

Documented Source Code for flowfram.sty v2.2

Nicola L. C. Talbot

2026-04-25

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

Contents

Terms	2
1 Main Package Code	3
1.1 Package Initialisation	3
1.2 Frame Attributes	35
1.3 Flow Frames	74
1.4 Static Frames	84
1.5 Dynamic Frames	103
1.6 Determining Dimensions and Locations	114
1.7 Determining the relative location of one frame from another	121
1.8 Initialise Flow Frames	145
1.9 Output Routine	158
1.10 Switching Frames On and Off For Subsequent Pages	210
1.11 Static versions of floats	242
1.12 Standard Layouts	242
1.12.1 Column Styles	242
1.12.2 Backdrop Effects	267
1.12.3 Lines Between Frames	276
1.13 Putting Chapter Headings in Dynamic Frames	284
1.14 Thumbtabs	292
1.15 Minitocs	311
2 Support for FlowframTk	322
3 Rollback v1.17 (flowfram-2014-09-30)	363
4 Rollback v1.18 (flowfram-2025-08-23)	518
4.1 Flow Frames	526
4.2 Static Frames	545
4.3 Dynamic Frames	560

4.4	Determining Dimensions and Locations	575
4.5	Determining the relative location of one frame from another	583
4.6	Initialise Flow Frames	601
4.7	Output Routine	603
4.8	Static versions of floats	671
4.9	Standard Layouts	671
4.9.1	Column Styles	671
4.9.2	Backdrop Effects	686
4.9.3	Lines Between Frames	693
4.10	Putting Chapter Headings in Dynamic Frames	702
4.11	Thumbtabs	703
4.12	Minitocs	714

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 L^AT_EX 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{v2.2}{2026-04-25}
```

Declare package:

```
\ProvidesPackage{flowfram}[2026/04/25 v2.2 (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_EX3 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}
```

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

```
\newcommand*{\fflabe
```

font}{\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
```

```
\showmargintrue
```

```
\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 { 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} {invalid-width}
{Invalid width #1 for #2 frame (using #3 instead)}
{A #2 frame width must be greater than Opt}

\msg_new:nnnn {flowfram} {invalid-height}
{Invalid height #1 for #2 frame (using #3 instead)}
{A #2 frame height must be greater than Opt}

\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

```

\if@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 }

\tl_const:Nn \c_flowfram_flow_tl { flow }
\tl_const:Nn \c_flowfram_dynamic_tl { dynamic }
\tl_const:Nn \c_flowfram_static_tl { static }

```

Integer constants:

```
\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 ] [ 1 ]
    {
      \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 \flowframetextcol
  \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 fempty 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_thumtbat_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 } ,
thumtbat-text .choice: ,
thumtbat-text / normal .code:n =
{
\cs_set_eq:NN \__flowfram_thumtbat_fmt:nn \use_i:nn
} ,
thumtbat-text / stack .code:n =
{
\cs_set_eq:NN
\__flowfram_thumtbat_fmt:nn
\__flowfram_thumtbat_fmt_stack:nn
} ,
thumtbat-text / rotate-right .code:n =
{
\cs_set_eq:NN
\__flowfram_thumtbat_fmt:nn
\__flowfram_thumtbat_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 ttbttitle, 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@absolute page`.

```

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@absolute page }
} ,
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@tffttextcol 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 = { \lefttorightcolumnfalse } ,
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 { \thedisplayedframe }
    \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 \l__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:Nn \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:Nc \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 \__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@html opts
        { \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@html opts
        { \int_use:N \g__flowfram_html_opts_int }
        { #2 } { #3 } { #1 }
        { \thepage }
        { \theabsolute page }
    }
}

```

`\flowfram@preamble@html opts` Syntax: $\{\langle opt-id \rangle\}\{\langle frame-type \rangle\}\{\langle idn \rangle\}\{\langle options \rangle\}\{\langle page \rangle\}\{\langle absolute-page \rangle\}$
`\newcommand*\flowfram@preamble@html opts[6]{}`

`\flowfram@doc@html opts` Syntax: $\{\langle opt-id \rangle\}\{\langle frame-type \rangle\}\{\langle idn \rangle\}\{\langle options \rangle\}\{\langle page \rangle\}\{\langle absolute-page \rangle\}$
`\newcommand*\flowfram@doc@html opts[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
```

Dimension checks are only required for flow frames. Static or dynamic frames may simply be decorative.

```
\tl_if_eq:NnTF \c_flowfram_flow_tl { #7 }
{
```

Sanity check on width. If all defined flow frames accidentally have the width set to zero it can trigger an infinite number of pages.

```
\dim_compare:nNnTF { #2 } > { \c_zero_dim }
{
  \flowfram_frame_new_dim:nnnn
  { #7 } { width } { \int_use:c { c@max #7 } } { #2 }
```

```

}
{
  \__flowfram_error:eeee { invalid-width }
  { \dim_eval:n { #2 } } { #7 } { \token_to_str:N \typeblockwidth }
  \flowfram_frame_new_dim:nnnn
  { #7 } { width } { \int_use:c { c@max #7 } } { \typeblockwidth }
}
\dim_compare:nNnTF { #3 } > { \c_zero_dim }
{
  \flowfram_frame_new_dim:nnnn
  { #7 } { height } { \int_use:c { c@max #7 } } { #3 }
}
{
  \__flowfram_error:eeee { invalid-height }
  { \dim_eval:n { #3 } } { #7 } { \token_to_str:N \typeblockheight }
  \flowfram_frame_new_dim:nnnn
  { #7 } { height } { \int_use:c { c@max #7 } } { \typeblockheight }
}
}
{
  \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 } { backcolor } { \int_use:c { c@max #7 } } { { none } }
\flowfram_frame_new_tl:nnnn
{ #7 } { angle } { \int_use:c { c@max #7 } } { 0 }
\clist_set:Nc \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:Nc \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:Nc \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 {<type>}{<label>}
\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 {<type>}{<label>}{<int-var>}
If true, the corresponding IDN is stored in the <int-var>

```

\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:nnne { flowfram } { show-all-frames-none }
    { static }
  }
  {
    \msg_show:nnne { 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:nneee { 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:nnne { flowfram } { show-all-frames-none }
        { dynamic }
    }
    {
        \msg_show:nnne { 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 _ \romannumeral #3 _ bool }
    { true }
    { false }
}
\cs_new:Nn \__flowfram_show_box_attribute:nnn
{
    \box_if_empty:cTF
    { g__flowfram_ #1 _ #2 _ \romannumeral #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
    {
        \tl_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:nnnn
{
    \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:nnnn
{
  \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_gc_clear: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: {\langle frame type \rangle}{\langle tag \rangle}{\langle idn \rangle}

\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: {\langle frame type \rangle}{\langle tag \rangle}{\langle idn \rangle}{\langle value \rangle}

\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 _ tl }
    { #4 }
    \__flowfram_add_frame_attribute:cen
    { g__flowfram_ #1 _attributes_prop }
    { #2 } { tl }
  }
}

```

```

    {
      \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 _ tl }
}

Syntax: {<frame type>}{<tag>}{<idn>}{<value>}
\cs_new:Nn \flowfram_frame_set_tl:nnnn
{
  \tl_if_exist:cTF
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ tl }
  {
    \__flowfram_message:nnnen
      { info-setting-frame-attribute }
      { #2 } { #1 } { \int_eval:n { #3 } } { #4 }
    \tl_gset:cn
      { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ tl }
      { #4 }
  }
  {
    \__flowfram_error:eeee{ frame-unknown-tag }
    { #1 } { #2 } { \int_eval:n { #3 } } { tl }
  }
}
\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 _ tl }
  {
    \__flowfram_message:nnnen
      { info-setting-frame-attribute }
      { #2 } { #1 } { \int_eval:n { #3 } } { }
    \tl_gclear:c
      { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ tl }
  }
  {
    \__flowfram_error:eeee{ frame-unknown-tag }
    { #1 } { #2 } { \int_eval:n { #3 } } { tl }
  }
}
Syntax: {<frame type>}{<tag>}{<idn>}{<value>}

```

```

\cs_new:Nn \flowfram_frame_put_right_tl:nnnn
{
  \tl_if_exist:cTF
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ tl }
    {
      \tl_gput_right:cn
        { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ tl }
        { #4 }
      \__flowfram_message:nnnev
        { info-setting-frame-attribute }
        { #2 } { #1 } { \int_eval:n { #3 } }
        { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ tl }
    }
    {
      \__flowfram_error:eeeeee { frame-unknown-tag }
      { #1 } { #2 } { \int_eval:n { #3 } } { tl }
    }
  }
}
\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 _ \romannumeral #4 _ tl }
    {
      \tl_set_eq:Nc
        #1
        { g__flowfram_ #2 _ #3 _ \romannumeral #4 _ tl }
    }
    {
      \__flowfram_error:eeeeee { frame-unknown-tag }
      { #2 } { #3 } { \int_eval:n { #4 } } { tl }
    }
  }
}
Syntax: {<frame type>}{<tag>}{<idn>}<tl>
\cs_new:Nn \flowfram_set_frame_tl_to_tl:Nnnn
{
  \tl_if_exist:cTF
    { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ tl }
    {
      \__flowfram_message:nnneV
        { info-setting-frame-attribute }
        { #2 } { #1 } { \int_eval:n { #3 } } #4
      \tl_gset_eq:cN
        { g__flowfram_ #1 _ #2 _ \romannumeral #3 _ tl }
        #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:eeeeee { 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: {\frame type}{\tag}{\idn}{\code}
\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: }
}

{\page num}{\frame type}{\idn}

\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: {\page num}{\frame type}{\idn} 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 }
{
  \flowfram_if_frame_allpages:nnTF { #2 } { #3 }
  { \prg_return_true: }
  {
    \flowfram_if_frame_nopages:nnTF { #2 } { #3 }
    { \prg_return_false: }
    {

```

```

\flowfram_if_frame_oddpages_only:nnTF { #2 } { #3 }
{
  \int_if_odd:nTF { #1 }
  { \prg_return_true: }
  { \prg_return_false: }
}
{
  \flowfram_if_frame_evenpages_only:nnTF { #2 } { #3 }
  {
    \int_if_even:nTF { #1 }
    { \prg_return_true: }
    { \prg_return_false: }
  }
  {
    \flowfram_if_in_frame_clist:nnnnTF
    { #1 } { #2 } { pagelist } { #3 }
    { \prg_return_true: }
    {
      \bool_set_false:N \l__flowfram_found_bool
      \flowfram_frame_clist_map_inline:nnnn
      { #2 } { pagelist } { #3 }
      {
        \flowfram_if_in_pagerange:nnT { #1 } { ##1 }
        {
          \clist_map_break:n
          {
            \bool_set_true:N \l__flowfram_found_bool
          }
        }
      }
      \bool_if:NTF \l__flowfram_found_bool
      { \prg_return_true: }
      { \prg_return_false: }
    }
  }
}
}
}
}
}
{<num>}{<range>}
\prg_new_conditional:Nnn \flowfram_if_in_pagerange:nn
{ T, F, TF }
{
  \__flowfram_get_range:n { #2 }
  \bool_lazy_or:nnTF
  {
    \int_compare_p:nNn { #1 } < { \l__flowfram_range_start_int }
  }
  {

```


Syntax: $\{\langle page\ num\rangle\}\{\langle frame\ type\rangle\}\{\langle idn\rangle\}$ Check if there is at least one more page in the page list after the given number.

Get the last range in the page list.

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

65

```

}
{
  \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 _ \romannumeral #3 _ clist }
          { \c_one_int }
      }
      \int_if_zero:nT { #1 }
      {
        \int_set_eq:NN #1 \c_one_int
      }
    }
  }
}
}
}

```

Gets the starting page (stored in \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
            {
                \bool_set_true:N \l__flowfram_found_bool
            }
        }
        \int_set_eq:NN \l__flowfram_tmpa_int \l__flowfram_range_end_int
    }
}

```

```

        \bool_if:NTF \l__flowfram_found_bool
        { \prg_return_false: }
        { \prg_return_true: }
    }
}
}
}

\cs_new:Nn \flowfram_set_frame_excludelist:nnn
{
    \flowfram_frame_set_clist:nnnn
    { #1 } { excludelist } { #2 } { #3 }
}
\cs_generate_variant:Nn \flowfram_set_frame_excludelist:nnn
{ nne , nnV }

\cs_new:Nn \flowfram_concat_frame_excludelist:nnN
{
    \flowfram_frame_concat_clist:nnnN
    { #1 } { excludelist } { #2 } #3
}
\cs_generate_variant:Nn \flowfram_concat_frame_excludelist:nnN { nnc }

\cs_new:Nn \flowfram_concat_frame_excludelist:nnn
{
    \clist_set:Nn \l__flowfram_exclude_pages_clist { #3 }
    \flowfram_concat_frame_excludelist:nnN
    { #1 } { #2 } \l__flowfram_exclude_pages_clist
}
\cs_generate_variant:Nn \flowfram_concat_frame_excludelist:nnn { nne }

Syntax: {<frame type>}{<idn>} If frame's page list is set to odd:
\prg_new_conditional:Nnn \flowfram_if_frame_oddpages_only:nn
{ p , T , F , TF }
{
    \flowfram_if_tl_eq_frame_clist:NnnnTF
    \c_flowfram_odd_tl
    { #1 } { pagelist } { #2 }
    { \prg_return_true: }
    { \prg_return_false: }
}

If frame's page list is set to even:
\prg_new_conditional:Nnn \flowfram_if_frame_evenpages_only:nn
{ p , T , F , TF }
{
    \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: $\{\langle frame\ type\rangle\}\{\langle tag\rangle\}\{\langle idn\rangle\}$

```

\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: {<frame type>}{<tag>}{<idn>}

```

\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: {<frame type>}{<tag>}{<idn>}
\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: {<frame type>}{<tag>}{<idn>}{<choice value>} The <choice value>
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:nnn { flow } { hasframe } { \c@maxflow }
}
}
```

`\@n@wflowframe` Version 2.0 removed `\@n@wflowframe`

`\@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`

```
{\langle page list \rangle}{\langle width \rangle}{\langle height \rangle}{\langle x \rangle}{\langle y \rangle}
\cs_new_nopar:Nn \__flowfram_new_flow:nnnnn
{
  \__flowfram_new_flow:nnnnnn
  { #1 } { #2 } { #3 } { #4 } { #5 } { \int_use:N \c@maxflow }
}
{\langle page list \rangle}{\langle width \rangle}{\langle height \rangle}{\langle x \rangle}{\langle y \rangle}{\langle label \rangle}
\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` `{\langle flow-id \rangle}{\langle label \rangle}` Check proposed label for flow frame is unique. Version 2.0 removed `\@ff@checkuniqueidl`.

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

```

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

```

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

```

\NewDocumentCommand \getflowid { m m }
{
  \__flowfram_get_flow_id:e { #2 }
  \tl_set:Nx #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_flow_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 O { \int_use:N \c@maxstatic } }
{
  \__flowfram_new_static:nnnnnn
  { #2 } { #3 } { #4 } { #5 } { #6 } { #7 }
  \IfBooleanT { #1 }
  {
    \flowfram_frame_set_bool_true:nnn
    { static } { hasframe } { \c@maxstatic }
  }
}

\cs_new:Nn \__flowfram_new_static:nnnnnn
{
  \__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:nnnnnnnn
  { #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:N#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 \l__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_thumtbat_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:cVVn
        { __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
{ cVVn , 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 `\enablethumtbs`:

```

\cs_new:Nn \__flowfram_add_thumtbs:n { }
\cs_new:Nn \__flowfram_actual_add_thumtbs:n
{
    \tl_if_eq:eeT \g__flowfram_thumtbs_type_tl { #1 }
    {
        \bool_if:NTF \l__flowfram_starred_section_bool
        {
            \__flowfram_ttb_nonum:n { \l__flowfram_section_thumtbs_title_tl }
        }
        {
            \flowfram_if_mainmatter:TF
            {
                \__flowfram_ttb_num:n
                { \l__flowfram_section_thumtbs_title_tl }
            }
            {
                \__flowfram_ttb_notmainmatter_num:n
                { \l__flowfram_section_thumtbs_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 thumtbs 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 }

```

```

    {
      \cs_set_eq:cc { __flowfram_org_end_ #1 : } { @end #1 }
      \cs_set:cpn { @end #1 }
      {
        \__flowfram_post_section_unit:n { #1 }
        \use:c { __flowfram_org_end_ #1 : }
      }
    }
  }
}

\__flowfram_provide_section_unit:n { part }
\__flowfram_provide_section_unit:n { chapter }
\__flowfram_provide_section_unit:n { section }
\__flowfram_provide_section_unit:n { subsection }
\__flowfram_provide_section_unit:n { subsubsection }
\__flowfram_provide_section_unit:n { paragraph }
\__flowfram_provide_section_unit:n { subparagraph }

```

`\@ff@setsecthead` Define command to switch to adjusted section headings: Version 2.0 renamed `\@ff@setsecthead`.

```

\cs_new:Nn \__flowfram_set_shaped_section_headings:
{
  \tl_set:Nn \l__flowfram_current_pre_section_tl
  {
    \FLFsimpar
    \begin { minipage } { \linewidth }
    \let \par \endgraf
  }
  \tl_set:Nn \l__flowfram_current_post_section_tl
  {
    \end { minipage }
    \FLFsimpar
  }
}

```

`\@ff@getshape` Determine what shape command is being used: Version 2.0 renamed `\@ff@getshape`.

```

\cs_new:Nn \__flowfram_get_shape:n
{
  \tl_if_head_eq_meaning:nNTF { #1 } \parshape
  {
    \int_set_eq:NN
    \l__flowfram_shape_int
    \c_flowfram_shape_type_parshape_int
  }
  {
    \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 } \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:nnnn { 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
{
  \l_flowfram_parshape_tl
}
\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 }
\l_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 \l_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 }
      }
    }
  }
}
{
  \tl_clear:N \l__flowfram_postcontinued_tl
}
}

\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
}
```

\ffdynamicpostcontinued

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

\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:nnnnnn
  { #2 } { #3 } { #4 } { #5 } { #6 } { #7 }
}
```

```

\IfBooleanT { #1 }
{
    \flowfram_frame_set_bool_true:nnn
    { 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:nnnnnnn
    { #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:nn { \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}(\langle idn \rangle)` 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 }
}
{ }

```

`\@dynamictok` Version 2.0 removed `\@dynamictok`.

`\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`.

`\@empty` Version 2.0 removed `\@empty`.

`\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 { 0{} 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:nnnn
    { 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

```

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

```

\NewDocumentCommand \computefflowframearea { 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 indictating type of frame : 1 (flow), 2 (static), 3 (dynamic). The second number is its IDN. Values are stored in \ffareax, \ffareay, \ffareawidth and \ffareaheight. 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 \ffareax, \ffareay, \ffareawidth 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
  \ffareax { #1 } { evenx } { #2 }
  \flowfram_set_dim_to_frame_dim:Nnnn
```

```

\ffareay { #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 }
}

```

`\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 }
}

```

`\getdynamicevenbounds` Even pages:

```

\NewDocumentCommand \getdynamicevenbounds { 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 }
  }
}

```

`\@sgetdynamicevenbounds` Version 2.0 removed `\@sgetdynamicevenbounds`.

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

```

\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

$\langle type2 \rangle$ with IDN given by $\langle id2 \rangle$. 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:nnnnn
  { \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 }

```

```

    {
      \FLFabove>true
    }
    {
      \FLFabove=false
    }
  }
}

```

`\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:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
  {
    \__flowframe_check_odd_if_frame_below_by_idn:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
}

```

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

```

\cs_new:Nn \__flowframe_check_odd_if_frame_below_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 \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:nnnn
{
  \__flowframe_get_frame_bounds_by_type:nn { #1 } { #2 }
  \dim_add:Nn \ffareay { \ffareaheight }
  \dim_set_eq:NN \l__flowfram_tmpa_dim \ffareay
  \__flowframe_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_tpa_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 }
  {
    \FLFlefttrue
  }
  {
    \FLFleftfalse
  }
}

```

`\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 }
  }
}
}

```

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

```
\cs_new:Nn \__flowframe_check_odd_if_frame_right_by_idl:nnnn
{
  \__flowframe_get_frame_id:nn { #1 } { #2 }
  \__flowframe_get_frame_bounds_by_type:nn { #1 } { \l__flowframe_id_int }
  \dim_set_eq:NN \l__flowframe_tmpa_dim \ffareax
  \__flowframe_get_frame_id:nn { #3 } { #4 }
  \__flowframe_get_frame_bounds_by_type:nn { #3 } { \l__flowframe_id_int }
  \dim_add:Nn \ffareax { \ffareawidth }
  \dim_compare:nNnTF { \l__flowframe_tmpa_dim } > { \ffareax }
  {
    \FLFrighttrue
  }
  {
    \FLFrightfalse
  }
}
```

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

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

`\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 }
  {
    \FLFabovetrue
  }
  {
    \FLFabovefalse
  }
}

```

\@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 }
  {
    \FLFabovetrue
  }
  {
    \FLFabovefalse
  }
}

```

\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:nnnn
{
    \__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:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
  {
    \__flowframe_check_even_if_frame_right_by_idn:nnnn
    { #2 } { #3 } { #4 } { #5 }
  }
}

```

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

```

\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
  }
}

```

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

```

\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 $\langle id1 \rangle$ and $\langle id2 \rangle$ 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:nnnnn
  { \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

```



```

\ifFLleft
  \FFleft
\else
  \ifFLright
    \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_eveny_dim
{ #5 } { eveny } { \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_eveny_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_eveny_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
\FLFrighfalse
\dim_compare:nNnTF
{ \ffareax } > { \l__flowfram_x_dim + \l__flowfram_width_dim }
{
  \FLFrightrue
}
{
  \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

```

```

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

\IfSavedRelativeLocationAbove

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

\IfSavedRelativeLocationBelow

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

\IfSavedRelativeLocationLeft

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

\IfSavedRelativeLocationRight

```

\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 }
  }
}

\@srelflowloc Version 2.0 removed \@srelflowloc.

\@relflowloc Version 2.0 removed \@relflowloc.

```

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_info: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: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 }
    \__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_i: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:
`[\langle header-type \rangle][\langle header-id \rangle]{\langle col-1 \rangle}[\langle header-col1 \rangle]{\langle col-2 \rangle}[\langle header-col-2 \rangle]`

```

\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_i_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 }
  {
    \tl_gclear:N \g__flowfram_headed_two_col_ii_id_tl
  }
  {
    \IfBooleanTF { #1 }
    {
      \__flowfram_get_flow_id:n { #7 }
      \tl_gset:Ne \g__flowfram_headed_two_col_ii_id_tl
      { \int_use:N \l__flowfram_id_int }
    }
    {
      \bool_lazy_or:nnTF
      {
        \int_compare_p:nNn { #7 } < { \c_one_int }
      }
      {
        \int_compare_p:nNn { #7 } > { \c_maxflow }
      }
      {
        \msg_error:nnnn { flowfram } { idn-undefined }
        { flow } { #7 }
      }
      {
        \tl_gset:Ne \g__flowfram_headed_two_col_ii_id_tl
        { \int_eval:n { #7 } }
      }
    }
  }
}
}

```

Switch to the column associated with \onecolumn:

```

\cs_new:Nn \flowfram_set_one_col:
{

```

```

\tl_if_empty:NTF \g__flowfram_one_col_id_tl
{
  \msg_warning:nnn { flowfram } { misplaced-col-cmd } { \onecolumn }
}
{
  \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 }
  \int_compare:nNnTF
    { ##1 } = { \g__flowfram_one_col_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 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_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@footer if preferred:

\flowfram@dynamicfooter

```
\newcommand \flowfram@dynamicfooter [2]
{
  \g__flowfram_footercolor_tl
  \__flowfram_footer:n { #2 }
}

\tl_new:N \g__flowfram_footercolor_tl
\tl_gset:Nn \g__flowfram_footercolor_tl { \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: \end@dbfloat
\cs_set_eq:NN \end@dbfloat \end@float

```

`\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 \end@dbfloat \__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{absolute page}%
    }
    \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 \end@dblfloat \end@float
    \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@@
  \fi
}

```

```

\else
\expl@@mark@update@singlecol@structures@@
\fi
\@outputdblcol
\global \mparbottom \z@
\global \textfloatsheight \z@
\@flowfram@adjust@fdepth
\@floatplacement
}

```

\@flowfram@adjust@fdepth TODO investigate why this doesn't work as expected

```

\newcommand\@flowfram@adjust@fdepth{%
%\def\f@depth{1sp}%
}

```

\@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@
\@begindvi
\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{displayed frame}{0}%
\@flowfram@update@col@count{\c@this frame}%
\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_sfdf_width_dim
    { dynamic } { width } { #1 }
\if@twoside
  \ifodd\c@page
    \flowfram@dynamicheader
      { \l__flowfram_sfdf_width_dim }
      { \@oddhead }
  \else
    \flowfram@dynamicheader
      { \l__flowfram_sfdf_width_dim }
      { \@evenhead }
  \fi
\else
  \flowfram@dynamicheader
    { \l__flowfram_sfdf_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_sfdf_width_dim
    { dynamic } { width } { #1 }
}

```

```

\if@twoside
  \ifodd\c@page
    \flowfram@dynamicheader
    { \l__flowfram_sdf_width_dim }
    { \@oddhead }
  \else
    \flowfram@dynamicheader
    { \l__flowfram_sdf_width_dim }
    { }
  \fi
\else
  \flowfram@dynamicheader
  { \l__flowfram_sdf_width_dim }
  { \@thehead }
\fi
}

\cs_new:Nn \flowfram_dynamic_even_header:n
{
  \flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_sdf_width_dim
  { dynamic } { width } { #1 }
  \if@twoside
    \ifodd\c@page
      \flowfram@dynamicheader
      { \l__flowfram_sdf_width_dim }
      { }
    \else
      \flowfram@dynamicheader
      { \l__flowfram_sdf_width_dim }
      { \@evenhead }
    \fi
  \else
    \flowfram@dynamicheader
    { \l__flowfram_sdf_width_dim }
    { }
  \fi
}

\cs_new:Nn \flowfram_dynamic_footer:n
{
  \flowfram_set_dim_to_frame_dim:Nnnn
  \l__flowfram_sdf_width_dim
  { dynamic } { width } { #1 }
  \if@twoside
    \ifodd\c@page
      \flowfram@dynamicfooter
      { \l__flowfram_sdf_width_dim } { \@oddfoot }
    \else
      \flowfram@dynamicfooter
      { \l__flowfram_sdf_width_dim } { \@evenfoot }
    \fi
  \else
    \flowfram@dynamicfooter
    { \l__flowfram_sdf_width_dim } { }
  \fi
}

```



```

\fi
\else
\flowfram@dynamicfooter
{ \l__flowfram_sdf_width_dim } { \@thefoot }
\fi
}
\cs_new:Nn \flowfram_dynamic_odd_footer:n
{
\flowfram_set_dim_to_frame_dim:Nnnn
\l__flowfram_sdf_width_dim
{ dynamic } { width } { #1 }
\if@twoside
\ifodd\c@page
\flowfram@dynamicfooter
{ \l__flowfram_sdf_width_dim } { \@oddfoot }
\else
\flowfram@dynamicfooter
{ \l__flowfram_sdf_width_dim } { }
\fi
\else
\flowfram@dynamicfooter
{ \l__flowfram_sdf_width_dim } { \@thefoot }
\fi
}
\cs_new:Nn \flowfram_dynamic_even_footer:n
{
\flowfram_set_dim_to_frame_dim:Nnnn
\l__flowfram_sdf_width_dim
{ dynamic } { width } { #1 }
\if@twoside
\ifodd\c@page
\flowfram@dynamicfooter
{ \l__flowfram_sdf_width_dim } { }
\else
\flowfram@dynamicfooter
{ \l__flowfram_sdf_width_dim } { \@evenfoot }
\fi
\else
\flowfram@dynamicfooter
{ \l__flowfram_sdf_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@specialempty` Used with `\thispagestyle`:

```
\newcommand \ps@specialempty { \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
    }
  }
}
{
  \int_incr:N \l__flowfram_col_int

```

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 }`
`}`

`\if@ff@nwp` Conditional to keep track of whether or not a new page is required. Version 2.0 replaced legacy conditional with `l3bool`.
`%\newif\if@ff@nwp`
`\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
  {

```



```

{
  \__flowfram_dynamic_check_if_this_page:nn
  { \g__flowfram_pagecounter_tl } { #1 }
}
\cs_new:Nn \__flowfram_dynamic_check_if_this_page:nn
{
  \bool_gset_true:N \g__flowfram_not_this_frame_bool
  \flowfram_set_clist_to_frame_clist:Nnnn
  \l__flowfram_pages_clist
  { dynamic } { pagelist } { #2 }

  \__flowfram_check_if_this_page:n { #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. Version 2.0 renamed `\@setcolbox`.

```

\cs_new:Nn \__flowfram_set_column_box:n
{
  \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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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 }
        }
        {
          \hbox_to_zero:n
          {
            \kern
            \flowfram_frame_use_dim:nnn { flow } { posx } { #1 }
            \__flowfram_do_flow_bounding_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_bounding_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_bounding_box:n {#1}
        \hss
        }
    }
}
}
}
}

```

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. Version 2.0 renamed \ff@s@t@doubleboxoffset

```

\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@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 \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
{
```

```
  \__flowframe_check_if_this_page:nn
  { \g__flowframe_pagecounter_tl } { #1 }
  \bool_if:NF \g__flowframe_not_this_frame_bool
  {
```

```
    \@killglue
```

```
    \legacy_if:nTF { @twoside }
```

```
    {
```

```
      \int_if_odd:nTF { \c@page }
```

```
      {
```

```
        \flowframe_set_dim_to_frame_dim:Nnnn
```

```
        \l__flowframe_x_dim
```

```
        { flow } { posx } { #1 }
```

```
        \flowframe_set_dim_to_frame_dim:Nnnn
```

```
        \l__flowframe_y_dim
```

```
        { flow } { posy } { #1 }
```

```
      }
```

```
    {
```

```
      \flowframe_set_dim_to_frame_dim:Nnnn
```

```
      \l__flowframe_x_dim
```

```
      { flow } { evenx } { #1 }
```

```
      \flowframe_set_dim_to_frame_dim:Nnnn
```

```
      \l__flowframe_y_dim
```

```
      { flow } { eveny } { #1 }
```

```
    }
```

```
  }
```

```
  {
```

```
    \flowframe_set_dim_to_frame_dim:Nnnn
```

```
    \l__flowframe_x_dim
```

```
    { flow } { posx } { #1 }
```

```
    \flowframe_set_dim_to_frame_dim:Nnnn
```

```
    \l__flowframe_y_dim
```

```
    { flow } { posy } { #1 }
```

```
  }
```

```
  \__flowframe_get_margin_pos:e
```

```
  {
```

```
    \flowframe_frame_use_tl:nnn { flow } { margin } { #1 }
```

```
  }
```

```
  \tl_if_eq:NNTF \l__flowframe_margin_tl \c_flowframe_left_tl
```

```
  {
```

```
    \dim_add:Nn \l__flowframe_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
    }
  }
}

```

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_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 { een }

```

\@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: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 }
      {
        \__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
    {
      \@dotheft \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
    }
    \hss
}
}

```

`\@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 \TeX 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
        \@dodynamicthefoot
        \vbadness=\@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
}
\beginingroup
  \@dblfloatplacement
  \@startdblcolumn
  \@whiles \if@fcolmade \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:nnnnF
  { \int_eval:n { \g__flowfram_pagecounter_tl } }
  { 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_tl } }
  { 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_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_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:Nn \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_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 \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
      }
    }
  }
}

```

\@sflowswitchonnexonly Version 2.0 removed \@sflowswitchonnexonly.

\@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:N\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:N\l__flowfram_pages_clist
{
  \int_eval:n { \g__flowfram_pagecounter_tl } ,
  \int_use:N \l__flowfram_tmpb_int
}
}
}
\tl_gput_right:N\l__flowfram_output_adjust_frames_tl
{
  \exp_not:N \flowsetpagelist
  { \int_eval:n { #1 } }
  { \l__flowfram_pages_clist }
}
}

```

\@sflowswitchonnextoddoneonly Version 2.0 removed \@sflowswitchonnextoddoneonly.

\@flowswitchonnextoddoneonly Version 2.0 removed \@flowswitchonnextoddoneonly.

\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 \flowadddexclusion
    { \int_eval:n { #1 } }
    { \int_use:N \l__flowfram_tmpb_int }
  }
}

```

\@sflowswitchoffnextonly Version 2.0 removed \@sflowswitchoffnextonly.

\@flowswitchoffnextonly Version 2.0 removed \@flowswitchoffnextonly.

\flowswitchoffnextoddonly Switch off the listed flow frames for just the next odd page

```

\NewDocumentCommand \flowswitchoffnextoddonly { 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 \flowadddexclusion
    { \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 } }
        }
    }
    \tl_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
        }
    }
}

```

\@sdynamicswitchonnextodd Version 2.0 removed \@sdynamicswitchonnextodd.

\@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 }
    }
}

```

\@dynamicsswitchoffnext Version 2.0 removed \@dynamicsswitchoffnext.

\@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.

```

\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 \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:Ne \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

\dynamicsswitchonnextoddonly Switch on the listed dynamic frames for just the next odd page

```

\NewDocumentCommand \dynamicsswitchonnextoddonly { 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
}
}
\__flowfram_dynamic_check_if_this_page:nn
{ \l__flowfram_tmpb_int } { #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 } }
        { \l__flowfram_pages_clist }
    }
}

\sdynamicswitchonnextoddoneonly Version 2.0 removed \@sdynamicswitchonnextoddoneonly

\@dynamicswitchonnextoddoneonly Version 2.0 removed \@dynamicswitchonnextoddoneonly.

\dynamicsswitchoffnextonly Switch off the listed dynamic frames for just the next page
\NewDocumentCommand \dynamicsswitchoffnextonly { 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 \dynamicaddeexclusion
        { \int_eval:n { #1 } }
        { \int_use:N \l__flowfram_tmpb_int }
    }
}

\@sdynamicswitchoffnextonly Version 2.0 removed \@sdynamicswitchoffnextonly.

\@dynamicsswitchoffnextonly Version 2.0 removed \@dynamicsswitchoffnextonly

\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 }
  }
}

```

\dynamicsswitchoffnextoddonly Version 2.0 removed \@dynamicsswitchoffnextoddonly.

\dynamicsswitchoffnextoddonly Version 2.0 removed \@dynamicsswitchoffnextoddonly.

\staticsswitchonnext Switch on the listed static frames from the next page onwards

```

\NewDocumentCommand \staticsswitchonnext { 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 }
      }
    }
  }
}

```

\@sstaticswitchonnext Version 2.0 removed \@sstaticswitchonnext.

\@staticswitchonnext Version 2.0 removed \@staticswitchonnext.

\staticswitchonnextodd Switch on the listed static frames from the next odd page onwards

```

\NewDocumentCommand \staticswitchonnextodd { 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.

\staticswitchonnexonly Switch on the listed static frames for just the next page

```

\NewDocumentCommand \staticswitchonnexonly { 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
      }
    }
  }
}

```

}

\@sstaticswitchonnexonly Version 2.0 removed \@sstaticswitchonnexonly.

\@staticswitchonnexonly Version 2.0 removed \@staticswitchonnexonly.

\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 }
}
}

```

\@sstaticswitchonnextoddoneonly Version 2.0 removed \@sstaticswitchonnextoddoneonly.

\@staticswitchonnextoddoneonly Version 2.0 removed \@staticswitchonnextoddoneonly.

\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 \staticaddeexclusion
{ \int_eval:n { #1 } }
{ \int_use:N \l__flowfram_tmpb_int }
}
}

```

`\sstaticswitchoffnextodonly` Version 2.0 removed `\@sstaticswitchoffnextodonly`.

`\@staticswitchoffnextodonly` Version 2.0 removed `\@staticswitchoffnextodonly`.

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 `\@capttype` so that the `\caption` command may be used. Version 2.0 added paragraph breaks and ignore spaces.

```

statictable (env.)
\newenvironment{statictable}
{\par\def\@capttype{table}\ignorespaces}
{\par\ignorespacesafterend}

staticfigure (env.)
\newenvironment{staticfigure}
{\def\@capttype{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
\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 `__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 { O{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 }
  \ifffvadjust
  \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:Ne \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 }
  \ifffvadjust
    \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:

$\backslash\text{Ncolumninarea}[\langle pages \rangle]\{\langle n \rangle\}\{\langle w \rangle\}\{\langle h \rangle\}\{\langle x \rangle\}\{\langle y \rangle\}$.

```
\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 $\langle n \rangle$ 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 \onecolumnstop
  {
    \__flowfram_one_column_with_top_in_area:nnnnnnn
    { #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 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:nnn
  { #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 { 0{all} 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[ $\langle pages \rangle$ ]{ $\langle H \rangle$ }{ $\langle w \rangle$ }{ $\langle h \rangle$ }{ $\langle x \rangle$ }{ $\langle y \rangle$ }.
\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[ $\langle pages \rangle$ ]{ $\langle H \rangle$ }{ $\langle w \rangle$ }{ $\langle h \rangle$ }{ $\langle x \rangle$ }{ $\langle y \rangle$ }.
\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 $\langle type \rangle$ frame above both of them. Syntax:

```

\twocolumntop[ $\langle pages \rangle$ ]{ $\langle type \rangle$ }{ $\langle H \rangle$ }

```

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 { O{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 { O{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_sdf_height_dim { #3 }

```

Work out where to put the header frame.

```

  \dim_set:Nn \l__flowfram_y_dim
  { #5 - \l__flowfram_sdf_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[ $\langle pages \rangle$ ]{ $\langle type \rangle$ }{ $\langle n \rangle$ }{ $\langle H \rangle$ }
\NewDocumentCommand \Ncolumnstop { O{all} m m m o }
{
    \__flowfram_only_preamble:Nn \Ncolumnstop
    {
        \__flowfram_n_column_with_top_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 }
}
}
}
}

```

`\Ncolumnstop` Special case for static frame.

```

\NewDocumentCommand \Ncolumnstop { O{all} m m m o }
{
    \__flowfram_only_preamble:Nn \Ncolumnstop
    {
        \__flowfram_n_column_with_top_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 }
}
}
}
}

```

`\Ncolumnstop` Special case for dynamic frame.

```

\NewDocumentCommand \Ncolumnstop { O{all} m m m o }
{
    \__flowfram_only_preamble:Nn \Ncolumnstop
    {
        \__flowfram_n_column_with_top_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 }
}
}

```

\Ncolumntopinarea Again test to make sure the user requested a sensible number. \Ncolumntopinarea

```

[<pages>]{<type>}{<n>}{<H>}{<w>}{<h>}{<x>}{<y>}
\NewDocumentCommand \Ncolumntopinarea { 0{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 }
    }
  }
}

{<pages>}{<type>}{<n>}{<H>}{<w>}{<h>}{<x>}{<y>}
\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 }
    }
  }
}
}

```


`\@Ncolumnstopinarea` 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 `\@Ncolumnstopinarea`.

```
\cs_new:Nn \__flowfram_N_column_with_top_in_area:nnnnnnnn
{
  \dim_set:Nn \l__flowfram_sdf_height_dim { #4 }
```

Work out where to put the static frame.

```
\dim_set:Nn \l__flowfram_y_dim
{ #6 - \l__flowfram_sdf_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`

```
\ifffvadjust
  \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 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:nnnnnnnn
    { #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:nnnnnnnn
    { #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:nnnnnnn
    { #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[$\langle pages \rangle$]{ $\langle type \rangle$ }{ $\langle H \rangle$ }{ $\langle w \rangle$ }{ $\langle h \rangle$ }{ $\langle x \rangle$ }{ $\langle y \rangle$ },`

where $\langle H \rangle$ is the $\langle type \rangle$ frame's height. The area is defined by bottom left co-ordinates $(\langle x \rangle, \langle y \rangle)$ width $\langle w \rangle$, and height $\langle h \rangle$.

```

\NewDocumentCommand \onecolumnbottominarea
{ 0{all} m m m m m m o }
{
  \__flowfram_only_preamble:Nn \onecolumnbottominarea
  {
    \__flowfram_one_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_one_column_with_bottom_in_area:nnnnnnn
{
  \dim_set:Nn \l__flowfram_sdf_height_dim { #3 }

```

```

\dim_set:Nn \l__flowfram_height_dim
{
  #5
  - \l__flowfram_sfdh_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 { 0{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[$\langle pages \rangle$]{ $\langle type \rangle$ }{ $\langle H \rangle$ }.`

```

\NewDocumentCommand \Ncolumnbottom { O{all} m 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
 $[\langle pages \rangle] \{ \langle type \rangle \} \{ \langle n \rangle \} \{ \langle H \rangle \} \{ \langle w \rangle \} \{ \langle h \rangle \} \{ \langle x \rangle \} \{ \langle y \rangle \}$

```

\NewDocumentCommand \Ncolumnbottominarea { O{all} m 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:nnnnnnnn
      { #1 } { #2 } { #4 } { #5 } { #6 } { #7 } { #8 }
    }
  } { 2 }
  {
    \_flowfram_two_column_with_bottom_in_area:nnnnnnnn
    { #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:nnne { flowfram } { invalid-num-frames }
    { flow } { \int_eval:n { #3 } }
  }
}
}

```

Syntax: {<pages>}{<type>}{<n>}{<H>}{<w>}{<h>}{<x>}{<y>}

```

\cs_new:Nn \_flowfram_N_column_with_bottom_in_area:nnnnnnnn
{
  \dim_set:Nn \l_flowfram_sdf_height_dim { #4 }

```

Work out height of the flow frames.

```

\dim_set:Nn \l_flowfram_height_dim
{ #6 - \l_flowfram_sdf_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_sdf_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 $\langle W1 \rangle$ 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}.

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:nnnnnnnnn
    }
}

```

```

    { #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:
 $[\langle pages \rangle][\langle xoffset \rangle]\{\langle H \rangle\}\{\langle W1 \rangle\}\{\langle C1 \rangle\}\{\langle L1 \rangle\}\{\langle W2 \rangle\}\{\langle C2 \rangle\}\{\langle L2 \rangle\}$ where the
 first frame starts at $\langle xoffset \rangle$.

```

\cs_new:Nn \__flowfram_v_two_tone_bottom:nnnnnnnnn
{
  \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
{
  \compteleftedgeodd { \l__flowfram_x_dim }
  \if@twoside
    \compteleftedgeeven { \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:nnnnnn
    { #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 $\langle W1 \rangle$ 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 $\langle Ln \rangle$.

```
\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_sdf_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_sdf_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_sfdf_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:nnnnnnnnn { #1 } { #3 } { #4 } { #5 } { #6 }
  \__flowfram_h_two_tone_left:nnnnnnnnn { #1 } { #3 } { #7 } { #8 } { #9 }
}

```

\htwotoneleft Two tone horizontal strips along left border Syntax: \htwotoneleft[$\langle pages \rangle$]
[$\langle y \text{ offset} \rangle$]{ $\langle width \rangle$ }{ $\langle H1 \rangle$ }{ $\langle C1 \rangle$ }{ $\langle L1 \rangle$ }{ $\langle H2 \rangle$ }{ $\langle C2 \rangle$ }{ $\langle L2 \rangle$ }

```

\NewDocumentCommand \htwotoneleft
{ 0{all} 0 {\c_zero_dim} m m m m m m }
{
  \__flowfram_only_preamble:Nn \htwotoneleft
  {
    \__flowfram_h_two_tone_left:nnnnnnnnn
    { #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 }
{
  \__flowfram_only_preamble:Nn \htwotoneright
  {
    \__flowfram_h_two_tone_right:nnnnnnnnn

```



```

        { #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:nnnnnnnnn
{
  \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 { #2 }
  \dim_sub:Nn \l__flowfram_x_dim { #3 }
  \dim_sub:Nn \l__flowfram_evenx_dim { #3 }
  \__flowfram_h_two_tone_left:nnnnnnnnn { #1 } { #3 } { #4 } { #5 } { #6 }
  \__flowfram_h_two_tone_left:nnnnnnnnn { #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_sfd_width_dim { #3 }
  \__flowfram_next_h_N_band:
}
```

`\@nextthband` Make next static frame. Syntax:

`\@nextthband{<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

`\@nextthband`.

```
\cs_new:Nn \__flowfram_next_h_band:nnnnn
{
  \dim_set:Nn \l__flowfram_sfd_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:nnnnnn
    { #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:nnnnn
      { #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[$\langle y \text{ top} \rangle$][$\langle y \text{ bottom} \rangle$]{ $\langle frame1 \text{ type} \rangle$ }{ $\langle IDN1 \rangle$ }{ $\langle frame2 \text{ type} \rangle$ }{ $\langle IDN2 \rangle$ }.

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 $\langle y \text{ bottom} \rangle$.

\ffcolumnseprule This has changed in v1.09. Define **\ffcolumnseprule** and use instead of **\columnseprule**

```

\newlength\ffcolumnseprule
\setlength{\ffcolumnseprule}{2pt}

```

\ffruleddeclarations 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*{\ffruleddeclarations}{}

```

\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***[$\langle y \text{-top} \rangle$][$\langle y \text{-bottom} \rangle$]{ $\langle type-1 \rangle$ }{ $\langle id \rangle$ }{ $\langle type-2 \rangle$ }{ $\langle id \rangle$ }

```

\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-top \rangle}{\langle y-bottom \rangle}\langle type-1 int \rangle{\langle idn \rangle}\langle type-2 int \rangle{\langle idn \rangle}` Version 2.0 replaced `\@@insert@vrule`

```

\cs_new:Nn \__flowfram_insert_v_rule:nnNnNn
{
  \__flowfram_get_frame_bounds_by_typeid:nn { #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:nn { #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: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_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 { }
}

```

`\@insertvrule` Version 2.0 removed `\@insertvrule`.

`\@@insertvrule` Version 2.0 removed `\@@insertvrule`.

`\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
        { \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_h_rule:nnNnNn
        { #2 } { #3 }
      \l__flowfram_type_int { \l__flowfram_id_int }
      \l__flowfram_type_ii_int { \l__flowfram_id_ii_int }
    }
    {
      \__flowfram_insert_h_rule:nnNnNn
        { #2 } { #3 }
      \l__flowfram_type_int { #5 }
      \l__flowfram_type_ii_int { #7 }
    }
  }
}
}
}

```

\@inserthrul Version 2.0 removed \@inserthrul.

\@@inserthrul Version 2.0 removed \@@inserthrul.

\@@insert@hrul 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:

{\langle x left \rangle}{\langle x right \rangle}{\langle type ID \rangle}{\langle IDN \rangle}{\langle type ID \rangle}{\langle IDN \rangle}.

```

\cs_new:Nn \__flowfram_insert_h_rule:nnNnNn
{
  \__flowfram_get_frame_bounds_by_typeid:nn { #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:nn { #5 } { #6 }
  \dim_compare:nNnT
    { \l__flowfram_left_y_dim } > { \ffareay }

```

```

{
  \__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_width_dim \ffareawidth
  \__flowfram_swap_dim:NN \l__flowfram_left_height_dim \ffareaheight
}
\dim_set:Nn \l__flowfram_y_dim
{
  \l__flowfram_left_y_dim
  + \l__flowfram_left_height_dim
}
\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_sfdwidth_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 }
\abox \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_t1 } }
        { #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_t1 { #3 }
        \@starttoc { ttb }
        \__flowfram_create_thumbtabs:nn { #1 } { #2 }
        \cs_set:Nn \__flowfram_write_backmatter_ttb:

```

```

    {
      \addtocontents {ttb}
      {
        \protect \thumtbat@backmatter
        { \theFramePageCounter }
      }
    }
  \cs_set:Nn \__flowfram_no_thumtbat_warn:
  {
    \msg_warning:nn { flowfram } { no-thumtbat-rerun }
  }
}
{
  \msg_error:nnn { flowfram } { section-unit-not-defined } { #3 }
}
}

```

Warning to produce if no thumtbat:

```

\cs_new:Nn \__flowfram_no_thumtbat_warn:
{
  \msg_warning:nn { flowfram } { no-thumtbat-missing-makethumtbat }
}

```

Write the page number to the ttb file when \backmatter is used to get the end page range if not thumtbat 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

\thumtbat The thumtbat file, .ttb, will have a series of \thumtbat 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{\thumtbat}[4]{%
  \stepcounter { maxthumtbat }
  \__flowfram_message:nnnn { info-found-thumtbat }
  { #1 } { #2 } { #3 } { #4 }
  \tl_new:c { g__thumtbat_ \romannumeral \c@maxthumtbat _pages_tl }
  \int_new:c { g__thumtbat_ \romannumeral \c@maxthumtbat _start_int }
  \int_gset:cn { g__thumtbat_ \romannumeral \c@maxthumtbat _start_int }
  { #1 }
  \tl_new:c { g__thumtbat_ \romannumeral \c@maxthumtbat _unitnum_tl }
  \tl_gset:cn
  { g__thumtbat_ \romannumeral \c@maxthumtbat _unitnum_tl }
}

```

```

    { #2 }
\tl_new:c { g__thumbstab_ \romannumeral \c@maxthumbtabs _title_tl }
\tl_gset:cn { g__thumbstab_ \romannumeral \c@maxthumbtabs _title_tl }
    { #3 }
\tl_new:c { g__thumbstab_ \romannumeral \c@maxthumbtabs _link_tl }
\tl_gset:cn { g__thumbstab_ \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__thumbstab_ \romannumeral #1 _start_int }
}
\cs_new:Nn \__flowfram_thumbtab_unitnum:n
{
    \tl_use:c { g__thumbstab_ \romannumeral #1 _unitnum_tl }
}
\cs_new:Nn \__flowfram_thumbtab_pages:n
{
    \tl_use:c { g__thumbstab_ \romannumeral #1 _pages_tl }
}
\cs_new:Nn \__flowfram_thumbtab_set_pages:nn
{
    \tl_gset:cn { g__thumbstab_ \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__thumbstab_ \romannumeral #1 _title_tl }
}

```

`\SetThumbTabTitle` The title for individual thumbtabs may be changed.

```

\NewDocumentCommand \SetThumbTabTitle { m m }
{
    \tl_if_exist:cTF { g__thumbstab_ \romannumeral #1 _title_tl }
    {
        \tl_gset:cn { g__thumbstab_ \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:Ne \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 `eventthumbtab1`, `eventthumbtab2`, etc, and `eventthumbtabindex1`, `eventthumbtabindex2`, 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: $\{\langle index \rangle\}\{\langle height \rangle\}$

```
\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:NVneen
\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:neee {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_thumtab_seq
    {
        \flowfram_frame_set_clist:nnnn
        { dynamic } { hide } { ##1 } \c_true_bool
    }
    \seq_map_inline:Nn \g__flowfram_even_thumtab_index_seq
    {
        \flowfram_frame_set_clist:nnnn
        { dynamic } { pagelist } { ##1 } { none }
    }
}
}

```

`\thumbtindex` Show thumtab indexes on current page (if no optional argument) or for the given page list. Version 2.0 added optional argument.

