthrule[#1]}{\@sinerthrule[#1][Opt]}%
}

\@@sinerthrule Find out the frame types and their IDN.
\def\@@sinerthrule[#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@ii}{\@ff@tmpN}%
{\@ff@type@ii}{\ff@id}%
}

```

4.10 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}%
    }%
    }%
}

\@dynamicchap Starred form.
    \newcommand{\@dynamicchap}[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?

4.11 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
      }%
    }%
    \ifthenelse{\equal{#3}{section}}{%
      {%
        \@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 variable 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@subsp@text` which should be set to empty before use.

```
\def\@flf@subsp#1 #2{%
  \expandafter\flf@ta\expandafter{\@flf@subsp@text}%
  \flf@tb{#1}%
  \edef\@flf@subsp@text{\the\flf@ta\the\flf@tb}%
  \def\@flf@tmp{#2}%
  \ifx\@flf@tmp\@nnil
    \let\@flf@donextsubsp=\@gobble
  \else
    \expandafter\flf@ta\expandafter{\@flf@subsp@text}%
    \edef\@flf@subsp@text{\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}%
        }%
      }%
  }%
\fi
\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 `tth` 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  
}
```

4.12 Minitocs

`\@ttb@minitoc` Sectioning type for the minitoc, by default it is the same as the thumbtabs

```
\newcommand*\@ttb@minitoc{\@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@minitoclevel}{%
    {%
      \@ttb@minitoclevel=6\relax
    }%
    {%
      \expandafter\@ttb@minitoclevel\expandafter
        =\csname \@ttb@\@ttb@minitoclevel\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@minitoclevel. 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
    \@contsline>true
  \else
    \@contsline>false
  \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
  \@storetoctfalse
  \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 \appendminitocdynamicframe 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\@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@minitoc@type}{#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
    }%
    {}%
  }%
}

```

Change History

- 1.10 – 2007/08/21
- \@ttb@stack: now uses
 - \mbox 298, 706
 - \@flf@subsp: new 297, 705
 - \@storefileconts: uses
 - \PackageInfo instead of
 - \typeout 311, 712
 - \@ttb@stack: spaces now replaced with \space . 297, 706
- 1.11 – 2008/06/27
- General: added relative location commands 120, 581
 - \@@setdynamicframe: fixed bug
 - in clear key 106, 563
 - removed unwanted space 106, 563
 - \@@setflowframe: removed unwanted spaces 80, 537
 - \@ff@getevendim: new . . . 117, 577
 - \@getflowbounds: fixed bug:
 - changed frame id from 2 to 1 (thanks to Lutz Goldmann for pointing it out) 119, 580
 - \@getflowevenbounds: new 119, 580
 - \@getstaticbounds: fixed bug:
 - changed frame id from 1 to 2 (thanks to Lutz Goldmann for pointing it out) 118, 579
 - \@getstatischevenbounds: new 118, 579
 - \@setcol: added displayedframe increment 157, 601
 - \@sgetflowevenbounds: new 119, 580
 - \checkifframeabove: new 120, 581
 - \checkifframebelow: new 122, 583
 - \checkifframeleft: new . 125, 585
 - \checkifframeright: new 127, 587
 - displayedframe: displayedframe counter added 31, 523
 - \evencheckifframeabove: new 129, 589
 - \evencheckifframebelow: new 130, 591
 - \evencheckifframeleft: new 131, 592
 - \evencheckifframeright: new 132, 594
 - \FFabove: new 134, 596
 - \FFaboveleft: new 133, 595
 - \FFaboveright: new 133, 595
 - \FFbelow: new 134, 596
 - \FFbelowleft: new 133, 595
 - \FFbelowright:
 - new 133, 134, 595, 596
 - \ffhrule: new 282, 698
 - \FFleft: new 133, 595
 - \FFoverlap: new 134, 596
 - \ffruleddeclarations: new 275, 691
 - \ffvrule: new 279, 694
 - \getdynamicevenbounds: new . 120
 - \getdynamicid: new 104, 561
 - \getdynamiclabel: new . . 104, 561
 - \getflowevenbounds: new 119, 580
 - \getflowid: new 79, 530
 - \getflowlabel: new 79, 530
 - \getstatischevenbounds:
 - new 118, 579
 - \getstaticid: new 84, 546
 - \getstaticlabel: new 84, 546
 - \labelflow: new 31, 523
 - \labelflowidn: new 31, 522
 - \oddcheckifframeabove:
 - new 121, 581
 - \oddcheckifframebelow:
 - new 124, 583
 - \oddcheckifframeleft:
 - new 126, 586
 - \oddcheckifframeright:
 - new 128, 588
 - \relativeframelocation:
 - new 134, 596
 - \reldynamicloc: new 143, 598
 - \relflowloc: new 143, 598
 - \relstaticloc: new 143, 598
- 1.12 – 2009/11/25
- \iflefttorightcolumns:
 - new 15, 521
- 1.14 – 2012-11-10
- General: added page exclusion list 45
 - changed color to boolean option 28
 - new 27
 - now loading xfor and etoolbox 3, 516

now loading xkeyval instead of keyval	515	replaced \c@page with \ff@pages@countreg	207
replaced \c@page with \ff@pages@countreg	207	\@newdynamicframe: added page exclusion list	559
\@newdynamicframe: added page exclusion list	559	\@newflowframe: added page exclusion list	528
\@newstaticframe: added page exclusion list	544	\@df@chckifthispg: added optional argument	179, 648
\@df@chckifthispg: added optional argument	179, 648	replaced \c@page with \ff@pages@countreg	179, 649
replaced \c@page with \ff@pages@countreg	179, 649	\@ff@checkifmoreframes: added check for exclusion list	173, 607
\@ff@checkifmoreframes: added check for exclusion list	173, 607	replaced \c@page with \ff@pages@countreg	174, 607
replaced \c@page with \ff@pages@countreg	174, 607	\@ff@checkpages: replaced \c@page with \ff@pages@countreg	175, 608
\@ff@checkpages: replaced \c@page with \ff@pages@countreg	175, 608	\@ff@disablecolor: new	15, 520
\@ff@disablecolor: new	15, 520	\@ff@enablecolor: new	14, 520
\@ff@enablecolor: new	14, 520	\@ff@output@adjustframes: new	175, 610
\@ff@output@adjustframes: new	175, 610	\@ff@pages@countreg: new	12, 519
\@ff@pages@countreg: new	12, 519	\@g@tnextcol: replaced \c@page with \ff@pages@countreg	177, 647
\@g@tnextcol: replaced \c@page with \ff@pages@countreg	177, 647	\@outputdblcol: replaced \c@page with \ff@pages@countreg	667, 668
\@outputdblcol: replaced \c@page with \ff@pages@countreg	667, 668	\@outputpage: added absolutepage increment	605
\@outputpage: added absolutepage increment	605	\@putcolbbox: replaced \c@page with \ff@pages@countreg	184, 652
\@putcolbbox: replaced \c@page with \ff@pages@countreg	184, 652	\@putcolbox: added check for non-void box	183, 651
\@putcolbox: added check for non-void box	183, 651	replaced \c@page with \ff@pages@countreg	183, 651
replaced \c@page with \ff@pages@countreg	183, 651	\@putmarginbox: replaced \c@page with \ff@pages@countreg	188, 656
\@putmarginbox: replaced \c@page with \ff@pages@countreg	188, 656	\@setcol: changed check to use tolerance rather than testing if hsize and columnwidth are equal	157, 601
\@setcol: changed check to use tolerance rather than testing if hsize and columnwidth are equal	157, 601	\@sf@chckifthispg: added optional argument	179, 648
\@sf@chckifthispg: added optional argument	179, 648	replaced \c@page with \ff@pages@countreg	179, 648
replaced \c@page with \ff@pages@countreg	179, 648	absolutepage: new	12, 519
absolutepage: new	12, 519	color: changed to boolean option	519
color: changed to boolean option	519	\dynamicaddexclusion: new	107, 567
\dynamicaddexclusion: new	107, 567	\dynamicsetexclusion: new	107, 567
\dynamicsetexclusion: new	107, 567	\dynamicsetpagelist: new	107, 566
\dynamicsetpagelist: new	107, 566	\dynamicswitchoffnext: new	222, 625
\dynamicswitchoffnext: new	222, 625	\dynamicswitchoffnextodd: new	223, 626
\dynamicswitchoffnextodd: new	223, 626	\dynamicswitchoffnextoddone: new	229, 633
\dynamicswitchoffnextoddone: new	229, 633	\dynamicswitchoffnextonly: new	228, 633
\dynamicswitchoffnextonly: new	228, 633	\dynamicswitchonnext: new	219, 622
\dynamicswitchonnext: new	219, 622	\dynamicswitchonnextodd: new	220, 623
\dynamicswitchonnextodd: new	220, 623	\dynamicswitchonnextoddone: new	226, 629
\dynamicswitchonnextoddone: new	226, 629	\dynamicswitchonnextonly: new	225, 628
\dynamicswitchonnextonly: new	225, 628	\ffaddtoadjustframeshook: new	175, 646
\ffaddtoadjustframeshook: new	175, 646	\ffprechapterhook: new	20, 522
\ffprechapterhook: new	20, 522	\fftolerance: new	156, 601
\fftolerance: new	156, 601	\finishthispage: change \c@page to \ff@pages@countreg	74, 524
\finishthispage: change \c@page to \ff@pages@countreg	74, 524	\flf@doifverbose: new	6, 517
\flf@doifverbose: new	6, 517	\flf@message: new	5, 517
\flf@message: new	5, 517	\flowaddexclusion: new	81, 540
\flowaddexclusion: new	81, 540	\flowsetexclusion: new	81, 540
\flowsetexclusion: new	81, 540	\flowsetpagelist: new	81, 540
\flowsetpagelist: new	81, 540	\flowswitchoffnext: new	211, 613
\flowswitchoffnext: new	211, 613	\flowswitchoffnextodd: new	212, 614
\flowswitchoffnextodd: new	212, 614	\flowswitchoffnextoddone: new	219, 621
\flowswitchoffnextoddone: new	219, 621	\flowswitchoffnextonly: new	218, 621
\flowswitchoffnextonly: new	218, 621	\flowswitchonnext: new	209, 610
\flowswitchonnext: new	209, 610	\flowswitchonnextodd: new	210, 611
\flowswitchonnextodd: new	210, 611	\flowswitchonnextoddone: new	215, 617
\flowswitchonnextoddone: new	215, 617		

\flowswitchonnextonly:	2.0 – 2025-11-24
new	General: no longer loading xfor,
\framebreak: made change to	ifthen, xkeyval 3
switch global	\@@ff@getrange: removed . . . 175
\if@ttb@rotate: changed to	\@@dofooter: renamed 204
boolean key	\@@doheader: renamed 203
\newpage: Modified definition of	\@@dominitoc: renamed 317
\newpage	\@@dynamicframeswapcoords:
pages: new	removed 107
\reldynamicloc: removed	\@@ff@chckifthispg: renamed . 178
unwanted spaces	\@@ff@getrange: removed . . . 175
\relflowloc: removed unwanted	\@@hNtone: replaced 272
spaces	\@@hNtoneright: replaced . . . 273
\relstaticloc: removed	\@@htwotoneleft: replaced . . . 270
unwanted spaces	\@@htwotoneright: replaced . . . 271
rotate: changed to boolean key	\@@insert@vrule: replaced . . . 277
.	\@@insert@hrule: removed . . . 280
\staticaddexclusion: new	\@@insert@vrule: removed . . . 276
.	\@@newdynamicframe: renamed . 103
\staticsetexclusion: new	\@@newstaticframe: renamed . . 83
.	\@@setdynamicframe: renamed . 106
\staticsetpagelist: new .	\@@setflowframe: renamed . . . 80
.	\@@setstaticframe: renamed . . 86
\staticswitchoffnext:	\@@sinserthrule: removed . . . 282
new	\@@sinserthrule: removed . . . 279
\staticswitchoffnextodd:	\@@staticframeswapcoords:
new	removed 99
\staticswitchoffnextoddoneonly:	\@@storefileconts: removed . . 312
new	\@@ttb@stack: removed 298
\staticswitchonnext:	\@@vNtone: renamed 269
new	\@@vNtonetop: renamed 270
\staticswitchonnextodd:	\@@vtwotonetop: replaced 267
new	\@Ncolumnbottominarea:
\staticswitchonnextoddoneonly:	replaced 263
new	\@Ncolumninarea: replaced . . . 246
\staticswitchonnextoddoneonly:	\@Ncolumntopinarea: replaced . 255
new	\@addtotoclist: replaced 312
\staticswitchonnextonly:	\@appenddynamic: renamed . . . 113
new	\@appendminitocdf: removed . . 318
thumbtabs: new	\@beginstaticcontents:
verbose: new	renamed 95
1.15 – 2014-05-15	\@checkcontentsline: removed 315
\@ff@getshape: added check for	\@checkifframeabove: renamed 121
\Shapepar	\@checkifframebelow: renamed 123
\@setcol: set \linewidth as well	\@checkifframeleft: renamed . 125
as \textwidth	\@checkifframeright: renamed 127
1.16 – 2014-06-04	\@computeffarea: removed . . . 116
\@@newdynamicframe: added	\@dblfloatplacement: removed
‘hide’ and ‘hidethis’	redefinition 209
attributes	\@df@chckifthispg: replaced . . 179
\@setcol: Removed assignment	
of \textwidth	
1.17 – 2014-09-30	
\@setcol: added check for	
list	

\@df@checkuniqueidl: removed	104	\@ff@checkifmoreframes:	
\@df@setoffset: renamed	188	renamed	173
\@doclearpage: no longer		\@ff@checknumrange: removed	175
redefining	158	\@ff@checkpages: removed	175
\@docolbbox: renamed	181	\@ff@checkthispage: removed	175
\@docolbox: renamed	180	\@ff@checkuniqueidl: removed	79
\@dodfparbox: renamed	196	\@ff@do@allframes: renamed	206
\@dodynamicbbox: renamed	199	\@ff@doalldynamics: renamed	205
\@dodynamicbox: renamed	198	\@ff@doalldynamicsbbox:	
\@dominitoc: renamed	317	renamed	205
\@dostaticbbox: renamed	191	\@ff@doallflowframes:	
\@dothehead: added tagging		renamed	205
socket	159	\@ff@doallflowframesbbox:	
\@dothumbtabs: replaced	295	renamed	205
\@dynamicchap: replaced	290	\@ff@doallstatics: renamed	205
\@dynamicframeswapcoords:		\@ff@doallstaticsbbox:	
renamed	108	renamed	205
\@dynamicswitchoffnext:		\@ff@dotypeblock: renamed	206
removed	223	\@ff@drawmargins: renamed	190
\@dynamicswitchoffnextodd:		\@ff@enablecolor: removed	14
removed	225	\@ff@fbox: renamed	182
\@dynamicswitchoffnextodddonly:		\@ff@getdim: renamed	116
removed	230	\@ff@getrange: renamed	
\@dynamicswitchoffnextonly:		\@ff@getrange command	69
removed	229	\@ff@getrangegreater:	
\@dynamicswitchonnext:		removed	175
removed	220	\@ff@getrangeless: removed	175
\@dynamicswitchonnextodd:		\@ff@getshape: renamed	93
removed	222	\@ff@getstaticpos: removed	190
\@dynamicswitchonnextodddonly:		\@ff@heading: removed	87
removed	228	\@ff@left@evenx: replaced	276
\@dynamicswitchonnextonly:		\@ff@left@eveny: replaced	276
removed	226	\@ff@left@height: replaced	276
\@dynamictok: removed	108	\@ff@left@width: replaced	276
\@endstaticcontents: renamed	97	\@ff@left@x: replaced	276
\@evencheckifframeabove:		\@ff@left@y: replaced	276
renamed	130	\@ff@output@adjustframes:	
\@evencheckifframebelow:		renamed	175
renamed	131	\@ff@pages@countreg: renamed	12
\@evencheckifframeleft:		\@ff@parshape: renamed	87
renamed	132	\@ff@s@t@Ovalboxoffset:	
\@evencheckifframeright:		renamed	186
renamed	133	\@ff@s@t@defaultoffset:	
\@ff@getrange: renamed		renamed	186
\@ff@getrange	70	\@ff@s@t@Ovalboxoffset:	
\@ff@addtolist: removed	113	renamed	186
\@ff@adjh: replaced	265	\@ff@sectionhead: removed	87
\@ff@box: renamed	182	\@ff@setoffset: renamed	186
\@ff@chckifthispg: renamed	177	\@ff@setsecthead: renamed	93
		\@ff@swaplen: renamed	116

\@ffdraft: renamed	4	\@inserthrule: removed	280
\@ffnodraft: renamed	4	\@insertvrule: removed	276
\@flf@checkcontinued:		\@empty: removed	108
removed	108	\@makecol: removed redefinition	173
\@flf@endxdynamiccontents:		\@makeminitocchapter:	
removed	109	removed	320
\@flf@find@end: removed	109	\@makeminitocpart: removed	320
\@flf@get@body: removed	108	\@makeminitocsection:	
\@flf@getcontargs: removed	109	removed	320
\@flf@subsp: removed	297	\@makethumbchapter: removed	293
\@flowfram@checkrelativeloc:		\@makethumbpart: removed	293
new	140	\@makethumbsection: removed	293
\@flowfram@saverrelativeloc:		\@makethumbtabs: replaced	292
new	138	\@n@wdynamicframe: removed	103
\@flowframe@warn@rerun: new	141	\@n@wflowframe: removed	78
\@flowframeid: renamed	79	\@n@wstaticframe: removed	83
\@flowframeswapcoords:		\@newdynamicframe: removed	103
removed	82	\@newflowframe: removed	78
\@flowswitchoffnext: removed	212	\@newstaticframe: removed	83
\@flowswitchoffnextodd:		\@nextnband: renamed	274
removed	214	\@nextband: replaced	273
\@flowswitchoffnextoddoneonly:		\@nextvband: renamed	269
removed	219	\@nextvband: renamed	268
\@flowswitchoffnextonly:		\@oddcheckifframeabove:	
removed	219	renamed	122
\@flowswitchonnext: removed	209	\@oddcheckifframebelow:	
\@flowswitchonnextodd:		renamed	124
removed	211	\@oddcheckifframeleft:	
\@flowswitchonnextoddoneonly:		renamed	126
removed	218	\@oddcheckifframeright:	
\@flowswitchonnextonly:		renamed	129
removed	215	\@onecolumn: replaced	241
\@g@tnextcol: renamed	176	\@onecolumninarea: replaced	242
\@getdynamicbounds: renamed	119	\@opcol: include mark	
\@getdynamicievenbounds:		structures	161
renamed	120	\@printalignedtoc: replaced	319
\@getflowbounds: renamed	119	\@printtoc: replaced	318
\@getflowievenbounds: renamed	119	\@putcolbbox: renamed	184
\@getframeid: renamed	47	\@putcolbox: renamed	182
\@getstaticbounds: renamed	118	\@putdynamicbbox: renamed	202
\@getstaticievenbounds:		\@putdynamicbox: renamed	200
renamed	118	\@putmarginbox: renamed	188
\@gettype: removed	315	\@putstaticbbox: removed	194
\@greyscale: replaced	298	\@putstaticbox: renamed	192
\@hNtone: removed	272	\@relativeframelocation:	
\@hNtoneleft: removed	272	renamed	134
\@hNtoneright: removed	273	\@reldynamicloc: removed	143
\@htwotone: removed	270	\@relflowloc: removed	144
\@htwotoneleft: removed	271	\@relstaticloc: removed	143
\@htwotoneright: removed	271	\@resetdyn@mics: renamed	195

\@resetst@tics: renamed	195	\@sevencheckifframeright:	renamed	132	
\@s@ff@heading: removed	87	\@sevencheckifframeright:	renamed	132	
\@s@tdynamicframeid: removed		104	\@sf@chckifthispg: replaced	..		179	
\@s@tffttextcol: renamed	13	\@sf@checkuniqueidl: removed			84	
\@s@tflowframeid: removed	...	79	\@sf@setoffset: renamed		187	
\@s@tfr@mes: renamed	204	\@sflowframeswapcoords:	removed	82	
