

THE LTX-TALK-THEME PACKAGE

v0.5.3 2026-10-05

ThemeEngine for LTX-TALK

Matthias WERNER¹

<https://github.com/werner-matthias/osglecture>

The package styles presentations made with LTX-TALK. It provides bundled layout, colour, and font themes, together with commands for customizing and developing themes. It belongs to the [osglecture](#) bundle and works independently of its document class.

Table of Contents

1. Introduction	2	4.3. Positioning titles	9
1.1. Who should read this documentation?	2	4.4. Section titles as frame titles . .	10
I. User Manual	3	4.5. Further customizations	10
2. Quick start	3	4.6. Slot names in bundled themes	11
3. Using themes	4	5. Accessibility	11
3.1. Order and scope	4	5.1. Readable design	11
3.2. Bundled themes	4	5.2. Graphics and logos	12
3.2.1. Comparison of the bundled themes	5	5.3. Tagging and testing	12
4. Customizing themes	8	II. Developing custom themes	12
4.1. Customizing the layout	8	6. The theme model	12
4.2. Customizing colours	8	6.1. Areas	12
		6.2. Slots	13
		6.3. Slot types	13
		6.4. Layout	13

¹. matthias.werner@informatik.tu-chemnitz.de

6.5. Styling	13	8.3. Layout configuration	18
6.6. Area decoration	13	8.4. Metadata	20
7. Developing a theme step by step	14	8.5. Navigation	21
7.1. Registering the theme and choosing its geometry	14	8.6. Content elements	21
7.2. Defining areas and slots	14	8.7. Colour configuration	22
7.3. Positioning slots	15	8.8. Font configuration	22
7.4. Styling slots	15	8.9. Inner theme	23
7.5. Decorating areas	16	8.10. Slot definition	24
7.6. Extending existing themes	16	8.11. Slot positioning	25
7.7. Coordinating inner elements	16	8.12. Title pages	25
7.8. Complete example	17	8.13. Slot styling	25
7.9. Recommendations	17	8.14. Area decoration	26
8. Reference	18	8.15. Title placement	27
8.1. Package options	18	8.16. Accessibility options	28
8.2. Theme management	18	8.17. Templates and sockets	28
		Index	29

1. Introduction

`LTX-TALK-THEME` styles presentations made with `LTX-TALK`: headers and footers, sidebars, titles, colours, fonts, blocks, and list markers. As with Beamer, you can select a bundled theme and add colour or font themes. Custom themes describe areas and the elements placed in them, called *slots* here. Use `LuaLATEX` and a current bundle installation including `osgstyler` for the examples. The package was last tested with `ltx-talk` v0.6.10 dated 13 October 2026 and requires at least v0.5.3 dated 5 August 2026. Older class releases cause an error; releases newer than the tested one produce a compatibility warning. Both the class and the theme system are still under development. Load the class first; theme packages do not replace a class that has already been loaded.

1.1. Who should read this documentation?

This documentation addresses two audiences. Users want to select an existing theme and, where necessary, adapt it to their requirements. Theme authors, on the other hand, want to create a new theme or extend an existing one.

Part I is intended for the former group. On the other hand, anyone who wants to create their own theme should also read Part II. The reference (Section 8) can serve as a handy reference for both groups.

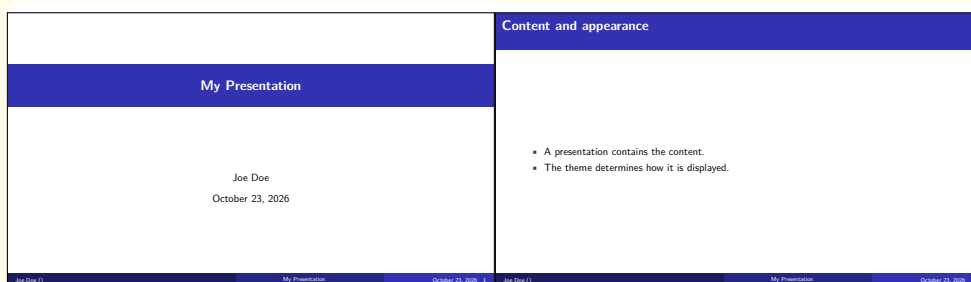
Part I.