```

\NewDocumentCommand \thumbtindex { o }
{
    \IfValueTF { #1 }
    {
        \clist_set:Ne \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:Ne \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_thumtbs_bool
{
    \seq_map_pairwise_function:NNN
    \g__flowfram_thumtbs_index_seq
    \g__flowfram_even_thumtbs_index_seq
    \__flowfram_set_thumtbs_pages:nn
}
{
    \seq_map_inline:Nn \g__flowfram_thumtbs_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 }
    }

```

```

    }
  }
}

```

`\setthumtabindex` Only change settings for the thumtab index. This can take a comma-separated number list.

```

\NewDocumentCommand \setthumtabindex { m m }
{
  \tl_if_eq:nnTF { #1 } { all }
  {
    \int_step_inline:nn { \c@maxthumtabs }
    {
      \__flowfram_set_thumtab_index:nn { ##1 } { #2 }
    }
  }
  {
    \clist_map_inline:nn { #1 }
    {
      \int_compare:nNnTF { ##1 } < { \c_one_int }
      {
        \msg_error:nnn { flowfram }
        { invalid-thumtab-index }
        { ##1 }
      }
      {
        \int_compare:nNnTF
        { ##1 } >
        { \seq_count:N \g__flowfram_thumtab_seq }
        {
          \msg_warning:nnn {flowfram} {cant-find-thumtab} { ##1 }
        }
        {
          \__flowfram_set_thumtab_index:nn { ##1 } { #2 }
        }
      }
    }
  }
}

```

`\@setthumtabindex` Change setting for individual thumtab index entry. Version 2.0 replaced `\@setthumtabindex`.

```

\cs_new:Nn \__flowfram_set_thumtab_index:nn
{

```

Find the ID for the index thumtab:

```

\int_set:Nn \l__flowfram_id_int
{
  \seq_item:Nn
  \g__flowfram_thumtab_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 }
        }
    }
}

```

`\tocandthumbtabindex` 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`type Sectioning type for the minitoc, by default it is the same as the thumbtabs Version 2.0 replaced `\@ttb@minitoc`type.

```

\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
\@nobeakfalse
\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: `\int-var\{toc line\}` 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:Nc \l__flowfram_remainder_tl
{
\tl_tail:n { #2 }
}
```

Now get the first argument:

```
\tl_set:Nc \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_mininitoc_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_mininitoc_ \romannumeral \c@maxminitoc _tl }
{
\tl_new:c { g__flowfram_mininitoc_ \romannumeral \c@maxminitoc _tl }
}
\tl_gput_right:cn
{ g__flowfram_mininitoc_ \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}{0pt}

\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}[2026/04/25 v2.2 (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 \LaTeX 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 \LaTeX :

```
\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 \jdrouline { 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
```

Provide command to restore full justification (may be used by import function if justification switches in blocks of text).

```
\let\@flowframtk@cr\
\skip_const:Nn \c__flowframtk_org_@right_skip { \@rightskip }
\skip_const:Nn \c__flowframtk_org_right_skip { \rightskip }
\skip_const:Nn \c__flowframtk_org_left_skip { \leftskip }
\int_const:Nn \c__flowframtk_org_finalhyphendemerits { \finalhyphendemerits }
\dim_const:Nn \c__flowframtk_org_parindent_dim { \parindent }
\skip_const:Nn \c__flowframtk_org_parfill_skip { \parfillskip }
```

`\flowframtkNoRagged`

```
\NewDocumentCommand\flowframtkNoRagged{}{%
\let\@flowframtk@cr
\@rightskip \c__flowframtk_org_@right_skip
\rightskip \c__flowframtk_org_right_skip
\leftskip \c__flowframtk_org_left_skip
\finalhyphendemerits \c__flowframtk_org_finalhyphendemerits
\parindent \c__flowframtk_org_parindent_dim
\parfillskip \c__flowframtk_org_parfill_skip
}
```

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 some convenient commands for use in the above.

```
\FlowframTkScopeIfGroup{<class name>}{<object>}
```

\FlowframTkScopeIfGroup

This command is designed for \flowframtkencapobject. If the <class name> is JDRGroup it will add scoping around <object> otherwise it will just do <object>.

```
\newcommand\FlowframTkScopeIfGroup[2]{
  \tl_if_eq:nnTF { #1 } { JDRGroup }
  {
    \begin { pgfscope }
    #2
    \end { pgfscope }
  }
  { #2 }
}
```

```
\FlowframTkBeginScopeIfGroup{<class name>}
```

\FlowframTkBeginScopeIfGroup

Similar to the above but designed for the paired markup commands.

```
\newcommand\FlowframTkBeginScopeIfGroup[1]{
  \tl_if_eq:nnT { #1 } { JDRGroup }
  {
    \begin { pgfscope }
  }
}
```

```
\FlowframTkEndScopeIfGroup{<class name>}
```

\FlowframTkEndScopeIfGroup

```
\newcommand\FlowframTkEndScopeIfGroup[1]{
  \tl_if_eq:nnT { #1 } { JDRGroup }
  {
    \end { pgfscope }
  }
}
```

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_flowframtkutils_annotate_bool
\bool_set_false:N \l_flowframtkutils_annotate_bool

```

Include arrows:

```

\bool_new:N \l_flowframtkutils_annotate_arrow_bool
\bool_set_true:N \l_flowframtkutils_annotate_arrow_bool

\dim_new:N \l_flowframtkutils_annotate_offsetx_dim
\dim_set:Nn \l_flowframtkutils_annotate_offsetx_dim { 1cm }
\dim_new:N \l_flowframtkutils_annotate_offsety_dim
\dim_set:Nn \l_flowframtkutils_annotate_offsety_dim { 1cm }

\int_new:N \l_flowframtkutils_annotate_location_int
\int_new:N \l_flowframtkutils_current_annotate_location_int
\int_const:Nn \c_flowframtkutils_annotate_location_auto_int { 0 }
\int_const:Nn \c_flowframtkutils_annotate_location_north_int { 1 }
\int_const:Nn \c_flowframtkutils_annotate_location_northeast_int { 2 }
\int_const:Nn \c_flowframtkutils_annotate_location_east_int { 3 }
\int_const:Nn \c_flowframtkutils_annotate_location_southeast_int { 4 }
\int_const:Nn \c_flowframtkutils_annotate_location_south_int { 5 }
\int_const:Nn \c_flowframtkutils_annotate_location_southwest_int { 6 }
\int_const:Nn \c_flowframtkutils_annotate_location_west_int { 7 }
\int_const:Nn \c_flowframtkutils_annotate_location_northwest_int { 8 }

```

```

\cs_new:Nn \__flowframtkutils_annotate_uncover:nn
{
  \flowframtkutils_uncover:nn { #1 } { #2 }
}

```

Syntax: $\langle tl\ var \rangle \{ \langle description \rangle \} \{ \langle tag \rangle \}$

```

\cs_new:Nn \__flowframtkutils_get_annotation_text:Nnn
{
  \tl_if_empty:nF { #2 }
  {
    \tl_set:Nn #1 { #2 }
  }
}

```

```

\bool_new:N \l_flowframtkutils_default_auto_bool
\bool_set_false:N \l_flowframtkutils_default_auto_bool
\tl_new:N \l_flowframtkutils_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 \flowframtkutils_get_default_overlay:Nnn
{
  \bool_if:NT \l_flowframtkutils_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__flowframtkutils_image_cy_dim
\dim_new:N \l__flowframtkutils_image_width_dim
\dim_new:N \l__flowframtkutils_image_height_dim

```

Object boundary:

```

\dim_new:N \l__flowframtkutils_object_minx_dim
\dim_new:N \l__flowframtkutils_object_miny_dim
\dim_new:N \l__flowframtkutils_object_maxx_dim
\dim_new:N \l__flowframtkutils_object_maxy_dim
\dim_new:N \l__flowframtkutils_object_cx_dim
\dim_new:N \l__flowframtkutils_object_cy_dim
\dim_new:N \l__flowframtkutils_object_width_dim
\dim_new:N \l__flowframtkutils_object_height_dim

\dim_new:N \l__flowframtkutils_x_dim
\dim_new:N \l__flowframtkutils_y_dim

```

Keep track of current group:

```

\int_new:N \l_flowframtkutils_group_level_int

\cs_new:Nn \flowframtkutils_uncover_encap:nnnnnnnn
{

```

Default to no overlay:

```

\cs_set_eq:NN \__flowframtkutils_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_flowframtkutils_overlay_int
\tl_clear:N \l_flowframtkutils_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_flowframtkutils_group_level_int
```

The bottom left position will be in the form `\pgfpoint{⟨x⟩}{⟨y⟩}`.

```
\pgfextractx \l_flowframtkutils_image_minx_dim { #5 }
\pgfextracty \l_flowframtkutils_image_miny_dim { #5 }
```

Width and height:

```
\dim_set:Nn \l_flowframtkutils_image_width_dim { #6 }
\dim_set:Nn \l_flowframtkutils_image_height_dim { #7 }
```

Calculate maximum.

```
\dim_set:Nn \l_flowframtkutils_image_maxx_dim
{
  \l_flowframtkutils_image_minx_dim
  + \l_flowframtkutils_image_width_dim
}
\dim_set:Nn \l_flowframtkutils_image_maxy_dim
{
  \l_flowframtkutils_image_miny_dim
  + \l_flowframtkutils_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_flowframtkutils_overlay_int
  \tl_set:NV
    \l__flowframtkutils_propval_tl
    \g_flowframtkutils_overlay_int
}
}
{

```

Tag setting takes precedence.

```

\tl_if_empty:nF { #4 }
{
  \prop_get:NnN \l_flowframtkutils_tag_overlay_prop { #4 }
  \l__flowframtkutils_propval_tl
}
\quark_if_no_value:NT
  \l__flowframtkutils_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_flowframtkutils_object_type_overlay_prop { #2 }
\l__flowframtkutils_propval_tl

```

If no object setting either, get the default:

```

\quark_if_no_value:NT

```

```

        \l__flowframtkutils_propval_tl
      {
        \flowframtkutils_get_default_overlay:Nnn
        \l__flowframtkutils_propval_tl
        { #1 } { #2 }
      }
    }
  }
}

```

Set the overlay and uncover functions as appropriate.

```

\quark_if_no_value:NTF
  \l__flowframtkutils_propval_tl
  {
    \tl_clear:N \l__flowframtkutils_overlay_tl
  }
  {
    \tl_if_empty:NF \l__flowframtkutils_propval_tl
    {
      \cs_set_eq:NN
      \__flowframtkutils_current_encap:nn
      \__flowframtkutils_uncover:nn
    }
    \tl_set_eq:NN
    \l__flowframtkutils_overlay_tl
    \l__flowframtkutils_propval_tl
  }
\bool_if:NT \l_flowframtkutils_annotate_bool
{

```

Annotation on. Store the current object's bounding box dimensions.

```

\pgfextractx \l__flowframtkutils_object_minx_dim { #5 }
\pgfextracty \l__flowframtkutils_object_miny_dim { #5 }
\dim_set:Nn \l__flowframtkutils_object_width_dim { #6 }
\dim_set:Nn \l__flowframtkutils_object_height_dim { #7 }

```

Calculate maximum point.

```

\dim_set:Nn \l__flowframtkutils_object_maxx_dim
{
  \l__flowframtkutils_object_minx_dim
  + \l__flowframtkutils_object_width_dim
}
\dim_set:Nn \l__flowframtkutils_object_maxy_dim
{
  \l__flowframtkutils_object_miny_dim
  + \l__flowframtkutils_object_height_dim
}

```

Calculate centre point.

```

\dim_set:Nn \l__flowframtkutils_object_cx_dim
{
  \l__flowframtkutils_object_minx_dim

```

```

        + 0.5 \l__flowframtkutils_object_width_dim
      }
    \dim_set:Nn \l__flowframtkutils_object_cy_dim
    {
      \l__flowframtkutils_object_miny_dim
      + 0.5 \l__flowframtkutils_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_flowframtkutils_group_level_int
  }

```

Encapsulate object.

```

    \exp_args:NV
    \__flowframtkutils_current_encap:nn
    \l__flowframtkutils_overlay_tl { #8 }
    \bool_if:NT \l_flowframtkutils_annotate_bool
    {

```

Add annotation (arrow, if applicable, and text).

```

    \tl_clear:N \l__flowframtkutils_annotation_tl
    \tl_set:Nn \l__flowframtkutils_propval_tl { \q_no_value }
    \__flowframtkutils_get_annotation_text:Nnn
    \l__flowframtkutils_propval_tl
    { #3 } { #4 }

```

If no text value, skip.

```

    \quark_if_no_value:NF
    \l__flowframtkutils_propval_tl
    {
      \int_if_zero:nTF
      { \l_flowframtkutils_annotate_location_int }
    }

```

Position set to auto so calculate suitable position id.

```

    \__flowframtkutils_auto_set_annotate_location:
  }
{

```

Specific position, so just store position id.

```

    \int_set_eq:NN
    \l_flowframtkutils_current_annotate_location_int
    \l_flowframtkutils_annotate_location_int
  }

```

Set the annotation content.

```

    \exp_args:NV
    \__flowframtkutils_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_flowframtkutils_current_annotate_location_int
        \c_flowframtkutils_annotate_location_northwest_int
    }
    {
        \dim_compare:nNnT
        { \l__flowframtkutils_object_maxy_dim }
        <
        { \l__flowframtkutils_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_flowframtkutils_current_annotate_location_int
        \c_flowframtkutils_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_flowframtkutils_current_annotate_location_int
        \c_flowframtkutils_annotate_location_east_int
    \dim_compare:nNnTF
    { \l__flowframtkutils_object_miny_dim }
    >
    { \l__flowframtkutils_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_flowframtkutils_current_annotate_location_int
        \c_flowframtkutils_annotate_location_northeast_int
    }
    {
        \dim_compare:nNnT
        { \l__flowframtkutils_object_maxy_dim }
        <
        { \l__flowframtkutils_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_flowframtkutils_current_annotate_location_int
        \c_flowframtkutils_annotate_location_southeast_int
    }
}

```



```

    }
}

```

Default arrow colour is black:

```

\tl_new:N \l_flowframtkutils_annotation_stroke_color_tl
\tl_set:Nn \l_flowframtkutils_annotation_stroke_color_tl
{
    \pgfsetstrokecolor { black }
}

```

Default start arrow head (at the bounding box edge):

```

\tl_new:N \l_flowframtkutils_annotation_start_arrow_tl
\tl_set:Nn \l_flowframtkutils_annotation_start_arrow_tl
{
    \pgfsetarrowsstart { < }
}

```

No end arrow head by default (the point where the text is placed).

```

\tl_new:N \l_flowframtkutils_annotation_end_arrow_tl
\cs_new:Nn \__flowframtkutils_set_annotate_content:n
{

```

The code to draw the arrow and text is stored in a token list variable.

```

\tl_set:Nn \l__flowframtkutils_annotation_tl
{
    \begin { pgfscope }
    \l_flowframtkutils_annotation_stroke_color_tl
    \l_flowframtkutils_annotation_start_arrow_tl
    \l_flowframtkutils_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_flowframtkutils_current_annotate_location_int }
{
    { \c_flowframtkutils_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_flowframtkutils_x_dim
    \l_flowframtkutils_object_cx_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_cx_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 ,
    at = {
      \exp_not:N \pgfpoint
      {
        \dim_use:N \l__flowframtkutils_x_dim
      }
      {
        \dim_use:N \l__flowframtkutils_y_dim
      }
    } ]
  { #1 }
}
}
{ \c_flowframtkutils_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__flowframtkutils_x_dim
{
  \l__flowframtkutils_object_maxx_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_maxx_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
  [
    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_offset_y_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_cx_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,
    at = {
      \exp_not:N \pgfpoint
      {
        \dim_use:N \l__flowframtkutils_x_dim
      }
      {
        \dim_use:N \l__flowframtkutils_y_dim
      }
    }
  ]
  { #1 }
}
}
{ \c_flowframtkutils_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__flowframtkutils_x_dim
{
  \l__flowframtkutils_object_minx_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_minx_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 , right,
    at = {
      \exp_not:N \pgfpoint
      {
        \dim_use:N \l__flowframtkutils_x_dim
      }
      {
        \dim_use:N \l__flowframtkutils_y_dim
      }
    }
  ]
  { #1 }
}
}
{ \c_flowframtkutils_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__flowframtkutils_x_dim
{
  \l__flowframtkutils_object_minx_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_minx_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
  [
    right,
    at = {
      \exp_not:N \pgfpoint
      {
        \dim_use:N \l__flowframtkutils_x_dim
      }
      {
        \dim_use:N \l__flowframtkutils_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 }
}
}

\tl_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:Nx \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 }
    }
  }
}
```

`\framtkSetDynamicOddEvenHead` 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:
  \@twosidettrue
}

```

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:
  \@twosidettrue
}

```

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
  \showmarginfalse
  \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*{\@ffbackground}[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{\@ffbackground}[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@ch@pter
}
}
\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{\thedisplayedframe}%
\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\sdfparindent
\newlength\flowframesep
\flowframesep=\fboxsep
\newlength\flowframerule
\flowframerule=\fboxrule
\newcommand*{\flowframeshowlayout}{%
\finishthispage
}%

```

```

\@ffdraft\mbox{}\finishthispage\clearpage
}%
}
\newif\ifusedframebreak
\newcommand{\framebreak}[1][4]{%
\global\usedframebreaktrue
{%
\parfillskip=Opt\pagebreak[#1]\parskip=Opt\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}%
\ifnotthiscol
\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}%
        \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
      }%
    }%
  }%
}

```

```

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

```

```

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

```

```

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

```

```

        {Key 'hidethis' not available for flow frames}{}%
    }%
}
\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*{@@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}%
}
\newcommand*{\ffswapoddeven}{%
    \@ifstar\@sflowframeswapcoords\@flowframeswapcoords
}
\newcommand*{@sflowframeswapcoords}[1]{%
    \@for\@ff@id:=#1\do
    {%

```

```

        \@flowframeid{\@ff@id}%
        \@@flowframeswapcoords{\ff@id}%
    }%
}
\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
                }%
            }%
        }%
    }%
}
\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\csname @staticframe@\romannumeral\c@maxstatic\endcsname
\expandafter
\newlength\csname @sf@\romannumeral\c@maxstatic @posx\endcsname
\expandafter
\newlength\csname @sf@\romannumeral\c@maxstatic @posy\endcsname
\expandafter\setlength
\csname @sf@\romannumeral\c@maxstatic @posx\endcsname{#4}%
\expandafter\setlength
\csname @sf@\romannumeral\c@maxstatic @posy\endcsname{#5}%
\expandafter\newlength
\csname @sf@\romannumeral\c@maxstatic @evenx\endcsname
\expandafter\newlength

```

```

\csname @sf@romannumeral@c@maxstatic @eveny\endcsname
\expandafter\setlength
\csname @sf@romannumeral@c@maxstatic @evenx\endcsname{#4}%
\expandafter\setlength
\csname @sf@romannumeral@c@maxstatic @eveny\endcsname{#5}%
{\@ff@tmp@x=#2\relax
\@ff@tmp@y=#3\relax
\expandafter
\edef\csname @sf@dim@\romannumeral@c@maxstatic\endcsname{%
[c][\the\@ff@tmp@y][c]{\the\@ff@tmp@x}}}%
\expandafter
\def\csname @sf@col@\romannumeral@c@maxstatic\endcsname{%
\flowframecol}%
\expandafter
\def\csname @sf@txtcol@\romannumeral@c@maxstatic\endcsname{%
\flowframetextcol}%
\expandafter
\def\csname @sf@backcol@\romannumeral@c@maxstatic\endcsname{%
{none}}}%
\expandafter
\edef\csname @sf@pages@\romannumeral@c@maxstatic\endcsname{#1}%
\expandafter
\gdef\csname @sf@xpages@\romannumeral@c@maxflow\endcsname{}%
\expandafter
\gdef\csname @sf@offset@\romannumeral@c@maxstatic\endcsname{%
compute}%
\expandafter
\gdef\csname @sf@angle@\romannumeral@c@maxstatic\endcsname{0}%
\expandafter
\gdef\csname @sf@shape@\romannumeral@c@maxstatic\endcsname{\relax}%
\expandafter
\def\csname @sf@frametype@\romannumeral@c@maxstatic\endcsname{%
fbox}%
\newboolean{@sf@clear@\romannumeral@c@maxstatic}%
\setboolean{@sf@clear@\romannumeral@c@maxstatic}{false}
\newboolean{@sf@hide@\romannumeral@c@maxstatic}%
\setboolean{@sf@hide@\romannumeral@c@maxstatic}{false}%
\newboolean{@sf@hidethis@\romannumeral@c@maxstatic}%
\setboolean{@sf@hidethis@\romannumeral@c@maxstatic}{false}%
\ifnextchar[{\@s@tstaticframeid{c@maxstatic}}%
{\@s@tstaticframeid{c@maxstatic}[\numberc@maxstatic}}%
}
\def\@s@tstaticframeid#1[#2]{%
\edef\ff@label{#2}%
\@sf@checkuniqueidl{#1}{\ff@label}%
\expandafter
\edef\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
\edef\csname @sf@dim@romannumeral#1\endcsname{%
  [c][\ff@height][\ff@valign]{\ff@width}}%
\ifdefempty{\ff@frametype}{}%
{%
  \expandafter
  \edef\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}%
\edef\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\noexpand\FLForgpar=\noexpand\par
    }%
\or
\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
\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@tfftextcol\noindent
    \@sf@mpg
    \setlength\parindent\@sfparindent
}
\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
\edef\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
\edef\csname @sf@xpages@\romannumeral#1\endcsname{#2}%
\flf@message{Setting exclusion for static frame
\number#1\space\space to "#2"}%
}
\newcommand*{\staticaddexclusion}[2]{%
\ifcsempy{@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"}%
}
\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\@staticframeswapcoords\@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
                    \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
                }%
            }%
        }%
    }%
}
\newcommand*{\newdynamicframe}{%
    \@n@wdynamicframe
}
\newcommand*{\@n@wdynamicframe}{%
    \global\advance\c@maxdynamic by 1\relax

```

```

\newboolean{dynamicframe\romannumeral\c@maxdynamic}
\@ifstar\@snewdynamicframe\@newdynamicframe
}
\newcommand*{\@snewdynamicframe}{%
\setboolean{dynamicframe\romannumeral\c@maxdynamic}{true}%
\@@newdynamicframe
}
\newcommand*{\@newdynamicframe}{%
\setboolean{dynamicframe\romannumeral\c@maxdynamic}{false}%
\@@newdynamicframe
}
\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}%
\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
    \@@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\@dynamictok
\def\xdynamiccontents#1{%
  \def\@flf@idn{#1}%
  \@dynamictok{}\@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\@dynamictok\the\flf@ta}%
    \@dynamictok\expandafter{\@flf@tmp}%
  \else
    \@dynamictok\expandafter{\the\@dynamictok#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\@tmpa{\the\@dynamictok{\@ff@text}}%
    \@dynamictok\expandafter{\@tmpa}%
  \fi
}

```

```

\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}%
\expandafter
\fi
\expandafter\end\expandafter{\@flf@}%
}
\newenvironment{dynamiccontents*}[1]{%
\def\@flf@{dynamiccontents*}%
\@dynamicframeid{#1}%
\xdynamiccontents{\ff@id}}{%
\enddynamiccontents
}
\newcommand{\setdynamiccontents}{%
\@ifstar\@sssetdynamiccontents\@setdynamiccontents
}
\newcommand{\@sssetdynamiccontents}[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][ ]{%
  \ffcontinuedtextlayout{#1}%
  \end{staticcontents*}%
  \begin{staticcontents*}{#2}\par\noindent\ignorespaces

```

```

}
\newcommand*{\@staticcontinueonframe}[2][{}]{%
  \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*{\computeffareaarea}{%
  \@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{\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}{\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
\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
}
\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
  \@@@getstaticpos
  \csname @sf@dim@\romannumeral#2\endcsname
  \setlength{\ffareawidth}{\@@@tmp@x}%
  \setlength{\ffareaheight}{\@@@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
  \@@@getstaticpos
  \csname @df@dim@\romannumeral#2\endcsname
  \setlength{\ffareawidth}{\@@@tmp@x}%
  \setlength{\ffareaheight}{\@@@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*\@getstaticbounds[1]{%
  \@staticframeid{#1}\getstaticbounds{\ff@id}%
}
\newcommand*\@getstaticbounds[1]{\@ff@getdim{2}{#1}}
\newcommand*\getstaticevenbounds{%
  \ifstar\sgetstaticevenbounds\getstaticevenbounds
}
\newcommand*\@getstaticevenbounds[1]{%
  \@staticframeid{#1}\getstaticevenbounds{\ff@id}%
}
\newcommand*\@getstaticevenbounds[1]{\@ff@getevendim{2}{#1}}
\newcommand*\getflowbounds{%
  \ifstar\sgetflowbounds\getflowbounds
}
\newcommand*\@getflowbounds[1]{%
  \@flowframeid{#1}\getflowbounds{\ff@id}%
}
\newcommand*\@getflowbounds[1]{\@ff@getdim{1}{#1}}
\newcommand*\getflowevenbounds{%
  \ifstar\sgetflowevenbounds\getflowevenbounds
}
\newcommand*\@getflowevenbounds[1]{%
  \@flowframeid{#1}\getflowevenbounds{\ff@id}%
}
\newcommand*\@getflowevenbounds[1]{\@ff@getevendim{1}{#1}}
\newcommand*\getdynamicbounds{%
  \ifstar\sgetdynamicbounds\getdynamicbounds
}
\newcommand*\@getdynamicbounds[1]{%
  \@dynamicframeid{#1}\getdynamicbounds{\ff@id}%
}
\newcommand*\@getdynamicbounds[1]{\@ff@getdim{3}{#1}}
\newcommand*\getdynamicevenbounds{%
  \ifstar\sgetdynamicevenbounds\getdynamicevenbounds
}
\newcommand*\@getdynamicevenbounds[1]{%
  \@dynamicframeid{#1}\getdynamicevenbounds{\ff@id}%
}
\newcommand*\@getdynamicevenbounds[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<\ffareax
    \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*{\@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}%
\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
}

```

```

\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
  \FLFrighttrue
\else
  \FLFrightfalse
\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
    \FLFrightrue
\else
    \FLFrightfalse
\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
\FLFabovetrue
\else
\FLFabovefalse
\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
\FLFabovetrue
\else
\FLFabovefalse
\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
  \FLFbelowtrue
\else
  \FLFbelowfalse
\fi
}
\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

```

```

        \FLbelowfalse
    \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#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<\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#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
  \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
  \FLFrighttrue
\else
  \FLFrightfalse

```



```

\fi
}
\newcommand*{\@evencheckifframeright}[4]{%
  \ifundefined{@get#1evenbounds}%
  {%
    \PackageError{flowfram}%
    {Unknown frame type ‘#1’}%
    {%
      Frame types may only be one of: static, dynamic or
      flow%
    }%
  }%
}%
{}%
\csname @get#1evenbounds\endcsname{#2}%
\edef\@ff@check{\the\ffareax}%
\ifundefined{@get#3evenbounds}%
{%
  \PackageError{flowfram}%
  {Unknown frame type ‘#3’}%
  {%
    Frame types may only be one of: static, dynamic or
    flow%
  }%
}%
{}%
\csname @get#3evenbounds\endcsname{#4}%
\advance\ffareax by \ffareawidth\relax
\expandafter\ifdim\@ff@check>\ffareax
  \FLFrightright
\else
  \FLFrightrightfalse
\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
    \ifFLFrightright
    \FFabove
  \fi
\fi
\else
  \ifFLFbelow
    \ifFLFleft
      \FFbelowleft
    \else
      \ifFLFrightright
      \FFbelowright
    \else
      \FFbelow
    \fi
  \fi
\else
  \ifFLFleft
    \FFleft
  \else
    \ifFLFrightright
    \FFright
  \else
    \FFoverlap
  \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
      \ifFLFrightright
      \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@mestrue
\edef\ff@txtcol{%
  \csname @ff@txtcol@\romannumeral\c@thisframe\endcsname}%
\@s@tff@txtcol
}
\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
  \fi
}

```

```

        \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>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
}
\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%
    }%
}
\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\vsizer\csname colheight\romannumeral#1\endcsname
    \global\@colht\vsizer
    \global\@colroom\@colht
    \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
    \@whiles\@ifcolmade \fi {\@opcol \@startcolumn }%
  \fi
}

```

```

\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
    \@makefcolumn\@deferlist
    \@whilesw \if@fcolmade \fi
  {%
    \@opcol
    \@makefcolumn\@deferlist
  }%
  \if@firstcolumn
    \xdef\@dbldeferlist{\@dbltoplist\@dbldeferlist}%
    \global\let\@dbltoplist\@empty
    \global\@colht\vsize
    \begingroup
      \@dblfloatplacement
      \@makefcolumn\@dbldeferlist
      \@whilesw \if@fcolmade \fi
    {%
      \@outputpage
      \@makefcolumn\@dbldeferlist
    }%
  \endgroup

```

```

\else
\ vbox{}%
\clearpage
\fi
\else
\setbox\@cclv\vbox{\box\@cclv\vfil}%
\@makecol\@opcol
\clearpage
\fi
}
\newcommand{\@dothehead}{%
\ vbox to \headheight
{%
\color@hbox\normalcolor\hbox to \textwidth{\@thehead}%
\color@endbox
}%
}
\newcommand{\@dotheft}{%
\color@hbox\normalcolor\hbox to \textwidth{\@theft}%
\color@endbox
}
\newcommand{\@dodynamicthehead}{%}
\newcommand{\@dodynamictheft}{%}
\def\@outputpage{%
\begingroup
\let\protect\noexpand
\@resetactivechars
\global\let\@ifnewlist\ifnewlist
\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{displayedframe}{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
    }%
    \s@tffttextcol
  }%
}
\renewcommand{\@makecol}{%
  \ifvoid\footins
    \setbox\@outputbox\box\@cclv
  \else
    \setbox\@outputbox\vbox
    {%
      \boxmaxdepth\@maxdepth\@tempdima\dp\@cclv
      \unvbox\@cclv
    }
  \fi
}

```

```

\vskip\skip\footins
\color@begingroup
\footnotecolor
\footnoterule
\unvbox\footins
\color@endgroup
}%
\fi
\edef\@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
}

```

```

\fi
}
\def\@@@ff@getrange#1-\relax\end{%
\def\@ff@numend{#1}%
}
\newcommand*{\@ff@output@adjustframes}{%
\newcommand*{\flowswitchonnext}{%
\@ifstar\@sflowswitchonnext\@flowswitchonnext
}
\newcommand{\@sflowswitchonnext}[1]{%
\@for\@ff@id:=#1\do{%
\@flowframeid{\@ff@id}%
\@ff@chckifthispg{\@ff@pages@countreg}{\@ff@id}%
\expandafter\toks@{\expandafter{\@ff@output@adjustframes}%
\if@notthiscol
\edef\@ff@output@adjustframes{%
\the\toks@
\noexpand\flowsetpagelist{\number\ff@id}{>\number\@ff@pages@countreg}%
}%
\else
\edef\@ff@output@adjustframes{%
\the\toks@
\noexpand\flowsetpagelist{\number\ff@id}%
{\number\@ff@pages@countreg,>\number\@ff@pages@countreg}%
}%
\fi
}%
}
\newcommand{\@flowswitchonnext}[1]{%
\@for\@ff@id:=#1\do{%
\@ff@chckifthispg{\@ff@pages@countreg}{\@ff@id}%
\expandafter\toks@{\expandafter{\@ff@output@adjustframes}%
\if@notthiscol
\edef\@ff@output@adjustframes{%
\the\toks@
\noexpand\flowsetpagelist{\number\@ff@id}{>\number\@ff@pages@countreg}%
}%
\else
\edef\@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}%
        \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*{\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
\expandafter\toks@\expandafter{\@ff@output@adjustframes}%
\xdef\@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}%
\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
\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
\expandafter\toks@\expandafter{\@ff@output@adjustframes}%
\xdef\@ff@output@adjustframes{%
  \the\toks@
  \noexpand\flowsetpagelist{\number\@ff@id}{\@ff@nextpages}%
}%
}%
}
\newcommand*{\flowswitchonnexonly}{%
  \ifstar\@sflowswitchonnexonly\@flowswitchonnexonly
}
\newcommand{\@sflowswitchonnexonly}[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}%
    \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{\@flowswitchonnexonly}[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
                \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}%
\edef\@ff@output@adjustframes{%
    \the\toks@
    \noexpand\flowsetpagelist{\number\ff@id}{\@ff@pages}%
}%
}%
}
\newcommand{\@flowswitchonnnextoddonly}[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
    }
}

```

```

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

```

```

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

```

```

\edef\@ff@output@adjustframes{%
  \the\toks@
  \noexpand\dynamicsetpagelist{\number\@ff@id}%
    {\number\@ff@pages@countreg,>\number\@ff@pages@countreg}%
}%
\fi
}%
}
\newcommand*\@dynamicswitchonnextodd{%
  \ifstar\@dynamicswitchonnextodd\@dynamicswitchonnextodd
}
\newcommand{\@dynamicswitchonnextodd}[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
    \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*{\dynamicsswitchoffnext}{%
\@ifstar\@dynamicsswitchoffnext\@dynamicsswitchoffnext
}
\newcommand{\@dynamicsswitchoffnext}[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\@dynamicsswitchoffnextodd\@dynamicsswitchoffnextodd
  }
  \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}%
      \@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
      \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}%
      \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\dynamicsetpagelist{\number\@ff@id}{\@ff@nextpages}%
      }%
    }%
  }
  \newcommand{\@dynamicsswitchoffnextodd}[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}%

```



```

\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}%}
\xdef\@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\@sdynamicswitchonnexoddonly\@dynamicswitchonnexoddonly
}
\newcommand{\@sdynamicswitchonnexoddonly}[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}%
\xdef\@ff@output@adjustframes{%
\the\toks@
\noexpand\dynamicsetpagelist{\number\ff@id}{\@ff@pages}%
}%
}%
}
\newcommand{\@dynamicswitchonnextoddonly}[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*{\dynamicsswitchoffnextonly}{%
\ifstar\@dynamicsswitchoffnextonly\@dynamicsswitchoffnextonly
}
\newcommand{\@dynamicsswitchoffnextonly}[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{\@dynamicsswitchoffnextonly}[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*{\dynamicsswitchoffnextoddoneonly}{%
\ifstar\@dynamicsswitchoffnextoddoneonly\@dynamicsswitchoffnextoddoneonly
}
\newcommand{\@dynamicsswitchoffnextoddoneonly}[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{\@dynamicswitchoffnextoddoneonly}[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}%
        \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
        \fi
    }
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
        \the\toks@
        \noexpand\staticsetpagelist{\number\@ff@id}{\@ff@nextpages}%
    }%
}%
}
\newcommand*{\staticswitchonnextonly}{%
    \ifstar\@staticswitchonnextonly\@staticswitchonnextonly
}
\newcommand{\@staticswitchonnextonly}[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
\edef\@ff@output@adjustframes{%
\the\toks@
\noexpand\staticsetpagelist{\number\ff@id}{\number\count@}%
}%
\else
\edef\@ff@output@adjustframes{%
\the\toks@
\noexpand\staticsetpagelist{\number\ff@id}%
{\number\@ff@pages@countreg,\number\count@}%
}%
\fi
}%
}
\newcommand{\@staticswitchonnexonly}[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
\edef\@ff@output@adjustframes{%
\the\toks@
\noexpand\staticsetpagelist{\number\@ff@id}{\number\count@}%
}%
\else
\edef\@ff@output@adjustframes{%
\the\toks@
\noexpand\staticsetpagelist{\number\@ff@id}%
{\number\@ff@pages@countreg,\number\count@}%
}%
\fi
}%
}
\newcommand*\@staticswitchonnextodonly{%
\ifstar\@sstaticswitchonnextodonly\@staticswitchonnextodonly
}
\newcommand{\@sstaticswitchonnextodonly}[1]{%
\@for\@ff@id:=#1\do{%
\@staticframeid{\@ff@id}%
\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{\@staticswitchonnnextodonly}[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@chkifthispg[\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\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*\@staticswitchoffnextoddone{\@staticswitchoffnextoddone}%
\ifstar\@staticswitchoffnextoddone\@staticswitchoffnextoddone
}
\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{%
    \@staticframeid{\@ff@id}%
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\staticaddexclusion{\number\@ff@id}{\number\count@}%
    }%
  }%
}%
}
\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@}%
    }%
  }%
}%
}
\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@chkifthispg{\c@curpg}{\@colN}%
  \if@notthiscol
  \repeat
  #1=\@colN\relax
}
\newcommand*{\@ff@chkifthispg}[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]{%
{%
\fbboxsep=\flowframesep
\fbboxrule=\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@tfftextcol #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".
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
}
\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
}
\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}%
                  {ovalbox}}{%
            }%
            {%
              \@ff@s@t@ovalboxoffset
            }%
            {%
              \@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
        {%
          \equal{\csname @sf@frametype@\romannumeral#1\endcsname}%
          {ovalbox}%
        }%
        {%
          \@ff@s@t@ovalboxoffset
        }%
        {%
          \ifthenelse
          {%
            \equal{\csname @sf@frametype@\romannumeral#1\endcsname}%
            {0valbox}%
          }%
          {%
            \@ff@s@t@0valboxoffset
          }%
          {%
            \@ff@s@t@defaultoffset
          }%
        }%
      }%
    }%
  }%
}

```

```

    }%
    {}%
  }%
  {%
    \setlength{\@ff@offset}%
    {\csname @sf@offset@\romannumeral#1\endcsname}%
  }%
}
\newcommand*{\@df@setoffset}[1]{%
  \ifthenelse
  {%
    \equal{\csname @df@offset@\romannumeral#1\endcsname}%
    {compute}%
  }%
  {%
    \setlength{\@ff@offset}{0pt}%
    \ifthenelse{\boolean{dynamicframe\romannumeral#1}}%
    {%
      \ifthenelse
      {%
        \equal{\csname @df@frametype@\romannumeral#1\endcsname}%
        {doublebox}%
      }%
      {%
        \@ff@s@t@doubleboxoffset
      }%
      {%
        \ifthenelse
        {%
          \equal{\csname @df@frametype@\romannumeral#1\endcsname}%
          {ovalbox}%
        }%
        {%
          \@ff@s@t@ovalboxoffset
        }%
        {%
          \ifthenelse
          {%
            \equal{\csname @df@frametype@\romannumeral#1\endcsname}%
            {ovalbox}%
          }%
          {%
            \@ff@s@t@ovalboxoffset
          }%
          {%
            \@ff@s@t@defaultoffset
          }%
        }%
      }%
    }%
  }%
}

```

```

    {}%
  }%
  {%
    \setlength{\@ff@offset}%
    {\csname @df@offset@romannumeral#1\endcsname}%
  }%
}
\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
}
\newcommand*{\@ff@drawmargins}{%
  \@colN=0\relax
  \whiledo{\@colN<\c@maxflow}%
  {%
    \advance\@colN by 1\relax
    \makebox[0pt][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@bbox{\@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
\fi
}
\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
}
\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*{\@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{}%
      }%
    }%
  }%
}
\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
      }%
    }%
  \or

```

```

    }}%
  }%
\fi
}
\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}%
        }%
      }%
    }%
  }
}
\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
\bgroup
\hbox

```

```

        \bgroup
        #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@doallstaticsbbox{%
    \@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@doalldynamicsbbox}{%
  \@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@tnxtcol{\@nxtcol}%
    \if@ff@nwpg
      \global\@firstcolumntrue
      \@setcolbox\c@thisframe
      \if@specialpage
        \global\@specialpagefalse
        \@nameuse{ps@\@specialstyle}\relax
      \fi
      \if@twoside
        \ifodd\count\z@
          \let\@thehead\@oddhead
          \let\@thefoot\@oddfoot
        \else
          \let\@thehead\@evenhead
          \let\@thefoot\@evenfoot
        \fi
      \else
        \let\@thehead\@oddhead
        \let\@thefoot\@oddfoot
      \fi
      \@begindvi
      \@dodynamicthehead\@dodynamicthefoot
      \vbadness=\@M
      \setbox\@outputbox\vbox{\hbox to \textwidth{\@ff@do@allframes}}%
      \@combinedblfloats
      \@outputpage
      \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 \@dblftop \@fpsep \@dblfpsep \@fpbot \@dblfpbot
}
\newenvironment{statictable}{\def\@capytype{table}}{}
\newenvironment{staticfigure}{\def\@capytype{figure}}{}
\newif\iffvadjust
\ffvadjusttrue
\renewcommand*{\onecolumn}{\@onecolumn}
\newcommand*{\@onecolumn}[1][all]{%
\@onecolumninarea[#1]{\textwidth}{\textheight}{Opt}{Opt}%
}
\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}{Opt}{Opt}%
}
\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{\onecolumnntop}
\newcommand*{\onecolumnnstop}[2][all]{%
\onecolumnntopinarea[#1]{static}{#2}{\textwidth}{\textheight}{Opt}{Opt}%
}
\newcommand*{\onecolumnDtop}[2][all]{%
\onecolumnntopinarea[#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*{\onecolumnntopinarea}[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{\onecolumnntopinarea}
\newcommand*{\onecolumnStopinarea}[6][all]{%
  \onecolumnntopinarea[#1]{static}{#2}{#3}{#4}{#5}{#6}%
}
\newcommand*{\onecolumnDtopinarea}[6][all]{%
  \onecolumnntopinarea[#1]{dynamic}{#2}{#3}{#4}{#5}{#6}%
}
\newcommand*{\twocolumnntop}[3][all]{%
  \twocolumnntopinarea[#1]{#2}{#3}{\textwidth}{\textheight}{Opt}{Opt}%
}
\@onlypreamble{\twocolumnntop}
\newcommand*{\twocolumnStop}[2][all]{%
  \@twocolumnntopinarea[#1]{static}{#2}{\textwidth}{\textheight}{Opt}{Opt}%
}
\newcommand*{\twocolumnDtop}[2][all]{%
  \twocolumnntop[#1]{dynamic}{#2}%
}
\newcommand*{\twocolumnntopinarea}{\@twocolumnntopinarea}
\newcommand*{\@twocolumnntopinarea}[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*{\Ncolumntop}[4][all]{%
  \Ncolumntopinarea[#1]{#2}{#3}{#4}{\textwidth}{\textheight}{0pt}{0pt}%
}
\@onlypreamble{\Ncolumntop}
\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
    \onecolumnmktopinarea[#1]{#2}{#4}{#5}{#6}{#7}{#8}%
  \or
    \twocolumnmktopinarea[#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{\Ncolumnmktopinarea}
\newcommand*{\@Ncolumnmktopinarea}[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}%
  \ifffvadjust
    \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]{%
\NcolumnTopinarea[#1]{static}{#2}{#3}{#4}{#5}{#6}{#7}%
}
\newcommand*\NcolumnDtopinarea[7][all]{%
\NcolumnTopinarea[#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
  \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}{0pt}{0pt}%
}

```



```

\@onlypreamble{\Ncolumnbottom}
\newcommand*{\NcolumnSbottom}[3][all]{%
  \Ncolumnbottom[#1]{static}{#2}{#3}%
}
\newcommand*{\NcolumnDbottom}[3][all]{%
  \Ncolumnbottom[#1]{dynamic}{#2}{#3}%
}
\newcommand*{\Ncolumnbottomminarea}[8][all]{%
  \ifnum#3>2\relax
    \@Ncolumnbottomminarea[#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
      \onecolumnbottomminarea[#1]{#2}{#4}{#5}{#6}{#7}{#8}%
    \or
      \twocolumnbottomminarea[#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{\Ncolumnbottomminarea}
\newcommand*{\@NcolumnSbottomminarea}[8][all]{%
  \setlength{\@ff@staticH}{#4}%
  \setlength{\columnheight}{#6}%
  \addtolength{\columnheight}{-\@ff@staticH}%
  \addtolength{\columnheight}{-\vcolumnsep}%
  \ifffvadjust
    \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}%
  \@nexthband{\ff@pages}{#2}{#3}{#4}{#5}%
  \@nexthband{\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}%
  \@nextthband{\ff@pages}{#2}{#3}{#4}{#5}%
  \@nextthband{\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}%
  \@nextthNband
}
\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*{\@nexthband}[5]{%
  \setlength{\@ff@staticH}{#3}%
  \ifthenelse{equal{#5}{}}{%
    %
    \newstaticframe[#1]{#2}{\@ff@staticH}{\@ff@tmp@x}{\@ff@tmp@y}%
  }%
  {%
    \newstaticframe[#1]{#2}{\@ff@staticH}{\@ff@tmp@x}{\@ff@tmp@y}[#5]%
  }%
  \expandafter\global\expandafter
    \setlength\csname @sf@romannumeral\c@maxstatic @even\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\ffcolumnseprule
\setlength{\ffcolumnseprule}{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}{\ffareax}%
    \setlength{\@ff@left@y}{\ffareay}%
    \setlength{\@ff@left@width}{\ffareaawidth}%
    \setlength{\@ff@left@height}{\ffareaaheight}%
    \@ff@getdim{#5}{#6}%
    \ifnum\@ff@left@x>\ffareax\relax
        \@ff@swaplen{\@ff@left@x}{\ffareax}%
        \@ff@swaplen{\@ff@left@y}{\ffareay}%
        \@ff@swaplen{\@ff@left@evenx}{\ffareaevenx}%
        \@ff@swaplen{\@ff@left@eveny}{\ffareaeveny}%
        \@ff@swaplen{\@ff@left@width}{\ffareaawidth}%
        \@ff@swaplen{\@ff@left@height}{\ffareaaheight}%
    \fi
    \setlength{\@ff@tmp@x}{\@ff@left@x}
    \addtolength{\@ff@tmp@x}{\@ff@left@width}%

```

```

\setlength{\@ff@staticW}{\ffareax}%
\addtolength{\@ff@staticW}{-\@ff@tmp@x}%
\setlength{\@ff@staticH}{\@ff@left@y}%
\addtolength{\@ff@staticH}{\@ff@left@height}%
\setlength{\@ff@tmp@y}{\ffareay}%
\addtolength{\@ff@tmp@y}{\ffareaheight}%
\ifnum\@ff@tmp@y>\@ff@staticH
  \setlength{\@ff@staticH}{\@ff@tmp@y}%
\fi
\ifnum\@ff@left@y<\ffareay\relax
  \setlength{\@ff@tmp@y}{\@ff@left@y}%
\else
  \setlength{\@ff@tmp@y}{\ffareay}%
\fi
\addtolength{\@ff@staticH}{-\@ff@tmp@y}%
\newstaticframe{\@ff@staticW}{\@ff@staticH}{%
  {\@ff@tmp@x}{\@ff@tmp@y}%
\addtolength{\@ff@staticH}{#1}%
\addtolength{\@ff@staticH}{#2}%
\setstaticcontents{\c@maxstatic}{%
\ffruledeclarations
\ffvrule{#2}{\ffcolumnseprule}{\@ff@staticH}}%
\ifcase#3\relax
  \or \edef\@ff@pages{\csname @ff@pages@romannumeral#4\endcsname}%
  \or \edef\@ff@pages{\csname @sf@pages@romannumeral#4\endcsname}%
  \or \edef\@ff@pages{\csname @df@pages@romannumeral#4\endcsname}%
\fi
\setstaticframe{\c@maxstatic}{pages=\@ff@pages}%
\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][0pt]{%
  \@ifnextchar[{\@sinsertvrule[#1]}{\@sinsertvrule[#1][0pt]}%
}
\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*{\insertthruler}{\ifstar\@sinsertthruler\insertthruler}
\newcommand*{\insertthruler}[1][Opt]{%