\@s@tstaticframeid: removed	..	84	\@sflowswitchoffnext:	removed	212	
\@sappenddynamic: renamed	..	112	\@sflowswitchoffnextodd:	removed	214	
\@sappendminitocdf: removed	..	318	\@sflowswitchoffnextoddoneonly:	removed	219	
\@scheckifframeabove:	renamed	121	\@sflowswitchoffnextonly:	removed	219
\@scheckifframebelow:	renamed	123	\@sflowswitchonnext:	removed		209
\@scheckifframeleft: renamed		125	\@sflowswitchonnextodd:	removed	211	
\@scheckifframeright:	renamed	127	\@sflowswitchonnextoddoneonly:	removed	218
\@scomputeffarea: removed	...	116	\@sflowswitchonnextonly:	removed	215	
\@sdynamicchap: removed	290	\@sgetdynamicbounds: removed			119	
\@sdynamicframeswapcoords:	removed	107	\@sgetdynamicsevenbounds:	removed	120
\@sdynamicswitchoffnext:	removed	223	\@sgetflowbounds: removed	...	119	
\@sdynamicswitchoffnextodd:	removed	225	\@sgetflowevenbounds:	removed	119
\@sdynamicswitchoffnextoddoneonly:	removed	230	\@sgetstaticbounds: removed	..	118	
\@sdynamicswitchoffnextonly:	removed	229	\@sgetstaticsevenbounds:	removed	118
\@sdynamicswitchonnext:	removed	220	\@sinserthrule: removed	282	
\@sdynamicswitchonnextodd:	removed	222	\@sinsertvrule: removed	279	
\@sdynamicswitchonnextoddoneonly:	removed	228	\@snewdynamicframe: removed	..	103	
\@sdynamicswitchonnextonly:	removed	226	\@snewstaticframe: removed	..	83	
\@setcol: renamed	157	\@soddcheckifframeabove:	renamed	122	
\@setcolbox: renamed	179	\@soddcheckifframebelow:	renamed	124	
\@setdynamiccontents:	renamed	112	\@soddcheckifframeleft:	renamed	126
\@setdynamicframe: removed	..	106	\@soddcheckifframeright:	renamed	128	
\@setflowframe: removed	80	\@srelativeframelocation:	renamed	134	
\@setframecol: renamed	43	\@sreldynamicloc: removed	...		143	
\@setstaticframe: removed	...	86	\@srelflowloc: removed		144	
\@setthumbtab: replaced	307	\@srelstaticloc: removed	...		143	
\@setthumbtabindex: replaced	..	309					
\@sevencheckifframeabove:	renamed					
		129					
\@sevencheckifframebelow:	renamed					
		130					

\@ssetdynamiccontents:		\@ttb@minitoclevel: replaced .	311
removed	112	\@ttb@minitoclevel: replaced ..	310
\@ssetdynamicframe: removed .	106	\@ttb@paragraph@level:	
\@ssetflowframe: removed ...	80	replaced	315
\@ssetstaticframe: removed ..	86	\@ttb@part@level: replaced ...	315
\@sstaticframeswapcoords:		\@ttb@section@level: replaced	315
removed	99	\@ttb@stack: replaced	297
\@sstaticswithchoffnext:		\@ttb@storetoc: replaced	311
removed	233	\@ttb@subparagraph@level:	
\@sstaticswithchoffnextodd:		replaced	315
removed	235	\@ttb@subsection@level:	
\@sstaticswithchoffnextoddoneonly:		replaced	315
removed	240	\@ttb@subsubsection@level:	
\@sstaticswithchoffnextonly:		replaced	315
removed	240	\@ttb@type: renamed	291
\@sstaticswithchonnext:		\@twocolumn: removed	243
removed	231	\@twocolumninarea: replaced ..	244
\@sstaticswithchonnextodd:		\@twocolumntopinarea: replaced	251
removed	232	\@vNtone: removed	268
\@sstaticswithchonnextoddoneonly:		\@vNtonebottom: removed	269
removed	239	\@vNtonetop: removed	270
\@sstaticswithchonnextonly:		\@vtwotone: removed	266
removed	236	\@vtwotonebottom: removed ...	267
\@staticcontinueonframe:		\@vtwotonetop: removed	267
renamed	100	\addtocontents: added	
\@staticframeid: renamed ...	84	redefinition	88
\@staticframeswapcoords:		\appenddynamiccontents: added	
renamed	99	optional argument	112
\@staticscontinueonframe:		changed to document	
renamed	100	command	112
\@staticswithchoffnext:		\@c@curpg: renamed	176
removed	233	\@c@ffrangenumber: removed	175
\@staticswithchoffnextodd:		\@ChapterInDynamic: new	284
removed	235	\checkifframeabove: changed to	
\@staticswithchoffnextoddoneonly:		document command	120
removed	240	\checkifframebelow: changed to	
\@staticswithchoffnextonly:		document command	122
removed	240	\checkifframeleft: changed to	
\@staticswithchonnext:		document command	125
removed	231	\checkifframeright: changed to	
\@staticswithchonnextodd:		document command	127
removed	232	\cleardoublepage: changed to	
\@staticswithchonnextoddoneonly:		document command	75
removed	239	\cleartoevenpage: new	75
\@staticswithchonnextonly:		\columnheight: removed	156
removed	236	\computebottomedge: changed to	
\@storefileconts: replaced ...	311	document command	113
\@thisstrip: renamed	268	\computeeflowframearea: changed	
\@ttb@chapter@level: replaced	315	to document command ...	114
\@ttb@level: removed	312		

<code>\computeleftedgeeven</code> : changed to document command ...	113	<code>\dynamicswitchonnextodonly</code> : changed to document command	226
<code>\computeleftedgeodd</code> : changed to document command ...	113	<code>\dynamicswitchonnextonly</code> : changed to document command	225
<code>\computerightedgeeven</code> : changed to document command ...	114	<code>\emulateonecolumn</code> : changed to document command	145
<code>\computerightedgeodd</code> : changed to document command ...	114	<code>\emulatetwocolumn</code> : changed to document command	145
<code>\computetopedge</code> : changed to document command	113	<code>\enableinitoc</code> : changed to document command	320
<code>\continueonframe</code> : changed to document command	99	<code>\enablethumbtabs</code> : changed to document command	302
<code>\dfchaphead</code> : changed to document command	283	<code>\endxdynamiccontents</code> : removed	109
<code>\dfEvenFooterStyle</code> : new	168	<code>\evencheckifframeabove</code> : changed to document command	129
<code>\dfEvenHeaderStyle</code> : new	168	<code>\evencheckifframebelow</code> : changed to document command	130
<code>\dfOddFooterStyle</code> : new	168	<code>\evencheckifframeleft</code> : changed to document command ...	131
<code>\dfOddHeaderStyle</code> : new	168	<code>\evencheckiframeright</code> : changed to document command	132
<code>\dfswapoddeven</code> : changed to document command	107	<code>\ff@st@doubleboxoffset</code> : renamed	185
<code>\disablethumbtabs</code> : changed to document command	303	<code>\ffaddtoadjustframeshook</code> : changed to document command	175
<code>\dominitoc</code> : changed to document command	316	<code>\ffdefaultcontinuetext</code> : new .	101
<code>\dynamicaddexclusion</code> : changed to document command ...	107	<code>\ffdefaultdynamiccontinuetext</code> : new	101
<code>dynamiccontents</code> : changed to document environment ...	108	<code>\ffdefaultpostcontinued</code> : new	102
<code>dynamiccontents*</code> : changed to document environment ...	108	<code>\ffdefaultstaticcontinuetext</code> : new	101
<code>\dynamicframeheight</code> : new ...	105	<code>\ffdynamicpostcontinued</code> : new	102
<code>\dynamicframewidth</code> : new	105	<code>\ffevenoffset</code> : removed	206
<code>\dynamicsetexclusion</code> : changed to document command ...	107	<code>\ffpshpar</code> : deprecated	87
<code>\dynamicsetpagelist</code> : changed to document command ...	107	<code>\ffstaticpostcontinued</code> : new .	101
<code>\dynamicswitchoffnextodd</code> : changed to document command	223	<code>\finishthispage</code> : changed to document command	74
<code>\dynamicswitchoffnextodonly</code> : changed to document command	229	<code>\flf@getcontargs</code> : removed ...	108
<code>\dynamicswitchoffnextonly</code> : changed to document command	228	<code>\flf@message</code> : removed	5
<code>\dynamicswitchonnext</code> : changed to document command ...	219	<code>\flf@ta</code> : removed	113
<code>\dynamicswitchonnextodd</code> : changed to document command	220	<code>\flf@tb</code> : removed	113
		<code>\FLForgpar</code> : deprecated	87
		<code>\FLFsimpar</code> : new	87

\flowaddexclusion: changed to document command	81	\getflowevenbounds: changed to document command	119
\flowfram@doc@htmllopts: new	43	\getflowid: changed to document command	79
\flowfram@dynamicfooter: new	159	\getstaticbounds: changed to document command	117
\flowfram@dynamicheader: new	158	\getstaticevenbounds: changed to document command	118
\flowfram@footer: new	159	\getstaticid: changed to document command	84
\flowfram@header: new	158	\globalnormalmargin: changed to document command	76
\flowfram@preamble@htmllopts: new	43	\globalreversemargin: changed to document command	76
\flowfram@thepagenumber: new	12	\hNtone: changed to document command	272
\flowframeshowlayout: changed to document command	73	\hNtoneleft: changed to document command	272
\FlowFramRestoreOR: new	160	\hNtoneright: changed to document command	272
\FlowFramSetDynamicHeadFootAttributes: new	165	\htwotone: changed to document command	270
\FlowFramUnrestoreOR: new	161	\htwotoneleft: changed to document command	271
\flowsetexclusion: changed to document command	81	\htwotoneright: changed to document command	271
\flowsetpagelist: changed to document command	81	\if@contsline: removed	312
\flowswitchoffnext: changed to document command	211	\if@ff@nwpg: replace	176
\flowswitchoffnextodd: changed to document command	212	\if@minitoc: replaced	320
\flowswitchoffnextoddoneonly: changed to document command	219	\if@notthiscol: replaced	176
\flowswitchoffnextonly: changed to document command	218	\if@storetoc: removed	310
\flowswitchonnext: changed to document command	209	\if@ttb@num: removed	11
\flowswitchonnextodd: changed to document command	210	\if@ttb@rotate: removed	10
\flowswitchonnextoddoneonly: changed to document command	215	\ifdfcontinued: removed	108
\flowswitchonnextonly: changed to document command	214	\IfSavedRelativeLocationAbove: new	141
\footnotecolor: deprecated	173	\IfSavedRelativeLocationBelow: new	142
\framebreak: changed to document command	74	\IfSavedRelativeLocationEq: new	141
\getdynamicbounds: changed to document command	119	\IfSavedRelativeLocationLeft: new	142
\getdynamicevenbounds: changed to document command	120	\IfSavedRelativeLocationRight: new	142
\getdynamicid: changed to document command	104	\insertvrule: changed to document command	275
\getflowbounds: changed to document command	118	\labelflow: changed to document command	31
		\labelflowidn: changed to document command	31

<code>\makebackgroundframe</code> : changed to document command . . .	274	<code>\onecolumn</code> : changed to document command	241
<code>\makedfheaderfooter</code> : changed to document command . . .	164	<code>\onecolumnbottom</code> : changed to document command	257
<code>\makethumbtabs</code> : changed to document command	292	<code>\onecolumnbottominarea</code> : changed to document command	258
<code>\Ncolumn</code> : changed to document command	245	<code>\onecolumnDbottom</code> : changed to document command	258
<code>\Ncolumnbottom</code> : changed to document command	262	<code>\onecolumnDbottominarea</code> : changed to document command	259
<code>\Ncolumnbottominarea</code> : changed to document command . . .	263	<code>\onecolumnDtop</code> : changed to document command	248
<code>\NcolumnDbottom</code> : changed to document command	263	<code>\onecolumnDtopinarea</code> : changed to document command . . .	250
<code>\NcolumnDbottominarea</code> : changed to document command . . .	265	<code>\onecolumninarea</code> : changed to document command	242
<code>\NcolumnDtop</code> : changed to document command	254	<code>\onecolumnSbottom</code> : changed to document command	257
<code>\NcolumnDtopinarea</code> : changed to document command	257	<code>\onecolumnSbottominarea</code> : changed to document command	259
<code>\Ncolumninarea</code> : changed to document command	246	<code>\onecolumnStop</code> : changed to document command	248
<code>\NcolumnSbottom</code> : changed to document command	262	<code>\onecolumnStopinarea</code> : changed to document command . . .	249
<code>\NcolumnSbottominarea</code> : changed to document command . . .	265	<code>\onecolumnstop</code> : changed to document command	247
<code>\NcolumnStop</code> : changed to document command	254	<code>\onecolumnstopinarea</code> : changed to document command . . .	249
<code>\NcolumnStopinarea</code> : changed to document command	256	<code>\ps@ffempty</code> : new	172
<code>\Ncolumnstop</code> : changed to document command	254	<code>\ps@ffplain</code> : new	172
<code>\Ncolumnstopinarea</code> : changed to document command	254	<code>\ps@specialffempty</code> : new	173
<code>\newdynamicframe</code> : changed to document command	102	<code>\RefSavedRelativeLocation</code> : new	141
<code>\newframe</code> : changed to document command	248	<code>\relativeframelocation</code> : changed to document command	134
<code>\newstaticframe</code> : changed to document command	83	<code>\reldynamicloc</code> : changed to document command	143
<code>\NoChapterInDynamic</code> : new . . .	290	<code>\relflowloc</code> : changed to document command	143
<code>\oddcheckifframeabove</code> : changed to document command . . .	121	<code>\relstaticloc</code> : changed to document command	143
<code>\oddcheckifframebelow</code> : changed to document command . . .	124	<code>\rotateframe</code> : changed to document command	180
<code>\oddcheckifframeleft</code> : changed to document command . . .	126	<code>\SaveRelativeFrameLocation</code> : new	136
<code>\oddcheckifframeright</code> : changed to document command . . .	128		

<code>\setalldynamicframes</code> : changed to document command . . .	105	<code>\staticswitchonnextodd</code> : changed to document command	231
<code>\setallflowframes</code> : changed to document command	80	<code>\staticswitchonnextoddoneonly</code> : changed to document command	236
<code>\setallstaticframes</code> : changed to document command	85	<code>\staticswitchonnextonly</code> : changed to document command	235
<code>\setdynamicframe</code> : changed to document command	105	<code>\theFramePageCounter</code> : new . .	12
<code>\setframes</code> : changed to document command	144	<code>thisframe</code> : replaced <code>\thepage</code> with <code>\theabsolutepage</code>	31
<code>\setinitialframe</code> : changed to document command	144	<code>\thumbtabhyperlinkformat</code> : new	16
<code>\setmargin</code> : changed to document command	77	<code>\thumbtabindex</code> : changed to document command and added optional argument . .	304
<code>\setstaticcontents</code> : changed to document command	98	<code>\thumbtabnumtitle</code> : new	297
<code>\setstaticframe</code> : changed to document command	85	<code>\thumbtabregularformat</code> : new .	17
<code>\setthumbtab</code> : changed to document command	306	<code>\twocolumn</code> : changed to document command	242
<code>\setthumbtabindex</code> : changed to document command	308	<code>\twocolumnbottom</code> : changed to document command	259
<code>\SetThumbTabTitle</code> : new	294	<code>\twocolumnbottominarea</code> : changed to document command	260
<code>\staticaddexclusion</code> : changed to document command	98	<code>\twocolumnDbottom</code> : changed to document command	260
<code>staticcontents</code> : changed to document environment	95	<code>\twocolumnDbottominarea</code> : changed to document command	262
<code>staticcontents*</code> : changed to document environment	95	<code>\twocolumnDtop</code> : changed to document command	251
<code>\staticframeheight</code> : new	85	<code>\twocolumnDtopinarea</code> : changed to document command . . .	253
<code>\staticframewidth</code> : new	85	<code>\twocolumninarea</code> : changed to document command	243
<code>\staticsetexclusion</code> : changed to document command	98	<code>\twocolumnSbottominarea</code> : changed to document command	262
<code>\staticsetpagelist</code> : changed to document command	98	<code>\twocolumnStop</code> : changed to document command	250
<code>\staticswitchoffnext</code> : changed to document command . . .	232	<code>\twocolumnStopinarea</code> : changed to document command . . .	253
<code>\staticswitchoffnextodd</code> : changed to document command	233	<code>\twocolumntop</code> : changed to document command	250
<code>\staticswitchoffnextoddoneonly</code> : changed to document command	240	<code>\twocolumntopinarea</code> : changed to document command . . .	251
<code>\staticswitchoffnextonly</code> : changed to document command	239	<code>\vNtone</code> : changed to document command	268
<code>\staticswitchonnext</code> : changed to document command . . .	230		

<code>\vNtonebottom</code> : changed to document command	269	<code>'hidethis'</code> keys	39, 535
<code>\vtwotone</code> : changed to document command	266	<code>\@@newstaticframe</code> : added <code>'hide'</code> and <code>'hidethis'</code> attributes . .	545
<code>\vtwotonebottom</code> : changed to document command	267	<code>\@putdynamicbox</code> : added check for <code>'hide'</code> and <code>'hidethis'</code> attributes	200, 663