User Manual

2. Quick start

The following example demonstrates a small but complete and executable presentation. It loads the provided theme `magpie`, generates a title slide and an additional slide with simple content.

```
1 \DocumentMetadata{}
2 \documentclass{ltx-talk}
3
4 \usepackage{ltxtalk-theme-magpie}
5 \useltxtalktheme{magpie}
6
7 \title{My Presentation}
8 \author{Joe Doe}
9 \date{\today}
10
11 \begin{document}
12
13 \begin{frame}
14   \maketitle
15 \end{frame}
16
17 \begin{frame}
18   \frametitle{Content and appearance}
19   \begin{itemize}
20     \item A presentation contains the content.
21     \item The theme determines how it is displayed.
22   \end{itemize}
23 \end{frame}
24
25 \end{document}
```



3. Using themes

A theme is selected in two steps. First, `\usepackage{ltxtalk-theme-magpie}` loads the theme file. Then, `\useltxtalktheme{magpie}` activates the theme for the presentation.

The commands for title, author, and date and the frame environment belong to the `ltx-talk` document class. The theme configures the public templates of the class and thus determines where and in which form these details appear on the slides. Typesetting and semantically tagging the content remain the responsibility of the class. The actual content therefore remains independent of the selected theme.

In the simplest case, changing the appearance only requires loading another bundled theme package and passing its name to `\useltxtalktheme`. The following overview lists the available themes. Section 7.6 explains how to combine a layout theme with an additive partial theme.

3.1. Order and scope

1. Load packages and activate the full theme.
2. Add colour and font themes.
3. Apply custom changes last.

Calling `\useltxtalktheme` again clears slots, area decorations, grids, and named layouts, among other settings. `\addltxtalktheme` extends the existing configuration. Slot definitions, slot positions, and grid changes are global; grouping a frame does not undo them. Layout geometry keys are local, but layout setup code may make global changes. For alternating layouts, define both states explicitly and apply the appropriate one before the frame.

3.2. Bundled themes

The six layout themes differ mainly in their division of the page. `minimal` uses restrained header and footer areas. `magpie` places the frame title in a coloured header and metadata in a three-part footer. `hawk` adds a navigation area on the left; `sparrow` arranges the section and subsection in the header and places the frame title below them. `bluebird` runs the section list, a row of dots, and a progress readout (see Section 8.5) in the header, and adds an alternative title layout `bluebird/band` plus a chrome-free layout `bluebird/plain`. `goose` raises a live table of contents on the right; its `goose/wide` layout drops that sidebar for wide figures. The corresponding package is named `ltxtalk-theme-⟨name⟩`.

Theme	Characteristic layout	Inspired by
<code>minimal</code>	plain header and footer	–
<code>magpie</code>	coloured header and footer	BEAMER Madrid
<code>hawk</code>	left navigation area	BEAMER Hannover
<code>sparrow</code>	two-level header navigation	BEAMER Szeged
<code>bluebird</code>	header section navigation and progress	BEAMER Berlin
<code>goose</code>	right table of contents	BEAMER Goettingen

3. Using themes

In contrast, anchovy and carp are colour-only themes. Add them after a layout theme with `\addltxtalktheme`; they change only the semantic colours. anchovy uses a blue-and-gold palette, whereas carp uses a red palette. Because neither defines areas or slots, the previously selected layout remains intact.

```
1 \usepackage{ltxtalk-theme-minimal}
2 \usepackage{ltxtalk-theme-anchovy}
3 \useltxtalktheme{minimal}
4 \addltxtalktheme{anchovy}
```