```

```

\@ifnextchar[{\@insertthrule[#1]}{\@insertthrule[#1][Opt]}}%
}
\def\@insertthrule[#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}%
}
\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}%
  {%
    \ffruleddeclarations
    \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][Opt]{%
\@ifnextchar[{\@sinserthrule[#1]}{\@sinserthrule[#1][Opt]}%
}
\def\@sinserthrule[#1][#2]#3#4#5#6{%
\ifthenelse{\equal{#3}{flow}}{%
{%
\def\@ff@type@i{1}%
\@flowframeid{#4}%
\@ff@tmpN=\ff@id
}%
{%
\ifthenelse{\equal{#3}{static}}{%
{%
\def\@ff@type@i{2}%
\@staticframeid{#4}%
\@ff@tmpN=\ff@id
}%
{%
\ifthenelse{\equal{#3}{dynamic}}{%
{%
\def\@ff@type@i{3}%
\@dynamicframeid{#4}%
\@ff@tmpN=\ff@id
}%
{%
\PackageError{flowfram}%
{Unknown frame type '3'}%
{%
Available frame types are: 'flow', 'static'
or 'dynamic'%
}%
}%
}%
}%
\ifthenelse{\equal{#5}{flow}}{%
{%
\def\@ff@type@ii{1}%
\@flowframeid{#6}%
}%
{%
\ifthenelse{\equal{#5}{static}}{%
{%
\def\@ff@type@ii{2}%
\@staticframeid{#6}%
}
}
}
}

```

```

}%
{%
  \ifthenelse{\equal{#5}{dynamic}}%
  {%
    \def\@ff@type@ii{3}%
    \@dynamicframeid{#6}%
  }%
  {%
    \PackageError{flowfram}%
    {Unknown frame type '5'}%
    {%
      Available frame types are: 'flow', 'static'
      or 'dynamic'%
    }%
  }%
}%
}%
\@@insert@hrule{#1}{#2}{\@ff@type@i}{\@ff@tmpN}%
{\@ff@type@ii}{\ff@id}%
}
\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*{\defaultthumtabtype}{section}%
}%
{%
  \newcommand*{\defaultthumtabtype}{chapter}%
}
\newcommand*{\@ttb@type}{\defaultthumtabtype}
\newcommand*{\makethumbtabs}[2][Opt]{%
  \@ifnextchar[%
    {\@makethumbtabs[#1]{#2}}%
    {%
      \@makethumbtabs[#1]{#2}[\defaultthumtabtype]%
    }%
}
\def\@makethumbtabs[#1]#2[#3]{%
  \@ifundefined{#3}%
  {%
    \PackageError{flowfram}%
    {%
      Unknown section type '#3'%
    }%
    {}%
  }%
  {%
    \renewcommand{\@ttb@type}{#3}%
    \ifthenelse{\equal{#3}{chapter}}%
    {%
      \@makethumbchapter
    }%
    {%
      \ifthenelse{\equal{#3}{part}}%
      {\@makethumbpart}%
      {%
        \@makethumbsection{#3}%
      }%
    }%
  }%
  \@starttoc{ttb}%
  \@dothumbtabs{#1}{#2}%
}

```



```

\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{\thumtbat}[4]{%
  \stepcounter{maxthumtbat}%
  \expandafter
    \gdef\csname thumtbat@pages@\romannumeral\c@maxthumtbat\endcsname{#1}%
  \expandafter
    \gdef\csname thumtbat@num@\romannumeral\c@maxthumtbat\endcsname{#2}%
  \expandafter
    \gdef\csname thumtbat@title@\romannumeral\c@maxthumtbat\endcsname{#3}%
  \expandafter
    \gdef\csname thumtbat@link@\romannumeral\c@maxthumtbat\endcsname{#4}%
}
\newcommand*{@dothumtbat}[2]{%
  \@colN=0\relax
  \whiledo{\@colN<\c@maxthumtbat}%
  {%
    \advance\@colN by 1\relax
    \edef\ff@pages{%
      \csname thumtbat@pages@\romannumeral\@colN\endcsname}%
    \ifnum\@colN=\c@maxthumtbat
      \expandafter
        \xdef\csname thumtbat@pages@\romannumeral\@colN\endcsname{%
          \ff@pages,>\ff@pages}%
    \else
      \advance\@colN by 1\relax
      \edef\ff@endpage{%
        \csname thumtbat@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 thumtbat@pages@\romannumeral\@colN\endcsname{%
            \ff@pages-\number\@ff@tmpN}%
      \fi
    \fi
  }%
  \@dothumtbat{#1}{#2}%
}
\newlength{\thumtbatwidth}
\setlength{\thumtbatwidth}{1cm}
\@ifundefined{hyperlink}%
{%
  \newcommand{\thumtbatindexformat}[3]{%
    \thumtbatformat{#2}{#3}%
  }%
}%
{%
  \newcommand{\thumtbatindexformat}[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
}
\def\@flf@subsp#1 #2{%
\expandafter\flf@ta\expandafter{\@flf@subsptext}%
\flf@tb{#1}%
\edef\@flf@subsptext{\the\flf@ta\the\flf@tb}%
\def\@flf@tmp{#2}%
\ifx\@flf@tmp\@nnil
\let\@flf@donextsubsp=\gobble
\else
\expandafter\flf@ta\expandafter{\@flf@subsptext}%
\edef\@flf@subsptext{\the\flf@ta\noexpand\space}%
\let\@flf@donextsubsp=\@flf@subsp
\fi
\@flf@donextsubsp{#2}%
}
\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{%
            \noexpand\thumbtabformat
              {%
                \csname thumbtab@num@\romannumeral\@ff@tmpN\endcsname\
                \csname thumbtab@title@\romannumeral\@ff@tmpN\endcsname
              }%
              {#2}%
            }%
          }%
        }%
      }%
    }%
  }%
  \if@ttb@num
    \expandafter
      \xdef\csname @dynamicframe@\romannumeral\c@maxdynamic\endcsname{%
        \noexpand\thumbtabformat
          {%
            \csname thumbtab@title@\romannumeral\@ff@tmpN\endcsname
          }%
          {#2}%
        }%
      }%
    }%
  }%
\fi
\if@ttb@num
  \expandafter

```

```

\edef\csname @dynamicframe@\romannumeral\c@maxdynamic\endcsname{%
  \noexpand\thumbtabformat
  {%
    \csname thumbtab@num@\romannumeral\@ff@tmpN\endcsname
  }%
  {#2}%
}%
\fi
}%
\expandafter
\edef\csname @df@backcol@\romannumeral\c@maxdynamic\endcsname
{[gray]{\@ff@greyscale}}
\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
\ifthenelse{\boolean{@ttb@title}\and\boolean{@ttb@num}}{%
  {%
    \expandafter
    \edef\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
  \edef\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
  \edef\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
\edef\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\edef\csname @df@pages@\romannumeral\ff@id\endcsname
{none}%
\advance\ff@id by 1\relax
\expandafter\edef\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{\@setthumbtab}[2]{%
  \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}%
    }%
  }
}
\newcommand{\setthumbtabindex}[2]{%
  \ifthenelse{\equal{#1}{all}}%

```

```

{%
  \ff@tmpN=0\relax
  \whiledo{\ff@tmpN<\c@maxthumtbs}%
  {%
    \advance\ff@tmpN by 1\relax
    \@setthumtbsindex{\ff@tmpN}{#2}%
  }%
}%
{%
  \for\@ttb@id:=#1\do{\@setthumtbsindex{\@ttb@id}{#2}}%
}%
}
\newcommand{\@setthumtbsindex}[2]{%
  \ifthenelse{(\c@maxthumtbs<#1)\ or \(#1<1\)}%
  {%
    \PackageWarning{flowfram}%
    {%
      Can't find thumbtab number '\number#1',
      ttb file may not be up-to-date%
    }%
  }%
  {%
    \@dynamicframeid{thumtbsindex\number#1}%
    \@setdynamicframe{\ff@id}{#2}%
  }%
}
\newcommand*{\tocandthumtbsindex}{%
  \aligntoctrue
  \tableofcontents
  \thumtbsindex
  \aligntofalse
}
\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@minitocctype}%
{%
\@ttb@minitoclevel=6\relax
}%
{%
\expandafter\@ttb@minitoclevel\expandafter
=\csname @ttb@\@ttb@minitocctype @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@contentsline
    \@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
      \@contentslinetrue
    \else
      \@contentslinefalse
    \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
  \@storetoctfalse
  \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@maxtocunits}%
  {%
    \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@maxtocunits}%
  {%
    \advance\@ff@tmpN by 1\relax
    \ifnum\@ff@tmpN>\c@maxthumtbs
      \csname @toc@\romannumeral\@ff@tmpN\endcsname
    \else
      \@dynamicframeid{thumbtabindex\@ff@tmpN}%
      \expandafter\expandafter\expandafter
        \@ff@getstaticpos\csname @df@dim@\romannumeral\@ff@tmpN\endcsname
      \vbox to \@ff@tmpNpt
      {%
        \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###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@tffcol}{}
\newcommand*{\@s@tffttextcol}{}%
```

`\@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@rotatefalse}
```

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

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

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


```

\if@ttb@num
    \newif\if@ttb@num
    \@ttb@numfalse
\if@ttb@title
    \newif\if@ttb@title
    \@ttb@titletrue

thumbtabs The thumbtabs option replaces the ttbtitle, ttbnotitle, ttbnum and ttbnonum op-
tions.
    \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@absolute page`.

```
\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@absolute page`):

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

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

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

absolute page

```
\newcounter{absolute page}
```

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

```

End of test if \chapter defined:

```
}
```

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

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

thisframe Next define a counter to keep track of the IDN of the current flow frame.

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

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

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

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

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

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

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

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

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

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

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

Define some temporary variables

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

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

```
\newlength\sdfparindent
```

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
  {%
    \@ffdraft\mbox{}\finishthispage\clearpage
  }%
}
```

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

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

`\finishthispage` The commands `\newpage` and `\pagebreak` can be used to move on to the next flow frame, but to finish the entire page, use `\finishthispage`. This is (loosely)

based on the code for `\clearpage`. (`\dbltopnum` not required as we can't have column-spanning floats.)

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

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

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

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

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

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

```
\@twocolumnfalse
```

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

```
\@mparswitchfalse
```

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

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

`\globalnormalmargin`

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

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

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

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

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



```

    {\globalreversemargin}%
    {\globalnormalmargin}%
}

```

`\newflowframe` Create a new flow frame. Syntax:

```
\newflowframe[<pages>]{<width>}{<height>}{<x>}{<y>}[<label>]
```

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

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

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

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

`\@n@wflowframe`

```

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

```

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

```

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

```

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

```

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

```

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

```

\newcommand{\@@newflowframe}[5][all]{%
  \expandafter\global\expandafter
  \newbox\csname column\romannumeral\c@maxflow\endcsname
  \expandafter\global\expandafter

```

```

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

```

Page exclusion list:

```

\expandafter
\gdef\csname @ff@xpages@\romannumeral\c@maxflow\endcsname{}%
\expandafter
\gdef\csname @ff@offset@\romannumeral\c@maxflow\endcsname{compute}
\expandafter
\gdef\csname @ff@angle@\romannumeral\c@maxflow\endcsname{0}%
\expandafter
\gdef\csname @ff@margin@\romannumeral\c@maxflow\endcsname{right}
\ifnum\c@thisframe=0\relax
\ifthenelse{\equal{#1}{all}}\TE@or\equal{#1}{odd}}%
{%
\c@thisframe=\c@maxflow
\global\setlength{\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{\hsize}{#2}%
\global\usedframebreaktrue
\fi
}%
}%
}%
\fi
\@ifnextchar[%
{\@s@tflowframeid{\c@maxflow}}%
{%
\@s@tflowframeid{\c@maxflow}[\number\c@maxflow]%
}%
}

```

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

```

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

```

\ff@checkuniqueidl Check 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{<idl>} 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{f\flowframe}{#2}%
    \ifdefempty{\ff@frame}{}%
    {%
        \setboolean{columnframe\romannumeral#1}{\ff@frame}%
    }
}

```

```

}%
\ifdefempty{\ff@width}{}%
{%
  \expandafter
  \setlength\csname colwidth\romannumeral#1\endcsname
  {\ff@width}%
}%
\ifdefempty{\ff@height}{}%
{%
  \expandafter
  \setlength\csname colheight\romannumeral#1\endcsname
  {\ff@height}%
}%
\ifdefempty{\ff@x}{}%
{%
  \expandafter
  \setlength\csname col@\romannumeral#1@posx\endcsname
  {\ff@x}%
  \expandafter
  \setlength\csname col@\romannumeral#1@evenx\endcsname
  {\ff@x}%
}
\ifdefempty{\ff@y}{}%
{%
  \expandafter
  \setlength\csname col@\romannumeral#1@posy\endcsname
  {\ff@y}%
  \expandafter
  \setlength\csname col@\romannumeral#1@eveny\endcsname
  {\ff@y}%
}%
\ifdefempty{\ff@evenx}{}%
{%
  \expandafter
  \setlength\csname col@\romannumeral#1@evenx\endcsname
  {\ff@evenx}%
}%
\ifdefempty{\ff@eveny}{}%
{%
  \expandafter
  \setlength\csname col@\romannumeral#1@eveny\endcsname
  {\ff@eveny}%
}%
\ifdefempty{\ff@oddx}{}%
{%
  \expandafter
  \setlength\csname col@\romannumeral#1@posx\endcsname
  {\ff@oddx}%
}%
\ifdefempty{\ff@oddy}{}%

```

```

{%
  \expandafter
    \setlength\csname col@\romannumeral#1@posy\endcsname
      {\ff@oddy}%
}%
\ifdefempty{\ff@label}{}%
{%
  \@s@tflowframeid{#1}[\ff@label]%
}%
\ifdefempty{\ff@frametype}{}%
{%
  \expandafter
    \edef\csname @ff@frametype@\romannumeral#1\endcsname{%
      \ff@frametype}%
}%
\ifdefempty{\ff@col}{}%
{%
  \expandafter\@setframecol\ff@col\end{#1}{col}{ff}%
}%
\ifdefempty{\ff@txtcol}{}%
{%
  \expandafter\@setframecol\ff@txtcol\end{#1}{txtcol}{ff}%
}%
\ifdefempty{\ff@backcol}{}%
{%
  \expandafter\@setframecol\ff@backcol\end{#1}{backcol}{ff}%
}%
\ifdefempty{\ff@margin}{}%
{%
  \expandafter
    \xdef\csname @ff@margin@\romannumeral#1\endcsname{%
      \ff@margin}%
}%
\ifdefempty{\ff@pages}{}%
{%
  \flowsetpagelist{#1}{\ff@pages}%
}%
\ifundef{\ff@xpages}{}%
{%
  \flowsetexclusion{#1}{\ff@xpages}%
}%
\ifdefempty{\ff@offset}{}%
{%
  \expandafter
    \xdef\csname @ff@offset@\romannumeral#1\endcsname{%
      \ff@offset}%
}%
\ifdefempty{\ff@angle}{}%
{%
  \expandafter

```

```

\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]{%
  \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"}%
}
```

`\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} \TEor \equal{#1}{even}}{%
    {%
      \ifthenelse{\equal{#1}{odd}}{\@colN=1}{\@colN=2}%
      \whiledo{\@colN<\c@maxflow\TEor \@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 \TEor \@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 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.

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

`\@n@wstaticframe`

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

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

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

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

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

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

```
\newcommand*{\@@newstaticframe}[5][all]{%
  \expandafter
    \newbox\csname @staticframe@\romannumeral\c@maxstatic\endcsname
  \expandafter
    \newlength\csname @sf@\romannumeral\c@maxstatic @posx\endcsname
  \expandafter
    \newlength\csname @sf@\romannumeral\c@maxstatic @posy\endcsname
  \expandafter\setlength
    \csname @sf@\romannumeral\c@maxstatic @posx\endcsname{#4}%
  \expandafter\setlength
    \csname @sf@\romannumeral\c@maxstatic @posy\endcsname{#5}%
  \expandafter\newlength
    \csname @sf@\romannumeral\c@maxstatic @evenx\endcsname
  \expandafter\newlength
    \csname @sf@\romannumeral\c@maxstatic @eveny\endcsname
  \expandafter\setlength
    \csname @sf@\romannumeral\c@maxstatic @evenx\endcsname{#4}%
  \expandafter\setlength
    \csname @sf@\romannumeral\c@maxstatic @eveny\endcsname{#5}%
  {\@ff@tmp@x=#2\relax
  \@ff@tmp@y=#3\relax
  \expandafter
    \xdef\csname @sf@dim@\romannumeral\c@maxstatic\endcsname{%
      [c][\the\@ff@tmp@y][c]{\the\@ff@tmp@x}}}%
  \expandafter
    \def\csname @sf@col@\romannumeral\c@maxstatic\endcsname{%
      \flowframecol}%
  \expandafter
    \def\csname @sf@txtcol@\romannumeral\c@maxstatic\endcsname{%
```

```

\flowframetextcol}%
\expandafter
\def\csname @sf@backcol@\romannumeral\c@maxstatic\endcsname{%
{none}}}%
\expandafter
\xdef\csname @sf@pages@\romannumeral\c@maxstatic\endcsname{#1}%
Page exclusion list:
\expandafter
\gdef\csname @sf@xpages@\romannumeral\c@maxflow\endcsname{}%
\expandafter
\gdef\csname @sf@offset@\romannumeral\c@maxstatic\endcsname{%
compute}%
\expandafter
\gdef\csname @sf@angle@\romannumeral\c@maxstatic\endcsname{0}%
\expandafter
\gdef\csname @sf@shape@\romannumeral\c@maxstatic\endcsname{\relax}%
\expandafter
\def\csname @sf@frametype@\romannumeral\c@maxstatic\endcsname{%
fbox}%
\newboolean{@sf@clear@\romannumeral\c@maxstatic}%
\setboolean{@sf@clear@\romannumeral\c@maxstatic}{false}

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

```

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

```

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

```

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

```

\newcommand*{\@sf@checkuniqueidl}[2]{%
\@colN=0\relax
\whiledo{\@colN<\c@maxstatic}%
{%
\advance\@colN by 1\relax
\ifnum\@colN=#1\relax
\else
\ifthenelse
{%
\equal{#2}{\csname @sf@id@\romannumeral\@colN\endcsname}%
}
}
}

```

```

}%
{%
  \PackageError{flowfram}%
  {Static frame IDL '#2' already defined}%
  {%
    You can't assign this label, as it is already defined
    for static frame \number\@colN
  }%
}%
{}%
\fi
}%
}

```

`\getstaticlabel` `\getstaticlabel{⟨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} \TEor \equal{#1}{even}}{%
      {%
        \ifthenelse{\equal{#1}{odd}}{\@colN=1}{\@colN=2}%
        \whiledo{\@colN<\c@maxstatic\TEor \@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
\edef\csname @sf@dim@\romannumeral#1\endcsname{%
[c][\ff@height][\ff@valign]{\ff@width}}%
\ifdefempty{\ff@frametype}{}%
{%
\expandafter
\edef\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\\[0pt]\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}{\langle idn \rangle}`.

```

\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*}{\langle label \rangle}`.

```

\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\@ff@parshape\@ff@parshape
}%
\or
\parshape:
\edef\@sf@mpg{%
\noexpand
\begin{minipage}\csname @sf@dim@\romannumeral#1\endcsname
\@ff@parshape
\begin{group}
\@ff@parshape
\let\@ff@parshape\@ff@parshape
\let\@ff@parshape\@ff@parshape
\@ff@setsecthead
}%
\or
\shapepar or \Shapepar:
\edef\@sf@mpg{%
\noexpand
\begin{minipage}\csname @sf@dim@\romannumeral#1\endcsname
\@ff@parshape
\let\@ff@parshape\@ff@parshape
}%
\fi
\edef\@sf@thisframe{\csname @staticframe@\romannumeral#1\endcsname}%
\begin{lrbox}{\@sf@thisframe}%
\edef\@ff@txtcol{\csname @sf@txtcol@\romannumeral#1\endcsname}%
\@sf@txtcol\noindent
\@sf@mpg
\setlength\@ff@parindent\@sf@parindent
}
End staticcontents stuff
\newcommand*{\@endstaticcontents}{%
\ifnum\ff@shape=2\relax
\par
\else
\FLForgapar
\fi
\endgroup
\end{minipage}%
\end{lrbox}%

```

```

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

\sfswapoddeven Allow user to specify flow frame either by IDN or IDL:
    \newcommand*{\sfswapoddeven}{%
        \ifstar\@sstaticframeswapcoords\@staticframeswapcoords
    }

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

\@staticframeswapcoords Unstarred form:
    \newcommand*{\@staticframeswapcoords}[1]{%
        \ifthenelse{\equal{#1}{all}}{%
            {%
                \ff@id=0\relax
                \whiledo{\ff@id<\c@maxflow}%
                {%
                    \advance\ff@id by 1\relax
                    \@@staticframeswapcoords{\ff@id}%
                }%
            }%
        }%
    }

```

```

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

```

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
IDN.

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

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

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

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

\newcommand*\@dynamicframeid[1]{%
    \@colN=0\relax
    \ff@id=0\relax
    \whiledo{\@colN<\c@maxdynamic}%
    {%
        \advance\@colN by 1\relax
        \ifthenelse
        {%
            \equal{#1}{\csname @df@id@\romannumeral\@colN\endcsname}%
        }%
        {%
            \ff@id=\@colN\relax
        }
    }
    Break out of loop
    \@colN=\c@maxdynamic
    }%
    {}%
}%
\ifnum\ff@id=0\relax
    \PackageError{flowfram}%
    {Can't find dynamic frame id '#1'}{}%
\fi

```

}

`\@getframeid \@getframeid{<type>}{<idl>}`

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

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

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

`\dynamicframex`

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

`\dynamicframey`

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

`\dynamicframeevenx`

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

`\dynamicframeeveny`

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

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

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

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

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

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

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

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

```
\newcommand*{\@setdynamicframe}[2]{%
  \ifthenelse{\equal{#1}{all}}{%
    {%
      \setalldynamicframes{#2}%
    }%
  }%
  {%
    \ifthenelse{\equal{#1}{odd} \TEor \equal{#1}{even}}{%
      {%
        \ifthenelse{\equal{#1}{odd}}{%
          {\@colN=1}%
          {\@colN=2}%
          \whiledo{\@colN<\c@maxdynamic\TEor\@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 \TEor \@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
\@@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

```

```

\edef\csname @df@offset@\romannumeral#1\endcsname{\ff@offset}%
}%
\ifdefempty{\ff@angle}%
{}%
{%
\expandafter
\edef\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
\edef\csname @df@style@\romannumeral#1\endcsname{\ff@style}%
}%
}%
\ifdefempty{\ff@clear}%
{}%
{%
\setboolean{@df@clear@\romannumeral#1}{\ff@clear}%
}%
\ifdefempty{\ff@margin}%
{}%
{%
\PackageError{flowfram}%
{%
Key 'margin' not available for dynamic frames%

```



```

    }%
    {dynamic frames don't have marginal notes}%
  }%
  \ifdefempty{\ff@hide}{}%
  {%
    \setboolean{@df@hide@\romannumeral#1}{\ff@hide}%
  }%
  \ifdefempty{\ff@hidethis}{}%
  {%
    \global\csletcs{if@df@hidethis@\romannumeral#1}{if\ff@hidethis}%
  }%
}

```

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

```

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

```

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

```

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

```

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

```

\newcommand*{\dynamicaddexclusion}[2]{%
  \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"}%
}

```

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

```

\newcommand*{\@@dynamicframeswapcoords}[1]{%

```

```

\setlength{\@ff@tmp@x}{%
  {\csname @df@romannumeral#1@evenx\endcsname}%
\expandafter\setlength
  \csname @df@romannumeral#1@evenx\endcsname
  {\csname @df@romannumeral#1@posx\endcsname}%
\expandafter\setlength
  \csname @df@romannumeral#1@posx\endcsname{\@ff@tmp@x}%
\setlength{\@ff@tmp@y}{%
  {\csname @df@romannumeral#1@eveny\endcsname}%
\expandafter\setlength
  \csname @df@romannumeral#1@eveny\endcsname
  {\csname @df@romannumeral#1@posy\endcsname}%
\expandafter\setlength\csname @df@romannumeral#1@posy\endcsname
  {\@ff@tmp@y}%
}

```

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

```

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

```

\@sdynamicframeswapcoords Starred form

```

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

```

\@dynamicframeswapcoords Unstarred form:

```

\newcommand*{\@dynamicframeswapcoords}[1]{%
  \ifthenelse{\equal{#1}{all}}{%
    {%
      \ff@id=0\relax
      \whiledo{\ff@id<\c@maxflow}%
      {%
        \advance\ff@id by 1\relax
        \@@dynamicframeswapcoords{\ff@id}%
      }%
    }%
  }%
  {%
    \ifthenelse{\equal{#1}{odd} \TE@or \equal{#1}{even}}{%
      {%
        \ifthenelse{\equal{#1}{odd}}{%
          {\@colN=1}%
          {\@colN=2}%
          \whiledo{\@colN<\c@maxflow\TE@or\@colN=\c@maxflow}%
          {%
            \@@dynamicframeswapcoords{\@colN}%
            \advance\@colN by 2\relax
          }%
        }%
      }%
    }%
  }%
}

```

```

}%
{%
  \@for\@ff@id:=#1\do{%
    \def\@ff@numstart{0}%
    \def\@ff@numend{10000}%
    \@ff@getrange{\@ff@id}%
    \ifnum\@ff@numstart=0\relax
      \def\@ff@numstart{1}%
    \fi
    \ifnum\@ff@numend>\c@maxflow
      \def\@ff@numend{\c@maxflow}%
    \fi
    \@colN=\@ff@numstart
    \whiledo{\@colN<\@ff@numend \TE@or \@colN=\@ff@numend}%
    {%
      \@@dynamicframeswapcoords{\@colN}%
      \advance\@colN by 1\relax
    }%
  }%
}%
}%
}

```

Set the contents of a dynamic frame.

`dynamiccontents` (*env.*) Syntax: `\begin{dynamiccontents}{\langle idn \rangle}`

The contents of the `dynamiccontents` environment needs to be stored in the control sequence `\@dynamicframe@ $\langle rn \rangle$` (where $\langle rn \rangle$ is the $\langle idn \rangle$ 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}%
  \fi
}

```

```

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

```

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

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

```

Find the end of the environment:

```

\def\@flf@find@end#1{%
    \def\@tempa{#1}%
    \global\let\flf@next=\relax
    \ifdfcontinued
        \@dynamictok\expandafter
            {\the\@dynamictok\ffcontinuedtextlayout}%
        \protected@edef\@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 $\langle id \rangle$. Can only be used inside `staticcontents` or `dynamiccontents` environments. If the starred version of the environment is used, $\{\langle id \rangle\}$ 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
  }%
}
```

```

\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{\ffareaawidth}{\flowframeawidth{#2}}%
  \setlength{\ffareaaheight}{\flowframeaheight{#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{\ffareaawidth}{\@ff@tmp@x}%
    \setlength{\ffareaaheight}{\@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{\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
}

```

`\@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
    \FLFabovetrue
  \else
    \FLFabovefalse
  \fi
}
```

The unstarred version


```

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

```

`\checkifframeright` `\checkifframeright{<type1>}{<id1>}{<type2>}{<id2>}` Checks if the first frame is to the right of the second frame where the first frame is of type *<type1>* with IDN given by *<id1>* and the second frame is of type *<type2>* with IDN given by *<id2>*. The starred version uses the IDL instead of the IDN. The first frame is not considered to be to the right of the second frame if they overlap. This code checks the page number to determine whether to use `\oddcheckifframeright`

or `\evencheckifframeright` so it should not be used in the first paragraph of the first flow frame on the page if the paragraph spans the page break.

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

Starred version:

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

Unstarred version:

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

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

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

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

The starred version

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

```

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

```

The unstarred version

```

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

```

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

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

The starred version

```
\newcommand*{\@sevencheckifframeabove}[4]{%
  \ifundefined{@sget#1evenbounds}%
  {%
    \PackageError{flowfram}%
    {Unknown frame type ‘#1’}%
    {%
      Frame types may only be one of: static, dynamic or
      flow%
    }%
  }%
}%
\csname @sget#1evenbounds\endcsname{#2}%
\edef\@ff@check{\the\ffareay}%
\ifundefined{@sget#3sevenbounds}%
{%
  \PackageError{flowfram}%
  {Unknown frame type ‘#3’}%
  {%
    Frame types may only be one of: static, dynamic or
    flow%
  }%
}%
\csname @sget#3sevenbounds\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#3evenbounds\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#3evenbounds}%

```

```

{%
  \PackageError{flowfram}%
  {Unknown frame type ‘#3’}%
  {%
    Frame types may only be one of: static, dynamic or
    flow%
  }%
}%
{}%
\csname @sget#3evenbounds\endcsname{#4}%
\expandafter\ifdim\@ff@check<\ffareay
  \FLFbelowtrue
\else
  \FLFbelowfalse
\fi
}

```

The unstarred version

```

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

```

`\evencheckifframeleft` `\evencheckifframeleft{<type1>}{<id1>}{<type2>}{<id2>}` Checks if the first

frame is to the left of the second frame where the first frame is of type $\langle type1 \rangle$ with IDN given by $\langle id1 \rangle$ and the second frame is of type $\langle type2 \rangle$ with IDN given by $\langle id2 \rangle$. 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
    \FLlefttrue
  \else
    \FLleftfalse
  \fi
}

```

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

```

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

```

The starred version

```

\newcommand*{\@sevencheckifframeright}[4]{%
  \@ifundefined{@sget#1evenbounds}%
  {%
    \PackageError{flowfram}%
    {Unknown frame type ‘#1’}%
    {%
      Frame types may only be one of: static, dynamic or
      flow%
    }%
  }%
  {}%
  \csname @sget#1evenbounds\endcsname{#2}%
  \edef\@ff@check{\the\ffareax}%
  \@ifundefined{@sget#3evenbounds}%
  {%
    \PackageError{flowfram}%

```

```

    {Unknown frame type ‘#3’}%
    {%
      Frame types may only be one of: static, dynamic or
      flow%
    }%
  }%
  {}%
  \csname @sget#3evenbounds\endcsname{#4}%
  \advance\ffareax by \ffareawidth\relax
  \expandafter\ifdim\@ff@check>\ffareax
    \FLFrighttrue
  \else
    \FLFrightfalse
  \fi
}

```

The unstarred version

```

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

```

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

```

\FFaboveleft
\newcommand*\FFaboveleft{above left}

\FFaboveright
\newcommand*\FFaboveright{above right}

\FFbelowleft
\newcommand*\FFbelowleft{below left}

\FFbelowright
\newcommand*\FFbelowright{below right}

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

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

\FFabove
\newcommand*\FFabove{above}

\FFbelow
\newcommand*\FFbelow{below}

\FFoverlap
\newcommand*\FFoverlap{overlap}

\relativeframelocation \relativeframelocation{<type1>}{<id1>}{<type2>}{<id2>} Displays one of
the above commands depending on the relative locations of the first frame to the
second frame. The arguments <id1> and <id2> refer to the IDN for the unstarred
version and to the IDL for the starred version.
\DeclareRobustCommand*\relativeframelocation{%
  \@ifstar\@srelativeframelocation\relativeframelocation
}
Starred version:
\newcommand*\@srelativeframelocation[4]{%
  \@scheckifframeabove{#1}{#2}{#3}{#4}%
  \@scheckifframebelow{#1}{#2}{#3}{#4}%
  \@scheckifframeleft{#1}{#2}{#3}{#4}%
  \@scheckifframeright{#1}{#2}{#3}{#4}%
  \ifFLFabove
    \ifFLFleft
      \FFaboveleft
    \else
      \ifFLFright
        \FFaboveright
      \else

```

```

        \FFabove
      \fi
    \fi
  \else
    \ifFLFbelow
      \ifFLFleft
        \FFbelowleft
      \else
        \ifFLFrigh
          \FFbelowright
        \else
          \FFbelow
        \fi
      \fi
    \else
      \ifFLFleft
        \FFleft
      \else
        \ifFLFrigh
          \FFright
        \else
          \FFoverlap
        \fi
      \fi
    \fi
  \fi
}

```

Unstarred version:

```

\newcommand*{\@relativeframelocation}[4]{%
  \@checkifframeabove{#1}{#2}{#3}{#4}%
  \@checkifframebelow{#1}{#2}{#3}{#4}%
  \@checkifframeleft{#1}{#2}{#3}{#4}%
  \@checkifframeright{#1}{#2}{#3}{#4}%
  \ifFLFabove
    \ifFLFleft
      \FFaboveleft
    \else
      \ifFLFrigh
        \FFaboveright
      \else
        \FFabove
      \fi
    \fi
  \else
    \ifFLFbelow
      \ifFLFleft
        \FFbelowleft
      \else
        \ifFLFrigh

```

```

        \FFbelowright
      \else
        \FFbelow
      \fi
    \fi
  \else
    \ifFLFleft
      \FFleft
    \else
      \ifFLFright
        \FFright
      \else
        \FFoverlap
      \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@setfr@mes
\@setfr@mesfalse
\newcommand*\setframes{%
  \ifnum\c@thisframe=0\relax
    \PackageWarning{flowfram}%
    {Can't find a flow frame on page 1.}
    \MessageBreak
    Attempting to find the first page with a flow frame%
  }%
  \@nxtcol=1\relax
  \c@curpg=1\relax
  \@g@tnextcol{\@nxtcol}%
}

```

Shipout pages without flow frames.

```

\advance\c@curpg by -1\relax
\whiledo{\c@curpg>0}%
{%
  \advance\c@curpg by -1\relax
  \setbox\@outputbox\vbox{\hbox to \textwidth{\@ff@do@allframes}}%
  \@outputpage
}

```

```

    }%
    \c@thisframe=\@nxtcol
  \fi
  \@setcol{\c@thisframe}\relax
  \@setfr@mesttrue
  \edef\ff@txtcol{%
    \csname @ff@txtcol@\romannumeral\c@thisframe\endcsname}%
  \@s@tffttextcol
}

```

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

```

\newcommand{\emulatetwocolumn}[1][\relax]{%
  \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][\relax]{%
  \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
  \fi
}

```

```

\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][1]{%
\PackageWarning{flowfram}%
{%
Ignoring \string\@onecolumn\space found in document environment.
Frames must be defined in the preamble%
}%
#1%
}%
\renewcommand{\@twocolumn}[1][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\vsizel\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 `\vsizel` instead of `\textheight`.

```

\output={%

```

```

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

```

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

```

\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
    \makefcolumn\@deferlist
    \@whilesw \if@fcolmade \fi
  {%
    \@opcol
    \makefcolumn\@deferlist
  }%
  \if@firstcolumn
    \xdef\@dbldeferlist{\@dbltoplist\@dbldeferlist}%
    \global\let\@dbltoplist\@empty
  \fi
}

```

```

\global\@colht\vsiz
\beginngroup
\@dblfloatplacement
\@makefcolumn\@dbldeferlist
\@whilesw \if@fcolmade \fi
{%
\@outputpage
\@makefcolumn\@dbldeferlist
}%
\endgroup
\else
\ vbox{}\%
\clearpage
\fi
\else
\setbox\@cclv\ vbox{\ box\@cclv\ vfil}\%
\@makecol\@opcol
\clearpage
\fi
}

```

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

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

```

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

```

`\@dothefoot` Same again for the footer.

```

\newcommand{\@dothefoot}{%
\color@hbox\normalcolor\ hbox to \textwidth{\@thefoot}%
\color@endbox
}
\newcommand{\@ddynamicthehead}{-}
\newcommand{\@ddynamicthefoot}{-}

```

`\@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{%
\beginngroup
\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{\@dotheft}{}%
  \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@tffttextcol
    }%
}

```

`\@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
}

```



```

\edef\@freelist{\@freelist\@midlist}%
\global\let\@midlist\@empty
\@combinefloats
\ifvbox\@kludgeins
  \@makespecialcolbox
\else
  \setbox\@outputbox\vbox to\@colht{%
    \@texttop\dimen@dp\@outputbox
    \unvbox \@outputbox
    \vskip -\dimen@\@textbottom
  }%
\fi
\global\maxdepth\@maxdepth
}

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

If found a frame, break out of loop.

```

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

```

```

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

```

Look ahead up to a maximum of 4 pages.

```

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

```

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

```

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

```

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

```

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

```

If found a frame, break out of loop.

```

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

```

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

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

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

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

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

```
\newcommand*{\@ff@checknumrange}[2]{%
  \def\@ff@numstart{0}%
  \def\@ff@numend{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}%
  \fi
}
```

```

        \@@@ff@getrange#2\end
    \fi
}

\@@@ff@getrange Extract the end value from the closed range.
    \def\@@@ff@getrange#1-\relax\end{%
        \def\@ff@numend{#1}%
    }

\@ff@output@adjustframes Provide a hook to adjust frame settings in the output routine.
    \newcommand*{\@ff@output@adjustframes}{}

\flowswitchonnext Switch on the listed flow frames from the next page onwards
    \newcommand*{\flowswitchonnext}{%
        \ifstar\@sflowswitchonnext\@flowswitchonnext
    }

\@sflowswitchonnext The starred version uses IDLs.
    \newcommand{\@sflowswitchonnext}[1]{%
        \@for\@ff@id:=#1\do{%
            \@flowframeid{\@ff@id}%
        }
        Is this frame already on?
        \@ff@chckifthispg{\@ff@pages@countreg}{\@ff@id}%
        \expandafter\toks@ \expandafter{\@ff@output@adjustframes}%
        \if@notthiscol
            \xdef\@ff@output@adjustframes{%
                \the\toks@
                \noexpand\flowsetpagelist{\number\@ff@id}{>\number\@ff@pages@countreg}%
            }%
        \else
            \xdef\@ff@output@adjustframes{%
                \the\toks@
                \noexpand\flowsetpagelist{\number\@ff@id}%
                {\number\@ff@pages@countreg,>\number\@ff@pages@countreg}%
            }%
        \fi
    }%
}

\@flowswitchonnext The unstarred version uses IDNs.
    \newcommand{\@flowswitchonnext}[1]{%
        \@for\@ff@id:=#1\do{%
            Is this frame already on?
            \@ff@chckifthispg{\@ff@pages@countreg}{\@ff@id}%
            \expandafter\toks@ \expandafter{\@ff@output@adjustframes}%
            \if@notthiscol
                \xdef\@ff@output@adjustframes{%
                    \the\toks@

```

```

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

\flowswitchonnextodd Switch on the listed flow frames from the next odd page onwards
\newcommand*{\flowswitchonnextodd}{%
    \ifstar\@sflowswitchonnextodd\@flowswitchonnextodd
}

\@sflowswitchonnextodd The starred version uses IDLs.
\newcommand{\@sflowswitchonnextodd}[1]{%
    \count@=\@ff@pages@countreg\relax
    \ifodd\count@\relax
        \advance\count@ by 1\relax
    \fi
    \@for\@ff@id:=#1\do{%
        \@flowframeid{\@ff@id}%
    }
    Is this frame already on?
    \iff@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?
    \iff@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 IDNs.

```
\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
```

```

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

```

`\@flowswitchoffnext` The unstarred version uses IDNs.

```

\newcommand{\@flowswitchoffnext}[1]{%
  \@for\@ff@id:=#1\do{%
    Is this frame already off on this page?
    \@ff@chckifthispg{\@ff@pages@countreg}{\@ff@id}%
    \if@notthiscol
      \def\@ff@pages{none}%
    \else
      \def\@ff@pages{\number\@ff@pages@countreg}%
    \fi
    \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
    \xdef\@ff@output@adjustframes{%
      \the\toks@
      \noexpand\flowsetpagelist{\number\@ff@id}{\@ff@pages}%
    }%
  }%
}

```

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

```

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

```

`\@sflowswitchoffnextodd` The starred version uses IDLs.

```

\newcommand{\@sflowswitchoffnextodd}[1]{%
  \count@=\@ff@pages@countreg\relax
  \ifodd\@ff@pages@countreg\relax
    \advance\count@ by 1\relax
  \fi
  \@for\@ff@id:=#1\do{%
    \@flowframeid{\@ff@id}%
    Is this frame already off on this page?
    \@ff@chckifthispg{\@ff@pages@countreg}{\@ff@id}%
    \if@notthiscol
      It's off on this page. Is it on or off on the next page, if this page is odd? First,
      is this page odd?
      \ifnum\@ff@pages@countreg=\count@\relax

```


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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

\@flowswitchoffnextodd The unstarred version uses IDNs.

```
\newcommand{\@flowswitchoffnextodd}[1]{%
  \count@=\@ff@pages@countreg\relax
  \ifodd\@ff@pages@countreg\relax
    \advance\count@ by 1\relax
```

```

\fi
\@for\@ff@id:=#1\do{%
Is this frame already off on this page?
\@ff@chckifthispg{\@ff@pages@countreg}{\@ff@id}%
\if@notthiscol
It's off on this page. Is it on or off on the next page, if this page is odd? First,
is this page odd?
\ifnum\@ff@pages@countreg=\count@\relax
This page is even and the frame is off on this page, so set to none.
\def\@ff@nextpages{none}%
\else
This page is odd. Is the frame on or off on the next page?
\@ff@chckifthispg{\count@}{\@ff@id}%
\if@notthiscol
Off on the next page as well, so set to none.
\def\@ff@nextpages{none}%
\else
Not off on the next page, so set to next page only.
\def\@ff@nextpages{\number\count@}%
\fi
\fi
\else
It's not off on this page. Is it on or off on the next page, if this page is odd?
First, is this page odd?
\ifnum\@ff@pages@countreg=\count@\relax
This page is even and the frame is not off on this page, so set to this page.
\def\@ff@nextpages{\number\@ff@pages@countreg}%
\else
This page is odd. Is the frame on or off on the next page?
\@ff@chckifthispg{\count@}{\@ff@id}%
\if@notthiscol
Off on the next page but not off on this page. So set to just this page.
\def\@ff@nextpages{\number\@ff@pages@countreg}%
\else
Not off on the next page as well, so set to this page and next page.
\def\@ff@nextpages{\number\@ff@pages@countreg,\number\count@}%
\fi
\fi
\fi
\expandafter\toks@\expandafter{\@ff@output@adjustframes}%
\xdef\@ff@output@adjustframes{%
\the\toks@

```

```

        \noexpand\flowsetpagelist{\number\@ff@id}{\@ff@nextpages}%
      }%
    }%
  }

```

`\flowswitchonnexonly` Switch on the listed flow frames for just the next page

```

\newcommand*{\flowswitchonnexonly}{%
  \ifstar\@sflowswitchonnexonly\@flowswitchonnexonly
}

```

`\@sflowswitchonnexonly` The starred version uses IDLs.

```

\newcommand{\@sflowswitchonnexonly}[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
}%
}

```

`\@flowswitchonnexonly` The unstarred version uses IDNs.

```

\newcommand{\@flowswitchonnexonly}[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
}%
}
```

`\flowswitchonnnextoddonly` Switch on the listed flow frames for just the next odd page

```
\newcommand*{\flowswitchonnnextoddonly}{%
  \@ifstar\@sflowswitchonnnextoddonly\@flowswitchonnnextoddonly
}
```

`\@sflowswitchonnnextoddonly` The starred version uses IDLs.

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

Is this frame already on?

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

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

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

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

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

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

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

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

```
\edef\@ff@pages{\number\count@}%
```

```

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

```

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

```

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

```

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

```

\ifodd\@ff@pages@countreg

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

```

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

```

```

}

```

`\@flowswitchonnextoddonly` The unstarred version uses IDNs.

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

Is this frame already on?

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```

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

```

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

```

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

```

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

```

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

\fi
\expandafter\toks@\expandafter{\@ff@output@adjustframes}%
\edef\@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}%
\edef\@ff@output@adjustframes{%
\the\toks@
\noexpand\flowaddexclusion{\number\@ff@id}{\number\count@}%
}%
}%
}

```

\@flowswitchoffnextonly The unstarred version uses IDNs.

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

\flowswitchoffnextoddonly Switch off the listed flow frames for just the next odd page

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

\@sflowswitchoffnextoddonly The starred version uses IDLs.

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

\@flowswitchoffnextoddonly The unstarred version uses IDNs.

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



```

    }%
  }%
}

\dynamicswitchonnext Switch on the listed dynamic frames from the next page onwards
  \newcommand*{\dynamicswitchonnext}{%
    \ifstar\@dynamicswitchonnext\dynamicswitchonnext
  }

\@dynamicswitchonnext The starred version uses IDLs.
  \newcommand{\@dynamicswitchonnext}[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
    }
  }

```

```

        \fi
      }%
    }

\dynamicswitchonnextodd Switch on the listed dynamic frames from the next odd page onwards
    \newcommand*{\dynamicswitchonnextodd}{%
      \ifstar\@sdynamicswitchonnextodd\@dynamicswitchonnextodd
    }

\@sdynamicswitchonnextodd The starred version uses IDLs.
    \newcommand{\@sdynamicswitchonnextodd}[1]{%
      \count@=\@ff@pages@countreg\relax
      \ifodd\count@\relax
        \advance\count@ by 1\relax
      \fi
      \@for\@ff@id:=#1\do{%
        \@dynamicframeid{\@ff@id}%
        Is this frame already on?
        \def\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?
        \def\chckifthispg[\count@]{\@ff@id}%
        \ifnum\count@=\@ff@pages@countreg\relax
        \else
          \if@notthiscol
          \else
            \edef\@ff@prepages{\@ff@prepages\number\count@,}%
          \fi
        \fi
        \expandafter\toks@\expandafter{\@ff@output@adjustframes}%
        \xdef\@ff@output@adjustframes{%
          \the\toks@
          \noexpand\dynamicsetpagelist{\number\@ff@id}%
            {\@ff@prepages>\number\count@}%
        }%
      }%
    }

\@dynamicswitchonnextodd The unstarred version uses IDNs.
    \newcommand{\@dynamicswitchonnextodd}[1]{%
      \count@=\@ff@pages@countreg\relax
      \ifodd\count@\relax
        \advance\count@ by 1\relax
      \fi
      \@for\@ff@id:=#1\do{%

```

Is this frame already on?

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

Is this frame already switched on for the next page?

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

`\dynamicsswitchoffnext` Switch off the listed dynamic frames from the next page onwards

```
\newcommand*{\dynamicsswitchoffnext}{%
\@ifstar\@dynamicsswitchoffnext\@dynamicsswitchoffnext
}
```

`\@dynamicsswitchoffnext` The starred version uses IDLs.

```
\newcommand{\@dynamicsswitchoffnext}[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\@sdynamicsswitchoffnextodd\dynamicsswitchoffnextodd
}
```

`\@sdynamicsswitchoffnextodd` The starred version uses IDLs.

```
\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}%
```

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}%
\edef\@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\@dynamicswitchonnexonly\@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
}%
}
```

`\@dynamicswitchonnexonly` The unstarred version uses IDNs.

```
\newcommand{\@dynamicswitchonnexonly}[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
}

```

`\dynamicsswitchonnextoddone` Switch on the listed dynamic frames for just the next odd page

```

\newcommand*{\dynamicsswitchonnextoddone}{%
  \ifstar\@dynamicsswitchonnextoddone\@dynamicsswitchonnextoddone
}

```

`\dynamicsswitchonnextoddone` The starred version uses IDLs.

```

\newcommand{\@dynamicsswitchonnextoddone}[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}%
}%
}%
}
```

\@dynamicswitchonnextoddonly The unstarred version uses IDNs.

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

Is this frame already on?