<code>\vtwotonetop</code> : changed to document command	267	<code>\@putstaticbox</code> : added check for <code>'hide'</code> and <code>'hidethis'</code> attributes	192, 658
2014-06-04 – 1.16			
General: added <code>'hide'</code> and			

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			
<code>\\$</code>	162	<code>\@@insert@vrule</code>	277, 495, 497, 693, <u>693</u> , 695
<code>\(</code>	509, 711	<code>\@@insert@hrule</code> . . .	280, 497, 695, <u>696</u>
<code>\)</code>	509, 711	<code>\@@insert@vrule</code> 276, 493, 494, 691, <u>692</u>	
<code>\@@@ff@getrange</code> . .	175, 434, 610, <u>610</u>	<code>\@@newdynamicframe</code> 103, 393, 558, <u>559</u>	
<code>\@@dofooter</code> . . .	204, 475, 477, <u>664</u> , 667	<code>\@@newflowframe</code> . . .	78, 367, 527, <u>527</u>
<code>\@@doheader</code> . . .	203, 474, 477, <u>664</u> , 667	<code>\@@newstaticframe</code>	83, 381, 543, 544, <u>544</u>
<code>\@@dominitoc</code>	317, 512, 513, 716, <u>716</u> , 717	<code>\@@next@Nband</code> .	274, <u>274</u> , 493, 690, <u>690</u>
<code>\@@dothumbtabs</code> 298, 504, 505, 704, <u>706</u>		<code>\@@next@vNband</code>	269, <u>269</u> , 489, 490, 686, <u>686</u>
<code>\@@dynamicframeswapcoords</code>	107, 401, 402, <u>567</u> , 568	<code>\@@par</code>	428, 602
<code>\@@ff@chckifthispg</code>	178, 460, 461, 647, 648, <u>648</u> , 649	<code>\@@setdynamicframe</code> 106, 396, 397, 507, 509, 562, 563, <u>563</u> , 709, 711	
<code>\@@ff@getrange</code> 175, 433, 434, 610, <u>610</u>		<code>\@@setflowframe</code>	80, 374, 375, 536, 537, <u>537</u>
<code>\@@flowfram@current@pre@section@title</code>	87, 88, 90, 91, 163	<code>\@@setfr@mecol</code>	44, 380, 543, <u>543</u>
<code>\@@flowfram@org@output@page</code> 160, <u>162</u>		<code>\@@setframecol</code>	44, 380, 543, <u>543</u>
<code>\@@flowfram@output@page</code> . .	161, <u>162</u>	<code>\@@setstaticframe</code>	86, 383, 384, 547, 548, <u>548</u>
<code>\@@flowframeswapcoords</code>	81, 379, 380, 541, <u>542</u>	<code>\@@sinsert@hrule</code> . .	282, 499, 698, <u>698</u>
<code>\@@hNtone</code> 272, 491, 492, 688, 689, <u>689</u>		<code>\@@sinsert@vrule</code> . .	279, 496, 694, <u>694</u>
<code>\@@hNtoneright</code> . . .	273, 492, 689, <u>689</u>	<code>\@@staticframeswapcoords</code>	99, 392, 393, 556, 557, <u>558</u>
<code>\@@htwotoneleft</code>	270, 490, 491, 687, <u>687</u> , 688	<code>\@@storefileconts</code> 312, 510, 712, <u>713</u>	
<code>\@@htwotoneright</code> .	271, 491, 688, <u>688</u>	<code>\@@ttb@stack</code>	298, 505, 706, <u>706</u>
<code>\@@if@newlist</code>	162, 163, 429, 430, 604, 605	<code>\@@vNtone</code> 269, 489, 490, 685, <u>685</u> , 686	
<code>\@@insert@hrule</code>	280, 498, 500, 697, <u>697</u> , 699	<code>\@@vNtonetop</code>	270, 490, 687, <u>687</u>
		<code>\@@vtwotonebottom</code>	488, 683, 684
		<code>\@@vtwotonetop</code>	267, 488, 684

<code>\@M</code> .. 207, 285–287, 428, 478, 602, 668	<code>\@contslinetrue</code> 512, 715
<code>\@Mi</code> 74, 365, 524	<code>\@currentHref</code> 291
<code>\@Miv</code> 428, 602	<code>\@currentlabel</code> ... 31, 364, 522, 523
<code>\@NcolumnSbottominarea</code> ... 487, 681	<code>\@currenttocentry</code> 289
<code>\@Ncolumnbottominarea</code> . 263, 486, 681	<code>\@currlist</code> 428, 603
<code>\@Ncolumninarea</code> 246, 479, 480, 671, 672	<code>\@dblarg</code> 387, 551
<code>\@Ncolumninarea</code> 255, 483, 676, 677	<code>\@dbldeferlist</code> 429, 603
<code>\@addtotoclist</code> ... 312, 511, 713, 713	<code>\@dblfloat</code> 160, 161
<code>\@afterheading</code> 91, 289, 502, 503, 514, 515, 702, 703, 718, 719	<code>\@dblfloatplacement</code> 208, 209, 429, 478, 603, 668, 669
<code>\@appenddynamic</code> 113, 404, 513, 571, 572, 572, 717	<code>\@dblfpbot</code> 478, 669
<code>\@appendminitocdf</code> 318, 513, 716, 716	<code>\@dblfpsep</code> 478, 669
<code>\@auxout</code> 43, 88, 137	<code>\@dblfpstop</code> 478, 669
<code>\@beginvdi</code> 163, 477, 668	<code>\@dbltoplist</code> 429, 603
<code>\@beginstaticcontents</code> 95, 389, 390, 554	<code>\@dbltopnum</code> 478, 669
<code>\@botlist</code> 428, 603	<code>\@dbltoproom</code> 478, 669
<code>\@capttype</code> 241, 478, 669	<code>\@deferlist</code> 428, 429, 603
<code>\@ccclv</code> 428, 429, 431, 603, 606	<code>\@df@chckifthispg</code> 179, 442–449, 461, 473, 474, 622–632, 648, 663, 664
<code>\@chapapp</code> 18, 285, 286	<code>\@df@checkuniqueidl</code> 104, 394, 560, 560
<code>\@chapter</code> 364, 502, 514, 522, 702, 718	<code>\@df@setoffset</code> 188, 466, 472, 473, 655, 662
<code>\@checkcontentsline</code> 315, 511, 512, 713, 715	<code>\@doclearpage</code> 158, 428, 603
<code>\@checkifframeabove</code> 121, 411, 424, 581, 597	<code>\@docolbbox</code> ... 181, 462–464, 650, 652
<code>\@checkifframebelow</code> 123, 412, 413, 424, 583, 597	<code>\@docolbox</code> ... 180, 461, 463, 649, 651
<code>\@checkifframeleft</code> 125, 414, 424, 585, 597	<code>\@dodfparbox</code> .. 196, 471, 472, 661, 662
<code>\@checkifframeright</code> 127, 416, 424, 587, 588, 597	<code>\@dodynamicbbox</code> 199, 473, 474, 662, 664
<code>\@colN</code> 365, 369, 370, 374, 375, 379, 380, 382–384, 392, 393, 395–397, 401, 402, 432, 459, 460, 468, 471, 475, 476, 480, 484, 487, 503, 504, 513, 523–525, 529, 530, 535–537, 541, 542, 545– 548, 557, 558, 560–563, 568, 607, 608, 647, 657, 660, 665, 666, 672, 677, 678, 682, 704, 717	<code>\@dodynamicbox</code> 198, 472–474, 661, 663
<code>\@colht</code> 157, 160, 161, 163, 427–429, 431, 478, 602, 603, 606, 668, 669	<code>\@dodynamicthefoot</code> 160, 164, 207, 358, 359, 429, 430, 477, 604, 605, 668
<code>\@colroom</code> 157, 161, 163, 427, 428, 602, 603	<code>\@dodynamicthehead</code> 160, 164, 207, 357, 358, 429, 430, 477, 604, 605, 668
<code>\@combinedblfloats</code> 207, 478, 668	<code>\@dominitoc</code> ... 317, 512, 513, 716, 717
<code>\@combinefloats</code> 431, 606	<code>\@dostaticbbox</code> 191, 469–471, 658, 659, 660
<code>\@computeffarea</code> .. 116, 406, 574, 575	<code>\@dostaticbox</code> . 190, 468, 470, 657, 659
<code>\@contslinefalse</code> 512, 715	<code>\@dothefoot</code> ... 160, 164, 204, 358, 359, 429, 430, 475, 604, 605, 665
	<code>\@dothehead</code> ... 159, 164, 204, 357, 358, 429, 430, 475, 604, 605, 664
	<code>\@dothumbtabs</code> . 294, 502, 503, 702, 704
	<code>\@dynamicchap</code> 290, 500, 501, 699, 700, 701
	<code>\@dynamicframeid</code> 104, 395, 396, 401, 404, 411, 430, 442–445, 447, 449, 450, 496, 497, 499–501, 507–509, 513, 561, 561, 562,

568, 571, 572, 580, 605, 622, 623, 625, 626, 628, 629, 633, 695, 698, 699, 701, 709–711, 717	\@dynamicframeswapcoords	\@ff@addtolist
..... 108, 401, 568, 568 113, 404, 405, 459, 572, 572, 646	\@ff@adjh
\@dynamicswitchoffnext 265, 487, 488, 683	\@ff@box
..... 223, 444, 445, 625, 625 182, 462, 468, 469, 472, 473, 475, 650, 650, 657, 658, 662, 664, 665	\@ff@chkifthispg 177, 365, 434– 441, 460, 462, 463, 467, 524, 611–620, 647, 647, 651, 652, 656
\@dynamicswitchoffnextodd 411–422, 582–595	\@ff@check
..... 225, 445, 446, 626, 627 173, 431, 459, 607, 646	\@ff@checkifmoreframes
\@dynamicswitchoffnextoddoneonly 175, 433, 609, 609	\@ff@checknumrange 175, 433, 609, 609
..... 230, 450, 633, 634 175, 432, 433, 607, 608, 608	\@ff@checkpages
\@dynamicswitchoffnextonly 175, 433, 608, 608 175, 432, 433, 607, 608, 608
..... 229, 449, 450, 633, 633	\@ff@checkthispage 175, 433, 608, 608	\@ff@checkuniqueidl 79, 369, 529, 529
\@dynamicswitchonnext	\@ff@disablecolor 15, 362, 363, 520, 520, 521
..... 220, 442, 443, 622, 623 94, 97, 197, 389, 390, 472, 553, 555, 661	\@ff@disablesec
\@dynamicswitchonnextodd 206, 425, 476, 478, 599, 666, 668	\@ff@do@allframes
..... 222, 443, 444, 623, 624	\@ff@doafter 508, 710
\@dynamicswitchonnextoddoneonly	\@ff@doalldynamics 205, 476, 477, 666, 667
..... 228, 447, 448, 629, 631	\@ff@doalldynamicsbbox 205, 476, 477, 666, 667
\@dynamicswitchonnextonly	\@ff@doallflowframes 204, 475, 477, 665, 667
..... 226, 447, 628, 629	\@ff@doallflowframesbbox 205, 475, 477, 665, 667
\@dynamictok .. 108, 402, 403, 569–571	\@ff@doallstatics 205, 475, 477, 665, 667
\@ehb 205, 476, 477, 666, 667	\@ff@doallstaticsbbox
428, 603 206, 476, 477, 666, 667 205, 476, 477, 666, 667
\@empty 428, 429, 431, 459, 603, 606, 646	\@ff@dotypeblock 206, 476, 477, 666, 667
\@emptycol 206, 476, 477, 666, 667	\@ff@drawmargins
428, 602 190, 468, 477, 657, 667 161
\@endfortrue .. 432, 460, 607, 608, 647	\@ff@enablecolor 14, 362, 363, 520, 520, 521
\@endpart 182, 461, 462, 469, 472, 649, 650, 657, 662	\@ff@fbox
502, 515, 703, 719 116, 407, 410, 411, 495, 498, 576, 579, 580, 693, 697	\@ff@getdim
\@endstaticcontents 117, 409–411, 577, 579, 580	\@ff@getevendim
..... 97, 389, 390, 554, 555 69, 368, 375, 379, 384, 392, 397,	\@ff@getrange
\@evencheckifframeabove 70, 433, 609, 609
..... 130, 411, 417, 418, 581, 589, 590	\@ff@getch@pter 364, 522
\@evencheckifframebelow	\@ff@getchapter 364, 522
..... 131, 413, 419, 583, 591, 592	\@ff@getchapter 364, 522
\@evencheckifframeleft	\@ff@OLD@makechapterhead 283, 501, 700	\@ff@OLD@makechapterhead
..... 132, 414, 420, 421, 585, 592, 593 283, 501, 700 283, 501, 700
\@evencheckifframeright		
..... 133, 416, 421, 422, 588, 594, 595		
\@evenfoot 162, 167–170, 172, 477, 668		
\@evenhead 162, 166–170, 172, 477, 668		
\@expl@@@mark@update@dblcol@structures@@		
..... 161		
\@expl@@@mark@update@singlecol@structures@@		
..... 161		
\@ff@@@getrange		
..... 70, 433, 609, 609		
\@ff@OLD@ch@pter		
..... 364, 522		
\@ff@OLD@chapter		
..... 364, 522		
\@ff@OLD@schapter		
..... 364, 522		
\@ff@OLD@makechapterhead 283, 501, 700		
\@ff@OLD@makeschapterhead		
..... 283, 501, 700		

402, 433, 461, 528, 536, 542,
 548, 558, 563, 568, 609, 609, 648
 \ff@getrangegreater
 175, 433, 434, 610, 610
 \ff@getrangeless
 175, 433, 434, 609, 610
 \ff@getshape
 . 93, 388, 390, 471, 553, 554, 661
 \ff@getstaticpos
 . 190, 384, 397, 408–410, 468,
 469, 472, 473, 514, 548, 563,
 577–579, 657, 657, 658, 662, 717
 \ff@greyscale ... 505–507, 707, 709
 \ff@heading .. 87, 387, 388, 551, 552
 \ff@id 374,
 375, 379, 383, 384, 392, 396,
 397, 401, 402, 406, 407, 434–
 459, 536, 541, 542, 547, 548,
 557, 558, 562, 563, 568, 574,
 575, 611–629, 631–641, 643–646
 \ff@left@evenx
276, 493, 495, 499, 692–694, 698
 \ff@left@eveny 276,
 493, 495, 496, 499, 692–694, 698
 \ff@left@height
276, 494, 495, 498, 692, 693, 697
 \ff@left@width
276, 494, 495, 498, 692, 693, 697
 \ff@left@x 276, 493,
 495, 498, 499, 692–694, 697, 698
 \ff@left@y 276, 493, 495,
 496, 498, 499, 692–694, 697, 698
 \ff@moreframesfalse 432, 607
 \ff@moreframestrue 433, 609
 \ff@nextid 403, 569, 570
 \ff@nextpages
 437, 438, 445, 446, 453–
 455, 614–616, 626–628, 638–640
 \ff@numend ... 368, 375, 379, 380,
 384, 392, 393, 397, 402, 433,
 434, 461, 528, 536, 537, 542,
 548, 558, 563, 568, 609, 610, 648
 \ff@numstart
 368, 375, 379, 380, 384,
 392, 393, 397, 402, 433, 434,
 460, 461, 528, 536, 537, 542,
 548, 558, 563, 568, 609, 610, 648
 \ff@nwpfalse 459, 647
 \ff@nwptrue 460, 647
 \ff@offset ... 365, 406, 407, 462,
 464–467, 523, 575, 650, 652–656
 \ff@oldparagraph 388, 552
 \ff@oldsection 388, 552
 \ff@oldsubparagraph 388, 552
 \ff@oldsubsection 388, 552
 \ff@oldsubsubsection 388, 552
 \ff@output@adjustframes
 175, 434–459, 610,
 611–617, 619, 621–625, 627–
 629, 631–638, 640, 641, 643–646
 \ff@pages 436, 437, 439–
 441, 444, 445, 448, 449, 453,
 456, 457, 495, 499, 507, 613,
 614, 618–621, 625, 630–632,
 637, 642–644, 694, 697, 698, 709
 \ff@pages@countreg 12,
 362, 365, 426, 427, 432–459,
 461–463, 467, 477, 478, 519,
519, 524, 600, 601, 607, 608,
 611–649, 651, 652, 656, 667, 668
 \ff@parshape 87, 387, 388, 390, 471,
 472, 551, 551, 552, 554, 555, 661
 \ff@pp 368, 432, 433,
 460, 461, 528, 607, 608, 647, 648
 \ff@prepages . 435, 436, 443, 444,
 452, 612, 613, 624, 625, 635, 636
 \ff@rest 403, 569, 570
 \ff@s@t@ovalboxoffset
186, 464–467, 653, 653, 654, 655
 \ff@s@t@defaultoffset
 186, 464–467, 653, 654–656
 \ff@s@t@doubleboxoffset
 464–466, 652, 653–655
 \ff@s@t@ovalboxoffset
 186, 464–467, 652, 653–655
 \ff@sectionhead
 87, 387, 388, 551, 552, 553
 \ff@setoffset
186, 461, 462, 464, 649, 650, 653
 \ff@setsecthead
 . 93, 388, 390, 472, 552, 555, 661
 \ff@staticH .. 426, 481–485, 487,
 489, 490, 492, 493, 495, 498,
 499, 600, 674, 675, 677–680,
 682, 686, 687, 690, 693, 694, 697
 \ff@staticW 485, 488, 489,
 492, 493, 495, 498, 499, 679,
 680, 683, 685, 689, 690, 693, 697

<code>\@ff@swaplen</code>	<code>\@flf@nr</code>	517–520
<u>116</u> , 407, 495, 498, <u>576</u> , 693, 697	<code>\@flf@subsp</code> ...	<u>297</u> , 504, 505, <u>705</u> , 706
<code>\@ff@text</code>	<code>\@flf@subsp@text</code> ..	504, 505, 705, 706
<code>\@ff@tmp@x</code>	<code>\@flf@tmp</code> 402, 504, 505, 569, 705, 706	
381, 384, 392, 394, 397, 401,	<code>\@flf@tmpa</code>	402, 569
406–410, 467–469, 472, 473,	<code>\@flf@tmpb</code>	402, 569
476, 477, 479, 480, 482, 484–	<code>\@flf@val</code>	517–520
493, 495, 496, 498, 499, 505,	<code>\@float</code>	160, 161
506, 523, 541, 544, 548, 556,	<code>\@float@placement</code>	161, 431, 607
557, 559, 563, 567, 574–579,	<code>\@flowfram@build@page@after</code> ...	
656–658, 662, 667, 671, 672,	160, 161, 163
675, 677, 678, 680, 682, 684–	<code>\@flowfram@build@page@before</code> ..	
691, 693, 694, 697, 698, 706–708	161, 163
<code>\@ff@tmp@x@even</code> ...	<code>\@flowfram@checkrelativeloc</code> <u>140</u> , <u>140</u>	
505, 506, 523, 684–691, 706–708	<code>\@flowfram@saverelativeloc</code> <u>137</u> , <u>138</u>	
<code>\@ff@tmp@y</code>	<code>\@flowfram@set@next@vsize</code>	163
379, 381, 384, 392, 394, 397,	<code>\@flowfram@standard@makechapterhead</code>	
401, 406–410, 426, 430, 467–	285
469, 472, 473, 475, 481–483,	<code>\@flowfram@update@col@count</code> ...	
485–493, 495, 496, 498, 499,	145, 161, 163, 208
505–507, 514, 523, 541, 544,	<code>\@flowframe@warn@rerun</code> <u>137</u> , <u>141</u> , <u>141</u>	
548, 557, 559, 563, 567, 574,	<code>\@flowframeid</code> <u>79</u> , <u>369</u> , <u>374</u> , <u>379</u> , <u>406</u> ,	
575, 577–579, 600, 605, 656–	<u>410</u> , <u>434–439</u> , <u>441</u> , <u>442</u> , <u>496</u> ,	
658, 662, 664, 665, 674, 675,	<u>499</u> , <u>500</u> , <u>530</u> , <u>530</u> , <u>536</u> , <u>541</u> ,	
677, 679, 680, 682, 684–691,	<u>574</u> , <u>579</u> , <u>580</u> , <u>611–614</u> , <u>616</u> ,	
693, 694, 697, 698, 706–709, 717	<u>618</u> , <u>621</u> , <u>622</u> , <u>694</u> , <u>695</u> , <u>698</u> , <u>699</u>	
<code>\@ff@tmpN</code>	<code>\@flowframeswapcoords</code>	
496, 497, 499, 500, 503–509,	<u>82</u> , <u>379</u> , <u>541</u> , <u>541</u>
513, 514, 523, 607, 608, 694,	<code>\@flowframtk@def@outlinelua</code> <u>325</u> , <u>327</u>	
695, 698, 699, 704, 706–711, 717	<code>\@flowframtk@def@outlinepdf</code> <u>325</u> , <u>327</u>	
<code>\@ff@type@i</code>	<code>\@flowframtk@def@outlinepst</code> <u>326</u> , <u>327</u>	
494–500, 692–699	<code>\@flowframtk@outline</code>	<u>326</u> , <u>327</u>
<code>\@ff@type@ii</code>	<code>\@flowframtk@provide@textpath</code> .	<u>327</u>
498, 500, 692, 693, 695–697, 699	<code>\@flowswitch@off@next</code> <u>212</u> , <u>436</u> , <u>613</u> , <u>613</u>	
<code>\@ff@background</code>	<code>\@flowswitch@off@next@odd</code>	
<u>13</u> ,	<u>214</u> , <u>437</u> , <u>438</u> , <u>614</u> , <u>615</u>
361, 363, 462, <u>517</u> , 520, 521, 650	<code>\@flowswitch@off@next@odd@only</code>	
<code>\@ff@draft</code>	<u>219</u> , <u>442</u> , <u>621</u> , <u>622</u>
<u>4</u> , 361, 365, <u>516</u> , 517, 524	<code>\@flowswitch@off@next@only</code>	
<code>\@ffnodraft</code>	<u>219</u> , <u>441</u> , <u>621</u> , <u>621</u>
<u>4</u> , 361, <u>516</u> , 517	<code>\@flowswitch@on@next</code> <u>209</u> , <u>434</u> , <u>610</u> , <u>611</u>	
<code>\@filef@und</code>	<code>\@flowswitch@on@next@odd</code>	
510, 712	<u>211</u> , <u>435</u> , <u>611</u> , <u>612</u>
<code>\@firstcolumn@false</code>	<code>\@flowswitch@on@next@odd@only</code>	
208, 478, 668	<u>218</u> , <u>439</u> , <u>440</u> , <u>618</u> , <u>619</u>
<code>\@firstcolumn@true</code>	<code>\@flowswitch@on@next@only</code>	
207, 477, 668	<u>215</u> , <u>438</u> , <u>439</u> , <u>616</u> , <u>617</u>
<code>\@firstofone</code>	<code>\@for</code> .	<u>368</u> , <u>374</u> , <u>375</u> , <u>379</u> , <u>383</u> , <u>384</u> ,
26	<u>392</u> , <u>396</u> , <u>397</u> , <u>401</u> , <u>402</u> , <u>406</u> ,	
<code>\@flf@</code>	<u>407</u> , <u>432–460</u> , <u>508</u> , <u>509</u> , <u>528</u> ,	
402–404, 569–571		
<code>\@flf@check@continued</code> ..		
<u>108</u> , 402, 569		
<code>\@flf@donext@subsp</code>		
504, 705		
<code>\@flf@end@dynamic@contents</code>		
.....		