Font themes are likewise layered over a layout theme. In addition to the simple font-sans, font-serif, and font-mixed variants, proper typeface families are provided: font-libertinus, font-newcm, and font-texgyre. Each family assigns coordinated faces to semantic roles for body copy, supporting text, headings, emphasis, code, mathematics, and symbols. All fonts used are distributed with T_EX Live.

```
1 \usepackage{ltxtalk-theme-sparrow}
2 \usepackage{ltxtalk-theme-font-libertinus}
3 \useltxtalktheme{sparrow}
4 \addltxtalktheme{font-libertinus}
```

The roles are named body, support, heading, emphasis, code, math, and symbol. Select them on a slot with `font-slot=heading`, or directly with `\UseLTXTalkFont{heading}`. The existing portable names main, companion, display, accent, and mono remain available as compatible synonyms. The mathematics slot is registered only as a resource for a later [unicode-math](#) adapter and is not activated as a text font.

3.2.1. Comparison of the bundled themes

The following example slides deliberately use identical content. This makes differences in text area, navigation, title placement, and semantic colours directly comparable. Each colour-only theme is shown over a layout that surfaces its palette: carp repaints primary and reads most clearly under magpie with its large header and footer fills; anchovy places its gold in the accent slots and needs a layout such as sparrow that colours secondary text from secondary.

```
1 \DocumentMetadata{}
2 \documentclass{ltx-talk}
```

3. Using themes

```
3
4 \usepackage{ltxtalk-theme-minimal}
5 \usepackage{ltxtalk-theme-magpie}
6 \usepackage{ltxtalk-theme-hawk}
7 \usepackage{ltxtalk-theme-sparrow}
8 \usepackage{ltxtalk-theme-bluebird}
9 \usepackage{ltxtalk-theme-goose}
10 \usepackage{ltxtalk-theme-anchovy}
11 \usepackage{ltxtalk-theme-carp}
12
13 \title[Theme comparison]{Theme comparison}
14 \author[A. Example]{Ada Example}
15 \institute{Example Institute}
16 \date{2 September 2026}
17
18 \newcommand*\ThemeExampleFrame[1]{
19   \begin{frame}
20     \frametitle{#1}
21     \framesubtitle{The same content in every theme}
22     A short paragraph demonstrates type, text colour, and the text area.
23     \begin{itemize}
24       \item first list item
25       \item second item with an \alert{alert}
26     \end{itemize}
27     \begin{block}{Example block}
28       Block content with the formula  $a^2+b^2=c^2$ .
29     \end{block}
30   \end{frame}
31 }
32
33 \begin{document}
34 \section{First topic}
35 \subsection{An aspect}
36 \useltxtalktheme{minimal}
37 \ThemeExampleFrame{minimal}
38 \useltxtalktheme{magpie}
39 \ThemeExampleFrame{magpie}
40 \section{Second topic}
41 \useltxtalktheme{hawk}
42 \ThemeExampleFrame{hawk}
43 \useltxtalktheme{sparrow}
44 \ThemeExampleFrame{sparrow}
45 \useltxtalktheme{bluebird}
46 \ThemeExampleFrame{bluebird}
47 \section{Third topic}
48 \subsection{Its detail}
49 \useltxtalktheme{goose}
50 \ThemeExampleFrame{goose}
```

3. Using themes

```

51 \useltxtalktheme{sparrow}
52 \addltxalktheme{anchovy}
53 \ThemeExampleFrame{sparrow + anchovy}
54 \useltxtalktheme{magpie}
55 \addltxalktheme{carp}
56 \ThemeExampleFrame{magpie + carp}
57 \end{document}