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```

```

\fi
\else

```

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

```

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

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

```

`\dynamicsswitchoffnextonly` Switch off the listed dynamic frames for just the next page

```

\newcommand*{\dynamicsswitchoffnextonly}{%
\ifstar\@dynamicsswitchoffnextonly\@dynamicsswitchoffnextonly
}

```

`\@dynamicsswitchoffnextonly` The starred version uses IDLs.

```

\newcommand{\@dynamicsswitchoffnextonly}[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@}%
}%
}%
}

```

`\@dynamicsswitchoffnextonly` The unstarred version uses IDNs.

```

\newcommand{\@dynamicsswitchoffnextonly}[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@}%
}%
}%
}

```

`\dynamicsswitchoffnextoddone` Switch off the listed dynamic frames for just the next odd page

```

\newcommand*{\dynamicsswitchoffnextoddone}{%
  \ifstar\@dynamicsswitchoffnextoddone\dynamicsswitchoffnextoddone
}

```

`\dynamicsswitchoffnextoddone` The starred version uses IDLs.

```

\newcommand{\@dynamicsswitchoffnextoddone}[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@}%
    }%
  }%
}

```

`\dynamicsswitchoffnextoddone` The unstarred version uses IDNs.

```

\newcommand{\@dynamicsswitchoffnextoddone}[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\@staticsswitchonnext\staticsswitchonnext
}

```

`\@staticsswitchonnext` The starred version uses IDLs.

```

\newcommand{\@staticsswitchonnext}[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
\edef\@ff@output@adjustframes{%
\the\toks@
\noexpand\staticsetpagelist{\number\ff@id}{>\number\@ff@pages@countreg}%
}%
\else
\edef\@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
\edef\@ff@output@adjustframes{%
\the\toks@
\noexpand\staticsetpagelist{\number\@ff@id}{>\number\@ff@pages@countreg}%
}%
\else
\edef\@ff@output@adjustframes{%
\the\toks@
\noexpand\staticsetpagelist{\number\@ff@id}%
{\number\@ff@pages@countreg,>\number\@ff@pages@countreg}%
}%
\fi
}%
}

```

`\staticswitchonnextodd` Switch on the listed static frames from the next odd page onwards

```

\newcommand*\staticswitchonnextodd{%
\@ifstar\@sstaticswitchonnextodd\staticswitchonnextodd
}

```

`\@sstaticswitchonnextodd` The starred version uses IDLs.

```

\newcommand{\@sstaticswitchonnextodd}[1]{%
\count@=\@ff@pages@countreg\relax
\ifodd\count@\relax
\advance\count@ by 1\relax

```

```

\fi
\@for\@ff@id:=#1\do{%
  \@staticframeid{\@ff@id}%
Is this frame already on?
  \@sf@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
  \def\@ff@prepages{}%
  \if@notthiscol
  \else
    \def\@ff@prepages{\number\@ff@pages@countreg,}%
  \fi
Is this frame already switched on for the next page?
  \@sf@chckifthispg[\count@]{\@ff@id}%
  \ifnum\count@=\@ff@pages@countreg\relax
  \else
    \if@notthiscol
    \else
      \edef\@ff@prepages{\@ff@prepages\number\count@,}%
    \fi
  \fi
  \expandafter\toks@{\expandafter{\@ff@output@adjustframes}%
  \xdef\@ff@output@adjustframes{%
    \the\toks@
    \noexpand\staticsetpagelist{\number\@ff@id}%
    {\@ff@prepages>\number\count@}%
  }%
}%
}
}

```

\@staticswitchonnextodd The unstarred version uses IDNs.

```

\newcommand{\@staticswitchonnextodd}[1]{%
  \count@=\@ff@pages@countreg\relax
  \ifodd\count@\relax
    \advance\count@ by 1\relax
  \fi
  \@for\@ff@id:=#1\do{%
Is this frame already on?
    \@sf@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
    \def\@ff@prepages{}%
    \if@notthiscol
    \else
      \def\@ff@prepages{\number\@ff@pages@countreg,}%
    \fi
Is this frame already switched on for the next page?
    \@sf@chckifthispg[\count@]{\@ff@id}%
    \ifnum\count@=\@ff@pages@countreg\relax
    \else
      \if@notthiscol

```

```

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

```

`\staticswitchoffnext` Switch off the listed static frames from the next page onwards

```

\newcommand*{\staticswitchoffnext}{%
\@ifstar\@sstaticswitchoffnext\@staticswitchoffnext
}

```

`\@sstaticswitchoffnext` The starred version uses IDLs.

```

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

```

Is this frame already off on this page?

```

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

```

`\@staticswitchoffnext` The unstarred version uses IDNs.

```

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

```

Is this frame already off on this page?

```

\@sf@chckifthispg[\@ff@pages@countreg]{\@ff@id}%
\if@notthiscol
\def\@ff@pages{none}%
\else
\def\@ff@pages{\number\@ff@pages@countreg}%
\fi
\expandafter\toks@\expandafter{\@ff@output@adjustframes}%

```

```

\xdef\@ff@output@adjustframes{%
  \the\toks@
  \noexpand\staticsetpagelist{\number\@ff@id}{\@ff@pages}%
}%
}%
}

```

`\staticswitchoffnextodd` Switch off the listed static frames from the next odd page onwards

```

\newcommand*{\staticswitchoffnextodd}{%
  \ifstar\@sstaticswitchoffnextodd\staticswitchoffnextodd
}

```

`\@sstaticswitchoffnextodd` The starred version uses IDLs.

```

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

```

Is this frame already off on this page?

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```


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

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

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

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

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

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

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

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

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

\@staticswitchoffnextodd The unstarred version uses IDNs.

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

Is this frame already off on this page?

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
\@sf@chkifthispg[\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}%
}%
}%
}
```

`\staticswitchonnexonly` Switch on the listed static frames for just the next page

```
\newcommand*{\staticswitchonnexonly}{%
  \@ifstar\@sstaticswitchonnexonly\@staticswitchonnexonly
}
```

`\@sstaticswitchonnexonly` The starred version uses IDLs.

```
\newcommand{\@sstaticswitchonnexonly}[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@chkifthispg[\@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
}%
}
```

`\@staticswitchonnexonly` The unstarred version uses IDNs.

```
\newcommand{\@staticswitchonnexonly}[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
}%
}
```

`\staticswitchonnextodonly` Switch on the listed static frames for just the next odd page

```
\newcommand*{\staticswitchonnextodonly}{%
  \@ifstar\@sstaticswitchonnextodonly\@staticswitchonnextodonly
}
```

@sstaticswitchonnnextodonly The starred version uses IDLs.

```
\newcommand{\@sstaticswitchonnnextodonly}[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}%
}%
}%
}
```

\@staticswitchonnextoddonly The unstarred version uses IDNs.

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

Is this frame already on?

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
\count@=\@ff@pages@countreg\relax
\advance\count@ by 1\relax
\@sf@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\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@}%
    }%
  }%
}

```

`\staticswitchoffnextoddonly` Switch off the listed static frames for just the next odd page

```

\newcommand*{\staticswitchoffnextoddonly}{%
  \ifstar\@sstaticswitchoffnextoddonly\@staticswitchoffnextoddonly
}

```

`\sstaticswitchoffnextodonly` The starred version uses IDLs.

```
\newcommand{\@sstaticswitchoffnextodonly}[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@}%
    }%
  }%
}
```

`\staticswitchoffnextodonly` The unstarred version uses IDNs.

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

`\ffaddtoadjustframeshook` Add stuff to the output hook.

```
\newcommand*{\ffaddtoadjustframeshook}[1]{%
  \@ff@addtolist\@ff@output@adjustframes\entry{#1}%
}
```

`\@g@tnextcol` Find the next flow frame. If there are no more flow frames, define a new one the size of the typeblock. (Otherwise the remaining document text will be lost.)

```
\newif\if@notthiscol
\newif\if@ff@nwpg
\newcount\c@curpg
\newcommand*{\@g@tnextcol}[1]{%
```

Do any frame adjustments

```
\@ff@output@adjustframes
```


Now clear the hook

```
\global\let\@ff@output@adjustframes\@empty
```

Now check for any more frames.

```
\@ff@checkifmoreframes  
\if@ff@moreframes  
\else
```

No more frames, add new frame

```
\PackageWarning{flowfram}%  
{Run out of flows frames on page \number\@ff@pages@countreg, adding new one}%  
\flf@doifverbose  
{%  
  \def\flf@messinfo{Here's the list of flow frames:}%  
  \count@=0\relax  
  \loop  
    \advance\count@ by 1\relax  
    \expandafter\toks@\expandafter{\flf@messinfo\MessageBreak}%  
    \edef\flf@messinfo{\the\toks@  
      \number\count@.  
      Pages: \csname @ff@pages@\romannumeral\count@\endcsname.  
      Exclusions: \csname @ff@xpages@\romannumeral\count@\endcsname.  
    }%  
    \ifnum\count@<\c@maxflow  
    \repeat  
    \PackageInfo{flowfram}{\flf@messinfo\@gobbletwo}%  
  }%  
  \@onecolumn  
  #1=\c@maxflow  
\fi  
\@notthiscoltrue  
\@ff@nwpgfalse  
\@colN=#1\relax  
  
\c@curpg=\@ff@pages@countreg  
\loop  
  \ifnum\@colN=\c@maxflow
```

Reached the end of the page. Try the next one.

```
\@colN=1\relax  
\@ff@nwpgtrue  
\advance\c@curpg by 1\relax  
\else
```

Move on to the next flow frame on this page.

```
\advance\@colN by 1\relax  
\fi  
\@ff@chckifthispg{\c@curpg}{\@colN}%  
\if@notthiscol  
\repeat  
#1=\@colN\relax  
}
```

`\@ff@chckifthispg` This is used to determine the next flow frame, since not all flow frames may be defined on every page. Checks to see if flow frame #2 is defined on page #1. First set up some variables.

```
\newcommand*{\@ff@chckifthispg}[2]{%
  \@notthiscolfalse
  \edef\ff@xpages{\csname @ff@xpages@romannumeral#2\endcsname}%
  \@for\@ff@pp:=\ff@xpages\do
  {%
    \ifnum0\@ff@pp=#1\relax
      \@notthiscoltrue
    \endfortrue
  }%
  \fi
  \if@notthiscol
  \else
    \@notthiscoltrue
    \edef\ff@pages{\csname @ff@pages@romannumeral#2\endcsname}%
    \@ff@chckifthispg{#1}%
  \fi
}
```

`\@ff@chckifthispg` Now go ahead and check.

```
\newcommand*{\@ff@chckifthispg}[1]{%
  \ifthenelse{\equal{\ff@pages}{none}}{%
    {}%
  }%
  \ifthenelse{\equal{\ff@pages}{all}}{%
    {}%
  }%
  \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 T_EX 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 \TeX box #3 of width #1 and height #2, and frame making command specified by #4.

```
\newcommand{\@ff@fbox}[4]{%
  {%
    \fboxsep=\flowframesep
    \fboxrule=\flowframerule
    \@s@tffcol
    \kern\@ff@offset
    #4{\@ff@box{#1}{#2}{#3}}%
  }%
}
```

`\@ff@box` Put the \TeX box #3 of width #1 and height #2 on the page.

```
\newcommand{\@ff@box}[3]{%
  {%
    \@ffbackground
    {%
      \vbox to#2 {\hb@xt@ #1{\hss{\@s@tffttextcol #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@ovalboxoffset` Compute offset for `\ovalbox`:

```

\newcommand*{\@ff@s@t@ovalboxoffset}{%
  \@ff@offset=-\fontdimen 8\tenlnw\relax
  \advance\@ff@offset by -\flowframesep\relax
}

```

`\@ff@s@t@defaultoffset` Compute default offset:

```

\newcommand*{\@ff@s@t@defaultoffset}{%
  \@ff@offset=-\flowframesep\relax
  \addtolength{\@ff@offset}{-\flowframerule}%
}

```

`\@ff@setoffset` Compute offset for flow frame #1. Stores offset value in `\ff@offset`.

```

\newcommand*{\@ff@setoffset}[1]{%
  \ifthenelse
    {equal{\csname @ff@offset@romannumeral#1\endcsname}{compute}}%
  {%
    \ifthenelse{\boolean{columnframe@romannumeral#1}}%
    {%
      \ifthenelse
        {%
          \equal{\csname @ff@frametype@romannumeral#1\endcsname}%
            {doublebox}%
        }
      }
    }
  }
}

```

```

}%
{%
  \@ff@s@t@doubleboxoffset
}%
{%
  \ifthenelse
  {%
    \equal{\csname @ff@frametype@\romannumeral#1\endcsname}%
    {ovalbox}%
  }%
  {%
    \@ff@s@t@ovalboxoffset
  }%
  {%
    \ifthenelse
    {%
      \equal{\csname @ff@frametype@\romannumeral#1\endcsname}%
      {0valbox}%
    }%
    {%
      \@ff@s@t@0valboxoffset
    }%
    {%
      \@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}%
        {0valbox}%
    }%
    {%
      \@ff@s@t@0valboxoffset
    }%
    {%
      \@ff@s@t@defaultoffset
    }%
  }%
}%
}%
{}%
}%
{%
  \setlength{\@ff@offset}%
  {\csname @sf@offset@\romannumeral#1\endcsname}%
}%
}

```

\@df@setoffset Compute offset for dynamic frame #1. Stores offset value in \@ff@offset.

```

\newcommand*{\@df@setoffset}[1]{%
  \ifthenelse
  {%
    \equal{\csname @df@offset@\romannumeral#1\endcsname}%
      {compute}%
  }%
  {%
    \setlength{\@ff@offset}{0pt}%
    \ifthenelse{\boolean{dynamicframe\romannumeral#1}}%
    {%
      \ifthenelse
      {%
        \equal{\csname @df@frametype@\romannumeral#1\endcsname}%

```

```

        {doublebox}%
    }%
    {%
        \@ff@s@t@doubleboxoffset
    }%
    {%
        \ifthenelse
        {%
            \equal{\csname @df@frametype@\romannumeral#1\endcsname}%
            {ovalbox}%
        }%
        {%
            \@ff@s@t@ovalboxoffset
        }%
        {%
            \ifthenelse
            {%
                \equal{\csname @df@frametype@\romannumeral#1\endcsname}%
                {ovalbox}%
            }%
            {%
                \@ff@s@t@ovalboxoffset
            }%
            {%
                \@ff@s@t@defaultoffset
            }%
        }%
    }%
    }%
    {}%
}%
{%
    \setlength{\@ff@offset}%
    {\csname @df@offset@\romannumeral#1\endcsname}%
}%
}

```

\@putmarginbox Draw box representing the margin for flow frame #1.

```

\newcommand*{\@putmarginbox}[1]{%
    \@ff@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}%
            \fi
        \fi
    \fi
}

```

```

\edef\ff@y{\csname col@\romannumeral#1@eveny\endcsname}%
\fi
\else
\edef\ff@x{\csname col@\romannumeral#1@posx\endcsname}%
\edef\ff@y{\csname col@\romannumeral#1@posy\endcsname}%
\fi
\setlength{\@ff@tmp@x}{\ff@x}%
\setlength{\@ff@tmp@y}{\ff@y}%
\@getmarginpos{\csname @ff@margin@\romannumeral#1\endcsname}%
\ifthenelse{\equal{\ff@margin}{left}}%
{%
\addtolength{\@ff@tmp@x}{-\marginparwidth}%
\addtolength{\@ff@tmp@x}{-\marginparsep}%
\ifthenelse{\boolean{columnframe\romannumeral#1}}%
{}%
{}%
}%
{%
\addtolength{\@ff@tmp@x}%
{\csname colwidth\romannumeral#1\endcsname}%
\addtolength{\@ff@tmp@x}{\marginparsep}%
\ifthenelse{\boolean{columnframe\romannumeral#1}}%
{}%
{}%
}%
\raise\@ff@tmp@y
\hb@xt@\z@
{%
\expandafter\kern\@ff@tmp@x
\@fr@meifdraft{\@ff@box{\marginparwidth}%
{\csname colheight\romannumeral#1\endcsname}{}}%
{M:\number#1}\hss
}%
\fi
\ignorespaces
}

```

\@ff@drawmargins Draw all the margins associated with the flow frames defined on the current page.

```

\newcommand*{\@ff@drawmargins}{%
\@colN=0\relax
\whiledo{\@colN<\c@maxflow}%
{%
\advance\@colN by 1\relax
\makebox[0pt][l]{\@putmarginbox{\@colN}}%
}%
}

```

\@ff@getstaticpos Extract the width and height for static or dynamic frame specified in the form [*c*] [*height*] [*valign*]{*width*}

```

\def\@ff@getstaticpos[#1][#2][#3]#4{%
  \@ff@tmp@x=#4\relax
  \@ff@tmp@y=#2\relax
  \def\ff@valign{#3}%
}

```

\@dostaticbox Display the savebox associated with static frame #1

```

\newcommand*\@dostaticbox[1]{%
  \edef\ff@frametype{%
    \csname @sf@frametype@\romannumeral#1\endcsname
  }%
  \edef\ff@col{\csname @sf@col@\romannumeral#1\endcsname}%
  \edef\ff@backcol{\csname @sf@backcol@\romannumeral#1\endcsname}%
  \@sf@setoffset{#1}%
  \expandafter\expandafter\expandafter
    \@ff@getstaticpos\csname @sf@dim@\romannumeral#1\endcsname
  \rotateframe
    {\csname @sf@angle@\romannumeral#1\endcsname}%
  {%
    \ifthenelse{\boolean{staticframe\romannumeral#1}}{%
      {%
        \@ff@fbox{\@ff@tmp@x}{\@ff@tmp@y}%
        {%
          \expandafter\usebox\csname @staticframe@\romannumeral#1\endcsname
        }
        {\csname\ff@frametype\endcsname}%
      }%
      {%
        \@ff@box{\@ff@tmp@x}{\@ff@tmp@y}%
        {%
          \expandafter\usebox\csname @staticframe@\romannumeral#1\endcsname
        }%
      }%
    }%
  }%
}

```

\@dostaticbbox Now for the bounding box:

```

\newcommand*\@dostaticbbox[1]{%
  \edef\ff@col{%
    \sf@setoffset{#1}%
  }%
  \expandafter\expandafter\expandafter
    \@ff@getstaticpos\csname @sf@dim@\romannumeral#1\endcsname
  \@fr@meifdraft
  {%
    \@ff@box{\@ff@tmp@x}{\@ff@tmp@y}%
    {%
      \expandafter\usebox\csname @staticframe@\romannumeral#1\endcsname
    }%
  }%
  {\S:\number#1;\csname @sf@id@\romannumeral#1\endcsname}%
}

```

}

\@putstaticbox Put the static box #1 at its given position, with its associated border.

\newcommand*{\@putstaticbox}[1]{%

Check the 'hide' and 'hidethis' attributes

\ifthenelse{\boolean{@sf@hidethis@romannumeral#1}}%

{%

\@notthiscoltrue

\global\csletcs@if@sf@hidethis@romannumeral#1}{iffalse}%

}%

{%

\ifthenelse{\boolean{@sf@hide@romannumeral#1}}%

{%

\@notthiscoltrue

}%

{%

Neither 'hide' nor 'hidethis' have been set so check the page list.

\@sf@chckifthispg{#1}%

}%

}%

\if@notthiscol

\else

\@killglue

\if@twoside

\ifodd\c@page

\expandafter\raise\csname @sf@romannumeral#1@posy\endcsname

\hb@xt@\z@

{%

\expandafter\kern \csname @sf@romannumeral#1@posx\endcsname

\@dostaticbox{#1}\hss

}%

\else

\expandafter\raise\csname @sf@romannumeral#1@eveny\endcsname

\hb@xt@\z@

{%

\expandafter\kern \csname @sf@romannumeral#1@evenx\endcsname

\@dostaticbox{#1}\hss

}%

\fi

\else

\expandafter\raise\csname @sf@romannumeral#1@posy\endcsname

\hb@xt@\z@

{%

\expandafter\kern \csname @sf@romannumeral#1@posx\endcsname

\@dostaticbox{#1}\hss

}%

\fi

\fi

}

\@putstaticbbox Now for the bounding box:

```

\newcommand*{\@putstaticbbox}[1]{%
  \@sf@chckifthispg{#1}%
  \if@notthiscol
  \else
    \@killglue
    \if@twoside
      \ifodd\c@page
        \expandafter\raise\csname @sf@romannumeral#1@posy\endcsname
        \hb@xt@\z@
        {%
          \expandafter\kern\csname @sf@romannumeral#1@posx\endcsname
          \@dostaticbbox{#1}\hss
        }%
        \ignorespaces
      \else
        \expandafter\raise\csname @sf@romannumeral#1@eveny\endcsname
        \hb@xt@\z@
        {%
          \expandafter\kern\csname @sf@romannumeral#1@evenx\endcsname
          \@dostaticbbox{#1}\hss
        }%
        \ignorespaces
      \fi
    \else
      \expandafter\raise\csname @sf@romannumeral#1@posy\endcsname
      \hb@xt@\z@
      {%
        \expandafter\kern\csname @sf@romannumeral#1@posx\endcsname
        \@dostaticbbox{#1}\hss
      }%
      \ignorespaces
    \fi
  \fi
}

```

\@resetst@tics Clear the contents of all the static frames that have the clear option set.

```

\newcommand*{\@resetst@tics}{%
  \@colN=0\relax
  \whiledo{\@colN<\c@maxstatic}%
  {%
    \advance\@colN by 1\relax

```

Has the clear flag been set?

```

    \ifthenelse{\boolean{@sf@clear@romannumeral\@colN}}{%
      {%

```

Set the contents of the box to empty

```

      \global\sbox
      {%

```

```

\csname @staticframe@\romannumeral\@colN\endcsname
}%
{}%
}%
{}%
}%
}

```

`\@resetdyn@mics` Clear the contents of the dynamic frames that have the `clear` option set.

```

\newcommand*{\@resetdyn@mics}{%
\@colN=0\relax
\whiledo{\@colN<\c@maxdynamic}%
{%
\advance\@colN by 1\relax
\ifthenelse{\boolean{@df@clear@\romannumeral\@colN}}{%
{%
\expandafter\global\expandafter
\gdef\csname @dynamicframe@\romannumeral\@colN\endcsname{}%
}%
}%
}%
}

```

`\@dodfparbox` Display contents of dynamic box (contents stored in `\ff@contents`, style given by `\ff@style`):

```

\newcommand*{\@dodfparbox}[1]{%
\expandafter\let\expandafter
\@ff@parshape\csname @df@shape@\romannumeral#1\endcsname
\expandafter\@ff@getshape\@ff@parshape\relax
\ifcase\ff@shape
no shape
\expandafter\expandafter\expandafter
\parbox\csname @df@dim@\romannumeral#1\endcsname
{%
\setlength\parindent\sdfparindent
\csname\ff@style\endcsname{\ff@contents}%
}%
\or
\parshape
\expandafter\expandafter\expandafter
\parbox\csname @df@dim@\romannumeral#1\endcsname
{%
\setlength\parindent\sdfparindent
\csname\ff@style\endcsname
{%
\let\oldpar=\par
\let\par=\ffpshpar
\ff@setsecthead

```

```

        \@ff@parshape
        \ff@contents\oldpar
    }}%
}%
\or
\shapepar
    \expandafter\expandafter\expandafter
    \parbox\csname @df@dim@\romannumeral#1\endcsname
    {%
        \setlength\parindent\sdfparindent
        \csname\ff@style\endcsname
        {%
            \@ff@disablesec\@ff@parshape
            \ff@contents\par
        }}%
    }%
\fi
}

```

\@dodynamicbox Typeset the dynamic box with its associated border.

```

\newcommand*{\@dodynamicbox}[1]{%
    \edef\ff@frametype{%
        \csname @df@frametype@\romannumeral#1\endcsname
    }%
    \edef\ff@col{\csname @df@col@\romannumeral#1\endcsname}%
    \edef\ff@txtcol{\csname @df@txtcol@\romannumeral#1\endcsname}%
    \edef\ff@backcol{\csname @df@backcol@\romannumeral#1\endcsname}%
    \edef\ff@style{\csname @df@style@\romannumeral#1\endcsname}%
    \def\ff@contents{\csname @dynamicframe@\romannumeral#1\endcsname}%
    \@df@setoffset{#1}%
    \expandafter\expandafter\expandafter
    \@ff@getstaticpos\csname @df@dim@\romannumeral#1\endcsname
    \rotateframe{\csname @df@angle@\romannumeral#1\endcsname}%
    {%
        \ifthenelse{\boolean{dynamicframe\romannumeral#1}}{%
            {%
                \@ff@fbox{\@ff@tmp@x}{\@ff@tmp@y}%
                {\@dodfparbox{#1}}%
                {\csname\ff@frametype\endcsname}%
            }%
            {%
                \@ff@box{\@ff@tmp@x}{\@ff@tmp@y}%
                {%
                    \@dodfparbox{#1}%
                }%
            }%
        }%
    }%
}

```


\@dodynamicbbox Now for the bounding box:

```

\newcommand*{\@dodynamicbbox}[1]{%
  \edef\ff@col{%
    \@df@setoffset{#1}%
    \expandafter\expandafter\expandafter
    \@ff@getstaticpos\csname @df@dim@\romannumeral#1\endcsname
    \@fr@meifdraft
  {%
    \@ff@box{\@ff@tmp@x}{\@ff@tmp@y}%
    {%
      \expandafter\expandafter\expandafter
      \parbox\csname @df@dim@\romannumeral#1\endcsname
      {}%
    }%
  }%
}%
}
\@D:number#1;\csname @df@id@\romannumeral#1\endcsname}%
}

```

\@putdynamicbox Put the dynamic frame #1 at its given position

```

\newcommand*{\@putdynamicbox}[1]{%
  Check the 'hide' and 'hidethis' attributes
  \ifthenelse{\boolean{@df@hidethis@\romannumeral#1}}{%
    {%
      \@notthiscoltrue
      \global\csletcs@if@df@hidethis@\romannumeral#1}{iffalse}%
    }%
  {%
    \ifthenelse{\boolean{@df@hide@\romannumeral#1}}{%
      {%
        \@notthiscoltrue
      }%
    }%
  }%
}

```

Neither 'hide' nor 'hidethis' have been set so check the page list.

```

  \@df@chckifthispg{#1}%
}%
}%
\if@notthiscol
\else
  \@killglue
  \if@twoside
  \ifodd\c@page
    \expandafter\raise\csname @df@\romannumeral#1@posy\endcsname
    \hb@xt@\z@
    {%
      \expandafter\kern\csname @df@\romannumeral#1@posx\endcsname
      \@dodynamicbox{#1}\hss
    }%
  \ignorespaces

```

```

\else
\expandafter\raise\csname @df@\romannumeral#1@eveny\endcsname
\hb@xt@\z@
{%
\expandafter\kern \csname @df@\romannumeral#1@evenx\endcsname
\@dodynamicbox{#1}\hss
}%
\ignorespaces
\fi
\else
\expandafter\raise\csname @df@\romannumeral#1@posy\endcsname
\hb@xt@\z@
{%
\expandafter\kern \csname @df@\romannumeral#1@posx\endcsname
\@dodynamicbox{#1}\hss
}%
\ignorespaces
\fi
\fi
}

```

\@putdynamicbbox Bounding box:

```

\newcommand*{\@putdynamicbbox}[1]{%
\@df@chckifthispg{#1}%
\if@notthiscol
\else
\@killglue
\if@twoside
\ifodd\c@page
\expandafter\raise\csname @df@\romannumeral#1@posy\endcsname
\hb@xt@\z@
{%
\expandafter\kern \csname @df@\romannumeral#1@posx\endcsname
\@dodynamicbbox{#1}\hss
}%
\ignorespaces
\else
\expandafter\raise\csname @df@\romannumeral#1@eveny\endcsname
\hb@xt@\z@
{%
\expandafter\kern \csname @df@\romannumeral#1@evenx\endcsname
\@dodynamicbbox{#1}\hss
}%
\ignorespaces
\fi
\else
\expandafter\raise\csname @df@\romannumeral#1@posy\endcsname
\hb@xt@\z@
{%
\expandafter\kern \csname @df@\romannumeral#1@posx\endcsname

```

```

        \@dodynamicbbox{#1}\hss
      }%
      \ignorespaces
    \fi
  \fi
}

```

`\@@doheader` Do standard header in the standard place.

```

\newcommand*{\@@doheader}{%
  \setlength\@ff@tmp@y{\textheight}%
  \addtolength{\@ff@tmp@y}{\headsep}%
  \def\ff@col{}%
  \def\ff@txtcol{}%
  \def\ff@backcol{{none}}%
  \@ff@box{0pt}{\@ff@tmp@y}{\makebox[0pt][l]{\@dothehead}}%
}

```

`\@@dofooter` Do standard footer in the standard place.

```

\newcommand*{\@@dofooter}{%
  \setlength\@ff@tmp@y{-\footskip}%
  \def\ff@col{}%
  \def\ff@txtcol{}%
  \def\ff@backcol{{none}}%
  \@ff@box{0pt}{\@ff@tmp@y}{\makebox[0pt][l]{\@dothefoot}}%
}

```

`\@s@tfr@mes` This is a modified version of the way the picture environment works:

```

\newcommand{\@s@tfr@mes}[1]{%
  {%
    \@picht\textheight
    \setbox\@picbox\hb@xt@ \textwidth
    \bgroup
      \hbox
        \bgroup
          #1\relax
        \egroup
      \hss
    \egroup
    \ht\@picbox\@picht
    \dp\@picbox\z@
    \mbox{\box\@picbox}%
  }%
}

```

`\@ff@doallflowframes` Puts all the flow frames defined on the current page

```

\newcommand*{\@ff@doallflowframes}{%
  \@colN=0\relax
  \whiledo{\@colN<\c@maxflow}%
  {%

```

```

        \advance\@colN by 1\relax
        \@putcolbox{\@colN}%
    }%
}

```

`\@ff@doallflowframesbbox` Flow frame bounding boxes:

```

\newcommand*{\@ff@doallflowframesbbox}{%
    \@colN=0\relax
    \whiledo{\@colN<\c@maxflow}%
    {%
        \advance\@colN by 1\relax
        \@putcolbbox{\@colN}%
    }%
}

```

`\@ff@doallstatics` Puts all static frames defined on the current page

```

\newcommand*{\@ff@doallstatics}{%
    \@colN=0\relax
    \whiledo{\@colN<\c@maxstatic}%
    {%
        \advance\@colN by 1\relax
        \@putstaticbox{\@colN}%
    }%
}

```

`\@ff@doallstaticsbbox` Static frame bounding boxes:

```

\newcommand*{\@ff@doallstaticsbbox}{%
    \@colN=0\relax
    \whiledo{\@colN<\c@maxstatic}%
    {%
        \advance\@colN by 1\relax
        \@putstaticbbox{\@colN}%
    }%
}

```

`\@ff@doalldynamics` Puts all the dynamic frames defined on the current page

```

\newcommand*{\@ff@doalldynamics}{%
    \@colN=0\relax
    \whiledo{\@colN<\c@maxdynamic}%
    {%
        \advance\@colN by 1\relax
        \@putdynamicbox{\@colN}%
    }%
}

```

`\@ff@doalldynamicsbbox` Dynamic frame bounding boxes:

```

\newcommand*{\@ff@doalldynamicsbbox}{%
    \@colN=0\relax
    \whiledo{\@colN<\c@maxdynamic}%

```

```

    {%
      \advance\@colN by 1\relax
      \@putdynamicbbox{\@colN}%
    }%
  }

\@ff@dotypeblock Draw typeblock frame if draft.
\newcommand*{\@ff@dotypeblock}{%
  \makebox[0pt][l]%
  {%
    \@fr@meifdraft[\setffdrafttypeblockcolor]%
    {%
      \vbox to \textheight{\hbox to \textwidth{}}%
    }%
    {}%
  }%
}

\@ff@do@allframes Put all frames defined on the current page.
\newlength\ffevenoffset
\newcommand*{\@ff@do@allframes}{%
  \ffevenoffset=0pt\relax
  \if@twoside
    \ifodd\c@page
    \else
      \ffevenoffset=-\oddsidemargin\relax
      \advance\ffevenoffset by \evensidemargin\relax
      \kern\ffevenoffset\relax
    \fi
  \fi
  \setlength{\@ff@tmp@x}{\textwidth}%
  \advance\@ff@tmp@x by -\ffevenoffset\relax
  \makebox[\@ff@tmp@x][l]%
  {%
    \@s@tfr@mes
    {%
      \@ff@doallstatics
      \@@doheader
      \@@dofooter
      \@ff@doallflowframes
      \@ff@doalldynamics
      \ifshowtypeblock
        \@ff@dotypeblock
      \fi
      \ifshowframebbox
        \@ff@doallstaticsbbox
        \@ff@doallflowframesbbox
        \@ff@doalldynamicsbbox
      \fi
      \ifshowmargins

```

```

        \off@drawmargins
      \fi
    }%
  }%
}

```

`\@outputdblcol` This was modified from the output routine for standard two column format. After `\@g@tnextcol`, the register `\c@curpg` contains the page that the next flow frame is on. If `\c@curpg` minus `\c@page` is greater than 1, then there is at least one page without a flow frame. These pages will have to be shipped before TeX can continue with the rest of the document.

```

\newcount\@nxtcol
\def\@outputdblcol{%
  \@nxtcol=\c@thisframe

  \c@curpg=\@ff@pages@countreg
  \@g@tnextcol{\@nxtcol}%
  \ifff@nwpg

```

Next flow frame starts on new page.

```

    \global\@firstcolumntrue
    \@setcolbox\c@thisframe
    \if@specialpage
      \global\@specialpagefalse
      \@nameuse{ps@\@specialstyle}\relax
    \fi
    \if@twoside
      \ifodd\count\z@
        \let\@thehead\@oddhead
        \let\@thefoot\@oddfoot
      \else
        \let\@thehead\@evenhead
        \let\@thefoot\@evenfoot
      \fi
    \else
      \let\@thehead\@oddhead
      \let\@thefoot\@oddfoot
    \fi
    \@begindvi
    \@dynamicthehead\@dynamicthefoot
    \vbadness=\@M
    \setbox\@outputbox\vbox{\hbox to \textwidth{\@ff@do@allframes}}%
    \@combinedblfloats
    \@outputpage

```

Shipout pages without flow frames.

```

\advance\c@curpg by -\@ff@pages@countreg\relax
\whiledo{\c@curpg>0}%
{
  \advance\c@curpg by -1\relax
}

```

```

        \setbox\@outputbox\vbox{\hbox to \textwidth{\@ff@do@allframes}}}%
        \@outputpage
    }
    \begingroup
        \@dblfloatplacement
        \@startdblcolumn
        \@whilesw \if@fcolmade \fi
            {\@outputpage \@startdblcolumn }%
    \endgroup
    \@resetst@tics
    \@resetdyn@mics
\else
Still on same page, save contents of box255
    \global\@firstcolumnfalse
    \@setcolbox\c@thisframe
    \fi
    \global\c@thisframe=\@nxtcol
    \@setcol{\c@thisframe}\relax
    \global\@colht\vsize
}

```

`\@dblfloatplacement` Modify `\@dblfloatplacement` replacing `\textheight` with `\vsize`.

```

\def\@dblfloatplacement{%
    \global\@dbltopnum\c@dbltopnumber
    \global\@dbltoproom\dbltopfraction\@colht\@textmin
    \@colht\advance\@textmin -\@dbltoproom
    \@fpmin\dblfloatpagefraction\vsize
    \@fptop \@dblfpsep \@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}{0pt}{0pt}%
}
```

Need a length to store the height of the flow frame so that it can be adjusted.

```
\newlength\columnheight
```

`\onecolumninarea` `\onecolumn` is in fact a special case of `\onecolumninarea` which sets up one flow frame in the specified area, given by bottom left corner (`⟨x⟩`, `⟨y⟩`), relative to the typeblock, with width `⟨w⟩` and height `⟨h⟩`. The only difference between `\onecolumninarea` and explicitly creating the flow frame using `\newflowframe` is the `\onecolumninarea` will adjust the vertical height 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[⟨pages⟩]{⟨w⟩}{⟨h⟩}{⟨x⟩}{⟨y⟩}.
\newcommand*{\onecolumninarea}{\@onecolumninarea}
\@onlypreamble{\onecolumninarea}
```

`\@onecolumninarea`

```
\newcommand*{\@onecolumninarea}[5][all]{%
  \setlength{\columnheight}{#3}%
  \iffvadjust
    \adjustheight{\columnheight}%
  \fi
  \@n@wflowframe[#1]{#2}{\columnheight}{#4}{#5}%
}
```

`\twocolumn` Set up two flow frames parallel to each other with a distance of `\columnsep` between them, to fill the entire typeblock (although the frames may end up marginally shorter than `\textheight` after they have been adjusted.) Again,

these commands may only be used in the preamble. Note that unlike the standard `\twocolumn` command, this one has an optional argument that indicates which pages the two flow frames should appear on. Syntax: `\twocolumn[<pages>]`.

```
\renewcommand*{\twocolumn}{\@twocolumn}
```

```
\@twocolumn
```

```
\newcommand*{\@twocolumn}[1][all]{%
  \@twocolumninarea[#1]{\textwidth}{\textheight}{0pt}{0pt}%
}
```

`\twocolumninarea` Again, `\twocolumn` is actually a special case of `\twocolumninarea`. Syntax:

```
\twocolumninarea[<pages>]{<w>}{<h>}{<x>}{<y>}.
\newcommand*{\twocolumninarea}{\@twocolumninarea}
\@onlypreamble{\twocolumninarea}
```

```
\@twocolumninarea
```

```
\newcommand*{\@twocolumninarea}[5][all]{%
  \setlength{\columnheight}{#3}%
  \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
}
```

`\Ncolumn` Again for an arbitrary number of columns (*<n>*). Syntax: `\Ncolumn[<pages>]{<n>}`.

```
\newcommand*{\Ncolumn}[2][all]{%
  \Ncolumninarea[#1]{#2}{\textwidth}{\textheight}{0pt}{0pt}%
}
```

\@onlypreamble{\Ncolumn}

\Ncolumninarea Check the number of flow frames requested, and do one of the special cases if available. Syntax:

\Ncolumninarea[$\langle pages \rangle$]{ $\langle n \rangle$ }{ $\langle w \rangle$ }{ $\langle h \rangle$ }{ $\langle x \rangle$ }{ $\langle y \rangle$ }.

```
\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!%
      }%
    {%
      It does not make much sense to ask to create 0 flow frames%
    }%
  \or
    \onecolumninarea[#1]{#3}{#4}{#5}{#6}%
  \or
    \twocolumninarea[#1]{#3}{#4}{#5}{#6}%
  \else
    \PackageError{flowfram}%
    {%
      Can't create a negative number of flow frames!%
    }%
    {%
      You have asked for \number#2 \space flow frames
      which really doesn't make sense%
    }%
  \fi
\fi
}
```

\@onlypreamble{\Ncolumninarea}

\@Ncolumninarea Set up $\langle n \rangle$ 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
}
```

```

\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 *<type>*) 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}{0pt}{0pt}%
}
\@onlypreamble{\onecolumnstop}

```

\onecolumnstop Special case for static frame. Syntax: **\onecolumnstop**[*<pages>*]{*<H>*}

```

\newcommand*{\onecolumnstop}[2][all]{%
\onecolumnstopinarea[#1]{static}{#2}{\textwidth}{\textheight}{0pt}{0pt}%
}

```

\onecolumnstop Special case for dynamic frame. Syntax: **\onecolumnstop**[*<pages>*]{*<H>*}

```

\newcommand*{\onecolumnstop}[2][all]{%
\onecolumnstopinarea[#1]{dynamic}{#2}{\textwidth}{\textheight}{0pt}{0pt}%
}

```

\newframe Create a frame of given type. Syntax:

```

\newframe[<pages>]{<type>}{<w>}{<h>}{<x>}{<y>}.

```

```

\newcommand*{\newframe}[6][all]{%
  \ifthenelse{\equal{#2}{flow}}{%
    {%
      \@nwflowframe[#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'%
  }%
}%
}

```

`\onecolumnntopinarea` Now for a specified area. Syntax:

```

\onecolumnntopinarea[⟨pages⟩]{⟨type⟩}{⟨H⟩}{⟨w⟩}{⟨h⟩}{⟨x⟩}{⟨y⟩}.
\newlength\@ff@staticH

```

```

\newcommand*{\onecolumnntopinarea}[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}%
  \@nwflowframe[#1]{#4}{\columnheight}{#6}{#7}%
}

```

```

\@onlypreamble{\onecolumnntopinarea}

```

`\onecolumnStopinarea` Special case for static frame. Syntax:

```

\onecolumnStopinarea[⟨pages⟩]{⟨H⟩}{⟨w⟩}{⟨h⟩}{⟨x⟩}{⟨y⟩}.
\newcommand*{\onecolumnStopinarea}[6][all]{%

```

```

\onecolumnDtopinarea Special case for dynamic frame. Syntax:
\onecolumnDtopinarea[pages]{H}{w}{h}{x}{y}.
\newcommand*\onecolumnDtopinarea[6][all]{%
  \onecolumnDtopinarea[#1]{dynamic}{#2}{#3}{#4}{#5}{#6}%
}

\twocolumnDtopinarea Now for two flow frames, with a single type frame above both of them. Syntax:
\twocolumnDtopinarea[pages]{type}{H}
  First the special case where the area is the entire typeblock:
  \newcommand*\twocolumnDtopinarea[3][all]{%
    \twocolumnDtopinarea[#1]{#2}{#3}{\textwidth}{\textheight}{Opt}{Opt}%
  }
  \@onlypreamble{\twocolumnDtopinarea}

\twocolumnDtopinarea Special case for static frame.
\newcommand*\twocolumnDtopinarea[2][all]{%
  \twocolumnDtopinarea[#1]{static}{#2}{\textwidth}{\textheight}{0pt}{0pt}%
}

\twocolumnDtopinarea Special case for dynamic frame.
\newcommand*\twocolumnDtopinarea[2][all]{%
  \twocolumnDtopinarea[#1]{dynamic}{#2}%
}

Now for a general area.

\twocolumnDtopinarea Syntax:
\twocolumnDtopinarea[pages]{type}{H}{w}{h}{x}{y}.
\newcommand*\twocolumnDtopinarea{\@twocolumnDtopinarea}
\newcommand*\@twocolumnDtopinarea[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}{-\vcolumnsep}%
  \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}%
}
```

`\Ncolumnntop` Similarly for an arbitrary number of flow frames. Special case where the area is the typeblock.

Syntax:

```
\Ncolumnntop[⟨pages⟩]{⟨type⟩}{⟨n⟩}{⟨H⟩}
\newcommand*{\Ncolumnntop}[4][all]{%
  \Ncolumnntopinarea[#1]{#2}{#3}{#4}{\textwidth}{\textheight}{0pt}{0pt}%
}
\@onlypreamble{\Ncolumnntop}
```

`\NcolumnStop` Special case for static frame.

```
\newcommand*{\NcolumnStop}[3][all]{%
  \Ncolumnntop[#1]{static}{#2}{#3}%
}
```

`\NcolumnDtop` Special case for dynamic frame.

```
\newcommand*{\NcolumnDtop}[3][all]{%
  \Ncolumnntop[#1]{dynamic}{#2}{#3}%
}
```

`\Ncolumnpinarea` Again test to make sure the user requested a sensible number.

```

\newcommand*{\Ncolumnpinarea}[8][all]{%
  \ifnum#3>2\relax
    \@Ncolumnpinarea[#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
      \onecolumnpinarea[#1]{#2}{#4}{#5}{#6}{#7}{#8}%
    \or
      \twocolumnpinarea[#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{\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`

```

  \ifffvadjust
    \adjustheight{\columnheight}%
  \fi

```

work out the widths of the flow frames

```
\@colN=#3\relax
\advance\@colN by -1\relax
\setlength{\columnwidth}{#5}%
\addtolength{\columnwidth}{- \@colN\columnsep}%
\divide\columnwidth by #3\relax
```

Set the x position of the first frame

```
\setlength{\@ff@tmp@x}{#7}%
\iflefttorightcolumns
\else
\addtolength{\@ff@tmp@x}{#5}%
\addtolength{\@ff@tmp@x}{-\columnwidth}%
\fi
\@colN=0\relax
\loop
\advance\@colN by 1\relax
\newflowframe[#1]{\columnwidth}{\columnheight}{\@ff@tmp@x}{#8}%
```

work out the offset for the next column

```
\iflefttorightcolumns
\addtolength{\@ff@tmp@x}{\columnwidth}%
\addtolength{\@ff@tmp@x}{\columnsep}%
\else
\addtolength{\@ff@tmp@x}{-\columnwidth}%
\addtolength{\@ff@tmp@x}{-\columnsep}%
\fi
\ifnum\@colN<#3
\repeat
}
```

`\NcolumnStopinarea` Specific case for static frame.

```
\newcommand*{\NcolumnStopinarea}[7][all]{%
\Ncolumnntopinarea[#1]{static}{#2}{#3}{#4}{#5}{#6}{#7}%
}
```

`\NcolumnDtopinarea` Specific case for dynamic frame.