<u>109</u> , 403, 570		
<code>\@flf@find@end</code> <u>109</u> , 402, 403, 569, 570		
<code>\@flf@get@body</code> <u>108</u> , 402, 403, 569, 570		
<code>\@flf@get@contargs</code>		
<u>109</u> , 403, 570		
<code>\@flf@idn</code>		
402, 403, 569–571		

	536, 541, 542, 547, 548, 557, 558, 562, 563, 568, 574, 575, 607, 608, 611–619, 621–629, 631, 633–641, 643–648, 710, 711		414, 416, 417, 419–421, 423– 425, 434–439, 441–445, 447, 449–451, 453, 455, 456, 458, 493, 497, 500, 513, 527, 536, 541, 543, 547, 551, 555, 557, 558, 562, 568, 571, 574, 579– 581, 583–589, 591, 592, 594, 596, 598, 610, 611, 613, 614, 616, 618, 621–623, 625, 626, 628, 629, 633–637, 640, 641, 644, 645, 691, 695, 699, 716
<code>\@fpbot</code>	478, 669	<code>\@ifundefined</code>	311, 360, 361, 364, 387, 388, 390, 411–422, 431, 500–502, 504, 510, 511, 513–516, 521, 551, 552, 554, 582, 584–595, 606, 700–702, 705, 713, 714, 717–719
<code>\@fpmin</code>	478, 669	<code>\@inserthrule</code>	280, 497, 695, 695
<code>\@fpsep</code>	478, 669	<code>\@insertvrule</code>	276, 493, 691, 691
<code>\@fptop</code>	478, 669	<code>\@killglue</code>	183, 184, 188, 192, 194, 201, 202, 463, 467, 469, 470, 473, 474, 651, 652, 656, 659, 663, 664
<code>\@fr@meifdraft</code>	13, 361, 462, 468, 469, 473, 476, 516, 650, 657, 658, 662, 666	<code>\@kludgeins</code>	431, 606
<code>\@freelist</code>	431, 606	<code>\@latex@warning@no@line</code>	428, 602
<code>\@g@tnextcol</code>	176, 425, 459, 477, 599, 646, 667	<code>\@latexerr</code>	428, 603
<code>\@getdynamicbounds</code>	119, 411, 580, 580	<code>\@l@empty</code>	108, 402, 403, 569, 570
<code>\@getdynamicievenbounds</code>	120, 411, 580, 580	<code>\@listdepth</code>	158, 428, 602
<code>\@getflowbounds</code>	119, 410, 579, 580	<code>\@makechapterhead</code>	283, 285, 286, 501, 700
<code>\@getflowevenbounds</code>	119, 410, 580, 580	<code>\@makecol</code>	173, 428, 429, 431, 602, 604, 606
<code>\@getframeid</code>	47, 396, 561	<code>\@makefcolumn</code>	429, 603
<code>\@getmarginpos</code>	76, 366, 467, 525, 526, 656	<code>\@makeminitocchapter</code>	320, 514, 718, 718
<code>\@getstaticbounds</code>	118, 410, 579, 579	<code>\@makeminitocpart</code>	320, 514, 718, 718
<code>\@getstaticievenbounds</code>	118, 410, 579, 579	<code>\@makeminitocsection</code>	320, 514, 515, 718, 719
<code>\@gettype</code>	315, 511, 512, 714, 715	<code>\@makeschapterhead</code>	283, 287, 501, 700
<code>\@gobble</code>	26, 168–170, 172, 399, 430, 504, 505, 566, 604, 705, 706	<code>\@makespecialcolbox</code>	431, 606
<code>\@gobble@om</code>	88	<code>\@makethumbchapter</code>	293, 502, 702, 702
<code>\@gobble@som</code>	88, 94	<code>\@makethumbpart</code>	293, 502, 702, 702
<code>\@gobble@with@sphack@om</code>	163	<code>\@makethumbsection</code>	293, 502, 503, 702, 703
<code>\@gobble@with@sphack@som</code>	163	<code>\@makethumbtabs</code>	292, 501, 701, 701
<code>\@gobbletwo</code>	168, 170, 172, 459, 647	<code>\@maxdepth</code>	431, 606
<code>\@greyscale</code>	298, 505, 706, 706, 707	<code>\@midlist</code>	431, 606
<code>\@hNtone</code>	272, 491, 688, 688	<code>\@minitocfalse</code>	514, 718
<code>\@hNtoneleft</code>	272, 492, 689, 689	<code>\@minitoctrue</code>	514, 718
<code>\@hNtoneright</code>	273, 492, 689, 689	<code>\@mkboth</code>	168–170, 172
<code>\@height</code>	4		
<code>\@htwotone</code>	270, 490, 687, 687		
<code>\@htwotoneleft</code>	271, 491, 688, 688		
<code>\@htwotoneright</code>	271, 491, 688, 688		
<code>\@ifdefined</code>	396, 561		
<code>\@ifnextchar</code>	43, 368, 380, 382, 394, 403, 493, 496, 497, 499, 501, 529, 543, 545, 560, 570, 691, 694, 695, 698, 701		
<code>\@ifpackageloaded</code>	364, 522, 523		
<code>\@ifstar</code>	367, 374, 379, 380, 383, 387, 391– 393, 396, 401, 404, 406, 410–		

<code>\@mparbottom</code>	76, 161, 366, 431, 525, 607	
<code>\@mparswitchfalse</code>	76, 366, 525	478–488, 490–492, 514, 527, 605, 670–681, 684, 686–689, 718
<code>\@mtoc@old@chapter</code>	514, 718	<code>\@opcol</code> 160, 161, 161, 428, 429, 431, 602–604, 607
<code>\@mtoc@old@endpart</code>	515, 719	<code>\@outputbox</code> 144, 163, 180, 207, 208, 425, 430, 431, 461, 478, 599, 605, 606, 649, 668
<code>\@mtoc@old@part</code>	515, 719	<code>\@outputdblcol</code> 160, 161, 206, 431, 477, 607, 667
<code>\@mtoc@old@sect</code>	515, 719	<code>\@outputpage</code> 144, 160, 161, 161, 207, 208, 425, 429, 478, 599, 603, 604, 668
<code>\@n@wdynamicframe</code>	103, 393, 481, 558, 673	<code>\@parboxrestore</code> 429, 604
<code>\@n@wflowframe</code>	78, 367, 479, 481, 482, 526, 527, 670, 671, 673–675	<code>\@part</code> 502, 515, 702, 703, 719
<code>\@n@wstaticframe</code>	83, 380, 481, 543, 543, 673	<code>\@parttitle</code> 502, 503, 703
<code>\@nameuse</code>	162, 207, 477, 668	<code>\@picbox</code> 475, 665
<code>\@newdynamicframe</code>	103, 393, 558, 558	<code>\@picht</code> 475, 665
<code>\@newflowframe</code>	78, 367, 527, 527	<code>\@printalignedtoc</code> 319, 512, 513, 715, 717
<code>\@newlistfalse</code>	162, 429, 604	<code>\@printtoc</code> 318, 512, 513, 715, 717
<code>\@newstaticframe</code>	83, 380, 381, 543, 544	<code>\@putcolbbox</code> 184, 463, 475, 652, 665
<code>\@nexthNband</code>	274, 492, 493, 689, 690, 690	<code>\@putcolbox</code> 182, 462, 475, 651, 665
<code>\@nexthband</code>	273, 491–493, 687, 688, 690, 690	<code>\@putdynamicbbox</code> 202, 474, 476, 664, 666
<code>\@nextvNband</code>	269, 489, 490, 686, 686, 687	<code>\@putdynamicbox</code> 200, 473, 476, 662, 666
<code>\@nextvband</code>	268, 488–490, 684, 685, 685, 686	<code>\@putmarginbox</code> 188, 467, 468, 656, 657
<code>\@nil</code>	402, 505, 569, 706	<code>\@putstaticbbox</code> 194, 470, 476, 659, 666
<code>\@nnil</code>	504, 505, 705, 706	<code>\@putstaticbox</code> 192, 469, 476, 658, 666
<code>\@nobreakfalse</code>	311, 510, 712	<code>\@relativeframelocation</code> 134, 423–425, 596–598
<code>\@notthiscolfalse</code>	460, 461, 463, 647, 648, 651	<code>\@reldynamicloc</code> 143, 424, 598
<code>\@notthiscoltrue</code>	459–461, 469, 473, 647–649, 658, 663	<code>\@relflowloc</code> 144, 425, 598
<code>\@nxtcol</code>	425, 477, 478, 599, 667, 668	<code>\@relstaticloc</code> 143, 425, 598
<code>\@oddcheckifframeabove</code>	122, 411, 412, 581, 582	<code>\@resetactivechars</code> 162, 429, 604
<code>\@oddcheckifframebelow</code>	124, 413, 583, 584	<code>\@resetdyn@emics</code> 195, 471, 478, 660, 668
<code>\@oddcheckifframeleft</code>	126, 414, 415, 585, 586	<code>\@resetst@tics</code> 195, 471, 478, 660, 668
<code>\@oddcheckifframeright</code>	129, 416, 417, 588, 589	<code>\@reversemarginfalse</code> 76, 366, 525
<code>\@oddfirst</code>	162, 167–170, 172, 477, 668	<code>\@reversemargintrue</code> 76, 366, 525
<code>\@oddfirst</code>	162, 166, 168–170, 172, 477, 668	<code>\@s@ff@heading</code> 87, 387, 551, 551
<code>\@onecolumn</code>	241, 426, 459, 478, 600, 647, 669, 669	<code>\@s@tdynamicframeid</code> 104, 394, 398, 560, 560, 565
<code>\@onecolumninarea</code>	242, 478, 669, 670, 670	<code>\@s@tffcol</code> 13, 361, 363, 462, 517, 520, 521, 650
<code>\@onlypreamble</code>	165, 268–273, 320, 367, 430,	<code>\@s@tffttextcol</code> 13, 361, 363, 390, 426, 431, 462, 517, 520, 521, 555, 599, 606, 650
		<code>\@s@tflowframeid</code> 79, 368, 369, 376, 529, 529, 538
		<code>\@s@tfr@mes</code> 204, 475, 477, 665, 667

<code>\@ststaticframeid</code>	<code>\@sevencheckifframeabove</code>
..... 84 , 382 , 386 , 545 , 545 , 550 129 , 411 , 417 , 581 , 589
<code>\@sappenddynamic</code>	<code>\@sevencheckifframebelow</code>
..... 112 , 404 , 513 , 571 , 572 , 716 130 , 413 , 419 , 583 , 591
<code>\@sappendminitocdf</code> 318 , 513 , 716 , 716	<code>\@sevencheckifframeleft</code>
<code>\@schapter</code> 131 , 414 , 420 , 585 , 592
..... 364 , 522	<code>\@sevencheckifframeright</code>
<code>\@scheckifframeabove</code> 132 , 416 , 421 , 587 , 594
..... 121 , 411 , 423 , 581 , 596	<code>\@sf@chckifthispg</code>
<code>\@scheckifframebelow</code> 179 , 451–457 , 461 ,
..... 123 , 412 , 413 , 423 , 583 , 596	469 , 470 , 634–644 , 648 , 658 , 659
<code>\@scheckifframeleft</code>	<code>\@sf@checkuniqueidl</code> 84 , 382 , 545 , 545
..... 125 , 414 , 423 , 585 , 596	<code>\@sf@mpg</code>
<code>\@scheckifframeright</code> 390 , 554 , 555
..... 127 , 416 , 423 , 587 , 596	<code>\@sf@setoffset</code>
<code>\@scomputeffarea</code> . 116 , 406 , 574 , 574 187 , 465 , 468 , 469 , 654 , 657 , 658
<code>\@sdynamicchap</code> 290 , 500 , 501 , 699 , 700	<code>\@sf@thisframe</code>
<code>\@sdynamicframeswapcoords</code> 390 , 391 , 555
..... 107 , 401 , 568 , 568	<code>\@sflowframeswapcoords</code>
<code>\@sdynamicswitchoffnext</code> 82 , 379 , 541 , 541
..... 223 , 444 , 625 , 625	<code>\@sflowswitchoffnext</code>
<code>\@sdynamicswitchoffnextodd</code> 212 , 436 , 613 , 613
..... 225 , 445 , 626 , 626	<code>\@sflowswitchoffnextodd</code>
<code>\@sdynamicswitchoffnextoddone</code> 214 , 437 , 614 , 614
..... 230 , 450 , 633 , 633	<code>\@sflowswitchoffnextoddone</code>
<code>\@sdynamicswitchoffnextonly</code> 219 , 442 , 621 , 622
..... 229 , 449 , 633 , 633	<code>\@sflowswitchoffnextonly</code>
<code>\@sdynamicswitchonnext</code> 219 , 441 , 621 , 621
..... 220 , 442 , 622 , 622	<code>\@sflowswitchonnext</code> 209 , 434 , 610 , 611
<code>\@sdynamicswitchonnextodd</code>	<code>\@sflowswitchonnextodd</code>
..... 222 , 443 , 623 , 623 211 , 435 , 611 , 612
<code>\@sdynamicswitchonnextoddone</code>	<code>\@sflowswitchonnextoddone</code>
..... 228 , 447 , 629 , 629 218 , 439 , 618 , 618
<code>\@sdynamicswitchonnextonly</code>	<code>\@sflowswitchonnextonly</code>
..... 226 , 447 , 628 , 628 215 , 438 , 616 , 616
<code>\@sect</code>	<code>\@sgetdynamicbounds</code> 119 , 411 , 580 , 580
..... 503 , 515 , 703 , 719	<code>\@sgetdynamicicevenbounds</code>
<code>\@setcol</code> 120 , 411 , 580 , 580
..... 156 , 425–427 , 478 , 599 , 600 , 601 , 668	<code>\@sgetflowbounds</code> . 119 , 410 , 579 , 579
<code>\@setcolbox</code> 179 , 461 , 477 , 478 , 649 , 668	<code>\@sgetflowevenbounds</code>
<code>\@setdynamiccontents</code> 119 , 410 , 580 , 580
..... 112 , 404 , 501 , 571 , 571 , 700	<code>\@sgetstaticbounds</code> 118 , 410 , 579 , 579
<code>\@setdynamicframe</code> 106 , 396 , 562 , 562	<code>\@sgetstaticicevenbounds</code>
<code>\@setflowframe</code> 80 , 374 , 536 , 536 118 , 410 , 579 , 579
<code>\@setfr@mesfalse</code>	<code>\@sinserthrule</code> 282 , 497 , 499 , 695 , 698
..... 144 , 425 , 599	<code>\@sinsertrule</code> 279 , 493 , 496 , 691 , 694
<code>\@setfr@mestrue</code>	<code>\@snewdynamicframe</code> 103 , 393 , 558 , 558
..... 145 , 426 , 599	<code>\@snewflowframe</code> ... 78 , 367 , 527 , 527
<code>\@setframecol</code>	<code>\@snewstaticframe</code> . 83 , 380 , 543 , 543
..... 43 , 377 , 380 , 386 , 399 , 489 ,	<code>\@soddcheckifframeabove</code>
493 , 539 , 543 , 550 , 565 , 685 , 690 122 , 411 , 581 , 582
<code>\@setstaticframe</code> .. 86 , 383 , 547 , 547	
<code>\@setthumbtab</code> . 307 , 508 , 509 , 710 , 710	
<code>\@setthumbtabindex</code> 309 , 509 , 711 , 711	

\@soddcheckifframebelow 557, 579, 634, 635, 637, 638,
..... 124, 413, 583, 584, 640, 641, 645, 694, 695, 698, 699
\@soddcheckifframeleft \@staticframeswapcoords
..... 126, 414, 415, 585, 586 99, 392, 557, 557
\@soddcheckifframeright \@staticscontinueonframe
..... 128, 416, 587, 588 100, 390, 405, 554, 573
\@specialoutput 428, 602 \@staticswitchhoffnext
\@specialpagefalse 162, 207, 477, 668 233, 453, 636, 637
\@specialstyle ... 162, 207, 477, 668 \@staticswitchhoffnextodd
\@srelativeframelocation \@staticswitchhoffnextoddonly ..
..... 134, 423–425, 596, 598 240, 458, 645, 645
\@sreldynamicloc 143, 424, 598 \@staticswitchhoffnextonly
\@srelflowloc 144, 425, 598 240, 458, 644, 645
\@srelstaticloc 143, 425, 598 \@staticswitchchonnext
\@ssetdynamiccontents \@staticswitchchonnextodd
..... 112, 404, 571, 571 231, 451, 634, 635
\@ssetdynamicframe 106, 396, 562, 562 \@staticswitchchonnextodd
\@ssetflowframe ... 80, 374, 536, 536 232, 451, 452, 635, 636
\@ssetstaticframe . 86, 383, 547, 547 \@staticswitchchonnextoddonly ...
\@sstaticconts ... 98, 391, 555, 555 239, 456, 457, 641, 643
\@sstaticframeswapcoords \@staticswitchchonnextonly
..... 99, 392, 557, 557 236, 455, 640, 640
\@sstaticswitchhoffnext \@storefileconts . 311, 510, 712, 712
..... 233, 453, 636, 637 \@storetofalse .. 510, 512, 712, 715
\@sstaticswitchhoffnextodd \@storetocttrue 512, 715
..... 235, 453, 637, 637 \@tempa 403, 570
\@sstaticswitchhoffnextoddonly . \@tempboxa 326, 428, 603
..... 240, 458, 645, 645 \@tempdima 431, 606
\@sstaticswitchhoffnextonly \@textbottom 431, 606
..... 240, 458, 644, 644 \@textfloatsheight 161, 431, 607
\@sstaticswitchchonnext \@textmin 478, 669
..... 231, 451, 634, 634 \@texttop 431, 606
\@sstaticswitchchonnextodd \@tfoot 160, 162, 167,
..... 232, 451, 635, 635 168, 429, 430, 477, 604, 605, 668
\@sstaticswitchchonnextoddonly .. \@thead 159, 162,
..... 239, 456, 641, 641 166, 429, 430, 477, 604, 605, 668
\@sstaticswitchchonnextonly \@themargin 162, 163, 430, 605
..... 236, 455, 640, 640 \@thisstrip 268, 489,
..... 490, 492, 493, 685–687, 689, 690
\@startcolumn 428, 602 \@tmpa 403, 570
\@startdblcolumn 208, 478, 668 \@toplist 428, 603
\@starttoc 19, 146, 292,
310, 315, 316, 502, 510, 702, 712 \@topnewpage 75
\@staticcontinueonframe \@ttb@chapter@level ... 315, 512, 715
..... 100, 389, 405, 554, 573 \@ttb@contstype .. 511, 512, 714, 715
\@staticconts 98, 391, 555, 556 \@ttb@id 508, 509, 710, 711
\@staticframeid \@ttb@level 312, 511, 713, 714
. 84, 382, 383, 389, 392, 410,
451, 453, 455, 456, 458, 496,
499, 500, 546, 546, 547, 554, \@ttb@minitoclevel
..... 311, 510, 511, 713, 714
\@ttb@minitoctype
..... 310, 510, 514, 712, 713, 718

<code>\@ttb@numfalse</code>	362, 518, 519	<code>_</code>	4-6, 10, 11, 13-15, 17-24, 26, 27, 29, 31, 33-35, 39-62, 64-66, 69-73, 75-86, 90-136, 138, 139, 143-148, 150-153, 155, 157-162, 164, 165, 168-173, 176-227, 229-281, 283, 284, 288-313, 315, 316, 318-320, 326, 328-332, 336, 338-342, 353-355, 357-360
<code>\@ttb@numtrue</code>	362, 518, 519	<code>\~</code>	162
<code>\@ttb@old@chapter</code>	502, 702	A	
<code>\@ttb@old@endpart</code>	502, 503, 703	<code>absolutepage</code> (counter)	12, 519
<code>\@ttb@old@part</code>	502, 702, 703	<code>\addcontentsline</code>	288, 289
<code>\@ttb@old@sect</code>	503, 703	<code>\addtocontents</code> 88, 291, 292, 502, 503, 702, 703
<code>\@ttb@old@starttoc</code>	510, 712	<code>\AddToHook</code> 16, 20, 30, 31, 140, 163, 288	
<code>\@ttb@old@tableofcontents</code>	512, 715	<code>\addtolength</code> 114, 145, 146, 405-407, 426, 430, 464, 467, 468, 475, 479-485, 487-493, 495, 496, 498, 499, 505, 507, 573-575, 600, 605, 652, 653, 656, 664, 671, 672, 674, 675, 677-680, 682-690, 693, 694, 697, 698, 706, 709
<code>\@ttb@paragraph@level</code>	315, 512, 715	<code>\adjustcolsep</code>	266, 488, 683
<code>\@ttb@part@level</code>	315, 512, 714	<code>\adjustheight</code> 242, 244, 247, 249, 252, 256, 258, 261, 264, 265, 478-482, 484, 485, 487, 670, 672, 674, 675, 677, 679, 680, 682, 683
<code>\@ttb@rotatefalse</code>	361, 518	<code>\advance</code>	365, 367, 369, 370, 374, 375, 379, 380, 382-384, 392, 393, 395-397, 401, 402, 412-415, 417-422, 425-427, 432-460, 464, 468, 471, 475-478, 480, 484, 487, 490, 493, 503-505, 507-509, 511, 513, 525, 527, 529, 530, 535-537, 541-543, 545-548, 557, 558, 560-563, 568, 582-595, 599-601, 607, 608, 610, 612, 614-624, 626-637, 639-647, 653, 657, 660, 665-669, 672, 677, 682, 686, 690, 704, 706, 707, 709-711, 714, 717
<code>\@ttb@rotatetrue</code>	361, 518	<code>\aftergroup</code>	162, 430, 604
<code>\@ttb@section@level</code>	315, 512, 715	<code>\afterminitocskip</code> 316, 317, 512, 513, 716, 716
<code>\@ttb@stack</code>	297, 504, 505, 705, 706	<code>\afterpage</code>	316, 508, 710
<code>\@ttb@storetoc</code>	311, 510, 712, 712		
<code>\@ttb@subparagraph@level</code> 315, 512, 715		
<code>\@ttb@subsection@level</code> 315, 512, 715			
<code>\@ttb@subsubsection@level</code> 315, 512, 715		
<code>\@ttb@titlefalse</code>	362, 518, 519		
<code>\@ttb@titletrue</code>	362, 518, 519		
<code>\@ttb@toc</code>	510, 511, 713		
<code>\@ttb@type</code>	291, 501, 502, 510, 511, 514, 701, 702, 712, 714, 718		
<code>\@twocolumn</code> 243, 426, 479, 600, 670, 670			
<code>\@twocolumnfalse</code>	160, 289, 366, 525		
<code>\@twocolumninarea</code> 244, 479, 670, 670			
<code>\@twocolumntopinarea</code> 251, 482, 674, 675		
<code>\@twocolumntrue</code>	209		
<code>\@twosidetrue</code>	358-360		
<code>\@vNtone</code>	268, 489, 685, 685		
<code>\@vNtonebottom</code>	269, 490, 686, 686		
<code>\@vNtonetop</code>	270, 490, 686, 687		
<code>\@vtwotone</code>	266, 488, 683		
<code>\@vtwotonebottom</code>	267, 488, 684		
<code>\@vtwotonetop</code>	267, 488, 684		