```

Theme comparison First topic minimal <i>The same content in every theme</i> A short paragraph demonstrates type, text colour, and the text area. - first list item - second item with an alert Example block Block content with the formula $a^2 + b^2 = c^2$.	magpie The same content in every theme A short paragraph demonstrates type, text colour, and the text area. - first list item - second item with an alert Example block Block content with the formula $a^2 + b^2 = c^2$.
A. Example Theme comparison A. Example Second topic hawk The same content in every theme A short paragraph demonstrates type, text colour, and the text area. - first list item - second item with an alert Example block Block content with the formula $a^2 + b^2 = c^2$.	A. Example (Example Institute) Theme comparison 2 September 2024 2 Second topic sparrow The same content in every theme A short paragraph demonstrates type, text colour, and the text area. - first list item - second item with an alert Example block Block content with the formula $a^2 + b^2 = c^2$. Example Institute Theme comparison 4
First topic • Second topic • Third topic 2 / 3 bluebird The same content in every theme A short paragraph demonstrates type, text colour, and the text area. - first list item - second item with an alert Example block Block content with the formula $a^2 + b^2 = c^2$. A. Example Theme comparison Example Institute 5	Its detail goose The same content in every theme A short paragraph demonstrates type, text colour, and the text area. - first list item - second item with an alert Example block Block content with the formula $a^2 + b^2 = c^2$. A. Example Theme comparison First topic Second topic Third topic Its detail 3 / 3
Third topic Its detail sparrow + anchovy The same content in every theme A short paragraph demonstrates type, text colour, and the text area. - first list item - second item with an alert Example block Block content with the formula $a^2 + b^2 = c^2$. Example Institute Theme comparison 7	magpie + carp The same content in every theme A short paragraph demonstrates type, text colour, and the text area. - first list item - second item with an alert Example block Block content with the formula $a^2 + b^2 = c^2$. A. Example (Example Institute) Theme comparison 3 September 2024 8

4. Customizing themes

4.1. Customizing the layout

After activation, the dimensions selected by the theme can be overridden with `\ltxtalksetup`. Only the specified properties are changed. The following example enlarges the header and footer and disables the right margin area:

```

1 \useltxtalktheme{hawk}
2 \ltxtalksetup{
3   header-height = 2.8cm,
4   footer-height = 0.8cm,
5   right-margin = false
6 }
```

Section 8.3 lists all keys of `\ltxtalksetup`. For simple adjustments, the switches `header`, `left-margin`, `right-margin`, and `footer` and the dimensions `header-height`, `footer-height`, and `margin-width` are usually sufficient. Changing an area's size does not reposition existing slots automatically. Larger changes should therefore be implemented as a theme or partial theme; Section 7 describes the procedure.

A named layout can temporarily hide a sidebar. With `goose`, explicitly return to `goose/standard` afterwards:

```

1 % Preamble:
2 \usepackage{ltxtalk-theme-goose}
3 \useltxtalktheme{goose}
4 \title{A presentation}
5 \author{A. Author}
6 % In the document:
7 \UseLTXTalkLayout{goose/wide}
8 \begin{frame}
9   \frametitle{A wide illustration}
10  Content using the extra width.
11 \end{frame}
12 \UseLTXTalkLayout{goose/standard}
```

4.2. Customizing colours

The eight semantic theme colours can be assigned with `\setltxtalkcolors`. As with `\definecolor`, every value specifies a colour model and value, here as a `model=...`, `value=...` pair:

4. Customizing themes

```
1 \setltxtalkcolors{
2   primary    = {model=HTML, value=005B50},
3   secondary  = {model=HTML, value=3C7A73},
4   accent     = {model=HTML, value=D6A800},
5   text       = {model=HTML, value=202020},
6   background = {model=HTML, value=FFFFFF},
7   structure  = {model=HTML, value=005B50},
8   alert      = {model=HTML, value=A00020},
9   example    = {model=HTML, value=166534}
10 }
```

`primary` defines the theme's main colour; `secondary` supports additional information and accent highlights details. `text` and `background` form the foreground/background pair. `structure`, `alert`, and `example` style lists and block variants. Colour names carry the `ltxtalk-` prefix, with `ltxtalk-bg` for the background. The final three are also available as `structure`, `alert`, and `example`. Changes take effect where the theme uses these names.