```
\newcommand*{\NcolumnDtopinarea}[7][all]{%
\Ncolumnntopinarea[#1]{dynamic}{#2}{#3}{#4}{#5}{#6}{#7}%
}
```

Now the same kind of thing but with the $\langle type \rangle$ frame at the bottom. Firstly, a single flow frame with a $\langle type \rangle$ frame below it.

`\onecolumnbottom` Syntax:

```
\onecolumnbottom[ $\langle pages \rangle$ ]{ $\langle type \rangle$ }{ $\langle H \rangle$ }
\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[$\langle pages \rangle$]{ $\langle type \rangle$ }{ $\langle H \rangle$ }{ $\langle w \rangle$ }{ $\langle h \rangle$ }{ $\langle x \rangle$ }{ $\langle y \rangle$ },`

where $\langle H \rangle$ is the $\langle type \rangle$ frame's height. The area is defined by bottom left co-ordinates ($\langle x \rangle$, $\langle y \rangle$) width $\langle w \rangle$, and height $\langle h \rangle$.

```
\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}{0pt}{0pt}%
}
\@onlypreamble{\Ncolumnbottom}

```

`\NcolumnSbottom` Special case for static frame.

```

\newcommand*{\NcolumnSbottom}[3][all]{%
\Ncolumnbottom[#1]{static}{#2}{#3}%
}

```

`\NcolumnDbottom` Special case for dynamic frame.

```

\newcommand*{\NcolumnDbottom}[3][all]{%
\Ncolumnbottom[#1]{dynamic}{#2}{#3}%
}

```

`\Ncolumnbottominarea` Again check the user has requested a sensible number.

```

\newcommand*{\Ncolumnbottominarea}[8][all]{%
\ifnum#3>2\relax
\@Ncolumnbottominarea[#1]{#2}{#3}{#4}{#5}{#6}{#7}{#8}%
}

```

```

\else
  \ifcase#3\relax
    \PackageError{flowfram}{%
      You have requested 0 flowframes!}%
    It does not make much sense to ask to create 0 flow frames}
  \or
    \onecolumnbottominarea[#1]{#2}{#4}{#5}{#6}{#7}{#8}%
  \or
    \twocolumnbottominarea[#1]{#2}{#4}{#5}{#6}{#7}{#8}%
  \else
    \PackageError{flowfram}{%
      {%
        Can't create a negative number of flow frames!%
      }%
      {%
        You have asked for \number#3 \space flow frames
        which really doesn't make sense%
      }%
    }%
  \fi
\fi
}
\@onlypreamble{\Ncolumnbottominarea}

```

`\NcolumnSbottominarea` An arbitrary number of columns with a static frame underneath them all, filling the specified area.

```

\newcommand*{\NcolumnSbottominarea}[8][all]{%
  \setlength{\@ff@staticH}{#4}%
work out height of the flow frames
  \setlength{\columnheight}{#6}%
  \addtolength{\columnheight}{-\@ff@staticH}%
  \addtolength{\columnheight}{-\vcolumnsep}%
adjust the flow frame height so that it is a multiple of \baselineskip
  \ifffvadjust
    \adjustheight{\columnheight}%
  \fi
  \newframe[#1]{#2}{#5}{\@ff@staticH}{#7}{#8}%
work out the y offset of the flow frames
  \setlength{\@ff@tmp@y}{#6}%
  \addtolength{\@ff@tmp@y}{-\columnheight}%
  \addtolength{\@ff@tmp@y}{#8}%
work out the widths of the flow frames
  \@colN=#3\relax
  \advance\@colN by -1\relax
  \setlength{\columnwidth}{#5}%
  \addtolength{\columnwidth}{-\@colN\columnsep}%
  \divide\columnwidth by #3\relax

```

Set the x offset of the first frame.

```
\setlength{\@ff@tmp@x}{#7}%
\iflefttorightcolumns
\else
  \addtolength{\@ff@tmp@x}{#5}%
  \addtolength{\@ff@tmp@x}{-\columnwidth}%
\fi
\@colN=0\relax
\loop
  \advance\@colN by 1\relax
  \newflowframe[#1]{\columnwidth}{\columnheight}%
    {\@ff@tmp@x}{\@ff@tmp@y}%
```

work out the offset for the next column

```
\iflefttorightcolumns
  \addtolength{\@ff@tmp@x}{\columnwidth}%
  \addtolength{\@ff@tmp@x}{\columnsep}%
\else
  \addtolength{\@ff@tmp@x}{-\columnwidth}%
  \addtolength{\@ff@tmp@x}{-\columnsep}%
\fi
\ifnum\@colN<#3
\repeat
}
```

`\NcolumnSbottominarea` Specific case for static frame.

```
\newcommand*\NcolumnSbottominarea[1][all]{%
  \Ncolumnbottominarea[#1]{static}%
}
```

`\NcolumnDbottominarea` Specific case for dynamic frame.

```
\newcommand*\NcolumnDbottominarea[1][all]{%
  \Ncolumnbottominarea[#1]{dynamic}%
}
```

`\adjustheight` Given a height #1 (a length), adjust it so that it is a multiple of `\baselineskip`.

```
\newcount\@ff@adjh
\newcommand*\adjustheight[1]{%
```

convert to an integer

```
\@ff@adjh=#1\relax
\divide\@ff@adjh by \baselineskip\relax
#1=\baselineskip\relax
\multiply#1 by \@ff@adjh\relax
}
```

`\adjustcolsep` Adjust the value of `\columnsep` so that the margins will fit between columns.

```
\newcommand*\adjustcolsep{%
  \multiply\columnsep by 2\relax
  \addtolength{\columnsep}{\marginparwidth}%
}
```

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][0pt]{\@@vtwotonebottom{#1}{\paperheight}}
```

`\vtwotonebottom` Vertical two tone effect along the bottom of the page, of height $\langle H \rangle$. Syntax:

`\vtwotonebottom[⟨pages⟩][⟨xoffset⟩]{⟨H⟩}{⟨W1⟩}{⟨C1⟩}{⟨L1⟩}{⟨W2⟩}{⟨C2⟩}{⟨L2⟩}` 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}%
}

\Ntone Similarly for N colours. Syntax:
\Ntone[<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[][]{

# }{}{}{}{}{}


```

```

\newcommand*{\htwotone}[1][all]{%
  \def\ff@pages{#1}%
  \@htwotone
}

\@htwotone
  \newcommand*{\@htwotone}[1][0pt]{\@htwotoneleft{#1}{\paperwidth}}

\@htwotoneleft This is all done in much the same way as the vertical strips.
  \newcommand*{\@htwotoneleft}[8]{%
    \computeleftedgeodd{\ff@tmp@x}%
    \if@twoside
      \computeleftedgeeven{\ff@tmp@x@even}%
    \else
      \setlength{\ff@tmp@x@even}{\ff@tmp@x}%
    \fi
    \computebottomedge{\ff@tmp@y}%
    \addtolength{\ff@tmp@y}{#1}%
    \@nextthband{\ff@pages}{#2}{#3}{#4}{#5}%
    \@nextthband{\ff@pages}{#2}{#6}{#7}{#8}%
  }

  \@onlypreamble{\htwotone}

\htwotoneleft Two tone horizontal strips along left border Syntax: \htwotoneleft[pages][y offset][width]{H1}{C1}{L1}{H2}{C2}{L2}
  \newcommand*{\htwotoneleft}[1][all]{%
    \def\ff@pages{#1}%
    \@htwotoneleft
  }
  \@onlypreamble{\htwotoneleft}

\@htwotoneleft
  \newcommand*{\@htwotoneleft}[2][0pt]{\@htwotoneleft{#1}{#2}}

\htwotoneright Two tone horizontal strips along right border
  \newcommand*{\htwotoneright}[1][all]{%
    \def\ff@pages{#1}%
    \@htwotoneright
  }
  \@onlypreamble{\htwotoneright}

\@htwotoneright
  \newcommand*{\@htwotoneright}[2][0pt]{\@htwotoneright{#1}{#2}}

\@htwotoneright
  \newcommand*{\@htwotoneright}[8]{%
    \computerightedgeodd{\ff@tmp@x}%
    \if@twoside

```

```

        \computerightedgeeven{\@ff@tmp@x@even}%
    \else
        \setlength{\@ff@tmp@x@even}{\@ff@tmp@x}%
    \fi
    \computebottomedge{\@ff@tmp@y}%
    \addtolength{\@ff@tmp@y}{#1}%
    \addtolength{\@ff@tmp@x}{-#2}%
    \addtolength{\@ff@tmp@x@even}{-#2}%
    \@nextband{\ff@pages}{#2}{#3}{#4}{#5}%
    \@nextband{\ff@pages}{#2}{#6}{#7}{#8}%
}

\hNtone Now for  $\langle N \rangle$  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\csname @sf@romannumeral\c@maxstatic @evenx\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 `⟨y top⟩` or continue below the lowest point by `⟨y bottom⟩`.

`\ffcolumnseprule` This has changed in v1.09. Define `\ffcolumnseprule` and use instead of `\columnseprule`

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

`\ffruleddeclarations` 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*{\ffruleddeclarations}{}%
```

`\insertvrule` Determine whether or not the starred version is being used.

```
\newcommand*{\insertvrule}{\ifstar\@insertvrule\@insertvrule}
```

`\@insertvrule` Two optional arguments required.

```
\newcommand*{\@insertvrule}[1][0pt]{%
  \@ifnextchar[{\@insertvrule[#1]}{\@insertvrule[#1][0pt]}}%
```

Need some lengths:

```
\newlength\@ff@left@x
\newlength\@ff@left@y
\newlength\@ff@left@evenx
\newlength\@ff@left@eveny
\newlength\@ff@left@width
\newlength\@ff@left@height
```

`\@@insertvrule` Arguments all accounted for. Convert the frame type into a number to make life easier

```
\def\@@insertvrule[#1][#2]#3#4#5#6{%
  \ifthenelse{\equal{#3}{flow}}{%
    {%
      \def\@ff@type@i{1}%
    }%
    {%
      \ifthenelse{\equal{#3}{static}}{%
        {%
          \def\@ff@type@i{2}%
        }%
        {%
          \ifthenelse{\equal{#3}{dynamic}}{%
            {%
              \def\@ff@type@i{3}%
            }%
            {%

```

```

\PackageError{flowfram}%
{Unknown frame type '#3'}%
{%
  Available frame types are: 'flow', 'static'
  or 'dynamic'%
}%
}%
}%
\ifthenelse{\equal{#5}{flow}}%
{%
  \def\@ff@type@ii{1}%
}%
{%
  \ifthenelse{\equal{#5}{static}}%
  {%
    \def\@ff@type@ii{2}%
  }%
  {%
    \ifthenelse{\equal{#5}{dynamic}}%
    {%
      \def\@ff@type@ii{3}%
    }%
    {%
      \PackageError{flowfram}%
      {Unknown frame type '#5'}%
      {%
        Available frame types are: 'flow', 'static'
        or 'dynamic'%
      }%
    }%
  }%
}%
\@@insert@vrule{#1}{#2}{\@ff@type@i}{#4}{\@ff@type@ii}{#6}%
}

```

\@@insert@vrule Insert a new static frame between the two specified frames. Check to make sure which one is on the left and which one is on the right. Syntax:

```

\@@insert@vrule{<y top>}{<y bottom>}{<type ID>}{<IDN>}{<type ID>}{<IDN>}.
\newcommand*{\@@insert@vrule}[6]{%
  \@ff@getdim{#3}{#4}%
  \setlength{\@ff@left@x}{\ffareax}%
  \setlength{\@ff@left@y}{\ffareay}%
  \setlength{\@ff@left@width}{\ffareawidth}%
  \setlength{\@ff@left@height}{\ffareaheight}%
  \@ff@getdim{#5}{#6}%
  \ifnum\@ff@left@x>\ffareax\relax
    \@ff@swaplen{\@ff@left@x}{\ffareax}%
    \@ff@swaplen{\@ff@left@y}{\ffareax}%
    \@ff@swaplen{\@ff@left@evenx}{\ffareaevenx}%
  \fi
}

```

```

\ff@swaplen{\ff@left@eveny}{\ffareaevenx}%
\ff@swaplen{\ff@left@width}{\ffareawidth}%
\ff@swaplen{\ff@left@height}{\ffareaheight}%
\fi
\setlength{\ff@tmp@x}{\ff@left@x}
\addtolength{\ff@tmp@x}{\ff@left@width}%
\setlength{\ff@staticW}{\ffareax}%
\addtolength{\ff@staticW}{-\ff@tmp@x}%
\setlength{\ff@staticH}{\ff@left@y}%
\addtolength{\ff@staticH}{\ff@left@height}%
\setlength{\ff@tmp@y}{\ffareay}%
\addtolength{\ff@tmp@y}{\ffareaheight}%
\ifnum\ff@tmp@y>\ff@staticH
  \setlength{\ff@staticH}{\ff@tmp@y}%
\fi
\ifnum\ff@left@y<\ffareay\relax
  \setlength{\ff@tmp@y}{\ff@left@y}%
\else
  \setlength{\ff@tmp@y}{\ffareay}%
\fi
\addtolength{\ff@staticH}{-\ff@tmp@y}%
\newstaticframe{\ff@staticW}{\ff@staticH}%
  {\ff@tmp@x}{\ff@tmp@y}%
\addtolength{\ff@staticH}{#1}%
\addtolength{\ff@staticH}{#2}%
\setstaticcontents{\c@maxstatic}{%
\ffruledeclarations
\ffvrule{#2}{\ffcolumnseprule}{\ff@staticH}}%
\ifcase#3\relax
  \or \edef\ff@pages{\csname @ff@pages@\romannumeral#4\endcsname}%
  \or \edef\ff@pages{\csname @sf@pages@\romannumeral#4\endcsname}%
  \or \edef\ff@pages{\csname @df@pages@\romannumeral#4\endcsname}%
\fi
\setstaticframe{\c@maxstatic}{pages=\ff@pages}%

```

Check the difference between odd and even page co-ordinates and shift new frame in same direction. (Assumes the two original frames stay in the same relative position.)

```

\addtolength{\ff@tmp@x}{\ff@left@evenx}%
\addtolength{\ff@tmp@x}{-\ff@left@x}%
\addtolength{\ff@tmp@y}{\ff@left@eveny}%
\addtolength{\ff@tmp@y}{-\ff@left@y}%
\setstaticframe{\c@maxstatic}{evenx=\ff@tmp@x,eveny=\ff@tmp@y}%
}

```

```

\ffvrule \ffvrule{<offset>}{<width>}{<height>}
  Draws the rule for \insertvrule
\newcommand*{\ffvrule}[3]{%
  \hfill \rule[-#1]{#2}{#3}\hfill\mbox{}%
}

```



```

\@sininsertvrule Starred version. Two optional arguments required.
\newcommand*{\@sininsertvrule}[1][0pt]{%
  \@ifnextchar[{\@sininsertvrule[#1]}{\@sininsertvrule[#1][0pt]}}%
}

\@@sininsertvrule Find out the frame types and their IDN.
\def\@@sininsertvrule[#1][#2]#3#4#5#6{%
  \ifthenelse{\equal{#3}{flow}}{%
    {%
      \def\@ff@type@i{1}%
      \@flowframeid{#4}%
      \@ff@tmpN=\ff@id
    }%
    {%
      \ifthenelse{\equal{#3}{static}}{%
        {%
          \def\@ff@type@i{2}\@staticframeid{#4}\@ff@tmpN=\ff@id
        }%
        {%
          \ifthenelse{\equal{#3}{dynamic}}{%
            {%
              \def\@ff@type@i{3}%
              \@dynamicframeid{#4}%
              \@ff@tmpN=\ff@id
            }%
            {%
              \PackageError{flowfram}%
                {Unknown frame type '#3'}%
                {%
                  Available frame types are: 'flow', 'static'
                  or 'dynamic'%
                }%
            }%
          }%
        }%
      }%
    }%
  }%
  \ifthenelse{\equal{#5}{flow}}{%
    {%
      \def\@ff@type@ii{1}\@flowframeid{#6}%
    }%
    {%
      \ifthenelse{\equal{#5}{static}}{%
        {%
          \def\@ff@type@ii{2}%
          \@staticframeid{#6}%
        }%
        {%
          \ifthenelse{\equal{#5}{dynamic}}{%
            {%
              \def\@ff@type@ii{3}%
              \@dynamicframeid{#6}%
            }%
          }%
        }%
      }%
    }%
  }%
}

```

```

}%
{%
  \PackageError{flowfram}%
  {Unknown frame type '#5'}%
  {%
    Available frame types are: 'flow', 'static'
    or 'dynamic'%
  }%
}%
}%
}%
\insert@vrule{#1}{#2}{\@ff@type@i}{\@ff@tmpN}%
  {\@ff@type@ii}{\ff@id}%
}

```

`\inserthrule` Now for a horizontal rule. Syntax similar to `\insertvrule`. Determine whether or not the starred version is being used.

```
\newcommand*{\inserthrule}{\@ifstar\@sinserthrule\@inserthrule}
```

`\@inserthrule` Two optional arguments required.

```

\newcommand*{\@inserthrule}[1][0pt]{%
  \@ifnextchar[{\@@inserthrule[#1]}{\@@inserthrule[#1][0pt]}}%
}

```

`\@@inserthrule` Arguments all accounted for. Convert the frame type into a number to make life easier

```

\def\@@inserthrule[#1][#2]#3#4#5#6{%
  \ifthenelse{\equal{#3}{flow}}{%
    {%
      \def\@ff@type@i{1}%
    }%
    {%
      \ifthenelse{\equal{#3}{static}}{%
        {%
          \def\@ff@type@i{2}%
        }%
        {%
          \ifthenelse{\equal{#3}{dynamic}}{%
            {%
              \def\@ff@type@i{3}}%
            {%
              \PackageError{flowfram}%
              {Unknown frame type '#3'}%
              {%
                Available frame types are: 'flow', 'static'
                or 'dynamic'%
              }%
            }%
          }%
        }%
      }%
    }%
  }%
}

```

```

}%
\ifthenelse{\equal{#5}{flow}}%
{%
  \def\@ff@type@ii{1}%
}%
{%
  \ifthenelse{\equal{#5}{static}}%
  {%
    \def\@ff@type@ii{2}%
  }%
  {%
    \ifthenelse{\equal{#5}{dynamic}}%
    {%
      \def\@ff@type@ii{3}%
    }%
    {%
      \PackageError{flowfram}%
      {Unknown frame type '5'}%
      {%
        Available frame types are: 'flow', 'static'
        or 'dynamic'%
      }%
    }%
  }%
}%
\@insert@hrule{#1}{#2}{\@ff@type@i}{#4}{\@ff@type@ii}{#6}%
}

```

\@insert@hrule Insert a new static frame between the two specified frames. Check to make sure which one is on the top and which one is on the bottom. Syntax:

\@insert@hrule{<*x left*>}{<*x right*>}{<*type ID*>}{<*IDN*>}{<*type ID*>}{<*IDN*>}.

```

\newcommand*{\@insert@hrule}[6]{%
  \@ff@getdim{#3}{#4}%
  \setlength{\@ff@left@x}{\ffareax}%
  \setlength{\@ff@left@y}{\ffareay}%
  \setlength{\@ff@left@width}{\ffareawidth}%
  \setlength{\@ff@left@height}{\ffareaheight}%
  \@ff@getdim{#5}{#6}%
  \ifnum\@ff@left@y>\ffareay\relax
    \@ff@swaplen{\@ff@left@x}{\ffareax}%
    \@ff@swaplen{\@ff@left@y}{\ffareay}%
    \@ff@swaplen{\@ff@left@width}{\ffareawidth}%
    \@ff@swaplen{\@ff@left@height}{\ffareaheight}%
  \fi
  \setlength{\@ff@tmp@y}{\@ff@left@y}%
  \addtolength{\@ff@tmp@y}{\@ff@left@height}%
  \setlength{\@ff@staticH}{\ffareay}%
  \addtolength{\@ff@staticH}{-\@ff@tmp@y}%
  \setlength{\@ff@staticW}{\@ff@left@x}%
  \addtolength{\@ff@staticW}{\@ff@left@width}%
}

```

```

\setlength{\@ff@tmp@x}{\ffareax}%
\addtolength{\@ff@tmp@x}{\ffareawidth}%
\ifnum\@ff@tmp@x>\@ff@staticW\relax
  \setlength{\@ff@staticW}{\@ff@tmp@x}%
\fi
\ifnum\@ff@left@x<\ffareax\relax
  \setlength{\@ff@tmp@x}{\@ff@left@x}%
\else
  \setlength{\@ff@tmp@x}{\ffareax}%
\fi
\addtolength{\@ff@staticW}{-\@ff@tmp@x}%
\newstaticframe{\@ff@staticW}{\@ff@staticH}%
  {\@ff@tmp@x}{\@ff@tmp@y}%
\addtolength{\@ff@staticW}{#1}%
\addtolength{\@ff@staticW}{#2}%
\setstaticcontents{\c@maxstatic}%
{%
  \ffruledeclarations
  \ffhrule{#1}{\@ff@staticW}{\ffcolumnseprule}%
}%
\ifcase#3\relax
  \or \edef\@ff@pages{\csname @ff@pages@romannumeral#4\endcsname}%
  \or \edef\@ff@pages{\csname @sf@pages@romannumeral#4\endcsname}%
  \or \edef\@ff@pages{\csname @df@pages@romannumeral#4\endcsname}%
\fi
\setstaticframe{\c@maxstatic}{pages=\@ff@pages}%
\addtolength{\@ff@tmp@x}{\@ff@left@evenx}%
\addtolength{\@ff@tmp@x}{-\@ff@left@x}%
\addtolength{\@ff@tmp@y}{\@ff@left@eveny}%
\addtolength{\@ff@tmp@y}{-\@ff@left@y}%
\setstaticframe{\c@maxstatic}{evenx=\@ff@tmp@x,eveny=\@ff@tmp@y}%
}

```

`\ffhrule` `\ffhrule{<offset>}{<width>}{<height>}`

Draws the rule for `\inserthrule`

```

\newcommand*{\ffhrule}[3]{%
  \hspace*{#1}\rule{#2}{#3}%
}

```

`\@sinserthrule` Starred version. Two optional arguments required.

```

\newcommand*{\@sinserthrule}[1][Opt]{%
  \@ifnextchar[{\@@sinserthrule[#1]}{\@sinserthrule[#1][Opt]}%
}

```

`\@@sinserthrule` Find out the frame types and their IDN.

```

\def\@@sinserthrule[#1][#2]#3#4#5#6{%
  \ifthenelse{\equal{#3}{flow}}{%
    {%
      \def\@ff@type@i{1}%
    }
  }
}

```

```

\@flowframeid{#4}%
\@ff@tmpN=\ff@id
}%
{%
\ifthenelse{\equal{#3}{static}}%
{%
\def\@ff@type@i{2}%
\@staticframeid{#4}%
\@ff@tmpN=\ff@id
}%
{%
\ifthenelse{\equal{#3}{dynamic}}%
{%
\def\@ff@type@i{3}%
\@dynamicframeid{#4}%
\@ff@tmpN=\ff@id
}%
{%
\PackageError{flowfram}%
{Unknown frame type '#3'}%
{%
Available frame types are: 'flow', 'static'
or 'dynamic'%
}%
}%
}%
\ifthenelse{\equal{#5}{flow}}%
{%
\def\@ff@type@ii{1}%
\@flowframeid{#6}%
}%
{%
\ifthenelse{\equal{#5}{static}}%
{%
\def\@ff@type@ii{2}%
\@staticframeid{#6}%
}%
{%
\ifthenelse{\equal{#5}{dynamic}}%
{%
\def\@ff@type@ii{3}%
\@dynamicframeid{#6}%
}%
{%
\PackageError{flowfram}%
{Unknown frame type '#5'}%
{%
Available frame types are: 'flow', 'static'
or 'dynamic'%
}
}
}
}

```

```

    }%
  }%
} %
} %
@@insert@hrule{#1}{#2}{\@ff@type@i}{\@ff@tmpN}%
{\@ff@type@ii}{\ff@id}%
}

```

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[ $\langle y \text{ offset} \rangle$ ]{ $\langle height \rangle$ }[ $\langle sec \text{ type} \rangle$ ].
```

First check to see if there is a second optional argument.

```
\newcommand*{\makethumbtabs}[2][Opt]{%
  \@ifnextchar[%
    {\@makethumbtabs[#1]{#2}}%
    {%
      \@makethumbtabs[#1]{#2}[\defaultthumbtabtype]%
    }%
}
```

`\@makethumbtabs` Now all arguments are known, first redefine the appropriate sectioning command, then input the `ttb` file, and create the thumbtabs.

```
\def\@makethumbtabs[#1]#2[#3]{%
  \ifundefined{#3}%
  {%
    \PackageError{flowfram}%
    {%
      Unknown section type '#3'%
    }%
    {}%
  }%
  {%
    \renewcommand{\@ttb@type}{#3}%
    \ifthenelse{\equal{#3}{chapter}}{%
      {%
        \@makethumbchapter
      }%
      {%
        \ifthenelse{\equal{#3}{part}}{%
          {\@makethumbpart}%
          {%
            \@makethumbsection{#3}%
          }%
        }%
      }%
    }%
    \starttoc{ttb}%
    \dothumbtabs{#1}{#2}%
  }
}
```

`\@makethumbchapter` If thumbtabs correspond to chapters, redefine `\@chapter` so that each unstarred chapter writes an entry to the `.ttb` file.

```
\newcommand{\@makethumbchapter}{%
  \let\@ttb@old@chapter\@chapter
  \def\@chapter[##1]##2{%
```



```

\@ttb@old@chapter[##1]{##2}%
\addtocontents{ttb}{\protect\thumbtab
  {\thepage}{\thechapter}{##1}{chapter.\thechapter}}%
\@afterheading
}%
}

```

\@makethumbpart For parts in books or reports, the thumbtab needs to be saved after the part counter has been incremented, but before the page break so that the page number and part numbers are correct. If `\@endpart` is not defined, then the document class probably does not start a new page after `\part`. (This can't be guaranteed for non standard class files, but there's nothing that can be done about that.) If this happens, just redefine `\@part`, and hope for the best.

```

\newcommand{\@makethumbpart}{%
  \let\@ttb@old@part\@part
  \ifundefined{\@endpart}%
  {%
    \def\@part[##1]##2{%
      \@ttb@old@part[##1]{##2}%
      \addtocontents{ttb}{\protect\thumbtab
        {\thepage}{\thepart}{##1}{part.\thepage}}%
      \@afterheading
    }%
  }%
  {%
    \let\@ttb@old@endpart\@endpart
    \def\@part[##1]##2{%
      \def\@parttitle{##1}%
      \@ttb@old@part[##1]{##2}%
    }%
    \def\@endpart{%
      \addtocontents{ttb}%
      {%
        \protect\thumbtab{\thepage}%
        {\thepart}{\@parttitle}{part.\thepage}%
      }%
      \@ttb@old@endpart
    }%
  }%
}

```

\@makethumbsection Thumbtabs defined for one of the remaining standard sectioning commands. Since these commands use `\@startsection`, it is necessary to redefine `\@sect` to add the thumbtab information to the .ttb file.

```

\newcommand*{\@makethumbsection}[1]{%
  \let\@ttb@old@sect=\@sect
  \def\@sect##1##2##3##4##5##6[##7]##8{%
    \@ttb@old@sect{##1}{##2}{##3}{##4}{##5}{##6}[##7]{##8}%
    \ifthenelse{\equal{##1}{#1}}%

```

```

    {%
      \addtocontents{ttb}%
      {%
        \protect\thumbtab{\thepage}{\csname the#1\endcsname}%
        {##7}{#1.\csname the#1\endcsname}%
      }%
      \@afterheading
    }%
  }%
}

```

`\thumbtab` The thumbtab file, `.ttb`, will have a series of `\thumbtab` commands, when this file is read in, just store the information for now.

```

\newcommand{\thumbtab}[4]{%
  \stepcounter{maxthumbtabs}%
  \expandafter
    \gdef\csname thumbtab@pages@\romannumeral\c@maxthumbtabs\endcsname{#1}%
  \expandafter
    \gdef\csname thumbtab@num@\romannumeral\c@maxthumbtabs\endcsname{#2}%
  \expandafter
    \gdef\csname thumbtab@title@\romannumeral\c@maxthumbtabs\endcsname{#3}%
  \expandafter
    \gdef\csname thumbtab@link@\romannumeral\c@maxthumbtabs\endcsname{#4}%
}

```

`\@dothumbtabs` Once the thumbtab information has been read in and stored in the thumbtab macros, create the thumbtabs using this information. First need to work out the page ranges between each thumbtab. If the following thumbtab starts on the same page as the previous one, leave the page 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@subsptext` which should be set to empty before use.

```

\def\@flf@subsp#1 #2{%
  \expandafter\flf@ta\expandafter{\@flf@subsptext}%
  \flf@tb{#1}%
  \edef\@flf@subsptext{\the\flf@ta\the\flf@tb}%
  \def\@flf@tmp{#2}%
  \ifx\@flf@tmp\@nnil
    \let\@flf@donextsubsp=\@gobble
  \else
    \expandafter\flf@ta\expandafter{\@flf@subsptext}%
    \edef\@flf@subsptext{\the\flf@ta\noexpand\space}%
    \let\@flf@donextsubsp=\@flf@subsp
  \fi
  \@flf@donextsubsp{#2}%
}

```

`\@ttb@stack` Stack letters vertically. Any spaces first need to be substituted with `\space`, otherwise they will be ignored.

```

\newcommand{\@ttb@stack}[1]{%
  \def\@flf@subsptext{}%
  \expandafter\@flf@subsp#1 \@nil\relax
  \begin{tabular}{l}%
    \expandafter\@ttb@stack\@flf@subsptext\@nil\relax
  \end{tabular}%
}

```

`\@@ttb@stack`

```

\def\@@ttb@stack#1#2{%
  \def\@flf@tmp{#1}%
  \ifx\@flf@tmp\@nnil
    \let\flf@next\relax
  \else
    #1\\%
    \def\@flf@tmp{#2}%
    \ifx\@nnil#2\relax
      \let\flf@next\@gobble
    \else
      \let\flf@next\@@ttb@stack
    \fi
  \fi
  \flf@next{#2}%
}

```

`\@greyscale` Count register to compute the grey scale.

```

\newcount\@greyscale

```

`\@@dothumbtabs` Once the page range have been sorted, create the dynamic frames associated with each thumbtab. Thumbtabs will initially have a grey background, but this can be changed by the user. Each thumbtab is given an IDL `thumbtab<n>` where `<n>` is the index of the thumbtab (starting from 1 for the topmost thumbtab.) Each frame in the thumbtab index is given an IDL `thumbtabindex<n>`, where `<n>` is as before.

```
\newcommand{\@@dothumbtabs}[2]{%
  \setlength{\@ff@tmp@y}{\textheight}%
  \addtolength{\@ff@tmp@y}{-#2}%
  \addtolength{\@ff@tmp@y}{-#1}%
  \computerightedgeodd{\@ff@tmp@x}%
  \addtolength{\@ff@tmp@x}{-thumbtabwidth}%
  \computeleftedgeeven{\@ff@tmp@x@even}%
  \@ff@tmpN=0\relax
  \whiledo{\@ff@tmpN<\c@maxthumbtabs}%
  {%
    \advance\@ff@tmpN by 1\relax
    \@greyscale=\@ff@tmpN\relax
    \multiply\@greyscale by 60\relax
    \divide\@greyscale by \c@maxthumbtabs
    \advance\@greyscale by 25\relax
    \edef\@ff@greyscale{0.\number\@greyscale}%
```

Thumbtab

```
\newdynamicframe[none]{thumbtabwidth}{#2}%
  {\@ff@tmp@x}{\@ff@tmp@y}[thumbtab\number\@ff@tmpN]%
\expandafter\global\expandafter
  \setlength\csname @df@romannumeral\c@maxdynamic @evenx\endcsname
  {\@ff@tmp@x@even}%
```

set the contents of the dynamic frame

```
\ifthenelse{\boolean{ttb@title}\and\boolean{ttb@num}}{%
  {%
    \expandafter
    \xdef\csname @dynamicframe@romannumeral\c@maxdynamic\endcsname{%
      \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{%
      \noexpandthumbtabformat
      {%
        \csname thumbtab@title@romannumeral\@ff@tmpN\endcsname
```

```

    }%
    {#2}%
  }%
\fi
\if@ttb@num
  \expandafter
  \xdef\csname @dynamicframe@\romannumeral\c@maxdynamic\endcsname{%
    \noexpand\thumbtabformat
    {%
      \csname thumbtab@num@\romannumeral\@ff@tmpN\endcsname
    }%
    {#2}%
  }%
\fi
}%
\expandafter
\xdef\csname @df@backcol@\romannumeral\c@maxdynamic\endcsname
{[gray]{\@ff@greyscale}}

```

Thumbtab index

```

\newdynamicframe[none]{\thumbtabwidth}{#2}%
  {\@ff@tmp@x}{\@ff@tmp@y}[thumbtabindex\number\@ff@tmpN]%
\expandafter\global\expandafter
\setlength\csname @df@\romannumeral\c@maxdynamic @evenx\endcsname
{\@ff@tmp@x@even}%
\expandafter

```

set the contents of the dynamic frame

```

\ifthenelse{\boolean{@ttb@title}\and\boolean{@ttb@num}}{%
  {%
    \expandafter
    \xdef\csname @dynamicframe@\romannumeral\c@maxdynamic\endcsname{%
      \noexpand\thumbtabindexformat
      {%
        \csname thumbtab@link@\romannumeral\@ff@tmpN\endcsname
      }%
      {%
        \csname thumbtab@num@\romannumeral\@ff@tmpN\endcsname\
        \csname thumbtab@title@\romannumeral\@ff@tmpN\endcsname
      }%
      {#2}%
    }%
  }%
}%
{%
  \if@ttb@title
    \expandafter
    \xdef\csname @dynamicframe@\romannumeral\c@maxdynamic\endcsname{%
      \noexpand\thumbtabindexformat
      {%
        \csname thumbtab@link@\romannumeral\@ff@tmpN\endcsname
      }%
    }%
  }%
}

```

```

        {%
        \csname thumbtab@title@\romannumeral\@ff@tmpN\endcsname
        }%
        {#2}%
    }%
\fi
\if@ttb@num
\expandafter
\edef\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
\edef\csname @df@backcol@\romannumeral\c@maxdynamic\endcsname
{[gray]{\@ff@greyscale}}
\addtolength{\@ff@tmp@y}{-#2}%
}%
}%

```

`\enablethumbtabs` Enable thumbtabs. Once the IDN is obtained for the first thumbtab, the rest can be found by incrementing the number by 2 (the frames in between correspond to the thumbtab index.)

```

\newcommand*\enablethumbtabs{%
\ifnum\c@maxthumbtabs>0\relax
\@ff@tmpN=0\relax
\@dynamicframeid{thumbtab1}%
\whiledo{\@ff@tmpN<\c@maxthumbtabs}%
{%
\advance\@ff@tmpN by 1\relax
thumbtab
\edef\@ff@pages{\csname thumbtab@pages@\romannumeral\@ff@tmpN\endcsname}%
\@@setdynamicframe{\ff@id}{pages=\@ff@pages}%
\advance\ff@id by 2\relax
}%
\else
\PackageWarning{flowfram}{No thumb tabs defined}%
\fi
}

```

`\disablethumbtabs` Disable all thumbtabs.

```

\newcommand*\disablethumbtabs{%

```

```

\ifnum\c@maxthumbtabs>0\relax
\@ff@tmpN=0\relax
\@dynamicframeid{thumbtab1}%
\whiledo{\@ff@tmpN<\c@maxthumbtabs}%
{%
\advance\@ff@tmpN by 1\relax

```

Thumbtab:

```

\expandafter\xdef\csname @df@pages@\romannumeral\ff@id\endcsname
{none}%
\advance\ff@id by 1\relax

```

Thumbtab index:

```

\expandafter\xdef\csname @df@pages@\romannumeral\ff@id\endcsname
{none}%
\advance\ff@id by 1\relax
}%
\fi
}

```

`\thumbtabindex` Show thumbtab index on current page. The `\@ff@doafter` bit circumvents the problem of duplicate page numbers, as the table of contents is quite frequently on page i while the first chapter starts on page 1.

```

\newcommand*{\thumbtabindex}{%
\ifnum\c@maxthumbtabs>0\relax
\@ff@tmpN=0\relax
\@dynamicframeid{thumbtabindex1}%
\whiledo{\@ff@tmpN<\c@maxthumbtabs}%
{%
\advance\@ff@tmpN by 1\relax
\expandafter
\xdef\csname @df@pages@\romannumeral\ff@id\endcsname{\c@page}%
\edef\@ff@doafter{%
\noexpand\afterpage
{%
\noexpand\setdynamicframe{\number\ff@id}{pages=none}%
}%
}%
\@ff@doafter
\advance\ff@id by 2\relax
}%
\fi
}

```

`\setthumbtab` Modify the settings for all the thumbtabs (including thumbtab index). Since the thumbtabs are dynamic frames you could just use `\setdynamicframe`, however, the thumbtabs will not be generated on the first run, as there will be no information in the ttb file, so `\setdynamicframe` would generate an error. `\setthumbtab` will only give a warning message if it can not find the thumbtab.

The argument #1 is the index of the thumbtab (starting from 1), the second argument #2 is the frame settings.

```
\newcommand{\setthumbtab}[2]{%
  \ifthenelse{\equal{#1}{all}}{%
    {%
      \ff@tmpN=0\relax
      \whiledo{\ff@tmpN<\c@maxthumbtabs}%
        {%
          \advance\ff@tmpN by 1\relax
          \@setthumbtab{\ff@tmpN}{#2}%
        }%
      }%
    }%
  }%
  \for\@ttb@id:=#1\do{\@setthumbtab{\@ttb@id}{#2}}%
}%
}
```

`\@setthumbtab` Set individual thumbtab and its index tab.

```
\newcommand{\@setthumbtab}[2]{%
Check if this thumbtab exists
  \ifthenelse{(\c@maxthumbtabs<#1\)\ or \(#1<1\)}%
    {%
      \PackageWarning{flowfram}%
      {%
        Can't find thumbtab number '#1', ttb file may not be
        up-to-date%
      }%
    }%
  }%
  {%
    \@dynamicframeid{thumbtab\number#1}%
    \@setdynamicframe{\ff@id}{#2}%
    \@dynamicframeid{thumbtabindex\number#1}%
    \@setdynamicframe{\ff@id}{#2}%
  }%
}
```

`\setthumbtabindex` Only change settings for the thumbtab index. This can take a comma-separated number list.

```
\newcommand{\setthumbtabindex}[2]{%
  \ifthenelse{\equal{#1}{all}}{%
    {%
      \ff@tmpN=0\relax
      \whiledo{\ff@tmpN<\c@maxthumbtabs}%
        {%
          \advance\ff@tmpN by 1\relax
          \@setthumbtabindex{\ff@tmpN}{#2}%
        }%
      }%
    }%
  }%
```

```

\@for\@ttb@id:=#1\do{\@setthumbtabindex{\@ttb@id}{#2}}%
}%
}

```

`\@setthumbtabindex` Change setting for individual thumbtab index entry.

```

\newcommand{\@setthumbtabindex}[2]{%
Check if this thumbtab exists
\ifthenelse{(\@c@maxthumbtabs<#1\)\ \or \(#1<1\)}%
{%
\PackageWarning{flowfram}%
{%
Can't find thumbtab number '\number#1',
ttb file may not be up-to-date%
}%
}%
{%
\@dynamicframeid{thumbtabindex\number#1}%
\@setdynamicframe{\ff@id}{#2}%
}%
}

```

`\tocandthumbtabindex` Do both the table of contents and the thumbtab index

```

\newcommand*\@tocandthumbtabindex{%
\aligntoctrue
\tableofcontents
\thumbtabindex
\aligntocfalse
}

```

4.12 Minitocs

`\@ttb@minitotype` Sectioning type for the minitoc, by default it is the same as the thumbtabs

```

\newcommand*\@ttb@minitotype{\@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@files
    \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@minitoc@type}%
  {%
    \@ttb@minitoc@level=6\relax
  }%
  {%
    \expandafter\@ttb@minitoc@level\expandafter
    =\csname \@ttb@\@ttb@minitoc@type @level\endcsname
  }%
  \newread\@ttb@toc
  \openin\@ttb@toc=#1\relax
  \c@maxtocunits=0\relax
  \c@maxminitoc=0\relax
  \whiledo{\not\boolean{eof}}{\@ttb@toc}%
  {%
```

```

\read\@ttb@toc to\tocline
\@addtotoclist{\tocline}{\c@maxtocunits}%
}%
\closein\@ttb@toc
}

```

`\@addtotoclist` Before each line is added to the contents list, it is first checked to see whether the line starts with `\contentsline`. If it does, then check to see if the sectioning type corresponds to the thumbtab level. If it does, then start a new list. There will be `\c@maxtocunits` lists, each one corresponding to each thumbtab group. In addition, each contents line needs to be added to the minitoclists, but only if the sectioning type level is greater than `\@ttb@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
  \@storetoctrue
  \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]{%
    \@sappendeddynamic{#1}{\@dominitoc{##1}}%
  }%
}
```

`\@appendminitocdf` Unstarred version

```
\newcommand*{\@appendminitocdf}[1]{%
  \renewcommand{\@dominitoc}[1]{%
    \@appendeddynamic{#1}{\@dominitoc{##1}}%
  }%
}
```

`\@printtoc` Do the table of contents, which has been stored in `\c@maxtocunits` macros. (or possibly `\c@maxtocunits + 1`, if information was added before the first group—which corresponds to `\@colN=0`.)

```
\newcommand*{\@printtoc}{%
  \@colN=0\relax
  \csname @toc@\romannumeral\@colN\endcsname
  \whiledo{\@colN<\c@maxtocunits}%
  {%
    \advance\@colN by 1\relax
    \csname @toc@\romannumeral\@colN\endcsname
  }%
}
```

`\@printalignedtoc` Print the table of contents so that each unit is has vertical height the same as the height of the thumbtabs. Note that you may have to adjust the vertical offset of the thumbtabs (in `\makethumbtabs`) in order to make them correctly aligned.

```
\newcommand{\@printalignedtoc}{%
  \@ff@tmpN=0\relax
  \@ifundefined{@toc@\romannumeral\@ff@tmpN}%
  {}%
  {%
    \csname @toc@\romannumeral\@ff@tmpN\endcsname
    \par\noindent\hrulefill
  }%
}
```

```

\whiledo{\@ff@tmpN<\c@maxtocunits}%
{%
  \advance\@ff@tmpN by 1\relax
  \ifnum\@ff@tmpN>\c@maxthumbtabs
    \csname @toc@\romannumeral\@ff@tmpN\endcsname
  \else
    \@dynamicframeid{thumbtabindex\number\@ff@tmpN}%
    \expandafter\expandafter\expandafter
      \@ff@getstaticpos\csname @df@dim@\romannumeral\ff@id\endcsname
    \vbox to \@ff@tmp@y
    {%
      \noindent\parbox{\linewidth}%
      {%
        \csname @toc@\romannumeral\@ff@tmpN\endcsname
      }%
      \vfill
      \par\noindent\hrulefill
    }%
  \fi
}%
}