<code>\@whilesw</code> 208, 428, 429, 478, 602, 603, 668		
<code>\@wholewidth</code>	4		
<code>\@width</code>	4		
<code>\@writefile</code>	88		
<code>\\</code> 10, 298, 387, 399, 505, 551, 566, 706			
<code>_</code>	6, 10, 18, 162, 403, 404, 506, 570, 571, 707, 708		
<code>\~</code>	162		

<code>\alignctofalse</code> ..	15, 25, 310, 510, 512, 712, 715
<code>\alignctrue</code>	25, 310, 509, 712
<code>\and</code>	505, 506, 707, 708
<code>\appenddfminitoc</code>	318, 513, 716
<code>\appenddynamiccontents</code>	112, 404, 571
<code>\appto</code>	17
<code>\arabic</code>	31, 291, 364, 522, 523
<code>\AtBeginDocument</code>	146, 426, 600
<code>\AtEndDocument</code>	141
B	
<code>\backmatter</code>	17
<code>\baselineskip</code>	163, 266, 316, 318, 428, 430, 487, 488, 512, 513, 602, 604, 683, 716
<code>\beforeminitocskip</code>	316, 317, 318, 512, 513, 715, 716
<code>\begin</code>	93, 96–98, 100, 196, 298, 326, 342, 390, 391, 403, 405, 505, 554–556, 570, 573, 706
<code>\begingroup</code> ...	162, 208, 311, 387, 388, 390, 429, 478, 510, 551, 552, 554, 555, 603, 604, 668, 712
<code>\bfseries</code>	285, 286
<code>\bgroup</code>	475, 665
<code>\bool</code>	10–12, 15–19, 22–28, 32, 37, 51, 63, 64, 66–69, 71, 72, 74, 75, 88–92, 140, 142, 144, 147–157, 171–179, 183, 184, 188, 192, 194, 200–202, 207, 209–213, 215–217, 220–227, 230, 232–238, 245, 252, 253, 275, 279, 292, 296, 298–304, 307–310, 314–317, 320, 321, 327–331, 333, 337, 339, 343–347, 349–351
<code>\boolean</code>	461, 464–466, 468, 469, 471–473, 505, 506, 510, 649, 653–658, 660, 662, 663, 707, 708, 713
<code>\bookmark</code>	430, 605
<code>\box</code> ..	51, 55, 56, 96, 163, 183–185, 189, 193–195, 201–204, 289, 353, 354, 428–431, 461, 462, 475, 603, 605, 606, 649, 650, 665
<code>\boxgs</code>	326
<code>\boxmaxdepth</code>	431, 606
<code>\break</code>	387, 551
C	
<code>\c</code>	7–9, 12–14, 33–35, 40, 44, 65–70, 76, 77, 79, 83, 93, 94, 96, 100, 101, 103, 110, 111, 116, 117, 144–146, 148–152, 155, 157, 158, 160, 165, 177, 178, 186–190, 196, 197, 204, 206–209, 214, 216–219, 225, 227–229, 235, 237–240, 243–246, 253–256, 259, 260, 262–264, 266–272, 274–276, 278–282, 292, 294, 296, 297, 303, 305, 306, 309, 313, 315, 329, 331, 341–349, 351
<code>\c@absolute page</code>	28, 74, 146, 362, 365, 426, 519, 524, 600
<code>\c@curpg</code>	176, 425, 459, 460, 477, 478, 599, 646, 647, 667, 668
<code>\c@dbltopnumber</code>	478, 669
<code>\c@displayed frame</code>	161, 364, 523
<code>\c@ffrangenumber</code> ..	175, 433, 434, 609, 610
<code>\c@maxdynamic</code>	50, 51, 102, 103, 105, 164, 196, 205, 299, 300, 365, 393–397, 403, 404, 408–410, 471, 476, 505–507, 523, 558–563, 570–572, 577, 578, 660, 666, 707–709
<code>\c@maxflow</code>	48, 75, 78–80, 145, 146, 148, 150–154, 157, 173–175, 177, 190, 205, 208, 241–263, 265, 364, 365, 367–370, 374, 375, 379–381, 392–394, 401, 402, 408, 409, 426, 427, 432, 459, 460, 468, 475, 479, 482, 485, 486, 522, 524, 527–530, 535–537, 541, 542, 544, 557–559, 568, 576, 578, 600, 601, 607, 608, 647, 657, 665, 671, 675, 680
<code>\c@maxminitoc</code>	312, 312, 314, 315, 510, 511, 713, 713, 714
<code>\c@maxstatic</code>	49, 83–85, 145, 146, 195, 205, 268, 273–275, 278, 279, 281, 282, 365, 380–384, 408, 409, 426, 471, 475, 476, 489, 492, 493, 495, 496, 499, 523, 543–548, 576, 578, 600, 660, 665, 666, 685, 690, 691, 694, 697, 698
<code>\c@maxthumbtabs</code>	

	293, 295, 296, 298, 306, 308,		
	319, 503, 505, 507–509, 513,		
	703, 704, 706, 707, 709–711, 717		
<code>\c@maxtocunits</code> 312, <u>312</u> ,	<code>\color@endbox</code> 158, 159, 429, 604
	319, 510, 511, 513, 713, <u>713</u> , 717	<code>\color@endgroup</code> 431, 606
<code>\c@minitoc</code> 316,	<code>\color@hbox</code> 158, 159, 429, 604
	320, 512, 514, 515, 715, 718, 719	<code>\colorbox</code> 14, 363, 520
<code>\c@page</code>	11, 12, 28, 75–77, 121, 123,	<code>\columnheight</code>	. <u>156</u> , 478–487, 670–
	125, 127, 128, 134, 166–168,		672, 674, 675, 677, 679, 680, 682
	183, 184, 188, 193, 194, 201,	<code>\columnsep</code> 244, 247, 252,
	202, 362, 366, 411, 413, 414,		256, 261, 264–266, 479, 480,
	416, 463, 467, 469, 470, 473,		482, 484, 485, 487, 488, 671–
	474, 476, 508, 519, 525, 526,		673, 675, 677, 678, 680, 682, 683
	581, 583, 585, 587, 588, 651,	<code>\columnwidth</code>
	652, 656, 659, 663, 664, 667, 710		157, 158, 160, 427, 428, 479,
<code>\c@secnumdepth</code>	... 169–171, 285, 286		480, 482, 484–487, 601, 602,
<code>\c@thisframe</code>	74, 75, 77, 79, 144–146,	<code>\computebottomedge</code> <u>113</u> , 267,
	160, 161, 163, 176, 207, 208,		269, 271–274, 405, 488, 489,
	364–366, 368, 425, 426, 431,		491–493, <u>574</u> , 684, 686–689, 691
	432, 477, 478, 522, 524, 526,	<code>\computeflowframearea</code>	. <u>114</u> , 406, <u>574</u>
	528, 599, 600, 606, 607, 667, 668	<code>\computeleftedgeeven</code>	.. <u>113</u> , 114,
<code>\catcode</code> 162		267, 269–272, 274, 295, 405,
<code>\centering</code> 11, 504, 705		406, 488–491, 493, 505, <u>573</u> ,
<code>\chapter</code> 18,		574, 684, 685, 687, 689, 691, 706
	19, 168, 283–285, 291, 364, <u>522</u>	<code>\computeleftedgeodd</code>
<code>\chapterfirstpagestyle</code> <u>113</u> , 114, 266, 267, 269–272,
	19, 20, 364, <u>521</u> , 522		274, 405, 406, 488–491, 493,
<code>\ChapterInDynamic</code> <u>284</u>		<u>573</u> , 574, 684, 685, 687, 689, 691
<code>\chaptermark</code> 19, 168, 169	<code>\computerightedgeeven</code>	. <u>114</u> , 271,
<code>\char</code> 6, 10		273, 406, 491, 492, <u>574</u> , 688, 689
<code>\checkifframeabove</code> <u>120</u> , 411, <u>581</u>	<code>\computerightedgeodd</code>
<code>\checkifframebelow</code> <u>122</u> , 412, <u>583</u>	 <u>114</u> , 271, 273, 295, 406,
<code>\checkifframeleft</code> <u>125</u> , 414, <u>585</u>		491, 492, 505, <u>574</u> , 688, 689, 706
<code>\checkifframeright</code> <u>127</u> , 416, <u>587</u>	<code>\computetopedge</code>
<code>\cleardoublepage</code>	<u>75</u> , 287, 289, 366, <u>525</u>	 <u>113</u> , 114, 267, 270, 405,
<code>\clearpage</code> 73, 75, 147,		406, 489, 490, <u>573</u> , 574, 684, 687
	153, 154, 156, 161, 287, 289,	<code>\contentsline</code> 313, 512, 715
	365, 366, 429, 524, 525, 603, 604	<code>\contentsname</code> 19
<code>\cleartoevenpage</code> <u>75</u>	<code>\continueonframe</code> 8,
<code>\clist</code> 30, 32, 34, 45, 60–67,		95, <u>99</u> , 109, 111, 112, 389, 390,
	80, 81, 85, 99, 105, 107, 114,		402, 403, 405, 554, 569, 570, <u>572</u>
	177–179, 209–212, 214–223,	<code>\copy</code> 326
	225–240, 243, 244, 268–270,	<code>\count</code> 162, 477, 668
	272, 282, 297, 304–306, 309, 322	<code>\count@</code> 365, 432, 433,
<code>\closein</code> 511, 713		435–459, 524, 607, 608, 612–647
<code>\closeout</code> 311	counters:	
<code>\col@number</code> 160, 208, 209	<code>absolute page</code> <u>12</u> , <u>519</u>
<code>\color</code> 3, 14, 360, 363, 516, 520	<code>displayed frame</code> <u>31</u> , <u>523</u>
<code>color (option)</code> <u>519</u>	<code>maxdynamic</code> <u>32</u> , <u>523</u>
<code>\color@begingroup</code> 431, 606	<code>maxflow</code> <u>31</u> , <u>522</u>
		<code>maxstatic</code> <u>31</u> , <u>523</u>
		<code>thisframe</code> <u>31</u> , <u>522</u>

<code>\cs</code>	4-6, 10, 11, 13-15, 17-23, 26, 27, 29, 30, 33-35, 39, 43-62, 65-67, 69-73, 76- 81, 83, 84, 86-88, 91-94, 96, 97, 99-101, 103, 104, 106, 108- 113, 115-135, 138, 140, 146, 147, 149, 152-154, 156-161, 165-173, 176-184, 186-188, 190-192, 194-196, 198-200, 202-207, 209-212, 214, 216, 218-223, 225, 226, 229-231, 233, 234, 236, 237, 239-246, 248, 249, 251, 255, 258, 260, 263, 264, 266-274, 277, 280, 283-285, 287-295, 297, 298, 301-303, 305, 307, 309-311, 313, 315-320, 326, 328-332, 336, 338, 341, 342, 353-358, 360	568-572, 603-605, 607, 609, 610, 612-616, 624-628, 635- 640, 646, 648, 650, 657, 662, 664, 665, 667, 669, 683-689, 692, 694-696, 698, 699, 701- 703, 705, 706, 714, 715, 718, 719
<code>\csletcs</code>	386, 400, 469, 473, 550, 566, 658, 663	
<code>\csname</code>	311, 366-370, 375-388, 390-401, 403, 404, 408-422, 425-427, 430-432, 459-474, 489, 492, 493, 495, 499, 503-508, 510, 511, 513, 514, 526-530, 537-550, 552, 554-557, 559-567, 571, 572, 577-579, 582-595, 599, 601, 602, 605-608, 646-664, 685, 690, 691, 694, 697, 698, 703, 704, 707-710, 712-714, 716, 717	
D		
<code>\dblfloatpagefraction</code>	478, 669	
<code>\dbltopfraction</code>	478, 669	
<code>\DeclareCurrentRelease</code>	3, 515	
<code>\DeclareOptionX</code>	361-363, 517-521	
<code>\DeclareRelease</code>	3, 515	
<code>\DeclareRobustCommand</code>	423-425, 596, 598	
<code>\def</code>	31, 75, 94, 162, 168-170, 172, 241, 361, 363, 364, 366, 368- 375, 379-382, 384, 387-389, 392-394, 397, 402-405, 428- 431, 433-438, 443-446, 452- 455, 459-462, 468, 472, 475, 477, 478, 488-492, 494, 496- 505, 512, 514-516, 520-523, 525, 526, 528-537, 542-545, 548, 551-554, 558, 560, 563,	
<code>\defaultthumbtabtype</code>	291, <u>291</u> , 292, 501, 701, <u>701</u>	
<code>\define@boolkey</code>	361, 518	
<code>\define@choicekey</code>	361, 362, 373, 374, 517-520, 534, 535	
<code>\define@key</code>	370-374, 530-535	
<code>\dfchaphead</code>	<u>282</u> , 500, <u>699</u>	
<code>\dfchapterclearpage</code>	<u>287</u> , 288	
<code>\DFchapterstyle</code>	283, <u>284</u> , 500, 501, 700, <u>700</u>	
<code>\dfcontinuedfalse</code>	402, 569	
<code>\dfcontinuedtrue</code>	403, 569	
<code>\dfEvenFooterStyle</code>	168, <u>168</u> , 169, 170, 172	
<code>\dfEvenHeaderStyle</code>	168, <u>168</u> , 169, 170	
<code>\dfOddFooterStyle</code>	168, <u>168</u> , 169, 170, 172	
<code>\dfOddHeaderStyle</code>	168, <u>168</u> , 169, 170	
<code>\DFschapterstyle</code>	283, <u>284</u> , 500, 501, 700, <u>700</u>	
<code>\dfswapoddeven</code>	107, 401, <u>567</u>	
<code>\dim</code>	14, 30, 32, 42, 52-54, 74, 97, 102, 113-116, 122, 124, 126-133, 136-139, 145, 146, 157, 158, 182, 186-189, 197, 198, 204, 242, 244, 246, 247, 249, 251, 252, 255, 256, 258, 261, 264- 274, 276-282, 288, 295, 301, 302, 329, 335-337, 339, 341-352	
<code>\dimen@</code>	427, 431, 601, 602, 606	
<code>\disablethumbtabs</code>	<u>303</u> , 508, <u>709</u>	
<code>displayedframe (counter)</code>	<u>31</u> , <u>523</u>	
<code>\divide</code>	479, 480, 482, 484, 485, 487, 505, 671, 672, 675, 677, 680, 682, 683, 707	
<code>\do</code>	368, 374, 375, 379, 383, 384, 392, 396, 397, 401, 402, 406, 407, 432-460, 508, 509, 528, 536, 541, 542, 547, 548, 557, 558, 562, 563, 568, 574, 575, 607, 608, 611-619, 621-629, 631, 633-641, 643-648, 710, 711	
<code>\document@default@language</code>	162	

<code>\dominitoc</code>	316,	646–649, 656–658, 662, 694,
	320, 512, 514, 515, <u>716</u> , 718, 719	697, 698, 704, 705, 707, 709, 710
<code>\dp</code>	204, 431, 475, 606, 665	<code>\egroup</code>
<code>draft</code> (option)	<u>517</u>	<code>\else</code> ... 11, 56, 75, 135, 139, 140,
<code>\dynamicaddexclusion</code>	<u>107</u> ,	146, 161, 162, 166–168, 244,
	229, 230, 400, 450, <u>567</u> , 633, 634	247, 252, 256, 261, 265, 267,
<code>dynamiccontents</code> (env.)	<u>108</u> , <u>569</u>	269–274, 362, 363, 365, 366,
<code>dynamiccontents*</code> (env.)	<u>108</u> , <u>571</u>	369, 382, 389, 391, 395, 402–
<code>\dynamicframeevenx</code>		404, 408–424, 426–429, 431–
	<u>105</u> , 396, 408, 410, <u>562</u> , 577, 578	460, 462–464, 467, 469, 470,
<code>\dynamicframeeveny</code>		473, 474, 476–480, 482–493,
	<u>105</u> , 396, 408, 410, <u>562</u> , 577, 578	495, 498, 503–505, 507, 510–
<code>\dynamicframeheight</code> <u>105</u> , 196, 197, 200		513, 518, 521, 524–526, 529,
<code>\dynamicframewidth</code> <u>105</u> , 197, 198, 200		545, 553, 555, 560, 569–572,
<code>\dynamicframex</code> <u>104</u> , 396, 408, <u>562</u> , 577		576–579, 581–597, 600–603,
<code>\dynamicframey</code> <u>104</u> , 396, 408, <u>562</u> , 577		606–620, 622–648, 651, 652,
<code>\dynamicsetexclusion</code>		656, 659, 660, 663, 664, 667,
 <u>107</u> , 399, 400, <u>566</u> , <u>567</u>	668, 671, 672, 675–678, 680–
<code>\dynamicsetpagelist</code>		682, 684, 686–691, 693, 697,
 <u>107</u> , 220–222, 225, 226,	704–706, 709, 712, 714, 715, 717
	228, 399, 400, 442–449, 565,	<code>\emph</code>
	<u>566</u> , 623–625, 627–629, 631, 632	18, 102, 405, 573
<code>\dynamicswitchoffnext</code> . <u>222</u> , <u>444</u> , <u>625</u>		<code>\empty</code>
<code>\dynamicswitchoffnextodd</code>		88, 163
 <u>223</u> , <u>445</u> , <u>626</u>	<code>\emulateonecolumn</code> <u>145</u> , <u>426</u> , <u>600</u>
<code>\dynamicswitchoffnextoddoneonly</code> ..		<code>\emulatetwocolumn</code> <u>145</u> , <u>426</u> , <u>599</u>
 <u>229</u> , <u>450</u> , <u>633</u>	<code>\enableminitoc</code> 320, <u>320</u> , 514, 718, <u>718</u>
<code>\dynamicswitchoffnextonly</code>		<code>\enablethumbtabs</code>
 <u>228</u> , <u>449</u> , <u>633</u>	<u>302</u> , <u>507</u> , <u>709</u>
<code>\dynamicswitchonnext</code> .. <u>219</u> , <u>442</u> , <u>622</u>		<code>\end</code> . 93, 97, 98, 100, 196, 298, 326,
<code>\dynamicswitchonnextodd</code> <u>220</u> , <u>443</u> , <u>623</u>		352, 377, 380, 386, 391, 399,
<code>\dynamicswitchonnextoddoneonly</code> ...		402–405, 433, 434, 489, 493,
 <u>226</u> , <u>447</u> , <u>629</u>	505, 511, 512, 539, 543, 550,
<code>\dynamicswitchonnextonly</code>		555, 556, 565, 569–571, 573,
 <u>225</u> , <u>447</u> , <u>628</u>	609, 610, 685, 690, 706, 713–715
		<code>\end@dblfloat</code>
		160, 161
		<code>\end@float</code>
		160, 161
		<code>\endcsname</code>
		311, 366–370,
		375–388, 390–401, 403, 404,
		408–422, 425–427, 430–432,
		459–474, 489, 492, 493, 495,
		499, 503–508, 510, 511, 513,
		514, 526–530, 537–550, 552,
		554–557, 559–567, 571, 572,
		577–579, 582–595, 599, 601,
		602, 605–608, 646–664, 685,
		690, 691, 694, 697, 698, 703,
		704, 707–710, 712–714, 716, 717
		<code>\enddynamiccontents</code>
		404, 571
		<code>\endgraf</code>
		93, 97, 197
		<code>\endgroup</code>
		162, 208,
		311, 388, 391, 429, 430, 478,
		510, 552, 555, 603, 604, 668, 712

E

<code>\edef</code>	369, 377,
	380, 382, 384, 387, 388, 390,
	394, 395, 397, 402, 403, 411–
	422, 426, 431, 432, 434–436,
	439–441, 443, 444, 448, 449,
	452, 456, 457, 459–461, 467–
	469, 472, 473, 495, 499, 503–
	505, 507, 508, 529, 530, 538,
	543, 545, 546, 548, 551, 552,
	554, 555, 560, 561, 563, 569,
	570, 582, 584–595, 599, 606–
	608, 610, 612, 613, 618–621,
	624, 630–632, 636, 642–644,

<code>\endxdynamiccontents</code>	<code>\ExplSyntaxOff</code>
..... 109 , 402 , 403 , 569 , 570 18 ,
<code>\entry</code> 30 , 161 , 285 , 320 , 321 , 327 , 360
environments:	<code>\ExplSyntaxOn</code>
<code>dynamiccontents</code> 18 , 30 , 163 , 285 , 321 , 322 , 327
<code>dynamiccontents*</code>	
<code>staticcontents</code>	
<code>staticcontents*</code>	
<code>staticfigure</code>	
<code>statictable</code>	
<code>\equal</code> 361 , 363 , 366–374 , 379 , 382–	
384 , 392 , 395 , 396 , 401 , 433 ,	
460 , 464–467 , 481 , 489 , 492 ,	
494 , 496–500 , 502 , 503 , 508 ,	
509 , 511 , 514–516 , 520 , 521 ,	
526 , 528–536 , 541 , 545–547 ,	
557 , 560 , 561 , 563 , 568 , 608 ,	
609 , 648 , 653–656 , 673 , 685 ,	
690 , 692 , 694–696 , 698 , 699 ,	
702 , 703 , 710 , 711 , 714 , 718 , 719	
<code>\evencheckifframeabove</code> 129 , 417 , 589	
<code>\evencheckifframebelow</code> 130 , 419 , 591	
<code>\evencheckifframeleft</code> . 131 , 420 , 592	
<code>\evencheckifframeright</code> 132 , 421 , 594	
<code>\evensidemargin</code>	
..... 113 , 162 , 405 , 476 , 573 , 667	
<code>\exp</code> .. 14 , 41 , 42 , 45 , 46 , 48–51 , 56 ,	
61 , 64 , 65 , 80 , 81 , 85–87 , 90 ,	
96 , 99 , 100 , 105–107 , 110 , 111 ,	
114 , 145 , 157 , 173 , 174 , 176 ,	
180 , 183 , 190 , 199 , 208–212 ,	
214 , 215 , 218–222 , 225 , 226 ,	
228–233 , 235 , 236 , 239 , 240 ,	
245 , 283 , 292 , 296 , 297 , 299–	
302 , 305 , 339 , 340 , 343–352 , 355	
<code>\expandafter</code>	
..... 311 , 363 , 367–	
369 , 375–382 , 384–386 , 390–	
394 , 397–405 , 408–410 , 412–	
422 , 427 , 430 , 433–459 , 461–	
464 , 468–474 , 489 , 492 , 493 ,	
503–508 , 510 , 511 , 513 , 520 ,	
527–529 , 537–541 , 543–545 ,	
548–550 , 554–557 , 559 , 560 ,	
563–567 , 569–572 , 577–579 ,	
582–595 , 601 , 602 , 605 , 609 ,	
611–617 , 619 , 621–625 , 627–	
629 , 631–638 , 640 , 641 , 643–	
646 , 649–652 , 657–664 , 685 ,	
690 , 691 , 703–710 , 712–714 , 717	
	F
	<code>\fboxrule</code> . 73 , 182 , 365 , 462 , 524 , 650
	<code>\fboxsep</code>
 14 , 73 ,
 182 , 363 , 365 , 462 , 520 , 523 , 650
	<code>\ff@angle</code>
 373 ,
 375 , 377 , 384 , 386 , 397 , 399 ,
 534 , 537 , 539 , 548 , 550 , 563 , 565
	<code>\ff@backcol</code>
 361 , 363 ,
 372 , 375 , 377 , 384 , 386 , 397 ,
 399 , 461 , 468 , 472 , 475 , 516 ,
 520 , 534 , 537 , 539 , 548 , 550 ,
 563 , 565 , 649 , 657 , 662 , 664 , 665
	<code>\ff@clear</code>
 373 ,
 375 , 377 , 384 , 387 , 397 , 400 ,
 535 , 537 , 539 , 548 , 550 , 563 , 566
	<code>\ff@col</code> ..
 363 , 372 , 375 , 377 , 384 ,
 386 , 397 , 399 , 461 , 462 , 468 ,
 469 , 472 , 473 , 475 , 520 , 533 ,
 537–539 , 548 , 550 , 563 , 565 ,
 649 , 650 , 657 , 658 , 662 , 664 , 665
	<code>\ff@contents</code>
 471 , 472 , 661 , 662
	<code>\ff@endpage</code>
 503 , 704
	<code>\ff@evenx</code>
 371 ,
 375 , 376 , 384 , 385 , 397 , 398 ,
 531 , 537 , 538 , 548 , 549 , 563 , 564
	<code>\ff@eveny</code>
 371 ,
 375 , 376 , 384 , 385 , 397 , 398 ,
 532 , 537 , 538 , 548 , 549 , 563 , 564
	<code>\ff@frame</code>
 372 , 375 , 384 ,
 385 , 397 , 533 , 537 , 548 , 563 , 564
	<code>\ff@frametype</code>
 372 , 375–377 ,
 384–386 , 397 , 398 , 461 , 468 ,
 469 , 472 , 533 , 537 , 538 , 548 ,
 549 , 563 , 565 , 649 , 657 , 658 , 662
	<code>\ff@height</code>

 370 , 375 , 376 , 384 , 385 , 397 ,
 398 , 531 , 537 , 548 , 549 , 563 , 564
	<code>\ff@hide</code>
 374 ,
 375 , 378 , 384 , 386 , 397 , 400 ,
 535 , 537 , 540 , 548 , 550 , 563 , 566
	<code>\ff@hidethis</code>
 374 ,
 375 , 378 , 384 , 386 , 397 , 400 ,
 535 , 537 , 540 , 548 , 550 , 563 , 566
	<code>\ff@id</code>
 365 , 369 , 370 ,
 374 , 379 , 382 , 383 , 390 , 392 ,

395, 396, 401, 404, 406, 407, 410, 411, 430, 434–448, 450– 458, 496, 497, 499–501, 507– 509, 514, 523, 530, 536, 541, 546, 547, 554, 557, 561, 562, 568, 571, 572, 574, 575, 579, 580, 605, 611–615, 617–619, 621–631, 633–643, 645, 694, 695, 698, 699, 701, 709–711, 717	\ff@width 370, 375, 384, 385, 397, 398, 530, 537, 548, 549, 563, 564
\ff@label 369, 371, 375, 376, 382, 384, 386, 394, 397, 398, 529, 532, 537, 538, 545, 548–550, 560, 563, 565	\ff@x 370, 375, 376, 384, 385, 397, 398, 467, 531, 537, 538, 548, 549, 563, 564, 656
\ff@margin 366, 367, 373, 375, 377, 384, 387, 397, 400, 467, 526, 534, 537, 539, 548, 550, 563, 566, 656	\ff@xpages 373, 375, 377, 384, 386, 397, 399, 432, 460, 534, 537, 539, 548, 550, 564, 566, 607, 608, 647
\ff@oddx 371, 375, 376, 384, 385, 397, 398, 532, 537, 538, 548, 549, 563, 564	\ff@y 370, 375, 376, 384, 385, 397, 398, 467, 531, 537, 538, 548, 549, 563, 564, 656
\ff@oddy 371, 375, 376, 384, 385, 397, 398, 532, 537, 538, 548, 549, 563, 564	\FFabove 134, 135, 139, 423, 424, 596, 596, 597
\ff@offset 373, 375, 377, 384, 386, 397, 399, 534, 537, 539, 548, 550, 563, 565	\FFaboveleft 133, 135, 139, 423, 424, 595, 596, 597
\ff@pages 368, 373, 375, 377, 384, 386, 397, 399, 432, 460, 461, 488–493, 503, 504, 528, 534, 537, 539, 548, 550, 563, 565, 607, 608, 647–649, 683–690, 704	\FFaboveright 133, 135, 139, 423, 424, 595, 596, 597
\ff@s@t@doubleboxoffset 185	\ffaddtoadjustframeshook 175, 459, 646
\ff@sehead 387, 388, 551, 552	\ffareaevenx 114, 116, 277, 406, 408, 495, 574, 576, 577, 693
\ff@shape 374, 375, 378, 384, 386, 388–390, 397, 399, 471, 535, 537, 539, 548, 550, 553–555, 564, 565, 661	\ffareaeveny 114, 117, 406, 408, 574, 576, 577
\ff@style 373–375, 377, 384, 387, 397, 399, 400, 471, 472, 535, 537, 539, 548, 551, 563, 566, 661, 662	\ffareaheight 114, 115, 117, 122, 124, 129–131, 138, 139, 277, 280, 406–410, 412– 414, 418–420, 495, 498, 574– 579, 582–585, 590–592, 693, 697
\ff@txtcol 363, 372, 375, 377, 384, 386, 390, 397, 399, 426, 431, 461, 462, 472, 475, 520, 533, 537, 539, 548, 550, 555, 563, 565, 599, 606, 649, 650, 662, 664, 665	\ffareaawidth 114, 115, 117, 126–129, 132, 133, 138, 139, 277, 280, 281, 406–410, 415, 417, 420–422, 495, 498, 574– 579, 586–589, 593–595, 693, 697
468, 535, 537, 540, 549, 564, 657	\ffareax 114–117, 126–129, 132, 133, 138, 139, 277, 280, 281, 406–410, 415– 417, 420–422, 495, 498, 574– 578, 586–589, 593–595, 693, 697
	\ffareay 114– 117, 122, 124, 129–131, 138, 139, 277, 278, 280, 281, 406– 414, 418–420, 495, 498, 574– 578, 582–585, 590–592, 693, 697
	\FFbelow 134, 135, 140, 423, 424, 596, 596, 597
	\FFbelowleft 133, 135, 139, 423, 424, 595, 596, 597
	\FFbelowright 133, 134, 135, 140, 423, 424, 595, 596, 596, 597

<code>\ffchapterdefaultfont</code>	. 286, 286, 287	393, 395, 397, 402–404, 406–
<code>\ffchapterheadstyle</code>	... 285, 286, 287	429, 431–460, 463, 464, 467,
<code>\ffchapternamenum</code>	... 286, 286	468, 470–472, 474, 476–493,
<code>\ffchapternamenumfont</code>	... 286, 286	495, 498, 499, 504–508, 510–
<code>\ffchapterpostheadskip</code>	285, 286, 287	512, 514, 517–521, 524–526,
<code>\ffchapterpostnamenum</code>	... 286, 286	528–530, 537, 542, 545, 546,
<code>\ffchapterpreheadskip</code>	. 285, 286–288	548, 553, 555, 558, 560, 561,
<code>\ffchaptertitlefont</code>	... 286, 286, 287	563, 568–572, 574–579, 581–
<code>\ffcolumseprule</code>	... 275, 278,	604, 606–648, 651, 652, 656,
	281, 493, 495, 499, 691, 694, 697	657, 659–661, 663, 664, 667,
<code>\ffcontinuedtextfont</code>	... 102, 102, 405, 573, 573	668, 670–672, 674, 675, 677–
	109, 110, 403, 405, 570, 573, 573	682, 684, 686–691, 693, 694,
<code>\ffcontinuedtextlayout</code>	100, 102,	697, 698, 704–710, 712, 714–717
<code>\ffdefaultcontinuetext</code>	... 101, 101	<code>\file</code> ... 353
<code>\ffdefaultdynamiccontinuetext</code>	... 101, 111	<code>final (option)</code> ... 517
<code>\ffdefaultpostcontinued</code>	... 102, 102	<code>\finishthispage</code> ... 73, 74, 145,
<code>\ffdefaultstaticcontinuetext</code>	... 101, 101	160, 365, 426, 524, 524, 599, 600
<code>\ffdynamicpostcontinued</code>	... 101, 102	<code>\firstmark</code> ... 430, 605
<code>\ffevenoffset</code>	. 206, 476, 477, 666, 667	<code>\flf@doifverbose</code> ... 5,
<code>\ffhrule</code>	... 281, 282, 499, 697, 698	6, 26, 176, 361, 459, 517, 517, 646
<code>\fflabelfont</code>	... 4, 13, 361, 516, 516	<code>\flf@getcontargs</code> . 108, 403, 569, 570
<code>\fflabelsep</code>	... 3, 13, 360, 361, 516, 516, 516	<code>\flf@message</code> ... 5, 361, 378, 391,
<code>\FFleft</code>	133, 135, 140, 423, 424, 595, 597	400, 401, 427, 461, 463, 517,
<code>\FFoverlap</code>	... 134, 135, 140, 423, 424, 596, 597	517, 540, 556, 567, 601, 649, 651
<code>\ffprechapterhook</code>	20, 20, 364, 522, 522	<code>\flf@messinfo</code> ... 459, 646, 647
<code>\ffpshpar</code>	... 87,	<code>\flf@next</code> ... 269, 274,
	387, 390, 472, 551, 551, 555, 661	387, 388, 403, 489, 490, 493,
<code>\FFright</code>	... 134, 135, 140, 423, 424, 596, 597	505, 551, 552, 570, 686, 690, 706
<code>\ffruledclarations</code>	... 275, 278,	<code>\flf@ta</code> ... 113, 402, 404, 405,
	281, 493, 495, 499, 691, 694, 697	504, 511, 512, 569, 572, 705, 714
<code>\ffstaticpostcontinued</code>	... 101, 101	<code>\flf@tb</code> ... 113, 404,
<code>\ffswapoddeven</code>	... 81, 379, 541	405, 504, 511, 512, 572, 705, 714
<code>\fftolerance</code>	... 156, 157, 427, 601, 601, 602	<code>\FLFabovefalse</code> ... 122, 130, 139,
<code>\ffvadjusttrue</code>	... 241, 478, 669	412, 418, 419, 582, 583, 590, 591
<code>\ffvrule</code>	. 278, 279, 495, 496, 694, 694	<code>\FLFabovetrue</code> ... 122, 130, 139,
<code>\fi</code>	... 11, 56, 75, 135, 139,	412, 418, 419, 582, 583, 590, 591
	140, 146, 161, 162, 166–171,	<code>\FLFbelowfalse</code> ... 124, 125, 131, 139, 413,
	206, 208, 242, 244, 245, 247,	414, 419, 420, 584, 585, 591, 592
	249, 252, 253, 256, 258, 261,	<code>\FLFbelowtrue</code> . 124, 131, 139, 413,
	264, 265, 267, 269–274, 285,	414, 419, 420, 584, 585, 591, 592
	286, 311, 361–366, 368–370,	<code>\FLFleftfalse</code> . 126, 127, 132, 139,
	375, 380, 382–384, 389–391,	415, 416, 421, 586, 587, 593, 594
		<code>\FLFlefttrue</code> . 126, 127, 132, 139,
		415, 416, 421, 586, 587, 593, 594
		<code>\FLForgpar</code> ... 87, 387,
		388, 390, 391, 551, 552, 554, 555
		<code>\FLFrightfalse</code> ... 129, 133, 139,
		417, 422, 423, 588, 589, 594, 595

<code>\FLFrightrue</code>	128, 129, 133,	<code>\FlowFramRestoreOR</code>	160
	139, 417, 422, 588, 589, 594, 595	<code>\flowframsectionheader</code>	169–171
<code>\FLFs impar</code>	87, <u>87</u> , 93, 96, 197	<code>\FlowFramSectionUnit</code>	<u>88</u> , 92
<code>\flowadde xclusion</code>	<u>81</u> , 218,	<code>\FlowFramSetDynamicHeadFootAttributes</code>	<u>165</u> , 171–173
	219, 378, 441, 442, <u>540</u> , 621, 622	<code>\flowframsetup</code>	29
<code>\flowfram</code>	4,	<code>\flowframsubsectionheader</code>	170, 171
	11, 13, 17–22, 30, 40–42, 44–	<code>\FlowFramTkDeclareKeys</code>	<u>322</u>
	50, 52–69, 71–73, 77–79, 81–	<code>\flowframtkencapobject</code>	<u>327</u> , <u>352</u>
	88, 90, 92, 96–101, 103–108,	<code>\flowframtkendobject</code>	<u>328</u>
	110–113, 116, 117, 136, 137,	<code>\flowframtkimageinfo</code>	<u>356</u>
	145, 147, 148, 152–154, 157,	<code>\flowframtkimgtitlechar</code>	<u>354</u> , <u>355</u>
	163, 164, 166–171, 173, 174,	<code>\flowframtkMakeDFHeaderFooter</code>	<u>360</u>
	176–203, 208, 245, 249, 268,	<code>\flowframtkNewDynamicStyle</code>	<u>357</u>
	273, 275, 278, 281, 282, 288,	<code>\flowframtkNewFrameBorder</code>	<u>356</u>
	290, 296, 297, 299, 300, 303,	<code>\flowframtkpdftransmod</code>	<u>326</u>
	304, 306, 315, 317–319, 357–360	<code>\flowframtkSetCreationDate</code> <u>355</u> , <u>356</u>	
<code>\flowfram@dfchap@sbox</code>	287–289	<code>\flowframtkSetDynamicEvenFoot</code>	<u>359</u>
<code>\flowfram@doc@html opts</code>	43, <u>43</u>	<code>\flowframtkSetDynamicEvenHead</code>	<u>358</u>
<code>\flowfram@dynamic footer</code> <u>159</u> , <u>167</u> , <u>168</u>		<code>\flowframtkSetDynamicOddEvenFoot</code>	<u>359</u>
<code>\flowfram@dynamic header</code> <u>158</u> , <u>166</u> , <u>167</u>		<code>\flowframtkSetDynamicOddEvenHead</code>	<u>357</u>
<code>\flowfram@ footer</code>	<u>159</u> , <u>160</u>	<code>\flowframtkSetDynamicOddFoot</code>	<u>358</u>
<code>\flowfram@ header</code>	<u>158</u> , <u>159</u>	<code>\flowframtkSetDynamicOddHead</code>	<u>357</u>
<code>\flowfram@ preamble@html opts</code>	43, <u>43</u>	<code>\FlowFramTkSetKey</code> <u>322</u> , <u>322</u> , <u>323</u> – <u>325</u>	
<code>\flowfram@ thepagenumber</code>	<u>12</u> , <u>137</u>	<code>\flowframtkSetTitle</code>	<u>355</u> , <u>356</u>
<code>\flowframchapter header</code>	<u>169</u>	<code>\flowframtkstartobject</code>	<u>328</u>
<code>\flowfram ecol</code>		<code>\flowframtkUseDynamicStyle</code>	<u>357</u>
 <u>12</u> , 14, 15, 44, 363, 368,	<code>\flowframtkUseDynamicStyleCsName</code>	<u>357</u>
	381, 394, 520, 521, 528, 544, 559	<code>\flowframtkUseFrameBorderCs</code>	<u>356</u>
<code>\flowframe evenx</code>		<code>\flowframtkUseFrameBorderCsName</code> <u>356</u>	
 <u>82</u> , 380, 408, 409, <u>542</u> , 576, 578	<code>\flowframtkutils</code>	
<code>\flowframe eveny</code> <u>328</u> – <u>331</u> , <u>336</u> , <u>338</u> , <u>340</u> , <u>352</u>	
 <u>82</u> , 380, 408, 409, <u>542</u> , 576, 578	<code>\FlowFramTkUtilsOverlayEncapSetup</code>	<u>352</u>
<code>\flowframe height</code>	<u>82</u> , 115,	<code>\FlowFramTkUtilsSetOverlayEncap</code> <u>352</u>	
	380, 406–409, <u>543</u> , 575, 576, 578	<code>\FlowFramUnrestoreOR</code>	<u>161</u>
<code>\flowframerule</code>	<u>73</u> , 182, 186,	<code>\flowsetexclusion</code>	
	365, 462, 464, <u>524</u> , 650, 652, 653 <u>81</u> , 377, 378, 539, <u>540</u>	
<code>\flowframesep</code>	<u>73</u> , 182, 186,	<code>\flowsetpagelist</code>	
	365, 462, 464, <u>523</u> , 650, 652, 653 <u>81</u> , 153–156, 209–212,	
<code>\flowframeshow layout</code> <u>73</u> , <u>365</u> , <u>524</u> , <u>524</u>		214, 215, 218, 377, 378, 434–	
<code>\flowfram textcol</code>		441, 539, <u>540</u> , 611–617, 619, 621	
 <u>12</u> , 14, 15, 44, 363, 368,	<code>\flowswitchoffnext</code>	<u>211</u> , 436, <u>613</u>
	381, 394, 520, 521, 528, 544, 559	<code>\flowswitchoffnextodd</code>	<u>212</u> , 437, <u>614</u>
<code>\flowfram width</code>	<u>82</u> , 115,	<code>\flowswitchoffnextoddone ly</code>	
	380, 406–409, <u>543</u> , 575, 576, 578 <u>219</u> , 442, <u>621</u>	
<code>\flowfram ex</code>	<u>82</u> ,	<code>\flowswitchoffnextonly</code> <u>218</u> , 441, <u>621</u>	
	115, 380, 406–408, <u>542</u> , 574–576		
<code>\flowframe y</code>	<u>82</u> ,		
	115, 380, 406–408, <u>542</u> , 574–576		
<code>\flowfram headerchapprefix</code>	18, 169		
<code>\flowfram headersep</code>	18, 169–171		

<code>\flowswitchonnext</code>	209 , 434 , 610	649 , 658 , 660 , 663 , 668 , 669 ,
<code>\flowswitchonnextodd</code> . .	210 , 435 , 611	685 , 690 , 691 , 707 , 708 , 714 , 715
<code>\flowswitchonnextoddone</code>	215 , 439 , 617	<code>\globalnormalmargin</code>
<code>\flowswitchonnextonly</code> .	214 , 438 , 616	76 , 77 , 366 , 367 , 525 , 526
<code>\fontdimen</code>	186 , 464 , 652 , 653	<code>\globalreversemargin</code>
<code>\footins</code>	428 , 431 , 603 , 606	76 , 77 , 366 , 367 , 525 , 526
<code>\footnotecolor</code> . . .	173 , 431 , 606 , 606	<code>\glossary</code>
<code>\footnoterule</code>	431 , 606	88 , 163 , 430 , 604
<code>\footskip</code>	164 , 204 , 430 , 475 , 605 , 664	<code>\group</code>
<code>\fp</code>	298 – 300	14 , 31 , 73 , 74 ,
<code>\frame</code>	361 , 516	102 , 109 , 182 , 204 , 317 , 352 , 353
<code>\framebreak</code> . .	7 , 74 , 365 , 427 , 524 , 602	
<code>\frontmatter</code>	17	
		H
		<code>\hangafter</code>
		387 , 388 , 551 , 552
		<code>\hangindent</code>
		387 , 388 , 551 , 552
		<code>\hb@xt@</code> 462 – 464 , 468 , 470 , 473 – 475 ,
		650 – 652 , 657 , 659 , 660 , 663 – 665
		<code>\hbox</code>
		4 , 74 , 75 , 158 , 159 ,
		182 – 185 , 189 , 193 – 195 , 201 –
		204 , 206 , 325 , 326 , 353 , 365 ,
		366 , 425 , 429 , 475 , 476 , 478 ,
		524 , 525 , 599 , 604 , 665 , 666 , 668
		<code>\headheight</code>
		30 , 159 , 164 ,
		204 , 405 , 429 , 430 , 573 , 604 , 605
		<code>\headsep</code>
		30 , 164 ,
		203 , 405 , 430 , 475 , 573 , 605 , 664
		<code>\hfil</code>
		387 , 551
		<code>\hfill</code>
		87 , 102 , 168 ,
		279 , 387 , 405 , 496 , 551 , 573 , 694
		<code>\hNtone</code>
		272 , 491 , 688
		<code>\hNtoneleft</code>
		272 , 492 , 689
		<code>\hNtoneright</code>
		272 , 492 , 689
		<code>\hoffset</code>
		113 , 405 , 573
		<code>\hrule</code>
		4
		<code>\hrulefill</code>
		319 , 320 , 513 , 514 , 717
		<code>\hsize</code>
		157 , 160 , 163 ,
		368 , 425 , 427 , 528 , 599 , 601 , 602
		<code>\hskip</code>
		4 , 13 , 361 , 516
		<code>\hspace</code>
		87 , 282 , 499 , 698
		<code>\hss</code>
		182 – 185 , 190 ,
		193 – 195 , 201 – 204 , 206 , 462 –
		464 , 468 , 470 , 471 , 473 – 475 ,
		650 – 652 , 657 , 659 , 660 , 663 – 665
		<code>\ht</code>
		204 , 475 , 665
		<code>\htwotone</code>
		270 , 490 , 491 , 687 , 687
		<code>\htwotoneleft</code>
		271 , 491 , 688
		<code>\htwotoneright</code>
		271 , 491 , 688
		<code>\Huge</code>
		285 , 286
		<code>\huge</code>
		285 , 286
		<code>\hyperlink</code>
		16 , 504 , 705
		<code>\hypersetup</code>
		355

I

<code>\if</code>	56	<code>\IfClassLoadedTF</code>	16
<code>\if@contsline</code>	312, 511, 713, 714	<code>\ifcsdef</code>	162