```

`\enableminitoc` Make mini tocs appear at the start of given sectional unit.

```

\newcounter{minitoc}
\newif\if@minitoc
\@minitocfalse

\newcommand*{\enableminitoc}[1][\@ttb@type]{%
  \@minitoctrue
  \setcounter{minitoc}{0}%
  \ifundefined{#1}%
  {%
    \PackageError{flowfram}{Sectioning type ‘#1’ not defined}{}%
  }%
  {%
    \renewcommand{\@ttb@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 299, 708
 \@flf@subsp: new 299, 708
 \@storefileconts: uses
 \PackageInfo instead of
 \typeout 312, 715
 \@ttb@stack: spaces now
 replaced with \space . 299, 708

1.11 – 2008/06/27

General: added relative location
 commands 121, 583
 \@@setdynamicframe: fixed bug
 in clear key 107, 566
 removed unwanted space 107, 566
 \@@setflowframe: removed
 unwanted spaces 81, 539
 \@ff@getevendim: new ... 118, 580
 \@getflowbounds: fixed bug:
 changed frame id from 2 to 1
 (thanks to Lutz Goldmann for
 pointing it out) 120, 582
 \@getflowevenbounds: new 120, 582
 \@getstaticbounds: fixed bug:
 changed frame id from 1 to 2
 (thanks to Lutz Goldmann for
 pointing it out) 119, 581
 \@getstaticevenbounds:
 new 119, 582
 \@setcol: added displayedframe
 increment 158, 603
 \@sgetflowevenbounds:
 new 120, 582
 \checkifframeabove: new 121, 583
 \checkifframebelow: new 124, 585
 \checkifframeleft: new . 126, 587
 \checkifframeright: new 128, 589
 displayedframe: displayedframe
 counter added 31, 525
 \evencheckifframeabove:
 new 130, 591
 \evencheckifframebelow:
 new 131, 593
 \evencheckifframeleft:
 new 132, 594
 \evencheckifframeright:
 new 133, 596

\FFabove: new 135, 598
 \FFaboveleft: new 134, 598
 \FFaboveright: new 135, 598
 \FFbelow: new 135, 598
 \FFbelowleft: new 135, 598
 \FFbelowright: new 135, 598
 \ffhrule: new 283, 700
 \FFleft: new 135, 598
 \FFoverlap: new 135, 598
 \ffruledclarations: new 276, 694
 \ffvrule: new 280, 696
 \getdynamicevenbounds: new . 121
 \getdynamicid: new 105, 563
 \getdynamiclabel: new .. 105, 563
 \getflowevenbounds: new 120, 582
 \getflowid: new 80, 532
 \getflowlabel: new 80, 532
 \getstaticevenbounds:
 new 119, 582
 \getstaticid: new 85, 548
 \getstaticlabel: new 85, 548
 \labelflow: new 32, 525
 \labelflowidn: new 31, 525
 \oddcheckifframeabove:
 new 123, 584
 \oddcheckifframebelow:
 new 125, 586
 \oddcheckifframeleft:
 new 127, 588
 \oddcheckifframeright:
 new 129, 590
 \relativeframelocation:
 new 135, 598
 \reldynamicloc: new ... 144, 600
 \relflowloc: new 145, 601
 \relstaticloc: new 144, 600

1.12 – 2009/11/25

\iflefttorightcolumns:
 new 15, 523

1.14 – 2012-11-10

General: added page exclusion list 46
 changed color to boolean
 option 29
 new 27
 now loading xfor and
 etoolbox 3, 518

now loading xkeyval instead of keyval	518	replaced \c@page with \@ff@pages@countreg .	208, 209
\@newdynamicframe: added page exclusion list	562	\@newflowframe: added page exclusion list	530
\@newstaticframe: added page exclusion list	547	\@df@chckifthispg: added optional argument	180, 651
replaced \c@page with \@ff@pages@countreg .	181, 651	\@ff@checkifmoreframes: added check for exclusion list	175, 609
replaced \c@page with \@ff@pages@countreg .	175, 610	\@ff@checkpages: replaced \c@page with \@ff@pages@countreg .	176, 611
\@ff@disablecolor: new . .	15, 523	\@ff@enablecolor: new . . .	14, 522
\@ff@output@adjustframes: new	177, 613	\@ff@pages@countreg: new	12, 522
\@g@tnextcol: replaced \c@page with \@ff@pages@countreg .	178, 649	\@outputdblcol: replaced \c@page with \@ff@pages@countreg	670
\@outputpage: added absolute page increment . . .	607	\@putcolbbox: replaced \c@page with \@ff@pages@countreg .	186, 654
\@putcolbox: added check for non-void box	184, 653	replaced \c@page with \@ff@pages@countreg .	184, 653
\@putmarginbox: replaced \c@page with \@ff@pages@countreg .	190, 658	\@setcol: changed check to use tolerance rather than testing if hsize and columnwidth are equal	158, 603
\@sf@chckifthispg: added optional argument	180, 651	replaced \c@page with \@ff@pages@countreg .	180, 651
absolute page: new	12, 522	color: changed to boolean option	522
\dynamicaddexclusion: new	108, 569	\dynamicsetexclusion: new	108, 569
\dynamicsetpagelist: new	108, 569	\dynamicswitchoffnext: new	223, 627
\dynamicswitchoffnextodd: new	224, 628	\dynamicswitchoffnextoddonly: new	230, 636
\dynamicswitchoffnextonly: new	230, 635	\dynamicswitchonnext: new	221, 625
\dynamicswitchonnextodd: new	222, 626	\dynamicswitchonnextoddonly: new	227, 632
\dynamicswitchonnextonly: new	226, 630	\ffaddtoadjustframeshook: new	177, 648
\ffprechapterhook: new . .	20, 524	\fftolerance: new	158, 603
\finishthispage: change \c@page to \@ff@pages@countreg . .	75, 527	\flf@doifverbose: new	6, 520
\flf@message: new	5, 520	\flowaddexclusion: new . .	82, 543
\flowsetexclusion: new . .	82, 542	\flowsetpagelist: new . . .	82, 542
\flowswitchoffnext: new	212, 615	\flowswitchoffnextodd: new	213, 616
\flowswitchoffnextoddonly: new	220, 624	\flowswitchoffnextonly: new	219, 623
\flowswitchonnext: new .	210, 613	\flowswitchonnextodd: new	211, 614
\flowswitchonnextoddonly: new	217, 620		

2.0 – 2025-11-24	
\flowswitchonnexonly:	General: no longer loading xfor,
new 215, 619	ifthen, xkeyval 3
\framebreak: made change to	\@@@ff@getrange: removed . . . 177
switch global 75, 526	\@@@dofooter: renamed 205
\if@ttb@rotate: changed to	\@@@doheader: renamed 205
boolean key 10	\@@@dominitoc: renamed 318
\newpage: Modified definition of	\@@dynamicframeswapcoords:
\newpage 76, 527	removed 108
pages: new 522	\@@ff@chckifthispg: renamed . 179
\reldynamicloc: removed	\@@ff@getrange: removed . . . 177
unwanted spaces 144, 600	\@@hNtone: replaced 273
\relflowloc: removed unwanted	\@@hNtoneright: replaced . . . 274
spaces 145, 601	\@@htwotoneleft: replaced . . . 272
\relstaticloc: removed	\@@htwotoneright: replaced . . 273
unwanted spaces 144, 600	\@@insert@vrule: replaced . . . 278
rotate: changed to boolean key 520	\@@insert@rule: removed . . . 281
\staticaddexclusion: new 99, 558	\@@insert@vrule: removed . . . 278
\staticsetexclusion: new 99, 558	\@@newdynamicframe: renamed . 104
\staticsetpagelist: new . 99, 558	\@@newstaticframe: renamed . . 84
\staticswitchoffnext:	\@@setdynamicframe: renamed . 107
new 233, 639	\@@setflowframe: renamed . . . 81
\staticswitchoffnextodd:	\@@setstaticframe: renamed . . 87
new 234, 640	\@@sininsert@rule: removed . . 284
\staticswitchoffnextoddoneonly:	\@@sininsert@vrule: removed . . 280
new 241, 647	\@@staticframeswapcoords:
\staticswitchoffnextonly:	removed 100
new 240, 647	\@@storefileconts: removed . . 313
\staticswitchonnex: new 231, 636	\@@ttb@stack: removed 299
\staticswitchonnexodd:	\@@vNtone: renamed 270
new 232, 637	\@@vNtonetop: renamed 271
\staticswitchonnexoddoneonly:	\@@vtwotonetop: replaced . . . 269
new 238, 643	\@Ncolumnbottominarea:
\staticswitchonnexonly:	replaced 265
new 237, 642	\@Ncolumninarea: replaced . . . 248
thumbtabs: new 521	\@Ncolumnnotinarea: replaced . 257
verbose: new 520	\@addtotoclist: replaced . . . 314
1.15 – 2014-05-15	\@appenddynamic: renamed . . . 114
\@ff@getshape: added check for	\@appendminitocdf: removed . . 320
\Shapepar 555	\@beginstaticcontents:
\@setcol: set \linewidth as well	renamed 96
as \textwidth 159, 604	\@checkcontentsline: removed 316
1.16 – 2014-06-04	\@checkifframeabove: renamed 122
\@@newdynamicframe: added	\@checkifframebelow: renamed 124
‘hide’ and ‘hidethis’	\@checkifframeleft: renamed . 127
attributes 562	\@checkifframeright: renamed 129
\@setcol: Removed assignment	\@computeeffarea: removed . . 117
of \textwidth 159, 604	\@dblfloatplacement: removed
1.17 – 2014-09-30	redefinition 210
\@setcol: added check for	\@df@chckifthispg: replaced . 180
list 159, 604	

\@df@checkuniqueidl: removed	105	\@ff@checkifmoreframes:	
\@df@setoffset: renamed	189	renamed	174
\@doclearpage: no longer		\@ff@checknumrange: removed	176
redefining	159	\@ff@checkpages: removed	176
\@docolbbox: renamed	183	\@ff@checkthispage: removed	176
\@docolbox: renamed	181	\@ff@checkuniqueidl: removed	80
\@dodfparbox: renamed	197	\@ff@doallframes: renamed	207
\@dodynamicbbox: renamed	201	\@ff@doalldynamics: renamed	207
\@dodynamicbox: renamed	199	\@ff@doalldynamicsbbox:	
\@dominitoc: renamed	318	renamed	207
\@dostaticbbox: renamed	193	\@ff@doallflowframes:	
\@dothehead: added tagging		renamed	206
socket	161	\@ff@doallflowframesbbox:	
\@dothumbtabs: replaced	296	renamed	206
\@dynamicchap: replaced	291	\@ff@doallstatics: renamed	206
\@dynamicframeswapcoords:		\@ff@doallstaticsbbox:	
renamed	109	renamed	206
\@dynamicswitchoffnext:		\@ff@dotypeblock: renamed	207
removed	224	\@ff@drawmargins: renamed	191
\@dynamicswitchoffnextodd:		\@ff@enablecolor: removed	14
removed	226	\@ff@fbox: renamed	183
\@dynamicswitchoffnextoddone:		\@ff@getdim: renamed	117
removed	231	\@ff@getrange: renamed	
\@dynamicswitchoffnextonly:		\@ff@getrange command	70
removed	230	\@ff@getrangegreater:	
\@dynamicswitchonnext:		removed	177
removed	222	\@ff@getrangeless: removed	177
\@dynamicswitchonnextodd:		\@ff@getshape: renamed	94
removed	223	\@ff@getstaticpos: removed	191
\@dynamicswitchonnextoddone:		\@ff@heading: removed	88
removed	230	\@ff@left@evenx: replaced	277
\@dynamicswitchonnextonly:		\@ff@left@eveny: replaced	278
removed	227	\@ff@left@height: replaced	278
\@dynamicctok: removed	109	\@ff@left@width: replaced	278
\@endstaticcontents: renamed	98	\@ff@left@x: replaced	277
\@evencheckifframeabove:		\@ff@left@y: replaced	277
renamed	131	\@ff@output@adjustframes:	
\@evencheckifframebelow:		renamed	177
renamed	132	\@ff@pages@countreg: renamed	12
\@evencheckifframeleft:		\@ff@parshape: renamed	88
renamed	133	\@ff@s@t@Ovalboxoffset:	
\@evencheckifframeright:		renamed	187
renamed	134	\@ff@s@t@defaultoffset:	
\@ff@getrange: renamed		renamed	187
\@ff@getrange	71	\@ff@s@t@Ovalboxoffset:	
\@ff@addtolist: removed	114	renamed	187
\@ff@adjh: replaced	267	\@ff@sectionhead: removed	88
\@ff@box: renamed	183	\@ff@setoffset: renamed	188
\@ff@chckifthispg: renamed	179	\@ff@setsecthead: renamed	94
		\@ff@swaplen: renamed	117

\@ffdraft: renamed	4	\@inserthrule: removed	281
\@ffnodraft: renamed	4	\@insertvrule: removed	277
\@flf@checkcontinued:		\@empty: removed	110
removed	110	\@makecol: removed redefinition	174
\@flf@endxdynamiccontents:		\@makeminitocchapter:	
removed	110	removed	322
\@flf@find@end: removed	110	\@makeminitocpart: removed	322
\@flf@get@body: removed	109	\@makeminitocsection:	
\@flf@getcontargs: removed	110	removed	322
\@flf@subsp: removed	299	\@makethumbchapter: removed	294
\@flowfram@checkrelativeloc:		\@makethumbpart: removed	294
new	142	\@makethumbsection: removed	294
\@flowfram@saverrelativeloc:		\@makethumbtabs: replaced	293
new	139	\@newdynamicframe: removed	104
\@flowframe@warn@rerun: new	142	\@newflowframe: removed	79
\@flowframeid: renamed	80	\@newstaticframe: removed	84
\@flowframeswapcoords:		\@newdynamicframe: removed	104
removed	83	\@newflowframe: removed	79
\@flowswitchoffnext: removed	213	\@newstaticframe: removed	84
\@flowswitchoffnextodd:		\@nextthNband: renamed	275
removed	215	\@nextthband: replaced	274
\@flowswitchoffnextoddonly:		\@nexttvNband: renamed	270
removed	221	\@nexttvband: renamed	269
\@flowswitchoffnextonly:		\@oddcheckifframeabove:	
removed	220	renamed	123
\@flowswitchonnext: removed	211	\@oddcheckifframebelow:	
\@flowswitchonnextodd:		renamed	126
removed	212	\@oddcheckifframeleft:	
\@flowswitchonnextoddonly:		renamed	128
removed	219	\@oddcheckifframeright:	
\@flowswitchonnextonly:		renamed	130
removed	217	\@onecolumn: replaced	243
\@g@tnextcol: renamed	177	\@onecolumninarea: replaced	243
\@getdynamicbounds: renamed	121	\@opcol: include mark	
\@getdynamicevenbounds:		structures	162
renamed	121	\@printalignedtoc: replaced	320
\@getflowbounds: renamed	120	\@printtoc: replaced	320
\@getflowevenbounds: renamed	120	\@putcolbbox: renamed	186
\@getframeid: renamed	48	\@putcolbox: renamed	184
\@getstaticbounds: renamed	119	\@putdynamicbbox: renamed	203
\@getstaticevenbounds:		\@putdynamicbox: renamed	201
renamed	119	\@putmarginbox: renamed	190
\@gettype: removed	316	\@putstaticbbox: removed	195
\@greyscale: replaced	299	\@putstaticbox: renamed	193
\@hNtone: removed	273	\@relativeframelocation:	
\@hNtoneleft: removed	274	renamed	136
\@hNtoneright: removed	274	\@reldynamicloc: removed	144
\@htwotone: removed	272	\@relflowloc: removed	145
\@htwotoneleft: removed	272	\@relstaticloc: removed	145
\@htwotoneright: removed	273	\@resetdyn@mics: renamed	197

\@resetst@tics: renamed	196	\@sevencheckifframeleft:	
\@s@ff@heading: removed	88	renamed	133
\@s@tdynamicframeid: removed	105	\@sevencheckifframeright:	
\@s@tffttextcol: renamed	14	renamed	134
\@s@tflowframeid: removed . . .	80	\@sf@chckifthispg: replaced . .	180
\@s@tfr@mes: renamed	205	\@sf@checkuniqueidl: removed	85
\@s@tstaticframeid: removed . .	85	\@sf@setoffset: renamed	188
\@sappenddynamic: renamed . .	114	\@sflowframeswapcoords:	
\@sappendminitocdf: removed .	320	removed	83
\@scheckifframeabove:		\@sflowswitchhoffnext:	
renamed	122	removed	213
\@scheckifframebelow:		\@sflowswitchhoffnextodd:	
renamed	124	removed	215
\@scheckifframeleft: renamed	126	\@sflowswitchhoffnextoddone:	
\@scheckifframeright:		removed	221
renamed	128	\@sflowswitchhoffnextonly:	
\@scomputeffarea: removed . .	117	removed	220
\@sdynamicchap: removed	292	\@sflowswitchonnext: removed	211
\@sdynamicframeswapcoords:		\@sflowswitchonnextodd:	
removed	109	removed	212
\@sdynamicswitchhoffnext:		\@sflowswitchonnextoddone:	
removed	224	removed	219
\@sdynamicswitchhoffnextodd:		\@sflowswitchonnextonly:	
removed	226	removed	216
\@sdynamicswitchhoffnextoddone:		\@sggetdynamicbounds: removed	121
removed	231	\@sggetdynamicsevenbounds:	
\@sdynamicswitchhoffnextonly:		removed	121
removed	230	\@sggetflowbounds: removed . .	120
\@sdynamicswitchonnext:		\@sggetflowsevenbounds:	
removed	222	removed	120
\@sdynamicswitchonnextodd:		\@sggetstaticbounds: removed .	119
removed	223	\@sggetstaticsevenbounds:	
\@sdynamicswitchonnextoddone:		removed	119
removed	230	\@sinserthrule: removed	284
\@sdynamicswitchonnextonly:		\@sinsertvrule: removed	280
removed	227	\@snewdynamicframe: removed .	104
\@setcol: renamed	158	\@snewstaticframe: removed . .	84
\@setcolbox: renamed	181	\@soddcheckifframeabove:	
\@setdynamiccontents:		renamed	123
renamed	113	\@soddcheckifframebelow:	
\@setdynamicframe: removed . .	107	renamed	125
\@setflowframe: removed	81	\@soddcheckifframeleft:	
\@setframecol: renamed	44	renamed	127
\@setstaticframe: removed . . .	87	\@soddcheckifframeright:	
\@setthumbtab: replaced	308	renamed	130
\@setthumbtabindex: replaced .	310	\@srelativeframelocation:	
\@sevencheckifframeabove:		renamed	135
renamed	131	\@sreldynamicloc: removed . .	144
\@sevencheckifframebelow:		\@srelflowloc: removed	145
renamed	132	\@srelstaticloc: removed . . .	145

\@ssetdynamiccontents:		\@ttb@minitoclevel: replaced .	312
removed	113	\@ttb@minitoctype: replaced ..	311
\@ssetdynamicframe: removed .	107	\@ttb@paragraph@level:	
\@ssetflowframe: removed ...	81	replaced	316
\@ssetstaticframe: removed ..	87	\@ttb@part@level: replaced ...	316
\@sstaticframeswapcoords:		\@ttb@section@level: replaced	316
removed	100	\@ttb@stack: replaced	299
\@sstaticswitchoffnext:		\@ttb@storetoc: replaced	312
removed	234	\@ttb@subparagraph@level:	
\@sstaticswitchoffnextodd:		replaced	316
removed	236	\@ttb@subsection@level:	
\@sstaticswitchoffnextoddoneonly:		replaced	316
removed	242	\@ttb@subsubsection@level:	
\@sstaticswitchoffnextonly:		replaced	316
removed	241	\@ttb@type: renamed	292
\@sstaticswitchonnext:		\@twocolumn: removed	244
removed	232	\@twocolumninarea: replaced ..	245
\@sstaticswitchonnextodd:		\@twocolumntopinarea: replaced	252
removed	233	\@vNtone: removed	270
\@sstaticswitchonnextoddoneonly:		\@vNtonebottom: removed	271
removed	240	\@vNtonetop: removed	271
\@sstaticswitchonnextonly:		\@vtwotone: removed	268
removed	238	\@vtwotonebottom: removed ...	268
\@staticcontinueonframe:		\@vtwotonetop: removed	268
renamed	101	\addtocontents: added	
\@staticframeid: renamed ...	85	redefinition	89
\@staticframeswapcoords:		\appenddynamiccontents: added	
renamed	100	optional argument	113
\@staticscontinueonframe:		changed to document	
renamed	101	command	113
\@staticswitchoffnext:		\c@curpg: renamed	177
removed	234	\c@ffrangenumber: removed	177
\@staticswitchoffnextodd:		\ChapterInDynamic: new	285
removed	236	\checkifframeabove: changed to	
\@staticswitchoffnextoddoneonly:		document command	121
removed	242	\checkifframebelow: changed to	
\@staticswitchoffnextonly:		document command	124
removed	241	\checkifframeleft: changed to	
\@staticswitchonnext:		document command	126
removed	232	\checkifframeright: changed to	
\@staticswitchonnextodd:		document command	128
removed	233	\cleardoublepage: changed to	
\@staticswitchonnextoddoneonly:		document command	76
removed	240	\cleartoevenpage: new	76
\@staticswitchonnextonly:		\columnheight: removed	158
removed	238	\computebottomedge: changed to	
\@storefilecontents: replaced ...	312	document command	115
\@thisstrip: renamed	269	\computeeflowframearea: changed	
\@ttb@chapter@level: replaced	316	to document command ...	115
\@ttb@level: removed	313		

<code>\computeleftedgeeven</code> : changed to document command ...	114	<code>\dynamicswitchonnextodonly</code> : changed to document command	227
<code>\computeleftedgeodd</code> : changed to document command ...	114	<code>\dynamicswitchonnextonly</code> : changed to document command	226
<code>\computerightedgeeven</code> : changed to document command ...	115	<code>\emulateonecolumn</code> : changed to document command	147
<code>\computerightedgeodd</code> : changed to document command ...	115	<code>\emulatetwocolumn</code> : changed to document command	146
<code>\computetopedge</code> : changed to document command	114	<code>\enableinitoc</code> : changed to document command	321
<code>\continueonframe</code> : changed to document command	100	<code>\enablethumbtabs</code> : changed to document command	304
<code>\dfchaphead</code> : changed to document command	284	<code>\endxdynamiccontents</code> : removed	110
<code>\dfEvenFooterStyle</code> : new	170	<code>\evencheckifframeabove</code> : changed to document command	130
<code>\dfEvenHeaderStyle</code> : new	169	<code>\evencheckifframebelow</code> : changed to document command	131
<code>\dfOddFooterStyle</code> : new	170	<code>\evencheckifframeleft</code> : changed to document command ...	132
<code>\dfOddHeaderStyle</code> : new	169	<code>\evencheckiframeright</code> : changed to document command	133
<code>\dfswapoddeven</code> : changed to document command	108	<code>\ff@s@t@doubleboxoffset</code> : renamed	187
<code>\disablethumbtabs</code> : changed to document command	304	<code>\ffaddtoadjustframeshook</code> : changed to document command	177
<code>\dominitoc</code> : changed to document command	318	<code>\ffdefaultcontinuetext</code> : new .	102
<code>\dynamicaddexclusion</code> : changed to document command ...	108	<code>\ffdefaultdynamiccontinuetext</code> : new	102
<code>dynamiccontents</code> : changed to document environment ...	109	<code>\ffdefaultpostcontinued</code> : new	103
<code>dynamiccontents*</code> : changed to document environment ...	109	<code>\ffdefaultstaticcontinuetext</code> : new	102
<code>\dynamicframeheight</code> : new ...	106	<code>\ffdynamicpostcontinued</code> : new	103
<code>\dynamicframewidth</code> : new	106	<code>\ffevenoffset</code> : removed	207
<code>\dynamicsetexclusion</code> : changed to document command ...	108	<code>\ffpshpar</code> : deprecated	88
<code>\dynamicsetpagelist</code> : changed to document command ...	108	<code>\ffstaticpostcontinued</code> : new .	102
<code>\dynamicswitchoffnextodd</code> : changed to document command	224	<code>\finishthispage</code> : changed to document command	75
<code>\dynamicswitchoffnextodonly</code> : changed to document command	230	<code>\flf@getcontargs</code> : removed ...	110
<code>\dynamicswitchoffnextonly</code> : changed to document command	230	<code>\flf@message</code> : removed	5
<code>\dynamicswitchonnext</code> : changed to document command ...	221	<code>\flf@ta</code> : removed	114
<code>\dynamicswitchonnextodd</code> : changed to document command	222	<code>\flf@tb</code> : removed	114
		<code>\FLForgpar</code> : deprecated	88
		<code>\FLFsimpar</code> : new	88

\flowaddexclusion: changed to document command	82	\getflowevenbounds: changed to document command	120
\flowfram@doc@htmlopts: new	43	\getflowid: changed to document command	80
\flowfram@dynamicfooter: new	160	\getstaticbounds: changed to document command	119
\flowfram@dynamicheader: new	160	\getstaticevenbounds: changed to document command	119
\flowfram@footer: new	160	\getstaticid: changed to document command	85
\flowfram@header: new	159	\globalnormalmargin: changed to document command	77
\flowfram@preamble@htmlopts: new	43	\globalreversemargin: changed to document command	77
\flowfram@thepagenumber: new	12	\hNtone: changed to document command	273
\flowframeshowlayout: changed to document command	74	\hNtoneleft: changed to document command	274
\FlowFramRestoreOR: new	161	\hNtoneright: changed to document command	274
\FlowFramSetDynamicHeadFootAttributes: new	166	\htwotone: changed to document command	271
\FlowFramUnrestoreOR: new	162	\htwotoneleft: changed to document command	272
\flowsetexclusion: changed to document command	82	\htwotoneright: changed to document command	272
\flowsetpagelist: changed to document command	82	\if@contsline: removed	313
\flowswitchoffnext: changed to document command	212	\if@ff@nwpg: replace	177
\flowswitchoffnextodd: changed to document command	213	\if@minitoc: replaced	321
\flowswitchoffnextoddonly: changed to document command	220	\if@notthiscol: replaced	177
\flowswitchoffnextonly: changed to document command	219	\if@storetoc: removed	312
\flowswitchonnext: changed to document command	210	\if@ttb@num: removed	12
\flowswitchonnextodd: changed to document command	211	\if@ttb@rotate: removed	10
\flowswitchonnextoddonly: changed to document command	217	\ifdfcontinued: removed	110
\flowswitchonnextonly: changed to document command	215	\IfSavedRelativeLocationAbove: new	143
\footnotecolor: deprecated	174	\IfSavedRelativeLocationBelow: new	143
\framebreak: changed to document command	75	\IfSavedRelativeLocationEq: new	142
\getdynamicbounds: changed to document command	120	\IfSavedRelativeLocationLeft: new	143
\getdynamicevenbounds: changed to document command	121	\IfSavedRelativeLocationRight: new	144
\getdynamicid: changed to document command	105	\insertvrule: changed to document command	276
\getflowbounds: changed to document command	120	\labelflow: changed to document command	32
		\labelflowidn: changed to document command	31

<code>\makebackgroundframe</code> : changed to document command . . .	275	<code>\onecolumn</code> : changed to document command	242
<code>\makedfheaderfooter</code> : changed to document command . . .	165	<code>\onecolumnbottom</code> : changed to document command	258
<code>\makethumbtabs</code> : changed to document command	293	<code>\onecolumnbottominarea</code> : changed to document command	259
<code>\Ncolumn</code> : changed to document command	247	<code>\onecolumnDbottom</code> : changed to document command	259
<code>\Ncolumnbottom</code> : changed to document command	263	<code>\onecolumnDbottominarea</code> : changed to document command	260
<code>\Ncolumnbottominarea</code> : changed to document command . . .	264	<code>\onecolumnDtop</code> : changed to document command	249
<code>\NcolumnDbottom</code> : changed to document command	264	<code>\onecolumnDtopinarea</code> : changed to document command . . .	251
<code>\NcolumnDbottominarea</code> : changed to document command . . .	266	<code>\onecolumninarea</code> : changed to document command	243
<code>\NcolumnDtop</code> : changed to document command	255	<code>\onecolumnSbottom</code> : changed to document command	258
<code>\NcolumnDtopinarea</code> : changed to document command	258	<code>\onecolumnSbottominarea</code> : changed to document command	260
<code>\Ncolumninarea</code> : changed to document command	247	<code>\onecolumnStop</code> : changed to document command	249
<code>\NcolumnSbottom</code> : changed to document command	264	<code>\onecolumnStopinarea</code> : changed to document command . . .	251
<code>\NcolumnSbottominarea</code> : changed to document command . . .	266	<code>\onecolumnstop</code> : changed to document command	248
<code>\NcolumnStop</code> : changed to document command	255	<code>\onecolumnstopinarea</code> : changed to document command . . .	250
<code>\NcolumnStopinarea</code> : changed to document command	257	<code>\ps@ffempty</code> : new	174
<code>\Ncolumnstop</code> : changed to document command	255	<code>\ps@ffplain</code> : new	173
<code>\Ncolumnstopinarea</code> : changed to document command	256	<code>\ps@specialffempty</code> : new	174
<code>\newdynamicframe</code> : changed to document command	103	<code>\RefSavedRelativeLocation</code> : new	142
<code>\newframe</code> : changed to document command	249	<code>\relativeframelocation</code> : changed to document command	135
<code>\newstaticframe</code> : changed to document command	84	<code>\reldynamicloc</code> : changed to document command	144
<code>\NoChapterInDynamic</code> : new . . .	292	<code>\relflowloc</code> : changed to document command	145
<code>\oddcheckifframeabove</code> : changed to document command . . .	123	<code>\relstaticloc</code> : changed to document command	144
<code>\oddcheckifframebelow</code> : changed to document command . . .	125	<code>\rotateframe</code> : changed to document command	181
<code>\oddcheckifframeleft</code> : changed to document command . . .	127	<code>\SaveRelativeFrameLocation</code> : new	137
<code>\oddcheckifframeright</code> : changed to document command . . .	129		

\setalldynamicframes: changed to document command . . .	106	\staticswitchonnextodd: changed to document command	232
\setallflowframes: changed to document command	81	\staticswitchonnextoddone: changed to document command	238
\setallstaticframes: changed to document command	86	\staticswitchonnextonly: changed to document command	237
\setdynamicframe: changed to document command	106	\theFramePageCounter: new . .	12
\setframes: changed to document command	145	thisframe: replaced \thepage with \theabsolute page . . .	31
\setinitialframe: changed to document command	145	\thumbtabhyperlinkformat: new	16
\setmargin: changed to document command	78	\thumbtabindex: changed to document command and added optional argument . .	305
\setstaticcontents: changed to document command	99	\thumbtabnumtitle: new	298
\setstaticframe: changed to document command	86	\thumbtabregularformat: new .	17
\setthumbtab: changed to document command	307	\twocolumn: changed to document command	244
\setthumbtabindex: changed to document command	310	\twocolumnbottom: changed to document command	261
\SetThumbTabTitle: new	295	\twocolumnbottominarea: changed to document command	261
\staticaddeexclusion: changed to document command	99	\twocolumnDbottom: changed to document command	261
staticcontents: changed to document environment	96	\twocolumnDbottominarea: changed to document command	263
staticcontents*: changed to document environment	96	\twocolumnDtop: changed to document command	252
\staticframeheight: new	86	\twocolumnDtopinarea: changed to document command . . .	254
\staticframewidth: new	86	\twocolumninarea: changed to document command	245
\staticsetexclusion: changed to document command	99	\twocolumnSbottominarea: changed to document command	263
\staticsetpagelist: changed to document command	99	\twocolumnStop: changed to document command	252
\staticswitchoffnext: changed to document command . . .	233	\twocolumnStopinarea: changed to document command . . .	254
\staticswitchoffnextodd: changed to document command	234	\twocolumnstop: changed to document command	251
\staticswitchoffnextoddone: changed to document command	241	\twocolumnstopinarea: changed to document command . . .	252
\staticswitchoffnextonly: changed to document command	240	\vNtone: changed to document command	270
\staticswitchonnext: changed to document command . . .	231		

<code>\vNtonebottom</code> : changed to document command	271	<code>\newstaticframe</code> : corrected function name	84
<code>\vtwotone</code> : changed to document command	267	2.2 – 2026-04-25	
<code>\vtwotonebottom</code> : changed to document command	268	<code>\FlowframTkBeginScopeIfGroup</code> : new	330
<code>\vtwotonetop</code> : changed to document command	268	<code>\FlowframTkEndScopeIfGroup</code> : new	330
2.1 – 2026-02-23		<code>\flowframtkNoRagged</code> : new . . .	329
General: add dimension check . .	44	<code>\FlowframTkScopeIfGroup</code> : new	330
added frame type label constants	12	2014-06-04 – 1.16	
<code>\@flowfram@adjust@fdepth</code> : new	163	General: added ‘hide’ and ‘hidethis’ keys	39, 538
<code>\newdynamicframe</code> : corrected function name	104	<code>\@@newstaticframe</code> : added ‘hide’ and ‘hidethis’ attributes . .	547
<code>\newflowframe</code> : corrected function name	79	<code>\@putdynamicbox</code> : added check for ‘hide’ and ‘hidethis’ attributes	202, 665
<code>\newframe</code> : corrected function name	250	<code>\@putstaticbox</code> : added check for ‘hide’ and ‘hidethis’ attributes	193, 661

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>	163
<code>\(</code>	511, 512, 713, 714
<code>\)</code>	511, 512, 713, 714
<code>\@@@ff@getrange</code> <u>177</u> , 436, 437, 613, <u>613</u>	
<code>\@@dofooter</code> . . .	205, 477, 479, <u>667</u> , 669
<code>\@@doheader</code> . . .	205, 477, 479, <u>667</u> , 669
<code>\@@dominitoc</code>	318, 515, 516, 718, <u>718</u> , 719
<code>\@@dothumbtabs</code> <u>299</u> , 506, 508, 707, <u>709</u>	
<code>\@@dynamicframeswapcoords</code>	108, 403, 404, <u>569</u> , 570, 571
<code>\@@ff@chckifthispg</code>	179, 463, 464, 650, <u>650</u> , 651
<code>\@@ff@getrange</code> . . .	177, 436, 612, <u>612</u>
<code>\@@flowfram@current@pre@section@title</code>	88, 89, 91, 92, 164
<code>\@@flowfram@org@output@page</code> <u>161</u> , 163	
<code>\@@flowfram@output@page</code> . .	162, 163
<code>\@@flowframeswapcoords</code>	82, 381, 382, 543, 544
<code>\@@hNtone</code>	273, 494, 691, <u>691</u>
<code>\@@hNtoneright</code> . . .	274, 495, 692, <u>692</u>
<code>\@@htwotoneleft</code> . .	272, 493, 690, <u>690</u>
<code>\@@htwotoneright</code> .	273, 494, 690, <u>690</u>
<code>\@@if@newlist</code> .	163, 164, 432, 433, 607
<code>\@@insert@hrule</code>	281, 501, 503, 699, <u>699</u> , 702
<code>\@@insert@vrule</code>	278, 497, 499, 695, <u>695</u> , 698
<code>\@@inserthrule</code> . . .	281, 500, 698, <u>698</u>
<code>\@@insertvrule</code> . . .	278, 496, 694, <u>694</u>
<code>\@@newdynamicframe</code> <u>104</u> , 396, 561, <u>561</u>	
<code>\@@newflowframe</code> <u>79</u> , 369, 370, 529, <u>529</u>	
<code>\@@newstaticframe</code> .	84, 383, 546, <u>546</u>
<code>\@@nexthNband</code> .	275, <u>275</u> , 495, 693, <u>693</u>
<code>\@@nextvNband</code> .	270, <u>270</u> , 492, 688, <u>688</u>
<code>\@@par</code>	430, 605
<code>\@@setdynamicframe</code>	

..... 107 , 399 , 400 , 510 – 512 , 564 , 565 , 566 , 711 , 713 , 714	<code>\@checkkifframeright</code> 129 , 418 , 419 , 426 , 590 , 599
<code>\@setflowframe</code> 81 , 377 , 378 , 538 , 539 , 539	<code>\@colN</code> 367 , 368 , 371 , 372 , 376 , 377 , 382 , 384 – 387 , 395 , 397 – 400 , 404 , 434 , 435 , 462 , 471 , 473 , 474 , 478 , 479 , 482 , 483 , 486 , 487 , 489 , 490 , 506 , 516 , 526 , 527 , 531 , 532 , 538 , 539 , 544 , 547 – 550 , 560 , 562 – 565 , 570 , 571 , 609 , 610 , 649 , 659 , 662 , 663 , 667 – 669 , 674 , 675 , 680 , 684 , 685 , 706 , 707 , 719
<code>\@setfr@mecol</code> 44 , 383 , 545 , 545	
<code>\@setframecol</code> 44 , 383 , 545 , 545	
<code>\@setstaticframe</code> 87 , 386 , 387 , 549 , 550 , 550	
<code>\@sinserthrule</code> .. 284 , 502 , 700 , 700	
<code>\@sinsertvrule</code> .. 280 , 498 , 697 , 697	
<code>\@staticframeswapcoords</code> 100 , 394 , 395 , 559 , 559 , 560	
<code>\@storefileconts</code> 313 , 513 , 715 , 715	<code>\@colht</code> 159 , 162 , 164 , 430 , 431 , 434 , 481 , 604 – 606 , 609 , 671
<code>\@ttb@stack</code> 299 , 507 , 708 , 708	<code>\@colroom</code> 159 , 162 , 164 , 430 , 431 , 604 , 605
<code>\@vNtone</code> 270 , 492 , 688 , 688 , 689	<code>\@combinedblfloats</code> 209 , 480 , 670
<code>\@vNtonetop</code> 271 , 493 , 689 , 689	<code>\@combinefloats</code> 434 , 609
<code>\@vtwotonebottom</code> 490 , 491 , 686 , 687	<code>\@computeffarea</code> 117 , 408 , 409 , 577 , 577
<code>\@vtwotonetop</code> 269 , 491 , 687	<code>\@contslinefalse</code> 514 , 717
<code>\@M</code> .. 208 , 286 , 288 , 430 , 480 , 605 , 670	<code>\@contslinetrue</code> 514 , 717
<code>\@Mi</code> 75 , 368 , 527	<code>\@currentHref</code> 293
<code>\@Miv</code> 431 , 605	<code>\@currentlabel</code> 31 , 32 , 367 , 525
<code>\@NcolumnSbottominarea</code> ... 489 , 684	<code>\@currenttocentry</code> 290
<code>\@Ncolumnbottominarea</code> . 265 , 489 , 683	<code>\@currlist</code> 431 , 605
<code>\@Ncolumninarea</code> .. 248 , 482 , 674 , 674	<code>\@dblarg</code> 390 , 553
<code>\@Ncolumnntopinarea</code> 257 , 485 , 486 , 679 , 679	<code>\@dbldeferlist</code> 431 , 605 , 606
<code>\@addtotoclist</code> ... 314 , 513 , 716 , 716	<code>\@dblfloat</code> 161 , 162
<code>\@afterheading</code> 92 , 290 , 505 , 517 , 518 , 705 , 706 , 721	<code>\@dblfloatplacement</code> 209 , 210 , 431 , 480 , 481 , 606 , 671 , 671
<code>\@appenddynamic</code> 114 , 407 , 516 , 574 , 574 , 719	<code>\@dblfpbot</code> 481 , 671
<code>\@appendminitocdf</code> 320 , 515 , 516 , 719 , 719	<code>\@dblfpsep</code> 481 , 671
<code>\@auxout</code> 43 , 89 , 138	<code>\@dblfpstop</code> 481 , 671
<code>\@beginndvi</code> 164 , 480 , 670	<code>\@dbltoplist</code> 431 , 605
<code>\@beginstaticcontents</code> . 96 , 392 , 556	<code>\@dbltopnum</code> 481 , 671
<code>\@botlist</code> 431 , 605	<code>\@dbltoproom</code> 481 , 671
<code>\@captype</code> 242 , 481 , 671	<code>\@deferlist</code> 431 , 605
<code>\@cclv</code> 431 – 433 , 605 , 606 , 608	<code>\@df@chckifthispg</code> . 180 , 445 – 451 , 463 , 476 , 625 – 634 , 651 , 665 , 666
<code>\@chapapp</code> 19 , 286 , 287	<code>\@df@checkuniqueidl</code> 105 , 397 , 562 , 562
<code>\@chapter</code> 366 , 505 , 517 , 524 , 704 , 721	<code>\@df@setoffset</code> 189 , 469 , 475 , 657 , 664 , 665
<code>\@checkcontentsline</code> 316 , 513 , 514 , 716 , 717	<code>\@doclearpage</code> 159 , 431 , 605
<code>\@checkkiffameabove</code> 122 , 413 , 414 , 426 , 583 , 584 , 599	<code>\@docolbbox</code> 183 , 464 , 466 , 652 , 654 , 655
<code>\@checkkiffamebelow</code> 124 , 415 , 426 , 585 , 586 , 599	<code>\@docolbox</code> ... 181 , 464 – 466 , 651 , 654
<code>\@checkkiffameleft</code> 127 , 417 , 426 , 587 , 588 , 599	<code>\@dodfparbox</code> .. 197 , 474 , 475 , 663 , 664
	<code>\@dodynamicbbox</code> 201 , 475 , 477 , 665 , 666 , 667
	<code>\@dodynamicbox</code> 199 , 475 , 476 , 664 , 665 , 666

<code>\@dodynamictheft</code>	<code>\@evencheckifframeabove</code>
..... 161, 166, 208, 361, 131, 414, 420, 421, 584, 592
362, 432, 433, 480, 606, 608, 670	<code>\@evencheckifframebelow</code>
<code>\@dodynamicthead</code> 132, 415, 421, 422, 586, 593, 594
..... 161, 166, 208, 360,	<code>\@evencheckifframeleft</code>
361, 432, 433, 480, 606, 607, 670 133, 417, 423, 588, 595
<code>\@dominitoc</code> ... 318, 515, 516, 718, 719	<code>\@evencheckifframeright</code>
<code>\@dostaticbbox</code> 193, 471, 473, 660, 662 134, 419, 424, 425, 590, 596, 597
<code>\@dostaticbox</code> . 191, 471, 472, 660, 661	<code>\@evenfoot</code>
<code>\@dothefoot</code> ... 161, 166, 205, 361, 164, 168–171, 173, 174, 480, 670
362, 432, 433, 477, 606, 607, 667	<code>\@evenhead</code>
<code>\@dothead</code> ... 161, 166, 205, 360, 168, 170, 171, 173, 174, 480, 670
361, 432, 433, 477, 606, 607, 667	<code>\@expl@@@mark@update@dblcol@structures@@</code>
<code>\@dothumbtabs</code> . 296, 504, 506, 704, 706 162
<code>\@dynamicchap</code>	<code>\@expl@@@mark@update@singlecol@structures@@</code>
..... 291, 503, 504, 702, 702, 703 163
<code>\@dynamicframeid</code> . 105, 398, 399,	<code>\@ff@@getrange</code> 71, 436, 612, 612
404, 406, 407, 413, 433, 445–	<code>\@ff@OLD@chapter</code> 366, 367, 524
450, 452, 453, 499, 502–504,	<code>\@ff@OLD@chapter</code> 366, 524
510–512, 516, 563, 563, 565,	<code>\@ff@OLD@chapter</code> 366, 524
570, 573, 574, 582, 583, 607,	<code>\@ff@OLDmakechapterhead</code> 284, 503, 702
608, 625–628, 631, 632, 635,	<code>\@ff@OLDmakeschapterhead</code>
636, 697, 701, 703, 711–714, 720 284, 503, 702
<code>\@dynamicframeswapcoords</code>	<code>\@ff@addtolist</code>
..... 109, 404, 570, 570 114, 407, 461, 574, 574, 648
<code>\@dynamicswitchoffnext</code>	<code>\@ff@adjh</code> 267, 490, 685
..... 224, 447, 627, 628	<code>\@ff@box</code> 183, 464,
<code>\@dynamicswitchoffnextodd</code>	465, 470–472, 475, 477, 652,
..... 226, 448, 628, 629	653, 653, 659, 660, 664, 665, 667
<code>\@dynamicswitchoffnextoddonly</code> .	<code>\@ff@chkifthispg</code> 179, 368, 437–
..... 231, 452, 453, 636, 636	443, 462, 465, 466, 470, 527,
<code>\@dynamicswitchoffnextonly</code>	613–622, 649, 650, 653, 654, 658
..... 230, 452, 635, 635	<code>\@ff@check</code> 414–425, 584–597
<code>\@dynamicswitchonnext</code>	<code>\@ff@checkifmoreframes</code>
..... 222, 445, 625, 625 174, 434, 462, 609, 649
<code>\@dynamicswitchonnextodd</code>	<code>\@ff@checknumrange</code> 176, 436, 611, 611
..... 223, 446, 626, 626	<code>\@ff@checkpages</code>
<code>\@dynamicswitchonnextoddonly</code> 176, 434, 435, 609, 610, 611
..... 230, 450, 451, 632, 633	<code>\@ff@checkthispage</code> 176, 435, 611, 611
<code>\@dynamicswitchonnextonly</code>	<code>\@ff@checkuniqueidl</code> 80, 371, 531, 531
..... 227, 449, 450, 630, 631	<code>\@ff@disablecolor</code>
<code>\@dynamicctok</code> .. 109, 405, 406, 571–573 15, 365, 366, 522, 523, 524
<code>\@ehb</code> 431, 605	<code>\@ff@disablesec</code> 95, 98,
<code>\@empty</code> .. 431, 434, 462, 605, 609, 649	199, 391, 393, 474, 555, 557, 664
<code>\@emptycol</code> 431, 605	<code>\@ff@do@allframes</code> 207,
<code>\@endfortrue</code>	428, 479, 480, 601, 669, 670, 671
..... 434, 435, 462, 609, 610, 650	<code>\@ff@doafter</code> 511, 712
<code>\@endpart</code> 505, 517, 705, 721	<code>\@ff@doalldynamics</code>
<code>\@endstaticcontents</code> 207, 478, 479, 668, 669
..... 98, 392, 393, 556, 557	

\@ff@doalldynamicsbbox	\@ff@left@x
..... 207, 479, 668, 669	277, 496–498, 501, 694–696, 699, 700
\@ff@doallflowframes	\@ff@left@y
..... 206, 478, 479, 667, 669	277, 496–498, 501, 502, 694–696, 699, 700
\@ff@doallflowframesbbox	\@ff@moreframesfalse
..... 206, 478, 479, 668, 669	434, 609
\@ff@doallstatics	\@ff@moreframestrue ...
..... 206, 478, 479, 668, 669	435, 436, 611
\@ff@doallstaticsbbox	\@ff@nextid
..... 206, 478, 479, 668, 669	405, 406, 572
\@ff@doallstaticsbbox	\@ff@nextpages
..... 206, 478, 479, 668, 669 440, 441, 448, 449, 456,
\@ff@dotypeblock . 207, 479, 669, 669	457, 617–619, 628–630, 640–642
\@ff@drawmargins	\@ff@numend
..... 191, 471, 479, 659, 670	371, 377,
\@ff@enablecolor	382, 386, 387, 395, 399, 404,
..... 14, 365, 366, 522, 522, 524	436, 437, 463, 531, 539, 544,
\@ff@fbox	550, 560, 565, 571, 611–613, 650
..... 183, 464, 465, 471, 475, 652, 653, 660, 664	\@ff@numstart
\@ff@getdim	371, 377, 382, 386, 387, 395, 399,
..... 117, 410, 413, 497, 501, 578, 581–583, 695, 699	404, 436, 463, 531, 539, 544,
\@ff@getevendim	550, 560, 565, 571, 611, 612, 650
..... 118, 411, 413, 580, 582, 583	\@ff@nwpgfalse
\@ff@getrange	462, 649
..... 70, 371, 377, 382, 386, 395, 399,	\@ff@nwpgtrue
404, 436, 463, 531, 539, 544,	462, 649
550, 560, 565, 571, 611, 612, 650	\@ff@offset ...
\@ff@getrangegreater	367, 409, 410, 465–470, 526, 577, 578, 653, 655–658
..... 177, 436, 612, 612	\@ff@oldparagraph
\@ff@getrangeless 177, 436, 612, 612	391, 555
\@ff@getshape	\@ff@oldsection
..... 94, 391, 392, 474, 555, 557, 663	391, 555
\@ff@getstaticpos	\@ff@oldsubparagraph