<code>\if@endfor</code>	432, 607, 608	<code>\ifcsempty</code> 378, 391, 400, 540, 556, 567	
<code>\if@fcolmade</code>		<code>\ifcsundef</code>	399, 566
208, 428, 429, 478, 602, 603, 668		<code>\ifdef</code>	18
<code>\if@ff@moreframes</code>		<code>\ifdefempty</code> 375–378, 384–387, 397–	
173, 431, 432, 459, 607, 608, 646		400, 537–540, 548–551, 564–566	
<code>\if@ff@nwpng</code> ...	176, 459, 477, 646, 667	<code>\ifdefequal</code>	388, 389, 553
<code>\if@filesw</code>	311, 510, 712	<code>\ifdfcontinued</code> 108, 402, 403, 569, 570	
<code>\if@firstcolumn</code>	429, 603	<code>\ifdim</code> ... 365, 407, 412–422, 426–	
<code>\if@mainmatter</code>	169, 285, 286	428, 524, 575, 582–595, 600–602	
<code>\if@minitoc</code>		<code>\iffalse</code>	367, 527
320, 511, 512, 514, 714, 716, 718		<code>\iffvadjust</code> 241, 242, 244, 247, 249,	
<code>\if@newlist</code> 162, 163, 429, 430, 604, 605		252, 256, 258, 261, 264, 478–	
<code>\if@notthiscol</code>		482, 484, 485, 487, 669, 670,	
... 176, 365, 434–449, 451–		672, 674, 675, 677, 679, 680, 682	
457, 459, 460, 462, 463, 467,		<code>\IfFileExists</code>	510, 712
469, 470, 473, 474, 524, 611–		<code>\ifFLFabove</code>	120, 135, 139,
620, 622–632, 634–644, 646,		140, 411, 423, 424, 581, 596, 597	
647, 651, 652, 656, 659, 663, 664		<code>\ifFLFbelow</code>	120, 135, 139,
<code>\if@openright</code>	287	140, 411, 423, 424, 581, 596, 597	
<code>\if@setfr@mes</code>	144, 425, 599	<code>\ifFLFleft</code>	120, 135, 139,
<code>\if@specialpage</code>	162, 477, 668	140, 411, 423, 424, 581, 596, 597	
<code>\if@storetoc</code>	310, 510, 712	<code>\ifFLFright</code>	120, 135, 139,
<code>\if@ttb@num</code>		140, 411, 423, 424, 581, 596, 597	
. 11, 362, 506, 507, 518, 707, 708		<code>\iflefttorightcolumns</code> ..	15, 244,
<code>\if@ttb@rotate</code> 10, 362, 504, 518, 705		247, 252, 256, 261, 264, 265,	
<code>\if@ttb@title</code>		363, 479, 480, 482, 484–487,	
. 12, 362, 506, 507, 518, 707, 708		521, 671, 672, 675, 677, 680, 682	
<code>\if@twocolumn</code>	146, 161	<code>\ifnum</code> 169–171, 285, 286, 365,	
<code>\if@twoside</code> 11, 75, 162, 166–		368–370, 375, 380, 382–384,	
168, 266, 267, 269–274, 366,		390, 392, 393, 395, 397, 402–	
463, 467, 469, 470, 473, 474,		404, 406–409, 425–428, 432,	
476, 477, 488–493, 525, 526,		433, 435–438, 443–446, 452–	
651, 652, 656, 659, 663, 664,		454, 459, 460, 479, 480, 483,	
667, 668, 684, 685, 687–689, 691		484, 486, 487, 489, 493, 495,	
<code>\ifaligntoc</code>	15, 512, 715	498, 503, 504, 507, 508, 511,	
<code>\IfBooleanT</code>	78, 83, 103, 249	513, 524, 528–530, 537, 542,	
<code>\IfBooleanTF</code>		545, 546, 548, 555, 558, 560,	
80, 85, 90, 98, 105, 109, 112,		561, 563, 568, 570–572, 574–	
114, 117–120, 122–132, 134,		578, 599–602, 607–609, 612–	
136, 143, 148–152, 209–212,		616, 624, 626–628, 635, 636,	
214, 215, 218, 219, 221–223,		638, 639, 647, 671, 672, 676,	
225, 226, 229–233, 235, 236,		678, 681, 682, 686, 690, 691,	
239, 240, 276, 280, 283, 284, 318		693, 697, 704, 709, 710, 714, 717	
<code>\ifcase</code>	361, 362, 390,	<code>\ifodd</code>	75, 162, 166–168,
407, 409, 471, 479, 483, 486,		366, 411, 413, 414, 416, 435–	
495, 499, 517–520, 554, 576,		446, 448–454, 456–460, 463,	
578, 661, 671, 676, 681, 694, 697		467, 469, 470, 473, 474, 476,	
		477, 525, 526, 581, 583, 585,	

587, 588, 612, 614, 615, 618–620, 622–624, 626, 627, 630–637, 639, 641–646, 648, 651, 652, 656, 659, 663, 664, 667, 668	<code>\ignorespacesafterend</code> 95, 241
<code>\IfPackageLoadedT</code> 30	<code>\immediate</code> 311, 510, 712
<code>\IfPackageLoadedTF</code> 322	<code>\includeteximage</code> 352
<code>\IfSavedRelativeLocationAbove</code> . 141	<code>\indent</code> 387, 551
<code>\IfSavedRelativeLocationBelow</code> . 142	<code>\index</code> 88, 163, 430, 604
<code>\IfSavedRelativeLocationEq</code> 141	<code>\input</code> 326
<code>\IfSavedRelativeLocationLeft</code> . . 142	<code>\input@path</code> 352
<code>\IfSavedRelativeLocationRight</code> . 142	<code>\inserthrule</code> 279, 497, 695
<code>\ifshowframebbox</code>	<code>\insertvrule</code> . 275, 275, 493, 691, 691
. 4, 206, 361, 477, 516, 667	<code>\int</code> 11, 12, 32–35, 42–51, 53–67, 69–80, 83–85, 93, 94, 96, 97, 101–105, 110, 111, 116, 117, 121, 123, 125, 127, 128, 134, 136, 138, 144–146, 148–158, 160, 161, 164, 165, 171, 173–178, 180, 182–184, 188, 190, 192–196, 198, 200–202, 205, 207–240, 242–248, 252, 253, 255, 256, 263–266, 268–270, 272–276, 278–281, 283, 284, 290, 291, 293–296, 299–302, 304–320, 328–331, 336, 338–343, 357–359
<code>\ifshowmargins</code>	<code>\interlinepenalty</code> 285–287
. 4, 206, 361, 477, 516, 667	<code>\ior</code> 311, 312
<code>\ifshowtypeblock</code>	
. 4, 206, 361, 477, 516, 667	
<code>\ifthenelse</code> 361, 363, 366–374, 379, 382–384, 392, 395, 396, 401, 433, 460, 461, 464–469, 471–473, 481, 489, 492, 494, 496–500, 502, 503, 505, 506, 508, 509, 511, 514–516, 520, 521, 526, 528–536, 541, 545–547, 557, 560, 561, 563, 568, 608, 609, 648, 649, 653–658, 660, 662, 663, 673, 685, 690, 692, 694–696, 698, 699, 702, 703, 707, 708, 710, 711, 714, 718, 719	
<code>\iftrue</code> 367, 527	
<code>\ifundef</code> 377, 378, 386, 399, 539, 550, 565, 566	
<code>\ifusedframebreak</code>	
. 74, 365, 427, 524, 601	
<code>\IfValueT</code> 89, 95, 98, 108, 109, 112, 149–151, 241–251, 253–260, 262, 263, 265, 352	
<code>\IfValueTF</code> 81, 88, 99, 100, 107, 109, 110, 147, 149, 248, 274, 304, 353	
<code>\ifvbox</code> 431, 606	
<code>\ifvmode</code> 74, 365, 524	
<code>\ifvoid</code> . . 428, 431, 462, 603, 606, 651	
<code>\ifx</code> 363, 389, 402, 403, 428, 433, 434, 504, 505, 512, 521, 553, 569, 570, 603, 609, 610, 705, 706, 715	
<code>\ignorespaces</code> . 102, 190, 195, 202, 203, 241, 389, 390, 403, 405, 468, 470, 471, 473, 474, 554, 570, 573, 657, 659, 660, 663, 664	
	J
	<code>\jdrimagebox</code> 325
	<code>\jdroutline</code> 326
	<code>\jobname</code> 311, 510, 712
	K
	<code>\kern</code> 182–185, 189, 193–195, 201–203, 462–464, 468, 470, 471, 473, 474, 476, 650–652, 657, 659, 660, 663, 664, 667
	<code>\keys</code> 16, 20, 29, 30, 35, 40, 89, 321, 322, 330, 331, 333, 335, 352, 353, 356
	L
	<code>\l</code> . . . 10–16, 19, 22, 23, 25, 27, 28, 32–42, 45–52, 54, 63–67, 69–71, 74–77, 79–82, 84, 86–101, 104–116, 118–120, 122, 124, 126–134, 136–139, 145, 146, 148–152, 157, 166–168, 171–182, 186–191, 196–199, 204, 205, 209–240, 242–247, 249, 251, 252, 255, 256, 258, 259,

261, 264–284, 288, 289, 295–
 302, 304–310, 313, 314, 316,
 318, 319, 321, 327–355, 357–359
 \label 31,
 88, 163, 364, 430, 522, 523, 604
 \labelflow 31, 364, 523
 \labelflowidn 31, 364, 522
 \language 162
 \leavevmode 4, 55
 \leftmark 168–170
 \lefttorightcolumnfalse 29, 363, 521
 \lefttorightcolumnstrue
 15, 29, 363, 521
 \legacy 17, 76, 77, 157, 183, 184, 188,
 192, 194, 201, 202, 207, 287, 316
 \let 21,
 26, 43, 87, 88, 93, 96, 137, 141,
 160–163, 165, 168–170, 172,
 197, 269, 274, 283, 288, 289,
 326, 327, 352, 360, 364, 367,
 375, 384, 386–390, 397, 399,
 403, 428–431, 459, 471, 472,
 477, 489, 490, 493, 501–505,
 510, 512, 514, 515, 522, 527,
 537, 548, 550–552, 554, 555,
 564, 565, 570, 602–606, 646,
 661, 668, 686, 690, 700, 702,
 703, 705, 706, 712, 715, 718, 719
 \lineskip 163, 430, 604
 \lineskiplimit 163, 430, 604
 \linewidth
 . 93, 158, 319, 428, 514, 602, 717
 \long 75, 402,
 403, 405, 512, 569, 570, 572, 715
 \loop 432, 459, 480, 484,
 487, 607, 646, 647, 672, 677, 682
 LR (option) 521

M

\m@ne . 74, 169, 171, 285, 286, 365, 524
 \mainmatter 17
 \makeatletter 311, 510, 712
 \makebackgroundframe .. 274, 493, 690
 \makebox 13, 190, 206, 361,
 468, 475–477, 516, 657, 664–667
 \makedfheaderfooter 164, 430, 605, 605
 \makethumbtabs 8, 292, 501, 701
 \MakeUppercase 21
 \marginparsep 189, 468, 656

\marginparwidth 189,
 266, 467, 468, 488, 656, 657, 683
 \markboth 169, 170, 288, 289
 \markright 169, 170, 288, 289
 \maxdepth 431, 606
 \maxdimen 428, 603
 maxdynamic (counter) 32, 523
 maxflow (counter) 31, 522
 maxstatic (counter) 31, 523
 \mbox 73,
 204, 279, 353, 365, 387, 388,
 475, 496, 524, 551, 552, 665, 694
 \MessageBreak
 176, 425, 427, 459, 599, 602, 646
 \minitocstyle
 317, 317, 512, 513, 716, 716
 \mod 326
 \moveright 163, 430, 605
 \msg 5–10, 35, 37, 45, 46,
 48–53, 55–57, 60, 66, 71, 73,
 91, 94, 97, 99, 116, 117, 138,
 141, 144, 146, 148–153, 157,
 165, 176, 177, 183, 246, 249,
 255, 264, 274, 283, 284, 292–
 294, 306, 309, 312, 320, 326, 360
 \multiply 488, 505, 683, 706

N

\Ncolumn 245, 479, 671
 \Ncolumnbottom ... 262, 486, 681, 681
 \Ncolumnbottominarea
 262, 263, 486, 487, 681, 681, 683
 \NcolumnDbottom 263, 486, 681
 \NcolumnDbottominarea
 265, 274, 487, 683
 \NcolumnDtop 254, 483, 676
 \NcolumnDtopinarea 257, 484, 678
 \Ncolumninarea
 245, 246, 479, 480, 671, 671
 \NcolumnSbottom 262, 486, 681
 \NcolumnSbottominarea . 265, 487, 682
 \NcolumnStop 254, 483, 676
 \NcolumnStopinarea 256, 484, 678
 \Ncolumnstoptop 253, 483, 676, 676
 \Ncolumnstoptopinarea
 254, 483, 484, 676, 676, 678
 \NeedsTeXFormat 3, 321, 360, 515
 \newboolean 380,
 382, 393, 394, 543, 545, 558, 560
 \newbox 95, 367, 381, 389, 527, 544, 554

<code>\newcommand</code>	3,	144, 176, 241, 361–363, 365,
	4, 6, 12, 17, 18, 20, 43, 79,	367, 402, 411, 425, 431, 459,
	82, 84, 85, 87, 94, 101, 102,	478, 510–512, 514, 516, 518,
	104, 105, 133, 134, 138, 140–	521, 524, 527, 569, 581, 599,
	142, 158–160, 162, 163, 168,	607, 646, 669, 712, 713, 715, 718
	169, 171–173, 208, 265, 266,	
	269, 274, 275, 279, 282, 284–	<code>\newlength</code>
	287, 291, 293, 294, 297, 317,	. 3, 30, 33, 73, 114, 156, 247,
	322, 325–328, 354, 356, 360–	275, 297, 316, 360, 365, 367,
	367, 369, 374, 375, 378–384,	381, 393, 406, 427, 476, 478,
	387–397, 400, 401, 404–407,	480, 481, 488, 493, 494, 504,
	409–427, 429–431, 433–476,	512, 516, 523, 524, 527, 544,
	478–493, 495–505, 507–527,	559, 574, 601, 666, 670, 673,
	529, 530, 535–537, 540–548,	674, 683, 691, 692, 704, 715, 716
	551–563, 566–568, 571–577,	<code>\newline</code> 87
	579–601, 604–619, 621–629,	<code>\NewMirroredHookPair</code> 97, 287
	631, 633–637, 639–641, 643–	<code>\newpage</code> 74,
	662, 664–667, 669–691, 693–	75, <u>75</u> , 365, 366, 524, 525, <u>525</u>
	695, 697–706, 709–713, 716–719	<code>\newread</code> 510, 713
<code>\newcount</code> 312, 365, 433, 459, 477,	487, 489, 505, 510, 511, 523,	<code>\newsavebox</code> 287
	609, 646, 667, 683, 685, 706, 713	<code>\newstaticframe</code>
<code>\newcounter</code> 12,		. <u>83</u> , 380, 489, 492, 493, 495,
	31, 32, 291, 320, 362, 364, 365,	499, <u>543</u> , 685, 690, 691, 693, 697
	501, 514, 519, 522, 523, 701, 718	<code>\newtoks</code> 402, 404, 569, 572
<code>\NewDocumentCommand</code> 16, 29, 31, 73–		<code>\newwrite</code> 311, 510, 712
	81, 83–85, 87, 88, 98, 99, 102,	<code>\nfss@catcodes</code> 162
	104, 105, 107, 109, 112–114,	<code>\nobreak</code> 285–287
	117–121, 123–132, 134, 136,	<code>\NoChapterInDynamic</code> <u>290</u>
	141, 143–145, 148, 149, 160,	<code>\noexpand</code> 162, 283,
	161, 164, 165, 175, 180, 209–	390, 403, 429, 434–459, 501,
	212, 214, 215, 218–220, 222,	504–508, 554, 555, 570, 604,
	223, 225, 226, 228–233, 235,	611–617, 619, 621–625, 627–
	236, 239, 240, 242, 243, 245–	629, 631–637, 639–641, 643–
	251, 253, 254, 256–260, 262,	646, 700, 705, 707, 708, 710
	263, 265–272, 274, 275, 279,	<code>\noindent</code> 74, 97, 102,
	283, 284, 290, 292, 294, 297,	319, 325, 365, 390, 403, 405,
	302–304, 306, 308, 310, 317,	513, 514, 524, 555, 570, 573, 717
	318, 320, 325, 326, 352, 356–360	<code>\normalcolor</code> . . 158, 159, 317, 363,
<code>\NewDocumentEnvironment</code> . . . 95, 108		429, 431, 512, 521, 604, 606, 716
<code>\newdynamicframe</code> <u>102</u> , 393,		<code>\normalfont</code> . . . 285, 286, 317, 512, 716
	430, 505, 506, <u>558</u> , 605, 707, 708	<code>\normalsfcodes</code> 163, 430, 604
<code>\newenvironment</code> 241, 389,		<code>\normalsize</code> 162, 317, 430, 512, 604, 716
	402, 404, 478, 554, 569, 571, 669	<code>\norotate</code> (option) <u>518</u>
<code>\newflowframe</code> <u>77</u> , 367, 480, 484–487,		<code>\not</code> 510, 713
	<u>526</u> , 527, 672, 677, 679, 680, 682	<code>\nr</code> 361, 362
<code>\newframe</code> <u>248</u> , 249, 252, 256, 259,		<code>\number</code> . . 369, 378, 382, 391, 394,
	261, 264, 481–483, 485, 487,	395, 400, 401, 403, 404, 407–
	<u>673</u> , 674, 675, 677, 679, 680, 682	410, 427, 434–459, 461–463,
<code>\newif</code> 4, 15, 74, 120,		468, 469, 473, 480, 483, 486,
		501, 504–506, 508, 509, 513,
		529, 530, 540, 545, 546, 556,

560, 561, 567, 570–572, 576–578, 601, 610–646, 649–651, 657, 658, 662, 672, 677, 681, 700, 704, 707, 708, 710, 711, 717

O

\oddcheckifframeabove . [121](#), [411](#), [581](#)
 \oddcheckifframebelow . [124](#), [413](#), [583](#)
 \oddcheckifframeleft .. [126](#), [414](#), [586](#)
 \oddcheckifframeright . [128](#), [416](#), [588](#)
 \oddsidemargin
 [113](#), [162](#), [405](#), [476](#), [573](#), [667](#)
 \oldpar [472](#), [661](#)
 \onecolumn [147](#), [152](#), [160](#),
[161](#), [241](#), [289](#), [427](#), [478](#), [601](#), [669](#)
 \onecolumnbottom
 [257](#), [257](#), [258](#), [484](#), [678](#), [678](#)
 \onecolumnbottominarea
[258](#), [484–486](#), [678](#), [678](#), [679](#), [681](#)
 \onecolumnDbottom [258](#), [484](#), [678](#)
 \onecolumnDbottominarea [259](#), [485](#), [679](#)
 \onecolumnDtop [248](#), [481](#), [673](#)
 \onecolumnDtopinarea .. [250](#), [482](#), [674](#)
 \onecolumninarea
 [242](#), [478](#), [480](#), [670](#), [671](#)
 \onecolumnSbottom [257](#), [484](#), [678](#)
 \onecolumnSbottominarea [259](#), [485](#), [679](#)
 \onecolumnStop [248](#), [426](#), [481](#), [600](#), [673](#)
 \onecolumnStopinarea
 [249](#), [250](#), [482](#), [674](#)
 \onecolumnstoptop [247](#), [480](#), [481](#), [673](#)
 \onecolumnstoptopinarea
[249](#), [480–483](#), [673](#), [674](#), [674](#), [676](#)
 \openin [510](#), [713](#)
 \openout [311](#), [510](#), [712](#)
 options:
 color [519](#)
 draft [517](#)
 final [517](#)
 LR [521](#)
 norotate [518](#)
 pages [519](#)
 RL [521](#)
 rotate [518](#)
 thumbtabs [518](#)
 ttbnonum [519](#)
 ttbnotitle [519](#)
 ttbnum [519](#)
 ttbtitle [519](#)
 verbose [517](#)

\OR [365](#), [524](#)
 \or .. [361](#), [362](#), [390](#), [407–409](#), [433](#),
[461](#), [471](#), [472](#), [480](#), [483](#), [486](#),
[495](#), [499](#), [509](#), [517–520](#), [554](#),
[555](#), [576–578](#), [608](#), [648](#), [661](#),
[671](#), [676](#), [681](#), [694](#), [697](#), [698](#), [711](#)
 \output [428](#), [602](#)
 \outputpenalty [428](#), [602](#)

P

\p@ [285](#), [286](#)
 \PackageError [369–373](#),
[377](#), [378](#), [382](#), [383](#), [387–390](#),