..... 191, 387, 400, 411, 412, 471, 475, 516, 550, 566,	\@ff@oldsubsection
579, 581, 659, 660, 664, 665, 720	391, 555
\@ff@greyscale ... 508–510, 709–711	\@ff@oldsubsubsection
\@ff@heading	391, 555
..... 88, 390, 553, 554	\@ff@output@adjustframes
\@ff@id 177, 437–462, 613, 613, 614–621, 623–631, 633, 635–643, 645, 647–649
..... 377, 381, 382, 386, 394, 395, 399, 404, 409, 437–461, 538, 539, 543, 544, 549, 550, 559, 560, 565, 570, 571, 577, 613–616, 618–620, 622–648	\@ff@pages
\@ff@left@evenx	439, 442–444, 447, 450–452, 455, 456, 459, 460, 498, 501, 510, 615, 616, 620–623, 627, 628, 632–635, 639, 640, 644–647, 696, 700, 711
..... 277, 496–498, 501, 694–696, 700	\@ff@pages@countreg
\@ff@left@eveny 12, 365, 368, 428–430, 434, 435, 437–466, 470, 480, 522, 522, 527, 602, 604, 609–611, 613–649, 651, 653, 654, 658, 670
..... 278, 496–498, 501, 694, 696, 700	\@ff@parshape ..
\@ff@left@height	88, 390–393, 474, 553, 553, 554, 556, 557, 663, 664
..... 278, 496–498, 501, 694–696, 699	\@ff@pp
\@ff@left@width	371, 434, 435, 462, 463, 531, 609–611, 650
..... 278, 496, 497, 501, 694–696, 699	\@ff@prepages
 438, 439, 446, 447, 454, 455, 614, 615, 626, 627, 638, 639
	\@ff@rest
	405, 406, 572

<code>\@ff@s@t@ovalboxoffset</code>	<code>\@ffbackground</code>
..... 187 , 467–469 , 655 , 656–658	. 14 , 364 , 366 , 465 , 519 , 523 , 653
<code>\@ff@s@t@defaultoffset</code>	<code>\@ffdraft</code> 4 , 363 , 364 , 368 , 519 , 519 , 526
..... 187 , 467–469 , 655 , 656–658	<code>\@ffnodraft</code> . 4 , 363 , 364 , 519 , 519 , 520
<code>\@ff@s@t@doubleboxoffset</code>	<code>\@filef@und</code>
..... 466–469 , 655 , 656–658	513 , 715
<code>\@ff@s@t@ovalboxoffset</code>	<code>\@firstcolumnfalse</code>
..... 187 , 467–469 , 655 , 656–658	209 , 481 , 671
<code>\@ff@sectionhead</code> 88 , 390 , 391 , 553 , 555	<code>\@firstcolumntrue</code>
<code>\@ff@setoffset</code> 188 , 464 , 467 , 652 , 655	208 , 480 , 670
<code>\@ff@setsecthead</code>	<code>\@firstofone</code>
. 94 , 391 , 393 , 474 , 555 , 557 , 663	26
<code>\@ff@staticH</code>	<code>\@flf@</code>
428 , 429 , 484–	405 , 406 , 571–573
489 , 492 , 493 , 495 , 498 , 501 ,	<code>\@flf@checkcontinued</code>
602 , 676 , 677 , 679 , 681 , 682 , 110 , 405 , 571 , 572
684 , 688 , 689 , 692 , 696 , 699 , 700	<code>\@flf@donextsubsp</code>
<code>\@ff@staticW</code>	507 , 708
488 , 490–	<code>\@flf@endxdynamiccontents</code>
492 , 494 , 495 , 498 , 501 , 682 , 110 , 406 , 572 , 573
686 , 687 , 691–693 , 696 , 699 , 700	<code>\@flf@find@end</code>
<code>\@ff@swaplen</code>	110 , 405 , 572
117 ,	<code>\@flf@get@body</code> 109 , 405 , 406 , 571–573
410 , 497 , 501 , 578 , 695 , 696 , 699	<code>\@flf@getcontargs</code>
<code>\@ff@text</code>	110 , 405 , 572
405 , 572	<code>\@flf@idn</code>
<code>\@ff@tmp@ex</code>	405 , 406 , 571 , 573
367 ,	<code>\@flf@nr</code>
381 , 384 , 387 , 394 , 396 , 400 ,	520–522
403 , 409–412 , 470–472 , 475 ,	<code>\@flf@subsp</code>
479 , 481–483 , 485–498 , 501 ,	299 , 507 , 708 , 708
502 , 508 , 509 , 526 , 543 , 546 ,	<code>\@flf@subsp@text</code>
550 , 559 , 561 , 566 , 570 , 577–	507 , 708
579 , 581 , 659 , 660 , 664 , 665 ,	<code>\@flf@tmp</code>
669 , 673–675 , 678 , 680 , 682 ,	405 , 507 , 571 , 572 , 708
683 , 685–693 , 696 , 700 , 709 , 710	<code>\@flf@tmpa</code>
<code>\@ff@tmp@ex@even</code> ...	405 , 571 , 572
367 , 490–496 ,	<code>\@flf@tmpb</code>
508 , 509 , 526 , 686–693 , 709 , 710	405 , 572
<code>\@ff@tmp@y</code>	<code>\@flf@val</code>
367 , 381 , 384 , 387 , 394 , 396 ,	520–522
400 , 403 , 404 , 409–412 , 428 ,	<code>\@float</code>
429 , 433 , 470–472 , 475 , 477 ,	161 , 162
484–496 , 498 , 501 , 502 , 508–	<code>\@floatplacement</code>
510 , 516 , 526 , 543 , 546 , 550 ,	163 , 434 , 609
559 , 561 , 566 , 570 , 577–579 ,	<code>\@flowfram@adjust@fdepth</code> ..
581 , 602 , 607 , 659 , 660 , 664 ,	163 , 163
665 , 667 , 676 , 677 , 679 , 681–	<code>\@flowfram@build@page@after</code> ...
693 , 696 , 699 , 700 , 709–711 , 720
<code>\@ff@tmpN</code>	161 , 162 , 165
367 , 435 ,	<code>\@flowfram@build@page@before</code> ..
498 , 499 , 502 , 503 , 506 , 508–
512 , 516 , 526 , 610 , 697 , 698 ,	162 , 165
701 , 702 , 707 , 709–713 , 719 , 720	<code>\@flowfram@checkrelative loc</code> 142 , 142
<code>\@ff@type@i</code> 496–503 , 694 , 695 , 697–702	<code>\@flowfram@saverrelative loc</code> 138 , 139
<code>\@ff@type@ii</code>	<code>\@flowfram@set@next@vsize</code> .
497 ,	164 , 165
499–503 , 695 , 697–699 , 701 , 702	<code>\@flowfram@standard@makechapterhead</code>

	286
	<code>\@flowfram@update@col@count</code> ...

	146 , 162 , 164 , 209
	<code>\@flowframe@warn@rerun</code> 139 , 142 , 142
	<code>\@flowframeid</code>
	80 , 372 , 377 ,
	382 , 409 , 413 , 437–439 , 441 ,
	442 , 444 , 498 , 499 , 502 , 532 ,
	532 , 538 , 543 , 577 , 582 , 613–
	616 , 619 , 620 , 623 , 624 , 697 , 701
	<code>\@flowframeswapcoords</code>

	83 , 381 , 382 , 543 , 543
	<code>\@flowframtk@cr</code>
	329
	<code>\@flowframtk@def@outline lua</code> 327 , 328
	<code>\@flowframtk@def@outline pdf</code> 327 , 328

<code>\@flowframtk@def@outlinepst</code>	327 , 328	<code>\@gobble@som</code>	89 , 95
<code>\@flowframtk@outline</code>	328 , 328	<code>\@gobble@with@sphack@om</code>	164
<code>\@flowframtk@provide@textpath</code>	328	<code>\@gobble@with@sphack@som</code>	164
<code>\@flowswitchoffnext</code>	213 , 439 , 615 , 616	<code>\@gobbletwo</code>	170 , 171 , 173 , 174 , 462 , 649
<code>\@flowswitchoffnextodd</code>	215 , 439 , 440 , 616 , 617	<code>\@greyscale</code>	299 , 508 , 708 , 709
<code>\@flowswitchoffnextoddoneonly</code>	221 , 444 , 445 , 624 , 624	<code>\@hNtone</code>	273 , 494 , 691 , 691
<code>\@flowswitchoffnextonly</code>	220 , 444 , 623 , 624	<code>\@hNtoneleft</code>	274 , 494 , 691 , 691
<code>\@flowswitchonnext</code>	211 , 437 , 613 , 613	<code>\@hNtoneright</code>	274 , 494 , 495 , 692 , 692
<code>\@flowswitchonnextodd</code>	212 , 437 , 438 , 614 , 615	<code>\@height</code>	4
<code>\@flowswitchonnextoddoneonly</code>	219 , 442 , 443 , 620 , 622	<code>\@htwotone</code>	272 , 493 , 690 , 690
<code>\@flowswitchonnextonly</code>	217 , 441 , 619 , 619	<code>\@htwotoneleft</code>	272 , 493 , 690 , 690
<code>\@for</code>	371 , 377 , 381 , 382 , 386 , 394 , 395 , 399 , 404 , 409 , 434 , 435 , 437 – 463 , 511 , 512 , 531 , 538 , 539 , 543 , 544 , 549 , 550 , 559 , 560 , 565 , 570 , 571 , 577 , 609 – 611 , 613 – 616 , 618 – 620 , 622 – 629 , 631 – 633 , 635 – 645 , 647 , 648 , 650 , 713 , 714	<code>\@htwotoneright</code>	273 , 493 , 494 , 690 , 690
<code>\@fpbot</code>	481 , 671	<code>\@ifdefined</code>	398 , 564
<code>\@fpmin</code>	481 , 671	<code>\@ifnextchar</code>	44 , 371 , 383 , 384 , 397 , 405 , 496 , 498 , 500 , 502 , 504 , 531 , 545 , 547 , 562 , 572 , 694 , 697 , 698 , 700 , 704
<code>\@fpsep</code>	481 , 671	<code>\@ifpackageloaded</code>	367 , 525
<code>\@fptop</code>	481 , 671	<code>\@ifstar</code>	369 , 377 , 381 , 383 , 386 , 390 , 393 , 394 , 396 , 399 , 404 , 406 – 408 , 413 – 415 , 417 – 421 , 423 – 425 , 427 , 437 , 439 , 441 , 442 , 444 – 450 , 452 – 458 , 460 , 461 , 496 , 499 , 503 , 515 , 529 , 538 , 543 , 546 , 549 , 553 , 558 – 560 , 565 , 570 , 573 , 574 , 577 , 581 – 588 , 590 , 592 , 593 , 595 , 596 , 598 , 600 , 601 , 613 – 616 , 619 , 620 , 623 – 628 , 630 , 632 , 635 – 637 , 639 , 640 , 642 , 643 , 647 , 694 , 698 , 702 , 719
<code>\@fr@meifdraft</code>	13 , 363 , 464 , 470 , 472 , 475 , 479 , 519 , 652 , 659 , 660 , 665 , 669	<code>\@ifundefined</code>	312 , 363 , 366 , 390 , 392 , 414 – 425 , 433 , 503 – 506 , 513 , 514 , 516 – 519 , 524 , 554 , 556 , 584 – 597 , 608 , 702 – 705 , 707 , 715 , 716 , 719 – 721
<code>\@freelist</code>	434 , 609	<code>\@inserthrul</code>	281 , 499 , 698 , 698
<code>\@g@tnextcol</code>	177 , 428 , 461 , 480 , 601 , 648 , 670	<code>\@insertvrul</code>	277 , 496 , 694 , 694
<code>\@getdynamicbounds</code>	121 , 413 , 582 , 583	<code>\@killglue</code>	184 , 186 , 190 , 194 , 195 , 202 , 204 , 465 , 466 , 470 , 472 , 473 , 476 , 653 , 654 , 658 , 661 , 662 , 665 , 666
<code>\@getdynamicsevenbounds</code>	121 , 413 , 583 , 583	<code>\@kludgeins</code>	434 , 609
<code>\@getflowbounds</code>	120 , 413 , 582 , 582	<code>\@latex@warning@no@line</code>	431 , 605
<code>\@getflowevenbounds</code>	120 , 413 , 582 , 582	<code>\@latexerr</code>	431 , 605
<code>\@getframeid</code>	48 , 398 , 564	<code>\@lempty</code>	110 , 405 , 572
<code>\@getmarginpos</code>	77 , 369 , 470 , 528 , 528 , 659	<code>\@listdepth</code>	159 , 430 , 604
<code>\@getstaticbounds</code>	119 , 413 , 581 , 581	<code>\@makechapterhead</code>	284 , 286 , 287 , 503 , 702 , 703
<code>\@getstaticsevenbounds</code>	119 , 413 , 582 , 582		
<code>\@gettype</code>	316 , 513 , 515 , 716 , 717		
<code>\@gobble</code>	26 , 170 , 171 , 173 , 174 , 402 , 432 , 507 , 568 , 607 , 708		
<code>\@gobble@om</code>	89		

<code>\@makecol</code>	<code>\@nobreakfalse</code>
174, 430, 432, 433, 605, 606, 608	312, 513, 715
<code>\@makefcolumn</code>	<code>\@notthiscolfalse</code>
431, 605, 606	462, 463, 465, 650, 651, 653
<code>\@makeminitocchapter</code>	<code>\@notthiscoltrue</code>
322, 517, 720, 721	462, 463, 472, 476, 649–651, 661, 665
<code>\@makeminitocpart</code> 322, 517, 720, 721	<code>\@nxtcol</code>
<code>\@makeminitocsection</code>	428, 480, 481, 601, 602, 670, 671
322, 517, 720, 721	<code>\@oddcheckifframeabove</code>
<code>\@makeschapterhead</code>	123, 414, 584, 585
284, 288, 503, 702, 703	<code>\@oddcheckifframebelow</code>
<code>\@makespecialcolbox</code>	126, 415, 416, 586, 587
434, 609	<code>\@oddcheckifframeleft</code>
<code>\@makethumbchapter</code>	128, 417, 418, 588, 589
294, 504, 505, 704, 704	<code>\@oddcheckifframeright</code>
<code>\@makethumbpart</code> 294, 504, 505, 704, 705	130, 419, 590, 591
<code>\@makethumbsection</code>	<code>\@oddfoot</code>
294, 504, 505, 704, 705	164, 168–171, 173, 174, 480, 670
<code>\@makethumbtabs</code> .. 293, 504, 704, 704	<code>\@oddhead</code>
<code>\@maxdepth</code>	164, 167, 168, 170, 171, 173, 174, 480, 670
433, 434, 608, 609	<code>\@onecolumn</code>
<code>\@midlist</code>	243, 429, 462, 481, 602, 603, 649, 672, 672
434, 609	<code>\@onecolumninarea</code> 243, 481, 672, 672
<code>\@minitocfalse</code>	<code>\@onlypreamble</code> 166, 270–274,
516, 720	322, 369, 433, 481–489, 491–
<code>\@minitoctrue</code>	495, 517, 529, 608, 672–679,
516, 720	681–684, 686, 687, 689–692, 721
<code>\@mkboth</code>	<code>\@opcol</code>
170, 171, 173, 174	161, 162, 162, 430–432, 434, 605, 606, 609
<code>\@mparbottom</code>	<code>\@outputbox</code>
77, 163, 368, 369, 434, 528, 609	146, 164, 181, 209, 428, 432–434, 464,
<code>\@mparswitchfalse</code> 76, 368, 527	480, 601, 607–609, 651, 670, 671
<code>\@mtoc@old@chapter</code>	<code>\@outputdblcol</code>
517, 721	161–163, 208, 434, 480, 609, 670
<code>\@mtoc@old@endpart</code>	<code>\@outputpage</code>
517, 721	146, 161, 162, 163, 209, 428, 431,
<code>\@mtoc@old@part</code>	432, 480, 601, 606, 606, 670, 671
517, 721	<code>\@parboxrestore</code>
<code>\@mtoc@old@sect</code>	432, 607 505, 517, 705, 721
518, 721	<code>\@part</code>
<code>\@n@wdynamicframe</code>	505, 705
104, 395, 483, 560, 676	<code>\@picbox</code>
<code>\@n@wflowframe</code> 79, 369, 481–	477, 478, 667
485, 529, 529, 672, 673, 676, 678	<code>\@picht</code>
<code>\@n@wstaticframe</code>	477, 478, 667
84, 383, 484, 545, 546, 676	<code>\@printalignedtoc</code>
<code>\@nameuse</code>	320, 515, 516, 718, 719
163, 208, 480, 670	<code>\@printtoc</code> ... 320, 515, 516, 718, 719
<code>\@newdynamicframe</code> 104, 396, 560, 561	<code>\@putcolbbox</code> .. 186, 466, 478, 654, 668
<code>\@newflowframe</code> 79, 369, 529, 529	<code>\@putcolbox</code> ... 184, 465, 478, 653, 668
<code>\@newlistfalse</code>	<code>\@putdynamicbbox</code>
163, 432, 607	203, 476, 479, 666, 669
<code>\@newstaticframe</code> .. 84, 383, 546, 546	<code>\@putdynamicbox</code> 201, 475, 478, 665, 668
<code>\@nexthNband</code>	<code>\@putmarginbox</code> 190, 470, 471, 658, 659
275, 494, 495, 691–693, 693	
<code>\@nexthband</code>	
274, 493–495, 690, 691, 692, 693	
<code>\@nextvNband</code>	
270, 492, 493, 688, 688, 689	
<code>\@nextvband</code>	
269, 491, 492, 686, 687, 687, 688	
<code>\@nil</code>	
405, 507, 571, 572, 708	
<code>\@nnil</code>	
507, 708	

<code>\@putstaticbbox</code>	195 , 473 , 478 , 662 , 668	<code>\@sdynamicswitchoffnextonly</code>	...
<code>\@putstaticbox</code>	193 , 472 , 478 , 661 , 668	 230 , 452 , 635 , 635
<code>\@relativeframelocation</code>	<code>\@sdynamicswitchonnext</code>
 136 , 425–427 , 598–601	 222 , 445 , 625 , 625
<code>\@reldynamicloc</code> 144 , 427 , 600	<code>\@sdynamicswitchonnextodd</code>
<code>\@relflowloc</code> 145 , 427 , 601	 223 , 446 , 626 , 626
<code>\@relstaticloc</code> 145 , 427 , 600	<code>\@sdynamicswitchonnextoddoneonly</code>	.
<code>\@resetactivechars</code> 163 , 432 , 606	 230 , 450 , 632 , 632
<code>\@resetdyn@mics</code>	197 , 474 , 480 , 663 , 671	<code>\@sdynamicswitchonnextonly</code>
<code>\@resetst@tics</code>	196 , 473 , 480 , 662 , 671	 227 , 449 , 630 , 631
<code>\@reversemarginfalse</code>	.. 77 , 369 , 528	<code>\@sect</code> 505 , 518 , 705 , 721
<code>\@reversemargintrue</code>	... 77 , 368 , 528	<code>\@setcol</code> 158 ,
<code>\@rightskip</code> 329		428–430 , 481 , 602 , 603 , 603 , 671
<code>\@s@ff@heading</code> 88 , 390 , 553 , 554	<code>\@setcolbox</code>
<code>\@s@tdynamicframeid</code>		181 , 464 , 480 , 481 , 651 , 670 , 671
 105 , 397 , 401 , 562 , 562 , 567	<code>\@setdynamiccontents</code>
<code>\@s@tffcol</code> 113 , 406 , 503 , 573 , 573 , 703
	. 14 , 364–366 , 465 , 519 , 523 , 653	<code>\@setdynamicframe</code>	107 , 399 , 565 , 565
<code>\@s@tfftextcol</code>	<code>\@setflowframe</code> 81 , 377 , 538 , 538
	. 14 , 364–366 , 393 , 428 , 433 ,	<code>\@setfr@mesfalse</code> 145 , 428 , 601
	465 , 519 , 523 , 557 , 602 , 608 , 653	<code>\@setfr@mesttrue</code> 146 , 428 , 602
<code>\@s@tflowframeid</code>	<code>\@setframecol</code>
 80 , 371 , 379 , 531 , 531 , 541		. 44 , 379 , 383 , 388 , 401 , 492 ,
<code>\@s@tfr@mes</code>	... 205 , 477 , 479 , 667 , 669		495 , 541 , 545 , 552 , 567 , 687 , 692
<code>\@s@tstaticframeid</code>	<code>\@setstaticframe</code>	.. 87 , 386 , 549 , 550
 85 , 384 , 388 , 547 , 547 , 552	<code>\@setthumbtab</code> 308 , 511 , 713 , 713
<code>\@sappenddynamic</code>	<code>\@setthumbtabindex</code>
 114 , 407 , 515 , 574 , 574 , 719	 310 , 512 , 713 , 714 , 714
<code>\@sappendminitocdf</code>	320 , 515 , 719 , 719	<code>\@sevencheckifframeabove</code>
<code>\@schapter</code> 366 , 524	 131 , 414 , 420 , 583 , 592
<code>\@scheckifframeabove</code>	<code>\@sevencheckifframebelow</code>
 122 , 413 , 414 , 425 , 583 , 598	 132 , 415 , 421 , 586 , 593
<code>\@scheckifframebelow</code>	<code>\@sevencheckifframeleft</code>
 124 , 415 , 425 , 585 , 598	 133 , 417 , 423 , 588 , 595
<code>\@scheckifframeleft</code>	<code>\@sevencheckifframeright</code>
 126 , 417 , 425 , 587 , 588 , 598	 134 , 418 , 424 , 590 , 596
<code>\@scheckifframeright</code>	<code>\@sf@chckifthispg</code>
 128 , 418 , 425 , 590 , 598	 180 , 453–460 , 463 ,
<code>\@scomputeffarea</code>		472 , 473 , 637–646 , 651 , 661 , 662
 117 , 408 , 409 , 577 , 577	<code>\@sf@checkuniqueidl</code>	85 , 384 , 547 , 547
<code>\@sdynamicchap</code>	292 , 503 , 504 , 702 , 703	<code>\@sf@mpg</code> 392 , 393 , 557
<code>\@sdynamicframeswapcoords</code>	<code>\@sf@setoffset</code>	188 , 468 , 471 , 656 , 660
 109 , 404 , 570 , 570	<code>\@sf@thisframe</code> 393 , 557 , 558
<code>\@sdynamicswitchoffnext</code>	<code>\@sflowframeswapcoords</code>
 224 , 447 , 627 , 627	 83 , 381 , 543 , 543
<code>\@sdynamicswitchoffnextodd</code>	<code>\@sflowswitchoffnext</code>
 226 , 448 , 628 , 628	 213 , 439 , 615 , 615
<code>\@sdynamicswitchoffnextoddoneonly</code>	<code>\@sflowswitchoffnextodd</code>
 231 , 452 , 636 , 636	 215 , 439 , 616 , 616

\@sflowswitchhoffnextoddonly . . .	\@sstaticswitchhoffnextodd
221, 444, 624, 624	236, 456, 640, 640
\@sflowswitchhoffnextonly	\@sstaticswitchhoffnextoddonly .
220, 444, 623, 623	242, 461, 647, 648
\@sflowswitchchonnext 211, 437, 613, 613	\@sstaticswitchhoffnextonly . . .
\@sflowswitchchonnextodd	241, 460, 647, 647
212, 437, 614, 614	\@sstaticswitchchonnext
\@sflowswitchchonnextoddonly	232, 453, 636, 636
219, 442, 620, 620	\@sstaticswitchchonnextodd
\@sflowswitchchonnextonly	233, 454, 637, 637
216, 441, 619, 619	\@sstaticswitchchonnextoddonly . .
\@sgetdynamicbounds 121, 413, 582, 582	240, 458, 643, 644
\@sgetdynamicicevenbounds	\@sstaticswitchchonnextonly
121, 413, 583, 583	238, 457, 642, 642
\@sgetflowbounds . 120, 413, 582, 582	\@startcolumn 430, 605
\@sgetflowevenbounds	\@startdblcolumn 209, 480, 671
120, 413, 582, 582	\@starttoc 19, 147,
\@sgetstaticbounds 119, 413, 581, 581	293, 312, 317, 504, 512, 704, 714
\@sgetstaticicevenbounds	\@staticcontinueonframe
119, 413, 582, 582	101, 392, 408, 556, 575
\@sinserthrul 284, 499, 502, 698, 700	\@staticconts 99, 393, 558, 558
\@sinsertvrul 280, 496, 498, 694, 697	\@staticframeid 85, 385, 386,
\@snewdynamicframe 104, 396, 560, 561	392, 395, 413, 453–458, 460,
\@snewflowframe . . . 79, 369, 529, 529	461, 499, 502, 548, 548, 549,
\@snewstaticframe . 84, 383, 546, 546	556, 559, 581, 582, 636, 638–
\@soddcheckifframeabove	640, 642, 644, 647, 648, 697, 701
123, 414, 583, 584	\@staticframeswapcoords
\@soddcheckifframebelow	100, 394, 395, 559, 559
125, 415, 585, 586	\@staticscontinueonframe
\@soddcheckifframeleft 127, 417, 588	101, 392, 407, 556, 575
\@soddcheckifframeright	\@staticswitchhoffnext
130, 418, 419, 590	234, 455, 639, 639
\@specialoutput 430, 605	\@staticswitchhoffnextodd
\@specialpagefalse 163, 208, 480, 670	236, 456, 457, 640, 641
\@specialstyle . . . 163, 208, 480, 670	\@staticswitchhoffnextoddonly . .
\@srelativeframelocation	242, 461, 647, 648
135, 425, 427, 598, 600, 601	\@staticswitchhoffnextonly
\@sreldynamicloc 144, 427, 600	241, 460, 647, 647
\@srelflowloc 145, 427, 601	\@staticswitchchonnext
\@srelstaticloc 145, 427, 600	232, 453, 454, 636, 637
\@ssetdynamiccontents	\@staticswitchchonnextodd
113, 406, 573, 573	233, 454, 455, 637, 638
\@ssetdynamicframe 107, 399, 565, 565	\@staticswitchchonnextoddonly . . .
\@ssetflowframe . . . 81, 377, 538, 538	240, 458, 459, 643, 645
\@ssetstaticframe . 87, 386, 549, 549	\@staticswitchchonnextonly
\@sstaticconts 99, 393, 558, 558	238, 457, 458, 642, 643
\@sstaticframeswapcoords	\@storefileconts
100, 394, 559, 559	312, 512, 513, 715, 715
\@sstaticswitchhoffnext	\@storetocfalse . . . 512, 515, 714, 718
234, 455, 639, 639	\@storetoctrue 515, 718

267, 481, 483–490, 672, 673,
 675–677, 679, 681, 682, 684, 685
 \advance 368, 369, 371, 372, 377, 382,
 383, 385–387, 395, 397–400,
 404, 414–425, 428–430, 434–
 436, 438–446, 448–462, 467,
 471, 473, 474, 478–481, 483,
 486, 489, 490, 492, 495, 506,
 508, 510–514, 516, 527, 529,
 531, 532, 538, 539, 544, 546–
 550, 559, 560, 562–565, 570,
 571, 584–589, 591–597, 601–
 604, 609, 610, 612, 614–617,
 619–624, 626, 628, 629, 631–
 638, 640–649, 655, 659, 662,
 663, 668–671, 674, 675, 680,
 684, 685, 688, 693, 706, 707,
 709, 711–713, 716, 717, 719, 720
 \aftergroup 163, 432, 607
 \afterminitocskip
 318, 319, 515, 718, 718
 \afterpage 317, 511, 712
 \alignctofalse
 .. 16, 25, 311, 512, 515, 714, 717
 \aligntoctrue 25, 311, 512, 714
 \and 508, 509, 709, 710
 \appenddfminitoc 319, 515, 719
 \appenddynamiccontents 113, 407, 574
 \appto 17, 18
 \arabic 31, 293, 367, 525
 \AtBeginDocument 147, 429, 603
 \AtEndDocument 142

B

\backmatter 17, 18
 \baselineskip
 164, 267, 318, 319, 431, 432,
 490, 515, 605, 607, 685, 718, 719
 \beforeminitocskip
 318, 318, 319, 515, 718, 718, 719
 \begin ... 94, 97–99, 101, 197, 299,
 327, 330, 345, 392, 393, 406–
 408, 507, 557, 558, 572, 575, 708
 \begingroup 163, 209,
 312, 390, 392, 393, 431, 432,
 480, 512, 554, 557, 606, 671, 715
 \bfseries 286, 287
 \bgroup 477, 478, 667
 \bool 10, 12, 15–
 19, 22–28, 32, 38, 52, 64, 67–
 69, 72, 73, 75, 76, 89–93, 141–
 144, 146, 148–158, 173–181,
 184, 186, 190, 193–195, 202–
 204, 208, 209, 211–218, 221–
 225, 227–229, 232–239, 246,
 253, 254, 277, 281, 293, 297,
 299–306, 308, 309, 311, 315,
 317, 318, 321, 322, 328, 331–
 333, 335, 339–342, 345, 347–353
 \boolean 464,
 467–476, 508, 509, 513, 652,
 655–657, 659–665, 709, 710, 715
 \botmark 433, 607
 \box 52, 55–57, 97, 164, 185,
 186, 191, 194–196, 202–206,
 290, 355, 356, 431–433, 464,
 465, 478, 605–608, 651, 652, 667
 \boxgs 327
 \boxmaxdepth 433, 608
 \break 389, 553

C

\c 7, 9, 12–14,
 33–35, 40, 44, 45, 65–71, 77,
 78, 80, 84, 94, 95, 97, 101, 102,
 104, 111, 112, 117, 118, 145–
 148, 150–153, 156, 158, 159,
 162, 166, 167, 178–180, 188–
 191, 198, 206, 207, 209, 210,
 216–220, 226, 228–230, 237–
 241, 244–248, 254–257, 261,
 264–268, 270–275, 277, 279–
 281, 283, 293, 295, 297, 298,
 305, 306, 308, 310, 315, 316,
 329, 331, 334, 343–347, 349–353
 \c@absolutepage 29, 75,
 148, 365, 368, 429, 522, 527, 603
 \c@curpg 177, 428,
 461, 462, 480, 601, 648, 649, 670
 \c@dbltopnumber 481, 671
 \c@displayedframe 162, 367, 525
 \c@ffrangenenum 177, 436, 611, 612
 \c@maxdynamic
 51, 52, 103–106, 165, 197, 207,
 300, 301, 367, 395–399, 406,
 407, 411, 412, 474, 478, 479,
 508–510, 525, 560–565, 573,
 574, 579, 581, 663, 668, 709–711
 \c@maxflow 49,
 75, 79–81, 146–148, 150–156,

158, 175, 176, 178, 191, 206,	\chaptermark 19, 170
210, 243–247, 249–256, 258–	\char 6, 10
267, 367–372, 376, 377, 382,	\checkiffframeabove 121, 413, 583
384, 395, 397, 404, 410, 412,	\checkiffframebelow 124, 415, 585
428–430, 434, 435, 462, 471,	\checkiffframeleft 126, 417, 587
478, 482, 485, 488, 525, 527,	\checkiffframeright 128, 418, 589
529–532, 538, 539, 544, 547,	\cleardoublepage 76, 288, 290, 368, 527
559, 560, 562, 570, 571, 579,	\clearpage 74, 76,
580, 602–604, 609, 610, 649,	148, 149, 154, 155, 157, 162,
659, 667, 668, 673, 678, 682, 683	288, 290, 368, 432, 526, 527, 606
\c@maxminitoc 313, 313,	\cleartoevenpage 76
315, 316, 513, 514, 715, 715, 717	\clist 30, 32–
\c@maxstatic	34, 45, 60–68, 81, 82, 86, 100,
. . . 50, 84–86, 146, 147, 197,	106, 108, 115, 179, 180, 210–
206, 269, 275, 276, 279, 280,	214, 216–224, 226–235, 237–
282, 283, 367, 383–387, 410,	241, 244, 245, 270, 271, 273,
412, 428, 429, 473, 478, 491,	274, 283, 298, 305–307, 310, 323
492, 495, 496, 498, 501, 502,	\closein 513, 716
525, 546–550, 579, 580, 602,	\closeout 312
662, 668, 687, 692, 693, 696, 700	\col@number 162, 209, 210
\c@maxthumbtabs . . . 294–297, 300,	\color 3, 15, 363, 365, 518, 523
307, 310, 320, 506, 508, 510–	color (option) 522
512, 516, 706, 709, 711–714, 720	\color@begingroup 434, 608
\c@maxtocunits . . . 313, 313, 320,	\color@endbox 159, 160, 432, 606
513, 516, 715, 715, 716, 719, 720	\color@endgroup 434, 608
\c@minitoc	\color@hbox 159, 160, 432, 606
317, 321, 515, 517, 518, 718, 721	\colorbox 15, 366, 523
\c@page 11, 12,	\columnheight
28, 76–78, 122, 124, 126, 127,	158, 481–490, 672, 673, 675–685
129, 136, 167–169, 185, 186,	\columnsep
190, 194, 195, 202, 204, 365,	245, 248, 253, 257, 262, 266,
368, 369, 414, 415, 417–419,	267, 481, 483, 485–490, 673–
465, 466, 470, 472, 473, 476,	675, 677, 678, 680, 682, 684, 685
477, 479, 511, 522, 527, 528,	\columnwidth 158, 159, 162,
583–586, 588, 590, 653, 654,	430, 481–483, 485–490, 604,
658, 661, 662, 665, 666, 669, 712	673–675, 677, 678, 680, 682–685
\c@secnumdepth . . . 171, 172, 286, 287	\computebottomedge 115,
\c@thisframe	268, 270, 272–274, 276, 408,
. 75, 76, 78–80, 145–147, 161,	491–496, 576, 686, 688, 690–693
162, 164, 177, 208, 209, 367–	\computeflowframearea . 115, 408, 577
369, 371, 428, 429, 433, 434,	\computeleftedgeeven
480, 481, 525, 527, 528, 530, 114, 115, 268–273,
531, 601–603, 608, 609, 670, 671	276, 296, 408, 490–494, 496,
\catcode 163	508, 576, 576, 686–691, 693, 709
\centering 11, 507, 707	\computeleftedgeodd 114,
\chapter 19,	115, 268–273, 276, 408, 490–
20, 170, 284–286, 292, 366, 524	494, 496, 575, 576, 686–691, 693
\chapterfirstpagestyle	\computerightedgeeven . 115, 273,
. 20, 20, 366, 524, 524	274, 408, 494, 495, 576, 691, 692
\ChapterInDynamic 285	

<code>\computerightedgeodd</code>	554, 556–559, 561–564, 566–
. . . . 115 , 273 , 274 , 296 , 408 ,	570, 573, 574, 579, 581, 584–
494 , 495 , 508 , 576 , 690 , 692 , 709	597, 601, 602, 604, 607–610,
<code>\computetopedge</code> 114 , 115 , 269 , 271 ,	649–666, 687, 692, 693, 696,
408 , 491 , 493 , 576 , 576 , 687 , 689	700, 706, 707, 709–712, 715–720
<code>\contentsline</code> 314 , 514 , 515 , 717	
<code>\contentsname</code> 19	
<code>\continueonframe</code>	
. . . 9 , 96 , 100 , 110 , 112 , 113 ,	
392 , 405 , 407 , 556 , 571 , 572 , 574	
<code>\copy</code> 327	
<code>\count</code> 164 , 480 , 670	
<code>\count@</code> 368 , 435 , 438 –	
462 , 527 , 610 , 614 – 624 , 626 – 649	
counters:	
<code>absolutepage</code> 12 , 522	
<code>displayedframe</code> 31 , 525	
<code>maxdynamic</code> 32 , 525	
<code>maxflow</code> 31 , 525	
<code>maxstatic</code> 32 , 525	
<code>thisframe</code> 31 , 525	
<code>\cs</code> 4 – 6 , 10 ,	
11 , 13 – 15 , 17 – 24 , 26 , 27 , 29 ,	
30 , 33 , 35 , 39 , 43 , 44 , 46 – 63 ,	
65 – 68 , 70 – 74 , 77 – 82 , 84 , 85 ,	
87 – 89 , 92 – 95 , 97 , 98 , 100 – 102 ,	
104 , 105 , 107 , 109 , 111 – 114 ,	
116 – 136 , 139 , 142 , 147 , 148 ,	
150 , 153 – 155 , 157 , 158 , 160 –	
162 , 166 – 171 , 173 , 174 , 177 ,	
179 – 181 , 183 , 184 , 186 – 191 ,	
193 , 195 , 197 , 199 , 201 – 203 ,	
205 – 208 , 211 – 214 , 216 , 217 ,	
220 – 224 , 226 , 228 , 230 – 235 ,	
237 , 238 , 241 – 246 , 248 – 250 ,	
253 , 256 , 257 , 259 , 262 , 265 ,	
268 – 275 , 278 , 281 , 284 – 286 ,	
288 – 296 , 298 – 300 , 302 – 304 ,	
306 – 308 , 310 , 312 – 314 , 317 –	
321 , 327 , 330 – 335 , 338 , 341 ,	
343 , 345 , 355 , 357 – 359 , 361 , 363	
<code>\csletcs</code> 389 ,	
403 , 472 , 476 , 553 , 569 , 661 , 665	
<code>\csname</code> 312 , 369 – 372 , 378 –	
394 , 396 – 398 , 400 – 404 , 406 ,	
407 , 411 , 412 , 414 – 425 , 428 ,	
430 , 433 – 435 , 462 – 477 , 491 ,	
495 , 496 , 498 , 501 , 505 , 506 ,	
508 – 516 , 528 – 532 , 540 – 552 ,	
	D
<code>\dblfloatpagefraction</code> 481 , 671	
<code>\dbltopfraction</code> 481 , 671	
<code>\DeclareCurrentRelease</code> 3 , 518	
<code>\DeclareOptionX</code> . . . 364 – 366 , 519 – 523	
<code>\DeclareRelease</code> 3 , 518	
<code>\DeclareRobustCommand</code>	
. 425 , 427 , 598 , 600 , 601	
<code>\def</code> 31 , 32 ,	
76 , 95 , 163 , 170 – 174 , 242 , 363 ,	
365 – 369 , 371 , 373 – 378 , 382 –	
384 , 386 , 387 , 390 – 392 , 395 ,	
397 , 399 , 400 , 404 – 407 , 431 –	
434 , 436 – 441 , 446 – 449 , 454 –	
457 , 462 – 464 , 471 , 475 , 477 ,	
480 , 481 , 490 – 494 , 496 – 500 ,	
502 – 505 , 507 , 514 , 515 , 517 –	
519 , 522 , 523 , 525 , 527 , 528 ,	
531 , 533 – 539 , 544 – 547 , 550 ,	
551 , 553 – 556 , 560 , 562 , 565 ,	
566 , 571 – 574 , 605 – 609 , 611 –	
618 , 626 – 630 , 638 – 642 , 649 ,	
650 , 652 , 660 , 664 , 667 , 670 ,	
671 , 686 – 692 , 694 , 695 , 697 –	
701 , 704 , 705 , 708 , 717 , 721	
<code>\defaultthumbtabtype</code>	
. . . . 292 , 292 , 293 , 504 , 703 , 704	
<code>\define@boolkey</code> 364 , 520	
<code>\define@choicekey</code>	
364 , 365 , 376 , 520 – 522 , 537 , 538	
<code>\define@key</code> 372 – 376 , 533 – 537	
<code>\dfchaphead</code> 284 , 503 , 702	
<code>\dfchapterclearpage</code> 288 , 289	
<code>\DFchapterstyle</code>	
. . . . 284 , 285 , 503 , 702 , 702 , 703	
<code>\dfcontinuedfalse</code> 405 , 572	
<code>\dfcontinuedtrue</code> 405 , 572	
<code>\dfEvenFooterStyle</code> 170 , 170 , 171 , 173	
<code>\dfEvenHeaderStyle</code> 169 , 170 , 171	
<code>\dfOddFooterStyle</code> 170 , 170 , 171 , 173	
<code>\dfOddHeaderStyle</code> 169 , 170 , 171	
<code>\DFschapterstyle</code>	
284 , 285 , 285 , 503 , 702 , 702 , 703	
<code>\dfswapoddeven</code> 108 , 404 , 570	

<code>\dim</code>	15, 30, 32, 42, 44, 45, 53–55, 75, 98, 103, 114–117, 123, 125–128, 130–134, 138– 140, 146, 147, 158, 159, 183, 187–191, 198, 199, 205, 206, 243, 245, 248, 250, 253, 257, 259, 260, 262, 265–283, 289, 296, 302, 303, 329, 331, 338–354	<code>\dynamicsswitchoffnextonly</code>	230, 452, 635
<code>\dimen@</code>	430, 434, 604, 609	<code>\dynamicsswitchonnext</code> . .	221, 445, 625
<code>\disablethumbtabs</code>	304, 510, 711	<code>\dynamicsswitchonnextodd</code> 222, 446, 626	
<code>displayedframe</code> (counter)	31, 525	<code>\dynamicsswitchonnextoddoneonly</code>	227, 450, 632
<code>\divide</code>	481, 483, 485, 486, 488–490, 508, 673, 674, 677, 680, 682, 684, 685, 709	<code>\dynamicsswitchonnextonly</code>	226, 449, 630
<code>\do</code>	371, 377, 381, 382, 386, 394, 395, 399, 404, 409, 434, 435, 437–463, 511, 512, 531, 538, 539, 543, 544, 549, 550, 559, 560, 565, 570, 571, 577, 609–611, 613–616, 618– 620, 622–629, 631–633, 635– 645, 647, 648, 650, 713, 714	E	
<code>\document@default@language</code>	163	<code>\edef</code>	371, 372, 379, 383–385, 387, 390, 392, 393, 397, 398, 400, 405, 406, 414–425, 428, 433–436, 438, 442, 443, 446, 447, 450– 452, 454, 455, 459, 460, 462– 464, 470, 471, 475, 498, 501, 506–508, 510, 511, 531, 532, 541, 545, 547, 548, 550, 553, 554, 557, 562, 563, 566, 571, 572, 584–597, 602, 608–610, 612, 614, 615, 620–623, 626, 627, 632–635, 638, 639, 644– 647, 649–651, 658–660, 664, 665, 696, 700, 706–709, 711, 712
<code>\dominitoc</code>	318, 321, 515, 517, 518, 718, 721	<code>\egroup</code>	478, 667
<code>\dp</code> . .	206, 433, 434, 478, 608, 609, 667	<code>\else</code>	11, 57, 76, 136, 137, 140, 141, 148, 163, 164, 167–169, 245, 246, 248, 254, 257, 263, 266, 268–274, 276, 364, 366, 368, 369, 371, 385, 391, 393, 397, 405–407, 410–412, 414–463, 465, 466, 470, 472, 473, 476, 477, 479, 480, 482, 483, 485–496, 498, 501, 506, 507, 510, 512, 514– 516, 520, 524, 527, 528, 531, 547, 555, 557, 562, 572–574, 579–581, 583–600, 602, 604– 606, 608–646, 648–650, 653, 654, 658, 659, 661, 662, 665, 666, 669–671, 673–675, 678– 680, 682–693, 696, 700, 706– 708, 711, 714, 717, 718, 720
<code>draft</code> (option)	519	<code>\emph</code>	18, 103, 408, 575
<code>\dynamicaddexclusion</code> . .	108, 230, 231, 403, 452, 453, 569, 635, 636	<code>\empty</code>	89, 164
<code>dynamiccontents</code> (env.)	109, 571	<code>\emulateonecolumn</code>	147, 429, 602
<code>dynamiccontents*</code> (env.)	109, 573	<code>\emulatetwocolumn</code>	146, 428, 602
<code>\dynamicframeevenx</code>	106, 398, 411, 412, 564, 579, 581	<code>\enableminitoc</code>	321, 322, 516, 517, 720, 721
<code>\dynamicframeeveny</code>	106, 398, 411, 412, 564, 579, 581		
<code>\dynamicframeheight</code> 106, 198, 199, 201			
<code>\dynamicframewidth</code> 106, 198, 199, 201			
<code>\dynamicframex</code> 105, 398, 411, 564, 579			
<code>\dynamicframey</code> 106, 398, 411, 564, 579			
<code>\dynamicsettexclusion</code>	108, 402, 403, 568, 569		
<code>\dynamicsetpagelist</code> 108, 221–224, 226, 227, 230, 402, 403, 445– 452, 568, 569, 625–631, 633, 635			
<code>\dynamicswitchoffnext</code> . 223, 447, 627			
<code>\dynamicswitchoffnextodd</code>	224, 448, 628		
<code>\dynamicswitchoffnextoddoneonly</code>	230, 452, 636		

<code>\enablethumtbs</code>	304 , 510 , 711	<code>\evensidemargin</code>	114 , 164 , 408 , 479 , 576 , 669
<code>\end</code>	94 , 98 , 99 , 101 , 197 , 299 , 327 , 330 , 354 , 379 , 383 , 388 , 393 , 394 , 401 , 405–408 , 436 , 437 , 492 , 495 , 507 , 513–515 , 541 , 545 , 552 , 557 , 558 , 567 , 571–573 , 575 , 612 , 613 , 687 , 692 , 708 , 716 , 717	<code>\exp</code>	14 , 41 , 42 , 45–47 , 49–52 , 57 , 62 , 65 , 66 , 81 , 82 , 86–88 , 91 , 97 , 100 , 101 , 106–108 , 111 , 112 , 115 , 146 , 158 , 175 , 176 , 178 , 181 , 182 , 184 , 192 , 200 , 210–213 , 215 , 216 , 219–224 , 226 , 227 , 230– 234 , 236 , 237 , 240–242 , 246 , 284 , 285 , 293 , 297 , 299 , 301– 303 , 306 , 342 , 343 , 346–354 , 358
<code>\end@dblfloat</code>	161 , 162	<code>\expandafter</code>	312 , 365 , 366 , 369–371 , 378–381 , 383 , 384 , 387–389 , 392–394 , 396 , 397 , 400–407 , 411 , 412 , 414– 425 , 430 , 433 , 436–441 , 443– 462 , 464–466 , 470–477 , 491 , 495 , 496 , 506–514 , 516 , 523 , 529–531 , 540–543 , 545–547 , 550–552 , 556–559 , 561 , 562 , 566–574 , 579 , 581 , 584–587 , 589 , 591–597 , 604 , 607 , 608 , 612–619 , 621 , 623–631 , 633 , 635–639 , 641–643 , 645 , 647– 649 , 651–655 , 659–666 , 687 , 692 , 693 , 706–712 , 715–717 , 720
<code>\end@float</code>	161 , 162	<code>\ExplSyntaxOff</code>	19 , 30 , 163 , 286 , 322 , 323 , 328 , 363
<code>\endcsname</code> . . .	312 , 369–372 , 378– 394 , 396–398 , 400–404 , 406 , 407 , 411 , 412 , 414–425 , 428 , 430 , 433–435 , 462–477 , 491 , 495 , 496 , 498 , 501 , 505 , 506 , 508–516 , 528–532 , 540–552 , 554 , 556–559 , 561–564 , 566– 570 , 573 , 574 , 579 , 581 , 584– 597 , 601 , 602 , 604 , 607–610 , 649–666 , 687 , 692 , 693 , 696 , 700 , 706 , 707 , 709–712 , 715–720	<code>\ExplSyntaxOn</code>	3 , 19 , 30 , 164 , 286 , 322 , 323 , 328
<code>\enddynamiccontents</code>	406 , 573	F	
<code>\endgraf</code>	94 , 98 , 198 , 199	<code>\f@depth</code>	163
<code>\endgroup</code>	163 , 209 , 312 , 390 , 391 , 393 , 431 , 432 , 480 , 513 , 554 , 557 , 606 , 607 , 671 , 715	<code>\fboxrule</code> .	74 , 183 , 367 , 465 , 526 , 653
<code>\endxdynamiccontents</code>	110 , 405 , 406 , 571 , 573	<code>\fboxsep</code>	15 , 74 , 183 , 366 , 367 , 465 , 523 , 526 , 653
<code>\entry</code>	407 , 461 , 574 , 648	<code>\ff@angle</code>	376 , 378 , 380 , 387 , 389 , 400 , 402 , 537 , 539 , 541 , 542 , 550 , 552 , 566 , 568
environments:		<code>\ff@backcol</code>	363 , 366 , 375 , 378 , 379 , 387 , 388 , 400 , 401 , 464 , 471 , 475 , 477 , 519 , 523 , 536 , 539 , 541 , 550 , 552 , 566 , 567 , 651 , 660 , 664 , 667
<code>dynamiccontents</code>	109 , 571	<code>\ff@clear</code>	376 , 378 , 380 , 387 , 389 , 400 , 402 , 537 , 539 , 542 , 550 , 553 , 566 , 568
<code>dynamiccontents*</code>	109 , 573	<code>\ff@col</code>	365 , 375 , 378 , 379 , 387 , 388 , 400 ,
<code>staticcontents</code>	96 , 556		
<code>staticcontents*</code>	96 , 556		
<code>staticfigure</code>	242 , 671		
<code>statictable</code>	242 , 671		
<code>\equal</code>	363 , 365 , 366 , 369 , 371–377 , 382 , 385 , 386 , 395 , 397–399 , 404 , 435 , 463 , 467– 470 , 483 , 491 , 495–500 , 502– 505 , 511 , 513 , 517–519 , 523 , 524 , 528 , 530–538 , 544 , 547 , 548 , 550 , 559 , 560 , 562 , 563 , 565 , 570 , 611 , 650 , 655–659 , 676 , 687 , 692 , 694 , 695 , 697– 701 , 704 , 705 , 713 , 716 , 720 , 721		
<code>\evencheckifframeabove</code>	130 , 420 , 591		
<code>\evencheckifframebelow</code>	131 , 421 , 593		
<code>\evencheckifframeleft</code> .	132 , 423 , 594		
<code>\evencheckifframeright</code>	133 , 424 , 596		

401, 464, 471, 475, 477, 523, 535, 539, 541, 550, 552, 566, 567, 651, 652, 660, 664, 665, 667	536, 539, 541, 550, 552, 566–568
\ff@contents 474, 475, 663, 664	\ff@pages 371, 375, 378, 380, 387, 389, 400, 402, 434, 435, 463, 464, 490–495, 506, 531, 536, 539, 541, 550, 552, 566, 568, 609, 610, 650, 651, 686–693, 706, 707
\ff@endpage 506, 707	\ff@s@t@doubleboxoffset <u>187</u>
\ff@evenx 373, 378, 379, 387, 388, 400, 534, 539, 540, 551, 566	\ff@sethead 390, 391, 553, 554
\ff@eveny 373, 378, 379, 387, 388, 400, 401, 534, 539, 540, 551, 566, 567	\ff@shape 376, 378, 380, 387, 389, 391–393, 400, 402, 474, 537, 539, 542, 551, 552, 555, 557, 566, 568, 663
\ff@frame 374, 378, 387, 400, 535, 539, 550, 551, 566	\ff@style 376, 378, 380, 387, 389, 400, 402, 474, 475, 537, 539, 542, 550, 553, 566, 568, 663, 664
\ff@frametype 374, 378, 379, 387, 388, 400, 401, 464, 471, 475, 535, 539, 541, 550, 552, 566, 567, 651, 652, 660, 664	\ff@txtcol . . . 365, 375, 378, 379, 387, 388, 393, 400, 401, 428, 433, 464, 475, 477, 523, 536, 539, 541, 550, 552, 557, 566, 567, 602, 608, 651, 652, 664, 667
\ff@height 373, 378, 387, 388, 400, 401, 533, 539, 540, 550, 552, 566, 567	\ff@valign 376, 378, 380, 388, 401, 471, 537, 539, 542, 552, 567, 660
\ff@hide 376, 378, 380, 387, 389, 400, 403, 538, 539, 542, 551, 553, 566, 569	\ff@width 373, 378, 387, 388, 400, 401, 533, 539, 540, 550, 552, 566, 567
\ff@hidethis 376, 378, 380, 387, 389, 400, 403, 538, 539, 542, 551, 553, 566, 569	\ff@x . 373, 378, 387, 400, 470, 533, 539, 540, 550, 551, 566, 658, 659
\ff@id . . . 367, 372, 377, 382, 385, 386, 392, 395, 398, 399, 404, 406, 407, 409, 410, 413, 433, 437–461, 498, 499, 502–504, 510–512, 516, 526, 532, 533, 538, 543, 544, 548, 549, 556, 559, 563, 565, 570, 573, 574, 577, 578, 581–583, 607, 608, 613–617, 619–621, 623–629, 631–633, 635–645, 647, 648, 697, 698, 701–703, 711–714, 720	\ff@xpages 375, 378, 380, 387, 389, 400, 402, 434, 435, 462, 536, 539, 541, 551, 552, 566, 568, 609, 610, 650
\ff@label 371, 374, 378, 379, 384, 387, 388, 397, 400, 401, 531, 535, 539, 541, 547, 551, 552, 562, 566, 567	\ff@y . 373, 378, 387, 400, 470, 533, 539, 540, 550, 551, 566, 658, 659
\ff@margin 369, 376, 378–380, 387, 389, 400, 402, 470, 528, 537, 539, 541, 550, 553, 566, 568, 659	\FFabove <u>135</u> , 136, 141, 425, 426, <u>598</u> , 599
\ff@oddx 374, 378, 379, 387, 388, 400, 401, 534, 539, 540, 551, 566, 567	\FFaboveleft <u>134</u> , 136, 140, 425, 426, 598, <u>598</u> , 599
\ff@oddy 374, 378, 379, 387, 388, 400, 401, 534, 539–541, 551, 552, 566, 567	\FFaboveright <u>135</u> , 136, 141, 425, 426, 598, <u>598</u> , 599
\ff@offset 375, 378, 380, 387, 389, 400, 401,	\ffaddtoadjustframeshook <u>177</u> , 461, <u>648</u>

422, 497, 498, 501, 576–581, 584–587, 592–594, 695, 696, 699	\fflabelsep . 3, 13, 363, 518, 518, 519
\ffareawidth 115, 116, 118, 119, 127, 128, 130, 133, 134, 140, 278, 281, 282, 408–412, 417–420, 423–425, 497, 501, 576–581, 588, 589, 591, 595–597, 695, 696, 699, 700	\FFleft 135, 137, 141, 425–427, 598, 599, 600
\ffareax 115, 116, 118, 127, 128, 130, 133, 134, 140, 278, 281, 282, 408–412, 417–420, 423–425, 497, 498, 501, 576–581, 588– 591, 595–597, 695, 696, 699, 700	\FFoverlap 135, 137, 141, 425–427, 598, 599, 600
\ffareay 115, 116, 118, 119, 123, 125, 126, 131, 132, 140, 278, 279, 281, 282, 408–412, 414–417, 420– 422, 497, 498, 501, 576–581, 584–587, 592–594, 695, 696, 699	\ffprechapterhook 20, 20, 366, 524, 524
\FFbelow 135, 136, 141, 425–427, 598, 599, 600	\ffpshpar 88, 390, 393, 474, 553, 553, 557, 663
\FFbelowleft 135, 136, 141, 425–427, 598, 599	\FFright 135, 137, 141, 425–427, 598–600
\FFbelowright 135, 136, 141, 425–427, 598, 599, 600	\ffruleddeclarations . . . 276, 279, 283, 496, 498, 501, 694, 696, 700
\ffchapterdefaultfont . 287, 287, 288	\ffstaticpostcontinued . . . 102, 102
\ffchapterheadstyle . . . 287, 287, 288	\ffswapoddeven 82, 381, 543
\ffchapternamenum 287, 287	\fftolerance 158, 158, 159, 429, 430, 603, 604
\ffchapternamenumfont 287, 287	\ffvadjusttrue 242, 481, 672
\ffchapterpostheadskip . . . 286, 288	\ffvrule 279, 280, 498, 696, 696
\ffchapterpostnamenum 287, 287	\fi 11, 57, 76, 136, 137, 141, 142, 148, 163, 164, 167–169, 171, 172, 208, 209, 243, 245, 246, 248, 251, 253, 254, 257, 260, 262, 263, 265, 266, 268– 274, 276, 286–288, 312, 364– 366, 368, 369, 371, 372, 377, 382, 385, 387, 391, 393, 395, 398, 399, 404–407, 409–432, 434–463, 465, 466, 470–473, 475–477, 479–498, 501, 506– 516, 520–522, 524, 527, 528, 531–533, 539, 544, 548–550, 555, 557, 560, 563, 565, 571– 574, 577–581, 583–597, 599, 600, 602–606, 608–650, 653– 655, 659, 661, 662, 664, 666, 667, 669–693, 696, 699, 700, 707, 708, 710–712, 714–718, 720
\ffchapterpreheadskip . 286, 287–289	\file 355
\ffchaptertitlefont 287, 288	final (option) 519
\ffcolumnseprule 276, 279, 283, 496, 498, 501, 694, 696, 700	\finalhyphendemerits 329
\ffcontinuedtextfont 103, 103, 408, 575, 575	\finishthispage 74, 75, 146, 147, 161, 367, 368, 428, 429, 526, 526, 602
\ffcontinuedtextlayout 101, 103, 111, 112, 405, 407, 408, 572, 575, 575	\firstmark 433, 607
\ffdefaultcontinuetext . . . 102, 102	\flf@doifverbose 5, 6, 26, 178, 364, 462, 520, 520, 649
\ffdefaultdynamiccontinuetext 102, 112	\flf@getcontargs 110, 405, 572
\ffdefaultpostcontinued . . 103, 103	\flf@message 5, 364, 381, 394, 403, 430, 464, 465, 520, 520, 542, 543, 558, 559, 569, 604, 651, 653
\ffdefaultstaticcontinuetext 102, 102	
\ffdynmicpostcontinued . . 102, 103	
\ffevenoffset 207, 479, 669	
\ffhrule . 283, 283, 501, 502, 700, 700	
\fflabelfont 4, 13, 363, 518, 519	

<code>\flf@messinfo</code>	462, 649	<code>\flowframecol</code>	
<code>\flf@next</code>	270, 275, 390, 391,		13 , 14 , 15 , 45 , 365 , 366 , 370 ,
	405, 406, 492, 495, 507, 508,		384 , 396 , 522 , 523 , 530 , 546 , 561
	553, 554, 572, 573, 688, 693, 708	<code>\flowframeevenx</code>	
<code>\flf@ta</code>	114 , 405, 407,		83 , 383 , 410 , 412 , 545 , 579 , 580
	507, 514, 571, 574, 708, 716, 717	<code>\flowframeeveny</code>	
<code>\flf@tb</code>	114 ,		83 , 383 , 410 , 412 , 545 , 579 , 580
	407, 507, 514, 574, 708, 716, 717	<code>\flowframeheight</code>	83 , 117 ,
<code>\FLFabovefalse</code>	123, 124, 131, 140,		383 , 409 , 410 , 412 , 545 , 577 – 580
	414, 415, 421, 584, 585, 592, 593	<code>\flowframerule</code>	74 , 183 , 187 ,
<code>\FLFabovetrue</code>	123, 124, 131, 140,		188 , 367 , 465 , 467 , 526 , 653 , 655
	414, 415, 421, 584, 585, 592, 593	<code>\flowframesep</code>	74 , 183 , 187 ,
<code>\FLFbelowfalse</code>	125, 126, 132,		188 , 367 , 465 – 467 , 526 , 653 , 655
	140, 416, 417, 422, 423, 587, 594	<code>\flowframeshowlayout</code> 74 , 367 , 526 , 526	
<code>\FLFbelowtrue</code>	125, 126,	<code>\flowframetextcol</code>	
	132, 140, 416, 417, 422, 587, 594		13 , 14 , 15 , 45 , 365 , 366 , 370 ,
<code>\FLFleftfalse</code>	128, 133,		384 , 396 , 523 , 524 , 530 , 547 , 561
	140, 418, 423, 424, 589, 595, 596	<code>\flowframewidth</code>	83 , 116 ,
<code>\FLFlefttrue</code>	128, 133,		383 , 409 , 410 , 412 , 545 , 577 – 580
	140, 418, 423, 424, 589, 595, 596	<code>\flowframex</code>	83 ,
<code>\FLForgpar</code>			116 , 382 , 409 , 410 , 544 , 577 – 579
	88 , 389 , 390 , 393 , 553 , 554 , 557	<code>\flowframey</code>	83 , 116 ,
<code>\FLFrightfalse</code>	130, 134,		117 , 382 , 409 , 410 , 545 , 577 – 579
	140, 419, 420, 424, 425, 591, 597	<code>\flowframheaderchapprefix</code> . .	19 , 171
<code>\FLFrighttrue</code>	130, 134,	<code>\flowframheadersep</code>	19 , 171 , 172
	140, 419, 420, 424, 425, 591, 597	<code>\FlowFramRestoreOR</code>	161
<code>\FLFsimpar</code>	88 , 88 , 94 , 97 , 198 , 199	<code>\flowframsectionheader</code> . . .	170 – 172
<code>\flowaddeclusion</code>	82 , 220 ,	<code>\FlowFramSectionUnit</code>	89 , 93
	221 , 381 , 444 , 445 , 543 , 623 , 624	<code>\FlowFramSetDynamicHeadFootAttributes</code>	
<code>\flowfram</code>	4, 12 , 13 ,		166 , 173 , 174
	17 – 22 , 30 , 31 , 40 – 42 , 44 – 51 ,	<code>\flowframsetup</code>	30
	53 – 70 , 72 – 74 , 78 – 80 , 82 – 89 ,	<code>\flowframesubsectionheader</code>	172
	91 , 93 , 97 – 102 , 104 – 109 , 111 –	<code>\FlowFramTkBeginScopeIfGroup</code> . .	330
	114 , 118 , 119 , 137 , 138 , 146 ,	<code>\FlowFramTkDeclareKeys</code>	323
	148 , 149 , 153 – 156 , 158 , 159 ,	<code>\flowframtkencapobject</code> . . .	329 , 355
	165 – 173 , 175 , 176 , 178 – 205 ,	<code>\flowframtkendobject</code>	329
	210 , 246 , 250 , 269 , 275 , 276 ,	<code>\FlowFramTkEndScopeIfGroup</code>	330
	279 , 280 , 283 , 289 – 291 , 298 –	<code>\flowframtkimageinfo</code>	358
	301 , 304 – 307 , 317 – 321 , 360 – 362	<code>\flowframtkimgtitlechar</code> . .	356 , 357
<code>\flowfram@dfchap@sbox</code>	288 , 290	<code>\flowframtkMakeDFHeaderFooter</code> .	362
<code>\flowfram@doc@htmlopts</code>	43 , 43	<code>\flowframtkNewDynamicStyle</code>	359
<code>\flowfram@dynamicfooter</code> 160 , 168 , 169		<code>\flowframtkNewFrameBorder</code>	359
<code>\flowfram@dynamicheader</code> 160 , 167 , 168		<code>\flowframtkNoRagged</code>	329
<code>\flowfram@footer</code>	160 , 161	<code>\flowframtkpdftransmod</code>	327
<code>\flowfram@header</code>	159 , 161	<code>\FlowFramTkScopeIfGroup</code>	330
<code>\flowfram@preamble@htmlopts</code> .	43 , 43	<code>\flowframtkSetCreationDate</code> 358 , 358	
<code>\flowfram@thepagenumber</code> . . .	12 , 138	<code>\flowframtkSetDynamicEvenFoot</code> .	362
<code>\flowframchapterheader</code> . . .	170 , 171	<code>\flowframtkSetDynamicEvenHead</code> .	361
		<code>\flowframtkSetDynamicOddEvenFoot</code>	361

<code>\flowframtkSetDynamicOddEvenHead</code>	254, 285, 289–293, 295–302,
.....	304–306, 308–317, 319–321,
<code>\flowframtkSetDynamicOddFoot</code> ..	323, 331, 332, 339, 340, 360–362
<code>\flowframtkSetDynamicOddHead</code> ..	360
<code>\FlowFramTkSetKey</code>	323, 324–326
<code>\flowframtkSetTitle</code>	357, 358
<code>\flowframtkstartobject</code>	329
<code>\flowframtkUseDynamicStyle</code>	359
<code>\flowframtkUseDynamicStyleCsName</code>	
.....	359
<code>\flowframtkUseFrameBorderCs</code> ...	359
<code>\flowframtkUseFrameBorderCsName</code>	359
<code>\flowframtkutils</code>	
....	330–333, 338, 341, 343, 355
<code>\FlowFramTkUtilsOverlayEncapSetup</code>	
.....	355
<code>\FlowFramTkUtilsSetOverlayEncap</code>	354
<code>\FlowFramUnrestoreOR</code>	162
<code>\flowsetexclusion</code>	
.....	82, 380, 381, 541, 542
<code>\flowsetpagelist</code>	
..	82, 154–157, 211–213, 215,
	216, 219, 380, 381, 437–444,
	541, 542, 613–617, 619–621, 623
<code>\flowswitchoffnext</code>	212, 439, 615
<code>\flowswitchoffnextodd</code> ..	213, 439, 616
<code>\flowswitchoffnextoddonly</code>	
.....	220, 444, 624
<code>\flowswitchoffnextonly</code> ..	219, 444, 623
<code>\flowswitchonnext</code>	210, 437, 613
<code>\flowswitchonnextodd</code> ..	211, 437, 614
<code>\flowswitchonnextoddonly</code>	
.....	217, 442, 620
<code>\flowswitchonnextonly</code> ..	215, 441, 619
<code>\fontdimen</code>	187, 467, 655
<code>\footins</code>	431, 433, 434, 605, 608
<code>\footnotecolor</code> ..	174, 433, 434, 608, 608
<code>\footnoterule</code>	434, 608
<code>\footskip</code> ..	165, 205, 433, 477, 607, 667
<code>\fp</code>	299, 300, 302
<code>\frame</code>	363, 519
<code>\framebreak</code> ..	7, 74, 368, 430, 526, 604
<code>\frontmatter</code>	17
G	
<code>\g</code>	12, 15–18, 24–26, 28–30,
	32, 35, 43, 48–50, 52, 76, 93,
	145–162, 165–167, 174–181,
	184, 186, 190, 193–195, 202–
	204, 208–241, 244, 246, 253,
	254, 285, 289–293, 295–302,
	304–306, 308–317, 319–321,
	323, 331, 332, 339, 340, 360–362
<code>\gdef</code>	166, 370, 384, 396,
	397, 407, 474, 506, 514, 530,
	547, 561, 562, 574, 663, 706, 717
<code>\getdynamicbounds</code>	120, 413, 582
<code>\getdynamicevenbounds</code> ..	121, 413, 583
<code>\getdynamicid</code>	105, 398, 563
<code>\getdynamiclabel</code>	105, 398, 563
<code>\getflowbounds</code>	120, 413, 582
<code>\getflowevenbounds</code>	120, 413, 582
<code>\getflowid</code>	80, 372, 532
<code>\getflowlabel</code>	80, 183, 372, 532
<code>\getstaticbounds</code>	119, 413, 581
<code>\getstaticevenbounds</code> ..	119, 413, 582
<code>\getstaticid</code>	85, 385, 548
<code>\getstaticlabel</code>	85, 385, 548
<code>\Ginput@path</code>	355
<code>\global</code>	30, 75–77, 80, 142,
	145, 159, 162–164, 208–210,
	368–371, 383, 387–389, 393,
	395, 400–403, 405, 428, 430–
	434, 462, 464, 472–474, 476,
	480, 481, 491, 495, 496, 508,
	509, 513–515, 526–531, 546,
	551–553, 558, 560, 566–569,
	572, 601, 604–607, 609, 649,
	651, 661–663, 665, 670, 671,
	687, 692, 693, 709, 710, 716–718
<code>\globalnormalmargin</code>	
.....	77, 78, 368, 369, 528, 529
<code>\globalreversemargin</code>	
.....	77, 78, 368, 369, 528, 529
<code>\glossary</code>	89, 164, 432, 607
<code>\group</code> ..	15, 31, 32, 74, 75, 103, 110,
	183, 184, 205, 206, 318, 319, 355
H	
<code>\hangafter</code>	390, 553, 554
<code>\hangindent</code>	390, 553, 554
<code>\hb@xt@</code>	465,
	466, 470, 472, 473, 476, 477,
	653, 654, 659, 661, 662, 665–667
<code>\hbox</code>	4,
	75, 76, 159, 160, 184–186, 191,
	194–196, 202–207, 326, 327,
	355, 368, 428, 432, 477, 479,
	480, 527, 601, 606, 667, 669–671

<code>\headheight</code>	30, 161, 165,	<code>\if@setfr@mes</code>	145, 428, 601
	205, 408, 432, 433, 576, 606, 607	<code>\if@specialpage</code>	163, 480, 670
<code>\headsep</code>	30, 165, 205,	<code>\if@storetoc</code>	312, 512, 714
	408, 432, 433, 477, 576, 607, 667	<code>\if@ttb@num</code>	
<code>\hfil</code>	389, 553		12, 364, 508, 509, 521, 710, 711
<code>\hfill</code>	88, 103, 169, 170,	<code>\if@ttb@rotate</code> 10, 364, 507, 520, 707	
	280, 389, 408, 498, 553, 575, 696	<code>\if@ttb@title</code>	12, 364, 508, 509, 521, 709, 710
<code>\hNtone</code>	273, 494, 691	<code>\if@twocolumn</code>	148, 162
<code>\hNtoneleft</code>	274, 494, 691	<code>\if@twoside</code>	11, 76, 164,
<code>\hNtoneright</code>	274, 494, 495, 692		167–169, 268–274, 276, 368,
<code>\hoffset</code>	114, 408, 576		369, 465, 466, 470, 472, 473,
<code>\hrule</code>	4		476, 477, 479, 480, 490–496,
<code>\hrulefill</code>	320, 321, 516, 719, 720		527, 528, 653, 654, 658, 661,
<code>\hsize</code>	158, 159, 162, 164,		662, 665, 666, 669, 670, 686–693
	371, 428, 430, 530, 531, 601, 604	<code>\ifaligntoc</code>	16, 515, 717, 718
<code>\hskip</code>	4, 13, 363, 519	<code>\IfBooleanT</code>	79, 84, 104, 250
<code>\hspace</code>	88, 283, 502, 700	<code>\IfBooleanTF</code>	81,
<code>\hss</code>	184–187, 191,		86, 91, 99, 106, 110, 113, 115,
	194–196, 202–206, 208, 465,		119–132, 134, 135, 137, 144,
	466, 470, 472, 473, 476–478,		145, 149–153, 210–212, 214,
	653–655, 659, 661, 662, 665–667		216, 217, 219–224, 226, 227,
<code>\ht</code>	206, 478, 667		230–232, 234, 235, 237, 238,
<code>\htwotone</code>	271, 493, 689, 690		240, 241, 277, 281, 284, 285, 319
<code>\htwotoneleft</code>	272, 493, 690	<code>\ifcase</code>	364, 365, 392,
<code>\htwotoneright</code>	272, 493, 494, 690		410, 411, 474, 482, 485, 489,
<code>\Huge</code>	286, 287		498, 501, 520–522, 557, 578,
<code>\huge</code>	286, 287		580, 663, 674, 679, 684, 696, 700
<code>\hyperlink</code>	17, 507, 707	<code>\IfClassLoadedTF</code>	16
<code>\hypersetup</code>	357, 358	<code>\ifcsdef</code>	163
		<code>\ifcsempy</code> 381, 394, 403, 543, 558, 569	
		<code>\ifcsundef</code>	402, 568
		<code>\ifdef</code>	19
		<code>\ifdefempty</code> 378–380, 387–389, 400–	
			403, 539–542, 551–553, 566–569
		<code>\ifdefequal</code>	391, 555
		<code>\ifdfcontinued</code>	110, 405, 571, 572
		<code>\ifdim</code>	368, 409, 410, 414–425,
			428–431, 527, 577, 578, 584–
			587, 589, 591–597, 602, 604, 605
		<code>\iffalse</code>	370, 529
		<code>\iffvadjust</code>	242, 243,
			245, 248, 250, 253, 257, 260,
			262, 265, 481, 483–489, 672,
			673, 675–677, 679, 681, 682, 684
		<code>\IfFileExists</code>	513, 715
		<code>\ifFLFabove</code>	121, 136,
			140, 141, 413, 426, 583, 598, 599
		<code>\ifFLFbelow</code>	121,
			136, 141, 413, 426, 427, 583, 599

I

<code>\if</code>	57
<code>\if@contsline</code>	313, 513, 514, 716
<code>\if@endfor</code>	434, 435, 609, 610
<code>\if@fcolmade</code>	
	209, 430, 431, 480, 605, 606, 671
<code>\if@ff@moreframes</code>	
	174, 434, 435, 462, 609, 610, 649
<code>\if@ff@nwpg</code>	177, 461, 480, 648, 670
<code>\if@files</code>	312, 512, 715
<code>\if@firstcolumn</code>	431, 605
<code>\if@mainmatter</code>	171, 286, 287
<code>\if@minitoc</code> 321, 514–516, 716, 718, 720	
<code>\if@newlist</code>	163, 164, 432, 433, 607
<code>\if@notthiscol</code> 177, 368, 437–443,	
	445–463, 465, 466, 470, 472,
	473, 476, 527, 613–622, 625–
	634, 637–646, 648–650, 653,
	654, 658, 661, 662, 665, 666
<code>\if@openright</code>	288

<code>\ifFLleft</code> . . .	121, 136, 137, 140,	<code>\ifthenelse</code>	363, 365, 366, 369, 371–
	141, 413, 426, 427, 583, 598–600		377, 382, 385, 386, 395, 397–
<code>\ifFLright</code> . . .	121, 136, 137, 141,		399, 404, 435, 463, 464, 467–
	142, 413, 426, 427, 583, 598–600		476, 483, 491, 495–500, 502–
<code>\iflefttorightcolumns</code>			505, 508, 509, 511–513, 517–
	15, 245, 246, 248,		519, 523, 524, 528, 530–538,
	253, 257, 262, 266, 366, 481–		544, 547, 548, 550, 559, 560,
	483, 485, 486, 488, 490, 523,		562, 563, 565, 570, 611, 650,
	673–675, 678, 680, 682, 683, 685		652, 655–665, 676, 687, 692,
<code>\ifnum</code>	171, 172,		694, 695, 697–701, 704, 705,
	286, 287, 368, 371, 372, 377,		709, 710, 713, 714, 716, 720, 721
	382, 385–387, 393, 395, 397–	<code>\iftrue</code>	369, 529
	399, 404, 406, 407, 409–412,	<code>\ifundef</code>	
	428–431, 434–436, 438, 440,		380, 389, 402, 541, 542, 552, 568
	441, 446–449, 454–457, 462,	<code>\ifusedframebreak</code>	
	482, 483, 485, 487, 489, 490,		75, 368, 430, 526, 604
	492, 495, 497, 498, 501, 506,	<code>\IfValueT</code>	90,
	510, 514, 516, 527, 530, 531,		96, 99, 109, 110, 113, 150, 151,
	533, 539, 544, 547, 548, 550,		153, 243–245, 247, 249–252,
	557, 560, 562, 563, 565, 571,		254–256, 258–264, 266, 267, 355
	573, 574, 577–581, 601, 603–	<code>\IfValueTF</code>	82, 89, 100, 101, 108,
	605, 609–611, 614–618, 626–		111, 148, 150, 250, 276, 305, 355
	630, 638, 640–642, 649, 650,	<code>\ifvbox</code>	434, 609
	674, 675, 679, 680, 683, 685,	<code>\ifvmode</code>	75, 368, 527
	688, 693, 695, 696, 699, 700,	<code>\ifvoid</code>	431, 433, 465, 605, 608, 653
	706, 707, 711, 712, 716, 717, 720	<code>\ifx</code>	366, 391,
<code>\ifodd</code>	76,		405, 406, 431, 436, 507, 514,
	164, 167–169, 368, 369, 414,		524, 555, 572, 605, 612, 708, 717
	415, 417–419, 438–440, 442–	<code>\ignorespaces</code>	103,
	446, 448, 450, 451, 453–461,		191, 196, 203, 205, 242, 392,
	463, 465, 466, 470, 472, 473,		406–408, 471, 473, 476, 477,
	476, 477, 479, 480, 527, 528,		556, 572, 575, 659, 662, 665–667
	583–586, 588, 590, 614–617,	<code>\ignorespacesafterend</code>	96, 242
	620–622, 624, 626, 628, 629,	<code>\immediate</code>	312, 512, 715
	632–634, 636–638, 640, 641,	<code>\includeteximage</code>	355
	644–646, 648, 650, 653, 654,	<code>\indent</code>	389, 390, 553
	658, 661, 662, 665, 666, 669, 670	<code>\index</code>	89, 164, 432, 607
<code>\IfPackageLoadedT</code>	30	<code>\input</code>	327
<code>\IfPackageLoadedTF</code>	323	<code>\input@path</code>	355
<code>\IfSavedRelativeLocationAbove</code> . .	143	<code>\inserthrule</code>	280, 499, 698
<code>\IfSavedRelativeLocationBelow</code> . .	143	<code>\insertvrule</code>	276, 276, 496, 693, 694
<code>\IfSavedRelativeLocationEq</code>	142	<code>\int</code> 11–13, 32–35, 43–52, 54–67, 70–	
<code>\IfSavedRelativeLocationLeft</code> . .	143		81, 84–86, 94, 95, 97, 98, 102–
<code>\IfSavedRelativeLocationRight</code> . .	144		106, 111, 112, 117, 118, 122,
<code>\ifshowframebbox</code>			124–127, 129, 136, 137, 139,
	4, 208, 363, 479, 519, 669		140, 145–159, 161, 162, 165–
<code>\ifshowmargins</code>			167, 172, 175–181, 183–186,
	4, 208, 363, 479, 519, 669		190, 191, 193–195, 197–199,
<code>\ifshowtypeblock</code>			201, 202, 204, 206–242, 244–
	4, 208, 363, 479, 519, 669		248, 250, 253, 254, 256, 257,

265–267, 269–271, 273–275, 277, 279, 281, 283–285, 291, 292, 294–297, 300–302, 304– 317, 319–321, 329, 331, 332, 334, 338–340, 342–345, 360–362	198, 199, 270, 275, 284, 289, 290, 327–329, 355, 362, 363, 366, 369, 370, 378, 387, 389– 393, 400, 402, 405, 406, 430– 434, 462, 474, 480, 492, 495, 503, 505, 507, 512, 515, 517, 518, 524, 529, 539, 551–557, 566, 568, 572, 573, 605–607, 609, 649, 663, 670, 688, 693, 702, 704, 705, 708, 714, 718, 721
\interlinepenalty 286, 288	\lineskip 164, 432, 607
\ior 312, 313	\lineskiplimit 164, 432, 607
J	
\jdrimagebox <u>326</u>	\linewidth 94, 159, 321, 430, 516, 604, 720
\jdroutline 327, 328	\long 76, 405, 407, 514, 571, 572, 574, 717
\jobname 312, 512, 715	\loop 435, 462, 483, 486, 490, 610, 649, 675, 680, 685
K	
\kern 183–187, 191, 194– 196, 202–205, 465, 466, 470, 472, 473, 476, 477, 479, 653– 655, 659, 661, 662, 665, 666, 669	LR (option) <u>523</u>
\keys 16, 20, 30, 35, 40, 90, 323, 332, 334–337, 355, 356, 358	
L	
\l 10, 12–14, 16, 19, 22, 23, 25–28, 32–43, 45–53, 55, 64–68, 70, 71, 75–78, 80–83, 85, 87–102, 105–117, 119–121, 123, 125– 128, 130–134, 136–140, 146, 147, 149–153, 158, 159, 167– 169, 173–176, 178–183, 187– 193, 198–201, 205–207, 210– 246, 248, 250, 251, 253, 254, 257, 259, 260, 262, 263, 265– 285, 289, 290, 296–303, 305– 311, 314, 315, 317, 319– 323, 328, 329, 331–358, 360–362	M
\label 31, 32, 89, 164, 367, 432, 525, 607	\m@ne . 75, 171, 172, 286, 287, 368, 527
\labelflow <u>32</u> , 367, <u>525</u>	\mainmatter 18
\labelflowidn <u>31</u> , 367, <u>525</u>	\makeatletter 312, 512, 715
\language 163	\makebackgroundframe . . <u>275</u> , 495, <u>693</u>
\leavevmode 4, 56	\makebox 13, 191, 207, 363, 471, 477, 479, 519, 659, 667, 669
\leftmark 170, 171	\makedfheaderfooter <u>165</u> , 433, <u>607</u> , 608
\leftskip 329	\makethumbtabs 8, <u>293</u> , 504, <u>704</u>
\lefttorightcolumnfalse 29, 366, <u>523</u>	\MakeUppercase 21, 22
\lefttorightcolumnstrue 15, 29, 366, <u>523</u>	\marginparsep 190, 191, 470, 659
\legacy . 17, 77, 158, 184, 186, 190, 194, 195, 202, 204, 208, 288, 317	\marginparwidth 190, 191, 267, 470, 490, 659, 685
\let 21, 26, 43, 88, 89, 94, 97, 138, 142, 161– 164, 166, 170, 171, 173, 174,	\markboth 170–172, 289, 290
	\markright 170, 172, 289, 290
	\maxdepth 434, 609
	\maxdimen 431, 605
	maxdynamic (counter) <u>32</u> , <u>525</u>
	maxflow (counter) <u>31</u> , <u>525</u>
	maxstatic (counter) <u>32</u> , <u>525</u>
	\mbox 74, 206, 280, 355, 368, 389–391, 478, 498, 526, 553, 554, 667, 696
	\MessageBreak 178, 428, 430, 462, 601, 604, 649
	\minitocstyle . 318, <u>318</u> , 515, 718, <u>718</u>
	\mod 327
	\moveright 164, 432, 607
	\msg 5–10, 35, 38, 46, 47, 49–54, 56–58, 61, 67, 72– 74, 92, 95, 98, 100, 117, 118,

139, 142, 145, 148, 150–155,
 158, 166, 178, 184, 247, 250,
 256, 265, 275, 285, 293–295,
 308, 310, 313, 321, 327, 328, 363
 \multiply 490, 508, 685, 709

N

\Ncolumn 246, 482, 673
 \Ncolumnbottom 263, 488, 489, 683, 683
 \Ncolumnbottominarea
 263, 264, 488–490, 683, 683, 685
 \NcolumnDbottom 264, 489, 683
 \NcolumnDbottominarea
 266, 275, 490, 685
 \NcolumnDtop 255, 485, 678
 \NcolumnDtopinarea 258, 487, 680
 \Ncolumninarea 247, 247, 482, 673, 674
 \NcolumnSbottom 264, 489, 683
 \NcolumnSbottominarea . 266, 490, 685
 \NcolumnStop 255, 485, 678
 \NcolumnStopinarea 257, 487, 680
 \Ncolumnstop 255, 485, 678, 678
 \Ncolumnstopinarea
 256, 485–487, 678, 679, 680
 \NeedsTeXFormat 3, 322, 363, 518
 \newboolean 383,
 384, 396, 397, 546, 547, 560, 562
 \newbox 96, 370, 383, 392, 529, 546, 556
 \newcommand . 3, 4, 6, 12, 13, 17, 19,
 20, 43, 80, 83, 85, 86, 88, 95,
 102, 103, 105, 106, 134, 135,
 139, 142–144, 159–161, 163,
 165, 169–174, 209, 267, 270,
 275, 276, 280, 283, 285–288,
 292, 294, 295, 299, 318, 323,
 327–330, 357, 359, 363–372,
 376–378, 381–387, 389–400,
 403, 404, 406–411, 413–430,
 432–479, 481–499, 501–508,
 510–513, 515–520, 522–529,
 531, 532, 538, 539, 542–550,
 553, 555–566, 569, 570, 573–
 578, 580–603, 606–609, 611–
 617, 619, 620, 622–633, 635–
 645, 647, 648, 650–669, 672–
 700, 702–709, 711–716, 718–721
 \newcount 313, 367, 436, 461, 480,
 490, 492, 508, 513, 526, 611,
 648, 670, 685, 688, 708, 715, 716

\newcounter
 12, 31, 32, 292, 293, 321, 365,
 367, 504, 516, 522, 525, 703, 720
 \NewDocumentCommand 16, 30–
 32, 74–82, 84–86, 88, 89, 99,
 100, 103, 105, 106, 108, 110,
 113–115, 119–133, 135, 137,
 142–147, 149, 150, 161, 162,
 165, 166, 177, 181, 210–213,
 215, 217, 219–224, 226, 227,
 230–232, 234, 237, 238, 240,
 241, 243, 245, 247–252, 254–
 256, 258–261, 263, 264, 266–
 268, 270–275, 277, 280, 284,
 285, 292, 293, 295, 298, 304,
 305, 307, 310, 311, 318, 319,
 321, 326–329, 354, 355, 358–362
 \NewDocumentEnvironment . . . 96, 109
 \newdynamicframe 103, 395,
 433, 508, 509, 560, 607, 709, 710
 \newenvironment 242, 392,
 405, 406, 481, 556, 571, 573, 671
 \newflowframe 78, 369, 483, 486–488,
 490, 529, 529, 675, 680–683, 685
 \newframe 249, 250, 253,
 257, 260, 262, 265, 483–489,
 675, 676, 677, 679, 681, 682, 684
 \newif . 4, 15, 16, 75, 121, 145, 177,
 242, 363, 364, 366, 368, 369,
 405, 413, 428, 434, 461, 481,
 512, 513, 515, 516, 519, 521,
 523, 526, 529, 572, 583, 601,
 609, 648, 672, 714, 716, 717, 720
 \newlength 3, 30,
 33, 74, 115, 158, 248, 276, 298,
 318, 363, 367, 370, 383, 396,
 408, 429, 479, 481, 483, 484,
 490, 496, 506, 515, 518, 526,
 530, 546, 561, 576, 603, 669,
 672, 675, 676, 686, 694, 707, 718
 \newline 88
 \NewMirroredHookPair . . 98, 288, 289
 \newpage 75, 76, 76, 368, 527, 527
 \newread 513, 715
 \newsavebox 288
 \newstaticframe
 . 84, 383, 491, 495, 496, 498,
 501, 545, 687, 692, 693, 696, 700
 \newtoks 405, 407, 571, 574
 \newwrite 312, 512, 715

<code>\nfss@catcodes</code>	163	<code>\onecolumninarea</code>	
<code>\nobreak</code>	286–288	243, 481, 482, 672, 674
<code>\NoChapterInDynamic</code>	292	<code>\onecolumnSbottom</code>	258, 487, 681
<code>\noexpand</code> 163, 284, 285, 392, 393,		<code>\onecolumnSbottominarea</code>	260, 487, 681
406, 432, 437–461, 503, 507–		<code>\onecolumnStop</code> 249, 429, 483, 602,	675
511, 557, 572, 606, 613–617,		<code>\onecolumnStopinarea</code>	
619–621, 623–631, 633, 635–		251, 251, 484, 676
643, 645, 647, 648, 703, 708–712		<code>\onecolumnstop</code>	248, 483, 675
<code>\noindent</code>	75, 98, 103, 320,	<code>\onecolumnstopinarea</code>	250,
321, 326, 368, 393, 406–408,		483, 484, 486, 675, 676, 677, 679	
516, 526, 557, 572, 575, 719, 720		<code>\openin</code>	513, 715
<code>\normalcolor</code>	160, 318, 366,	<code>\openout</code>	312, 512, 715
432, 433, 515, 524, 606, 608, 718		options:	
<code>\normalfont</code> ...	286, 287, 318, 515, 718	color	522
<code>\normalsfcodes</code>	164, 432, 607	draft	519
<code>\normalsize</code> 164, 318, 432, 515, 607, 718		final	519
<code>norotate</code> (option)	520	LR	523
<code>\not</code>	513, 715	norotate	520
<code>\nr</code>	364, 365	pages	522
<code>\number</code>	371, 372, 381,	RL	523
384, 385, 394, 397, 398, 403,		rotate	520
406, 407, 410–412, 430, 436–		thumbtabs	521
462, 464, 465, 470, 472, 475,		ttbnonum	521
482, 486, 489, 503, 506, 508,		ttbnotitle	521
509, 511, 512, 516, 531, 532,		ttbnum	521
542, 543, 547, 548, 558, 559,		ttbtitle	521
562, 563, 569, 573, 574, 578–		verbose	520
581, 604, 612–649, 651–653,		<code>\OR</code>	368, 527
659, 660, 665, 674, 679, 684,		<code>\or</code> ..	364, 365, 393, 410–412, 435,
703, 707, 709, 710, 712–714, 720		463, 474, 482, 486, 489, 498,	
		501, 511, 512, 520–522, 557,	
		578–581, 611, 650, 663, 664,	
		674, 679, 684, 696, 700, 713, 714	
		<code>\output</code>	430, 604
		<code>\outputpenalty</code>	430, 431, 605
O			
<code>\oddcheckifframeabove</code> .	123, 414, 584	P	
<code>\oddcheckifframebelow</code> .	125, 415, 586	<code>\p@</code>	286, 287
<code>\oddcheckifframeleft</code> ..	127, 417, 588	<code>\PackageError</code>	
<code>\oddcheckifframeright</code> .	129, 419, 590	372–376, 380, 385, 389–
<code>\oddsidemargin</code>		392, 397, 398, 402, 406, 407,	
.....	114, 164, 408, 479, 576, 669	410–412, 414–425, 430, 482,	
<code>\oldpar</code>	474, 663, 664	484–486, 489, 496, 497, 499,	
<code>\onecolumn</code>	148, 154,	500, 502–504, 517, 532–537,	
162, 242, 290, 429, 481, 603, 672		542, 548, 553–556, 563, 564,	
<code>\onecolumnbottom</code>		568, 573–575, 578–581, 584–	
.....	258, 259, 487, 680, 681	597, 604, 674, 676, 679, 684,	
<code>\onecolumnbottominarea</code>		695, 697–699, 701, 702, 704, 720	
259, 487, 489, 680, 681, 681, 684		<code>\PackageInfo</code>	
<code>\onecolumnDbottom</code>	259, 487, 681	364, 462, 513, 520, 649, 715
<code>\onecolumnDbottominarea</code>	260, 487, 681		
<code>\onecolumnDtop</code>	249, 483, 675		
<code>\onecolumnDtopinarea</code> ..	251, 484, 677		

<code>\PackageWarning</code> . . .	428–430, 462, 465, 495, 510–512, 601, 603, 604, 649, 653, 693, 711, 713, 714
<code>\pagebreak</code>	75, 368, 526
<code>pages (option)</code>	522
<code>\pagestyle</code>	166, 363
<code>\pagetotal</code>	75, 368, 527
<code>\paperheight</code>	115, 268, 270, 276, 408, 409, 490, 492, 496, 576, 577, 686, 688, 693
<code>\paperwidth</code>	115, 272, 273, 276, 408, 409, 493, 494, 496, 576, 577, 690, 691, 693
<code>\par</code> . . .	75, 88, 94, 97, 103, 198, 199, 242, 286–288, 320, 321, 368, 389, 390, 393, 406–408, 430, 474, 516, 526, 553, 554, 557, 572, 575, 605, 663, 664, 719, 720
<code>\paragraph</code> . . .	95, 391, 392, 555, 556
<code>\parbox</code>	11, 290, 321, 474, 475, 507, 516, 663–665, 707, 720
<code>\parfillskip</code>	75, 103, 329, 368, 408, 526, 575
<code>\parindent</code> . . .	88, 98, 198, 199, 286, 287, 329, 393, 474, 557, 663, 664
<code>\parshape</code> . . .	88, 94, 390, 391, 553, 555
<code>\parskip</code>	75, 368, 389, 526, 553
<code>\pdfannot@link@off@@</code>	160
<code>\pdfannot@link@on@@</code>	160
<code>\pdfextension</code>	357
<code>\pdfinfo</code>	357
<code>\penalty</code>	75, 368, 527
<code>\pgfextractx</code>	339, 341
<code>\pgfextracty</code>	339, 341
<code>\pgfpathlineto</code> . . .	346–351, 353, 354
<code>\pgfpathmoveto</code>	346–352, 354
<code>\pgfpoint</code>	346–354
<code>\pgfsetarrowsend</code>	338
<code>\pgfsetarrowsstart</code>	338, 345
<code>\pgfsetstrokecolor</code>	337, 345
<code>\pgftext</code>	346–349, 351–354
<code>\pgfusepath</code>	346–350, 352–354
<code>\preto</code>	76, 368, 527
<code>\prevgraf</code>	390, 391, 553, 554
<code>\prg</code> . . .	14, 17, 18, 47, 48, 57, 60, 62–65, 67–70, 72, 80, 81, 85, 105, 291
<code>\ProcessKeyOptions</code>	30, 323
<code>\ProcessOptionsX</code>	366, 524
<code>\prop</code>	48–50, 52, 53, 55, 57, 60, 72, 331, 335–337, 340
<code>\protect</code>	163, 292–294, 432, 505, 606, 705, 706
<code>\protected@edef</code>	405, 572
<code>\protected@write</code>	43, 89, 138
<code>\providecommand</code>	76, 88
<code>\ProvideDocumentCommand</code>	355, 357, 358
<code>\ProvidesPackage</code> . . .	3, 322, 363, 518
<code>\ps@empty</code>	166, 174, 363
<code>\ps@ffempty</code>	166, 174, 363
<code>\ps@ffheadings</code>	166, 173, 363
<code>\ps@ffmyheadings</code>	166, 173, 362
<code>\ps@ffplain</code>	166, 173, 363
<code>\ps@headings</code>	166, 363
<code>\ps@myheadings</code>	166, 362
<code>\ps@plain</code>	166, 363
<code>\ps@specialeempty</code>	174
<code>\ps@specialffempty</code>	174
<code>\pscharpath</code>	327
Q	
<code>\q</code>	41, 42, 44, 70, 71, 112, 113, 269, 275, 340, 342
<code>\quad</code>	19
<code>\quark</code>	71, 112, 340–342
R	
<code>\raggedright</code>	286, 287
<code>\raise</code>	465, 466, 470, 472, 473, 476, 477, 653, 654, 659, 661, 662, 665, 666
<code>\read</code>	513, 716
<code>\RefSavedRelativeLocation</code>	142
<code>\refstepcounter</code>	293
<code>\relativeframelocation</code>	135, 425, 598
<code>\relax</code>	43, 85, 89, 95, 104, 138, 142, 147, 163, 208, 270, 275, 289, 290, 312, 364–369, 371, 372, 376, 377, 382–387, 391–393, 395–400, 404–406, 409–412, 414–425, 428–430, 434–462, 467, 471, 473, 474, 478–483, 485, 486, 488–490, 492–495, 497, 498, 501, 506–508, 510–516, 520–522, 524, 525, 527, 529–533, 538, 539, 544, 546–550, 555, 557, 559–565, 570–573, 577–581, 584–589, 591–597, 601–604, 609–624, 626–638, 640–650, 655, 659, 660, 662, 663, 667–671, 673–675,

677, 679, 680, 682–685, 688,
 689, 691–693, 695, 696, 699,
 700, 706–709, 711–713, 715–720
 \reldynamicloc [144](#), [427](#), [600](#)
 \relflowloc [145](#), [427](#), [601](#)
 \relstaticloc [144](#), [427](#), [600](#)
 \renewcommand
 . [19–21](#), [24](#), [25](#), [89](#), [161](#), [162](#),
 [208](#), [284](#), [287](#), [288](#), [319](#), [355](#),
 [360–362](#), [364–366](#), [429](#), [433](#),
 [481](#), [503](#), [504](#), [512](#), [515–517](#),
 [520](#), [522–524](#), [603](#), [607](#), [608](#),
 [672](#), [673](#), [702](#), [704](#), [714](#), [718–720](#)
 \RenewDocumentCommand [17](#),
 [76](#), [96](#), [110](#), [148](#), [162](#), [242](#), [244](#)
 \repeat [435](#), [462](#), [483](#),
 [487](#), [490](#), [610](#), [649](#), [675](#), [680](#), [685](#)
 \RequirePackage [3](#),
 [30](#), [323](#), [327](#), [363](#), [366](#), [518](#), [524](#)
 \RerunFileCheck [312](#)
 \reset@font [164](#), [432](#), [607](#)
 \rightmark [170](#), [171](#)
 \rightskip [329](#)
 RL (option) [523](#)
 \romannumeral
 [48](#), [50–63](#), [65](#), [66](#), [72](#), [73](#), [97](#),
 [294–296](#), [315](#), [316](#), [318](#), [320](#),
 [321](#), [369–372](#), [378–389](#), [392–](#)
 [394](#), [396–398](#), [400–404](#), [406](#),
 [407](#), [411](#), [412](#), [428](#), [430](#), [433–](#)
 [435](#), [462–477](#), [491](#), [495](#), [496](#),
 [498](#), [501](#), [506](#), [508–511](#), [513–](#)
 [516](#), [528–532](#), [539–553](#), [556–](#)
 [564](#), [566–570](#), [573](#), [574](#), [579](#),
 [581](#), [601](#), [602](#), [604](#), [607–610](#),
 [649–666](#), [687](#), [692](#), [693](#), [696](#),
 [700](#), [706](#), [707](#), [709–712](#), [716–720](#)
 rotate (option) [520](#)
 \rotatebox [11](#), [181](#), [364](#), [507](#), [520](#), [707](#)
 \rotateframe [181](#), [182](#), [192](#), [200](#), [364](#),
 [464](#), [471](#), [475](#), [520](#), [652](#), [660](#), [664](#)
 \rule [280](#), [283](#), [498](#), [502](#), [696](#), [700](#)

S

\SaveRelativeFrameLocation [137](#)
 \sbox [290](#), [393](#), [473](#), [558](#), [662](#)
 \sdfparindent [33](#), [85](#), [104](#),
 [367](#), [393](#), [474](#), [526](#), [557](#), [663](#), [664](#)
 \section [19](#), [95](#), [391](#), [555](#)
 \sectionmark [19](#), [170–174](#)
 \seq [35](#), [46](#), [47](#),
 [178](#), [297–302](#), [304–306](#), [308–311](#)
 \set@typeset@protect .. [163](#), [432](#), [607](#)
 \setalldynamicframes
 [106](#), [399](#), [564](#), [565](#)
 \setallflowframes [81](#), [146](#), [147](#), [246](#),
 [376](#), [377](#), [428](#), [429](#), [538](#), [538](#), [602](#)
 \setallstaticframes [86](#), [386](#), [549](#), [550](#)
 \setboolean [378](#),
 [383](#), [384](#), [387](#), [389](#), [396](#), [397](#),
 [400](#), [402](#), [403](#), [539](#), [546](#), [547](#),
 [551](#), [553](#), [561](#), [562](#), [566](#), [568](#), [569](#)
 \setbox [146](#), [206](#), [209](#), [327](#), [428](#), [431–](#)
 [434](#), [464](#), [477](#), [480](#), [601](#), [605](#),
 [606](#), [608](#), [609](#), [651](#), [667](#), [670](#), [671](#)
 \setcounter
 .. [18](#), [23](#), [164](#), [433](#), [516](#), [607](#), [720](#)
 \setdynamiccontents
 [9](#), [109](#), [110](#), [406](#), [573](#)
 \setdynamicframe
 [106](#), [399](#), [511](#), [565](#), [712](#)
 \setffdraftcolor [3](#), [13](#), [363](#), [518](#), [519](#)
 \setffdrafttypeblockcolor
 [3](#), [207](#), [363](#), [479](#), [518](#), [669](#)
 \setflowframe [81](#), [245](#), [246](#),
 [253](#), [254](#), [262](#), [263](#), [377](#), [482](#),
 [485](#), [488](#), [538](#), [673](#), [678](#), [682](#), [683](#)
 \setframes [145](#), [148](#), [428](#), [429](#), [601](#), [603](#)
 \setinitialframe [145](#), [428](#), [601](#)
 \setkeys . [378](#), [387](#), [400](#), [539](#), [551](#), [566](#)
 \setlength [3](#), [30](#), [115](#), [158](#),
 [248](#), [276](#), [298](#), [318](#), [370](#), [371](#),
 [378](#), [379](#), [381](#), [383](#), [384](#), [387](#),
 [388](#), [393](#), [394](#), [396](#), [400](#), [401](#),
 [403](#), [404](#), [408–412](#), [428](#), [429](#),
 [433](#), [466](#), [468–470](#), [474](#), [477](#),
 [479](#), [481](#), [483–498](#), [501](#), [506](#),
 [508](#), [509](#), [515](#), [530](#), [531](#), [540](#),
 [541](#), [543](#), [546](#), [551](#), [557](#), [559](#),
 [561](#), [566](#), [567](#), [570](#), [575–581](#),
 [601](#), [603](#), [607](#), [655–659](#), [663](#),
 [664](#), [667](#), [669](#), [672–682](#), [684–](#)
 [696](#), [699](#), [700](#), [707](#), [709](#), [710](#), [718](#)
 \setmargin [78](#), [159](#), [369](#), [430](#), [528](#), [604](#)
 \SetOneColumnFrame [149](#)
 \setstaticcontents [9](#), [99](#),
 [146](#), [147](#), [279](#), [282](#), [393](#), [428](#),
 [429](#), [498](#), [501](#), [558](#), [602](#), [696](#), [700](#)
 \setstaticframe ... [86](#), [280](#), [283](#),
 [386](#), [498](#), [501](#), [502](#), [549](#), [696](#), [700](#)

<code>\setthumbtab</code>	307 , 511 , 712	<code>\staticswithchoffnextonly</code>	
<code>\setthumbtabindex</code>	310 , 511 , 713		240 , 460 , 647
<code>\SetThumbTabTitle</code>	295	<code>\staticswithchonnext</code>	231 , 453 , 636
<code>\settodepth</code>	146 , 147 , 428 , 429 , 602	<code>\staticswithchonnextodd</code>	232 , 454 , 637
<code>\settoheight</code>	146 , 147 , 428 , 429 , 602	<code>\staticswithchonnextoddone</code>	238 , 458 , 643
<code>\SetTwoColumnFrames</code>	150	<code>\staticswithchonnextonly</code>	237 , 457 , 642
<code>\sffamily</code>	4 , 363 , 518	<code>statictable (env.)</code>	242 , 671
<code>\sfswapoddeven</code>	100 , 394 , 559	<code>\stepcounter</code>	159 , 162 , 164 , 294 , 321 , 430 , 433 , 506 , 517 , 518 , 604 , 607 , 706 , 721
<code>\Shapepar</code>	95 , 391 , 555	<code>\string</code>	43 , 89 , 138 , 391 , 392 , 402 , 407 , 429 , 430 , 555 , 556 , 568 , 575 , 603 , 604
<code>\shapepar</code>	94 , 95 , 391 , 392 , 555 , 556	<code>\subparagraph</code>	95 , 391 , 392 , 555 , 556
<code>\shipout</code>	163 , 432 , 607	<code>\subsection</code>	95 , 391 , 392 , 555
<code>\showframebboxfalse</code>	4 , 363 , 519	<code>\subsectionmark</code>	171 – 174
<code>\showframebboxtrue</code>	4 , 363 , 519	<code>\subsubsection</code>	95 , 391 , 392 , 555 , 556
<code>\showmarginfalse</code>	4 , 363 , 519	<code>\sys</code>	328
<code>\showmargintrue</code>	4 , 363 , 519		
<code>\showtypeblockfalse</code>	4 , 363 , 519		
<code>\showtypeblocktrue</code>	4 , 363 , 519		
<code>\simpar</code>	88 , 389 , 553		
<code>\skip</code>	329 , 434 , 608		
<code>\small</code>	4 , 103 , 363 , 408 , 518 , 575		
<code>\space</code>	286 , 287 , 381 , 394 , 402 , 403 , 406 , 407 , 429 – 431 , 464 , 465 , 482 , 486 , 489 , 507 , 542 , 543 , 558 , 559 , 568 , 569 , 573 – 575 , 603 – 605 , 651 , 653 , 674 , 679 , 684 , 708		
<code>\staticaddeexclusion</code>	99 , 241 , 242 , 394 , 460 , 461 , 558 , 647 , 648		
<code>staticcontents (env.)</code>	96 , 556		
<code>staticcontents* (env.)</code>	96 , 556		
<code>staticfigure (env.)</code>	242 , 671		
<code>\staticframe</code>	96 , 98 , 392 , 393 , 556 – 558		
<code>\staticframeevenx</code>	86 , 386 , 411 , 412 , 549 , 579 , 581		
<code>\staticframeeveny</code>	86 , 386 , 411 , 412 , 549 , 579 , 581		
<code>\staticframeheight</code>	86 , 97		
<code>\staticframewidth</code>	86 , 97		
<code>\staticframex</code>	86 , 385 , 410 , 549 , 579		
<code>\staticframey</code>	86 , 386 , 411 , 549 , 579		
<code>\staticsetexclusion</code>	99 , 389 , 394 , 552 , 558		
<code>\staticsetpagelist</code>	99 , 232 – 234 , 236 , 237 , 240 , 389 , 394 , 453 – 460 , 552 , 558 , 637 – 643 , 645 , 647		
<code>\staticswithchoffnext</code>	233 , 455 , 639		
<code>\staticswithchoffnextodd</code>	234 , 456 , 640		
<code>\staticswithchoffnextoddone</code>	241 , 461 , 647		
<code>\staticswithchoffnextonly</code>	240 , 460 , 647		
<code>\staticswithchonnext</code>	231 , 453 , 636		
<code>\staticswithchonnextodd</code>	232 , 454 , 637		

<code>\theHdisplayedframe</code> . . .	31, 367, 525	<code>\topskip</code>	75, 368, 527
<code>\theHthisframe</code>	31, 367, 525	<code>ttbnonum</code> (option)	521
<code>\thepage</code> . . .	31, 43, 170, 171, 173, 367, 431, 505, 525, 605, 705, 706	<code>ttbnotitle</code> (option)	521
<code>\thepart</code>	505, 705	<code>ttbnum</code> (option)	521
<code>\thesection</code>	171, 172	<code>ttbtitle</code> (option)	521
<code>\thesubsection</code>	172	<code>\twocolumn</code>	148, 155, 162, 244, 429, 481, 603, 672
<code>\thethisframe</code>	31, 367, 525	<code>\twocolumnbottom</code>	260, 487, 488, 682, 682
<code>\thethumbtabnonum</code>	293	<code>\twocolumnbottominarea</code>	261, 488, 489, 682, 683, 684
<code>thisframe</code> (counter)	31, 525	<code>\twocolumnDbottom</code>	261, 261, 488, 682
<code>\thispagestyle</code>	20, 366, 524	<code>\twocolumnDbottominarea</code>	263, 488, 683
<code>\thr@@</code>	163	<code>\twocolumnDtop</code>	252, 484, 677
<code>\thumbtab</code>	292, 293, 294, 505, 506, 705, 706, 706	<code>\twocolumnDtopinarea</code>	254, 485, 678
<code>\thumbtab@backmatter</code>	294, 295	<code>\twocolumninarea</code>	244, 481, 482, 673, 674
<code>\thumbtabformat</code>	17, 24, 25, 298, 506–509, 707, 707, 709, 710	<code>\twocolumnSbottom</code>	261, 488, 682
<code>\thumbtabhyperlinkformat</code>	16, 17, 24	<code>\twocolumnSbottominarea</code>	263, 487, 488, 682, 683
<code>\thumbtabindex</code>	305, 317, 510, 512, 712, 714	<code>\twocolumnStop</code>	252, 428, 484, 602, 677
<code>\thumbtabindexformat</code>	17, 24, 302, 506, 509, 510, 707, 710, 711	<code>\twocolumnStopinarea</code>	254, 254, 485, 678
<code>\thumbtabnumtitle</code>	298, 303	<code>\twocolumntop</code>	251, 484, 677, 677
<code>\thumbtabregularformat</code>	17, 24, 25, 300, 301	<code>\twocolumntopinarea</code>	252, 484–486, 677, 677, 678, 679
<code>thumbtabs</code> (option)	521	<code>\typeblockheight</code> . . .	30, 45, 115, 146, 147, 162, 165, 205–207, 209, 243, 244, 247, 249, 251, 252, 255, 258, 259, 261, 264, 296
<code>\thumbtabwidth</code> . . .	11, 296, 298, 300, 301, 506–509, 707, 707, 709, 710	<code>\typeblockoffsety</code>	30, 115, 164
<code>\title</code>	358	<code>\typeblockwidth</code>	30, 45, 146, 147, 161, 162, 165, 206, 207, 243, 244, 247, 249, 251, 252, 255, 258, 259, 261, 264
<code>\tl</code>	12–16, 19, 22, 23, 28, 29, 32–34, 36, 38–42, 44, 46–48, 50, 51, 53, 55–60, 62, 67, 69– 71, 73, 77, 78, 80, 82, 85, 87– 95, 97, 98, 101, 102, 105, 107, 108, 112, 138–143, 149–157, 160, 166, 177–180, 183, 190, 193, 201, 211–216, 219–221, 223–227, 229–237, 240, 241, 244, 246, 253, 254, 269, 274, 289–296, 299, 302, 303, 307, 310, 311, 314–318, 320, 321, 329–332, 334–343, 345–358, 362	U	
<code>\tocandthumbtabindex</code>	311, 714		
<code>\tocandthumbtabindex</code>	311, 512, 714	<code>\uncover</code>	330, 331
<code>\tocline</code>	513, 716	<code>\undefined</code>	378, 387, 390, 391, 400, 539, 551, 554, 566
<code>\token</code>	7, 9, 45	<code>\unvbox</code>	431, 433, 434, 605, 608, 609
<code>\toks@</code>	406, 437–462, 572, 613–621, 623– 631, 633, 635–643, 645, 647–649	<code>\use</code>	15, 21, 26, 52, 53, 94, 148, 150, 157, 182, 192, 198–200, 250, 277, 281, 304, 319, 338, 357, 359
<code>\topmargin</code> . . .	30, 408, 432, 576, 607	<code>\usebox</code>	393, 471, 472, 558, 660
		<code>\usedframebreakfalse</code>	159, 430, 604
		<code>\usedframebreaktrue</code>	75, 76, 80, 145, 368, 371, 428, 526, 527, 531, 601
		<code>\UseHook</code>	98, 163, 164, 290

<code>\usepgflibrary</code>	328		
<code>\UseTaggingSocket</code>	160		
V			W
<code>\vadjust</code>	389, 553	<code>\whiledo</code> 368, 371, 372, 376, 377, 382,	
<code>\val</code>	364, 365	385–387, 395, 397–399, 404,	
<code>\value</code>	33, 34, 250	428, 434, 435, 471, 473, 474,	
<code>\vbadness</code>	208, 480, 670	478–480, 506, 508, 510–513,	
<code>\vbox</code>	4, 75, 146,	516, 527, 531, 532, 538, 539,	
161, 163, 164, 184, 205, 207,		544, 547–550, 559, 560, 562–	
209, 321, 326, 368, 428, 432–		565, 570, 571, 601, 609, 610,	
434, 465, 479, 480, 516, 527,		659, 662, 663, 667, 668, 670,	
601, 606–609, 653, 669–671, 720		706, 709, 711–713, 715, 719, 720	
<code>\vcolumnsep</code>	248, 250, 253,	<code>\write</code>	75, 368, 527
257, 260, 262, 265, 483–489,		X	
675, 676, 677, 679, 681, 682, 684		<code>\xdef</code>	284,
verbose (option)	520	371, 379–381, 384, 388–391,	
<code>\vfil</code>	161, 432, 606	394, 396, 397, 401–403, 406,	
<code>\vfill</code>	321, 433, 516, 608, 720	407, 431, 434, 437–461, 503,	
<code>\vNtone</code>	269, 492, 687, 689	506, 508–511, 514, 531, 541–	
<code>\vNtonebottom</code>	271, 492, 689	543, 546, 547, 552, 554, 558,	
<code>\vNtonetop</code>	271, 492, 493, 689	561, 562, 567–569, 573, 574,	
<code>\voffset</code>	31, 408, 576	605, 609, 613–621, 623–631,	
<code>\vrule</code>	4	633, 635–643, 645, 647, 648,	
<code>\vsize</code>	159, 162, 164,	703, 706, 707, 709–712, 716, 717	
165, 430, 431, 481, 604–606, 671		<code>\xdynamiccontents</code>	
<code>\vskip</code>	4, 164, 286, 318, 319, 389,	109, 405, 406, 571, 573	
432, 434, 515, 553, 607–609, 718		Z	
<code>\vspace</code>	286, 287	<code>\z@</code> 77, 163, 164, 171, 184, 286, 287,	
<code>\vsplit</code>	431, 605	368, 369, 431, 432, 434, 465,	
<code>\vss</code>	184, 205, 465, 653	466, 470, 472, 473, 476–478,	
<code>\vtwotone</code>	267, 490, 491, 686, 686	480, 528, 605, 607, 609, 653,	
<code>\vtwotonebottom</code>	268, 268, 491, 686	654, 659, 661, 662, 665–667, 670	
<code>\vtwotonetop</code>	268, 491, 687	<code>\z@skip</code>	164, 432, 607