[395](#), [396](#), [399](#), [400](#), [403–405](#),
[407–422](#), [427](#), [479–481](#), [483](#),
[486](#), [494](#), [496–498](#), [500](#), [501](#),
[514](#), [529–535](#), [539](#), [540](#), [545](#),
[546](#), [551–554](#), [560](#), [561](#), [566](#),
[570–572](#), [576–579](#), [582](#), [584–](#)
[595](#), [601](#), [671–673](#), [676](#), [681](#),
[692](#), [693](#), [695](#), [696](#), [698–701](#), [718](#)
 \PackageInfo
 [361](#), [459](#), [510](#), [517](#), [647](#), [713](#)
 \PackageWarning [425–](#)
[427](#), [459](#), [462](#), [493](#), [507](#), [509](#),
[599–602](#), [646](#), [651](#), [691](#), [709](#), [711](#)
 \pagebreak [74](#), [365](#), [524](#)
 pages (option) [519](#)
 \pagestyle [165](#), [360](#)
 \pagetotal [74](#), [365](#), [524](#)
 \paperheight [114](#), [266](#),
[268](#), [274](#), [275](#), [406](#), [407](#), [488](#),
[489](#), [493](#), [574](#), [575](#), [683](#), [685](#), [691](#)
 \paperwidth [114](#), [270](#),
[272](#), [274](#), [275](#), [406](#), [407](#), [490](#),
[491](#), [493](#), [574](#), [575](#), [687](#), [688](#), [691](#)
 \par [74](#), [87](#), [93](#), [96](#), [102](#), [197](#),
[241](#), [285–287](#), [319](#), [365](#), [387](#),
[388](#), [390](#), [391](#), [403](#), [405](#), [428](#),
[472](#), [513](#), [514](#), [524](#), [551](#), [552](#),
[554](#), [555](#), [570](#), [573](#), [602](#), [661](#), [717](#)
 \paragraph ... [94](#), [388](#), [389](#), [552](#), [553](#)
 \parbox [11](#), [289](#), [319](#), [471–](#)
[473](#), [504](#), [514](#), [661](#), [662](#), [705](#), [717](#)
 \parfillskip [74](#), [102](#), [365](#), [405](#), [524](#), [573](#)
 \parindent [87](#), [97](#), [197](#), [198](#),
[285](#), [286](#), [390](#), [471](#), [472](#), [555](#), [661](#)
 \parshape .. [87](#), [93](#), [387](#), [388](#), [551](#), [553](#)
 \parskip [74](#), [365](#), [387](#), [524](#), [551](#)
 \pdfannot@link@off@@ [158](#), [159](#)

<code>\pdfannot@link@on@@</code>	159	<code>\raise</code>	463, 464, 468–470, 473, 474, 651, 652, 657, 659, 660, 663, 664
<code>\pdfextension</code>	354	<code>\read</code>	511, 713
<code>\pdfinfo</code>	354	<code>\RefSavedRelativeLocation</code>	141
<code>\penalty</code>	74, 365, 524	<code>\refstepcounter</code>	292
<code>\pgfextractx</code>	336, 339	<code>\relativeframelocation</code>	134, 423, 596
<code>\pgfextracty</code>	336, 339	<code>\relax</code>	43, 84, 88, 94, 103, 137, 141, 145, 146, 162, 207, 269, 274, 288, 289, 311, 361– 365, 367–370, 374, 375, 379– 384, 388–390, 392–397, 401– 403, 407–409, 412–415, 417– 422, 425–428, 432–460, 464, 468, 471, 475–480, 482–490, 492, 493, 495, 498, 499, 503– 505, 507–513, 517–525, 527– 530, 535–537, 541–548, 553– 555, 557–563, 568, 570, 575– 578, 582–595, 599–602, 607– 610, 612–624, 626–647, 652, 653, 657, 660, 661, 665–668, 671, 672, 675–677, 680–683, 686, 687, 689–691, 693, 694, 697, 704, 706, 707, 709–715, 717
<code>\pgfpathlineto</code> ...	343–345, 347–351	<code>\reldynamicloc</code>	143, 424, 598
<code>\pgfpathmoveto</code> ...	343–346, 348–351	<code>\relflowloc</code>	143, 425, 598
<code>\pgfpoint</code>	343–352	<code>\relstaticloc</code>	143, 425, 598
<code>\pgfsetarrowsend</code>	335	<code>\renewcommand</code>	18, 20, 24, 88, 160, 161, 207, 283, 286, 287, 318, 352, 357–359, 361–364, 427, 430, 431, 478, 479, 501, 502, 510, 512–514, 517, 519–522, 601, 605, 606, 669, 670, 700, 702, 712, 715–718
<code>\pgfsetarrowsstart</code>	335, 342	<code>\RenewDocumentCommand</code>	16, 75, 95, 109, 147, 161, 241, 243
<code>\pgfsetstrokecolor</code>	335, 342	<code>\repeat</code>	433, 459, 460, 480, 484, 487, 608, 647, 672, 678, 682
<code>\pgftext</code>	343, 345–351	<code>\RequirePackage</code>	3, 30, 322, 325, 326, 360, 364, 515, 516, 521
<code>\pgfusepath</code>	343, 344, 346–351	<code>\RerunFileCheck</code>	311
<code>\preto</code>	75, 366, 525	<code>\reset@font</code>	162, 430, 604
<code>\prevgraf</code>	387, 388, 551, 552	<code>\rightmark</code>	168–170
<code>\prg</code> 14, 17, 18, 46, 47, 56, 59, 62–64, 66–69, 71, 72, 79, 80, 84, 104, 290		RL (option)	521
<code>\ProcessKeyOptions</code>	30, 321	<code>\romannumeral</code>	47– 62, 64, 65, 71, 72, 96, 293, 294, 314, 315, 317, 319, 366– 370, 375–387, 390–401, 403, 404, 408–410, 425–427, 430– 432, 459–474, 489, 492, 493,
<code>\ProcessOptionsX</code>	363, 521		
<code>\prop</code>	47, 48, 50–53, 55, 56, 59, 71, 328, 332–334, 338		
<code>\protect</code>	162, 291, 292, 429, 502, 503, 604, 702, 703		
<code>\protected@edef</code>	403, 570		
<code>\protected@write</code>	43, 88, 137		
<code>\providecommand</code>	75, 87		
<code>\ProvideDocumentCommand</code> ..	352, 355		
<code>\ProvidesPackage</code> ...	3, 321, 360, 515		
<code>\ps@empty</code>	165, 173, 360		
<code>\ps@ffempty</code>	165, 172, 360		
<code>\ps@ffheadings</code>	165, 171, 360		
<code>\ps@ffmyheadings</code>	165, 171, 360		
<code>\ps@ffplain</code>	165, 172, 360		
<code>\ps@headings</code>	165, 360		
<code>\ps@myheadings</code>	165, 360		
<code>\ps@plain</code>	165, 360		
<code>\ps@specialeempty</code>	173		
<code>\ps@specialffempty</code>	173		
<code>\ps@scharpath</code>	326		
Q			
<code>\q</code>	41, 44, 70, 71, 111, 112, 268, 273, 337, 339		
<code>\quad</code>	18		
<code>\quark</code>	70, 111, 338, 340		
R			
<code>\raggedright</code>	285, 286		

495, 499, 503–508, 511, 513,
514, 526–530, 537–550, 554–
567, 571, 572, 577–579, 599,
601, 602, 605–608, 646–664,
685, 690, 691, 694, 697, 698,
703, 704, 707–710, 714, 716, 717

rotate (option) [518](#)

\rotatebox
.. 10, 11, 180, 362, 504, 518, 705

\rotatframe 180, [180](#), 190, 199, 362,
461, 468, 472, [518](#), 649, 657, 662

\rule 279, 282, 496, 499, 694, 698

S

\SaveRelativeFrameLocation [136](#)

\sbox 288, 391, 471, 555, 660

\sdfparindent 33, 84, 103,
365, 390, 471, 472, [523](#), 555, 661

\section ... 19, 94, 388, 389, 552, 553

\sectionmark 19, 169, 170, 172

\seq 35, 45–47,
176, 296–301, 303, 304, 306–309

\set@typeset@protect .. 162, 430, 604

\setalldynamicframes
..... [105](#), 396, [562](#), 563

\setallflowframes [80](#), 145,
245, 374, 426, [535](#), 536, 599, 600

\setallstaticframes
..... 85, 383, 384, 547, [547](#)

\setboolean 375, 380–382, 385–387,
393, 394, 397, 400, 537, 543–
545, 548, 550, 558, 560, 564, 566

\setbox .. 144, 204, 207, 208, 326,
425, 428, 429, 431, 461, 475,
478, 599, 603, 606, 649, 665, 668

\setcounter
.. 18, 23, 163, 430, 514, 605, 718

\setdynamiccontents
..... 9, 108, [109](#), 404, [571](#)

\setdynamicframe
..... [105](#), 396, 508, [562](#), 710

\setffdraftcolor 3, 13, 360, 361, 516

\setffdraftttypeblockcolor
..... 3, 206, 360, 476, 516, 666

\setflowframe [80](#), 244,
245, 252, 253, 261, 374, 479,
482, 485, 486, [536](#), 671, 675, 680

\setframes [144](#), 146, 425, 427, [599](#), 601

\setinitialframe [144](#), 425, [599](#)

\setkeys . 375, 384, 397, 537, 548, 564

\setlength .. 3, 30, 114, 156, 247,
275, 297, 316, 367, 368, 375,
376, 379, 381, 385, 390, 392,
393, 397, 398, 401, 405–410,
425, 427, 430, 464–467, 471,
472, 475, 476, 478–485, 487–
493, 495, 498, 504–506, 512,
527, 528, 537, 538, 541, 544,
549, 555–557, 559, 564, 567,
573–579, 599, 601, 605, 652,
654–656, 661, 664, 667, 670–
675, 677–680, 682, 684–691,
693, 697, 704, 706–708, 715, 716

\setmargin 77, 158, 366, 428, [526](#), 602

\SetOneColumnFrame [148](#)

\setstaticcontents 9,
98, 145, 146, 278, 281, 391,
426, 495, 499, [555](#), 600, 694, 697

\setstaticframe 85, 278, 279, 282,
383, 495, 496, 499, [547](#), 694, 698

\setthumbtab [306](#), 508, [710](#)

\setthumbtabindex [308](#), 509, [711](#)

\SetThumbTabTitle [294](#)

\settodepth 145, 426, 600

\settoheight 145, 426, 600

\SetTwoColumnFrames [149](#)

\sffamily 4, 361, 516

\sfswapoddeven 99, 392, [557](#)

\Shapepar 94, 389, 553

\shapepar .. 93, 94, 388, 389, 553, 554

\shipout 162, 429, 604

\showframebboxfalse 4, 361, 516

\showframebboxtrue 4, 361, 516

\showmarginfalse 4, 361, 516

\showmargintrue 4, 361, 516

\showtypeblockfalse 4, 361, 516

\showtypeblocktrue 4, 361, 516

\simpar 87, 387, [551](#)

\skip 431, 606

\small 4, 102, 361, 405, 516, 573

\space ... 285, 286, 378, 391, 400,
401, 403–405, 427, 428, 461–
463, 480, 483, 486, 504, 540,
556, 566, 567, 570–572, 601,
602, 649, 651, 672, 677, 681, 705

\staticaddexclusion [98](#),
240, 391, 458, 459, [556](#), 645, 646

staticcontents (env.) [95](#), [554](#)

staticcontents* (env.) [95](#), [554](#)

staticfigure (env.) [241](#), [669](#)

`\staticframe` 95, 97, 389–391, 554, 555
`\staticframeevenx`
 . 85, 383, 408, 409, 547, 577, 578
`\staticframeeveny`
 . 85, 383, 408, 409, 547, 577, 578
`\staticframeheight` 85, 96
`\staticframewidth` 85, 96
`\staticframex` . 85, 383, 408, 546, 576
`\staticframey` . 85, 383, 408, 546, 577
`\staticsetexclusion`
 98, 386, 391, 550, 556
`\staticsetpagelist`
 98, 231–233, 235, 236,
 239, 386, 391, 451–457, 550,
 556, 634–637, 639–641, 643, 644
`\staticswithchoffnext` .. 232, 453, 636
`\staticswithchoffnextodd` 233, 453, 637
`\staticswithchoffnextoddoneonly` ...
 240, 458, 645
`\staticswithchoffnextonly`
 239, 458, 644
`\staticswithchonnnext` ... 230, 451, 634
`\staticswithchonnnextodd` 231, 451, 635
`\staticswithchonnnextoddoneonly`
 236, 456, 641
`\staticswithchonnnextonly` 235, 455, 640
`statictable` (env.) 241, 669
`\stepcounter` 158, 160,
 163, 293, 320, 428, 430, 503,
 514, 515, 602, 605, 703, 718, 719
`\string` . 43, 88, 137, 389, 399, 405,
 427, 553, 554, 566, 572, 601, 602
`\subparagraph` .. 94, 388, 389, 552–554
`\subsection` ... 94, 388, 389, 552, 553
`\subsectionmark` 170, 172
`\subsubsection` 94, 388, 389, 552, 553
`\sys` 327

T

`\tableofcontents` 18–
 21, 310, 315, 509, 512, 712, 715
`\TE@or` 368, 374, 375, 379,
 380, 384, 392, 393, 396, 397,
 401, 402, 528, 536, 537, 541,
 542, 547, 548, 557, 558, 563, 568
`\tenln` 186, 464, 652
`\tenlnw` 186, 464, 653
`\text` 297
`\textheight`
 . 30, 157, 160, 163, 325, 405,
 430, 475, 476, 478–486, 505,
 573, 605, 664–666, 669–671,
 673, 674, 676, 678, 679, 681, 706
`\textwidth` 30, 160, 163, 325,
 425, 428–430, 475, 476, 478–
 486, 599, 602, 604, 605, 665–
 671, 673, 674, 676, 678, 679, 681
`\the` 157, 381, 384, 387,
 388, 394, 397, 402, 403, 405,
 411–422, 427, 434–455, 457–
 459, 504, 511, 512, 544, 548,
 551, 552, 559, 563, 569–572,
 582, 584–595, 602, 611–617,
 619, 621–625, 627–629, 631–
 638, 640, 641, 643–646, 705, 714
`\theabsolute page` 31, 43
`\thechapter` ... 169, 285, 286, 502, 702
`\thedisplayedframe` 31, 364, 523
`\theFramePageCounter` 12, 88, 291, 292
`\theHdisplayedframe` ... 31, 364, 523
`\theHthisframe` 31, 364, 522
`\thepage`
 31, 43, 168–170, 172, 364, 428,
 502, 503, 522, 523, 602, 702, 703
`\thepart` 502, 503, 702, 703
`\thesection` 170, 171
`\thesubsection` 171
`\thethisframe` 31, 364, 522
`\thethumbtabnonum` 292
`thisframe` (counter) 31, 522
`\thispagestyle` 20, 364, 522
`\thr@@` 162
`\thumbtab` 291,
 292, 293, 502, 503, 702, 703, 703
`\thumbtab@backmatter` 292, 294
`\thumbtabformat` 16,
 17, 24, 297, 504–506, 705, 705, 707
`\thumbtabhyperlinkformat`
 16, 16, 17, 24
`\thumbtabindex`
 304, 316, 508, 509, 710, 712
`\thumbtabindexformat` 16, 24,
 300, 301, 504, 506, 507, 705, 708
`\thumbtabnumtitle` 297, 302
`\thumbtabregularformat`
 17, 24, 299, 300
`thumbtabs` (option) 518
`\thumbtabwidth` 11, 295, 297,
 299, 300, 504–506, 704, 705–708
`\title` 355

<code>\tl</code>	12–16, 19, 22, 28, 32–34, 36–42, 45–50, 52, 54–59, 62, 66, 68–70, 72, 76, 77, 79, 81, 84, 86–94, 96, 97, 100, 101, 104, 106, 111, 136– 141, 148–156, 158, 159, 165, 175, 176, 178, 181, 189, 191, 199, 209–215, 218–226, 228– 236, 239, 240, 242, 245, 252, 253, 268, 273, 288–295, 297, 298, 301, 302, 306, 308, 310, 313–317, 319, 320, 327–330, 332, 333, 335–340, 342–355, 360	<code>\typeblockoffsety</code>	30, 113, 163
<code>\tocandhumbtabindex</code>	310, 712	<code>\typeblockwidth</code>	30, 145, 146, 159, 160, 164, 204, 206, 242, 243, 245, 247, 248, 250, 251, 254, 257–260, 262, 263
<code>\tocandthumbtabindex</code> ..	310, 509, 712	U	
<code>\tocline</code>	511, 713	<code>\uncover</code>	328
<code>\token</code>	7–9	<code>\undefined</code>	375, 384, 388, 397, 537, 548, 552, 564
<code>\toks@</code>	403, 434–459, 570, 611–617, 619, 621–625, 627– 629, 631–638, 640, 641, 643–646	<code>\unvbox</code>	428, 431, 603, 606
<code>\topmargin</code> ...	30, 405, 430, 573, 605	<code>\use</code> ...	15, 21, 26, 52, 93, 146, 149, 156, 181, 191, 197–199, 248, 276, 280, 303, 318, 336, 355–357
<code>\topskip</code>	74, 365, 524	<code>\usebox</code>	391, 469, 555, 658
<code>ttbnonum</code> (option)	519	<code>\usedframebreakfalse</code> ..	157, 427, 602
<code>ttbnotitle</code> (option)	519	<code>\usedframebreaktrue</code>	74, 75, 79, 144, 365, 366, 368, 425, 524, 525, 528, 599
<code>ttbnum</code> (option)	519	<code>\UseHook</code>	97, 162, 163, 288, 289
<code>ttbtitle</code> (option)	519	<code>\usepgflibrary</code>	327
<code>\twocolumn</code>	147, 153, 160, 161, 242, 427, 479, 601, 670	<code>\UseTaggingSocket</code>	158, 159
<code>\twocolumnbottom</code> .	259, 485, 679, 679	V	
<code>\twocolumnbottominarea</code>	260, 485, 486, 679, 680, 681	<code>\vadjust</code>	387, 551
<code>\twocolumnDbottom</code> .	260, 260, 485, 679	<code>\val</code>	361, 362
<code>\twocolumnDbottominarea</code> .	262, 486, 680	<code>\value</code>	33, 34, 249
<code>\twocolumnDtop</code>	251, 482, 675	<code>\vbadness</code>	207, 478, 668
<code>\twocolumnDtopinarea</code> ..	253, 483, 676	<code>\vbox</code>	4, 74, 144, 159, 160, 162, 163, 182, 204, 206– 208, 319, 325, 365, 425, 429– 431, 462, 476, 478, 514, 524, 599, 603–606, 650, 666, 668, 717
<code>\twocolumninarea</code>	243, 479, 480, 670, 671	<code>\vcolumnsep</code>	247, 249, 252, 256, 258, 261, 264, 480–482, 484, 485, 487, 672, 674, 675, 677, 679, 680, 682
<code>\twocolumnSbottom</code>	260, 485, 679	<code>verbose</code> (option)	517
<code>\twocolumnSbottominarea</code>	262, 485, 486, 679, 680	<code>\vfil</code>	159, 429, 603
<code>\twocolumnStop</code> .	250, 426, 482, 600, 674	<code>\vfill</code>	319, 430, 514, 605, 717
<code>\twocolumnStopinarea</code>	253, 253, 482, 675	<code>\vNtone</code>	268, 489, 490, 685, 686
<code>\twocolumnntop</code>	250, 482, 674, 675	<code>\vNtonebottom</code>	269, 490, 686
<code>\twocolumnntopinarea</code>	251, 482, 483, 674, 675, 676	<code>\vNtonetop</code>	270, 490, 686
<code>\typeblockheight</code> .	30, 113, 144–146, 160, 164, 203, 204, 206–208, 242, 243, 245, 247, 248, 250, 251, 254, 257–260, 262, 263, 295	<code>\voffset</code>	30, 405, 573
		<code>\vrule</code>	4
		<code>\vsize</code>	157, 160, 163, 427–429, 478, 602, 603, 668, 669
		<code>\vskip</code>	4, 163, 285, 317, 387, 430, 431, 513, 551, 605, 606, 716
		<code>\vspace</code>	285, 286