Unspecified colours are derived along a fixed chain: `secondary`, `accent`, and `structure` inherit from `primary`; `text` and `background` use their neutral defaults, while `alert` and `example` use their semantic defaults. A derived colour is fixed when the command is executed; changing its source later does not recompute it automatically.

Attention!

`\setltxtalkcolors` is a *complete* redeclaration, not an incremental update. Omitted entries do not retain their previous value but follow the inheritance rules described above. Only `primary` is mandatory. If it is missing, or if any supplied colour specification is incomplete, the entire declaration fails and the previously valid palette remains unchanged. To change a single colour while preserving all others, repeat the complete most recent declaration and alter the desired entry.

4.3. Positioning titles

`\setltxtalktitleplacement` declares in which area a theme provides frame and section titles. A frame title can occur in the content area, the header, or both; a section title can be shown or hidden in the header:

```
1 \setltxtalktitleplacement{
2   frametitle = both,
3   section    = header
4 }
```

4. Customizing themes

The theme must provide suitable slots for the selected placement. The option alone does not create an additional visible element when the corresponding slot is absent.

4.4. Section titles as frame titles

With `\setltxtalkheadingbehavior{auto-section-title = true}`, the first `\frametitle` frame after `\section` or `\subsection` automatically takes over its numbered heading as the title – independently of whether the same theme also shows sections in the header (`\setltxtalktitleplacement`'s section key). If that same frame also gives its own text, it moves to the subtitle slot; a separately given `\framesubtitle` is then dropped, because the supplied title already occupies that slot:

```
1 \setltxtalkheadingbehavior{auto-section-title = true}
2 ...
3 \section{Introduction}
4 \begin{frame}
5   \frametitle{} % title becomes "1 Introduction"
6   ...
7 \end{frame}
8 \section{Related Work}
9 \begin{frame}
10  \frametitle{A Short History} % title becomes "2 Related Work",
11                               % subtitle becomes "A Short History"
12  ...
13 \end{frame}
```

The flag is consumed by exactly one `\frametitle` call: later frames in the same section are unaffected, and if `\section` and `\subsection` follow each other with no frame in between, the last (more specific) one wins. A frame with no `\frametitle` call at all stays titleless as before; an empty `\frametitle{}` is enough to pick up just the section title.

4.5. Further customizations

Individual slots can be restyled after activating the theme with `\styleltxtalkslot`. This requires knowing the slot name used by that theme; slot names are therefore part of a particular theme's interface, not of the general user interface. Likewise, `\decorateltxtalkarea` can, for example, change a rule above the footer.

```
1 \usepackage{ltxtalk-theme-minimal}
2 \useltxtalktheme{minimal}
3 \styleltxtalkslot{min-ftitle}{
```

```

4   color = ltxtalk-primary,
5   font = \Large\bfseries,
6   align = left
7 }
8 \decorationltxtalkarea{footer}{
9   frame = top,
10  rule-color = ltxtalk-accent,
11  rule-thickness = 0.6pt
12 }

```

Place such customizations immediately after `\useltxtalktheme`, because activating another theme later resets all slots and decorations. For reusable or structural additions, a partial theme activated with `\addltxtalktheme` is more appropriate.

4.6. Slot names in bundled themes

Style the following names with `\styleltxtalkslot` after activating the corresponding theme. Combine the prefix with a listed suffix, for example `min-ftitle`. Colour and font themes do not add slots.

minimal Prefix `min-`: title, section, page, ftitle, fsubtitle, author, footer-page.

magpie Prefix `magpie-`: frametitle, framesubtitle, author, title, page.

hawk Prefix `hawk-`: title, author, section, subsection, frametitle, framesubtitle.

sparrow Prefix `sparrow-`: section, subsection, frametitle, framesubtitle, institute, title, page.

bluebird Prefix `bluebird-`: navband, titleband, seclist, dots, progress, ftitle, fsub, footline1, footline2, fauthor, first, ftitle-foot, fpage.

goose Prefix `goose-`: sidemeta, toc, progress, subs, ftitle, fsub.

5. Accessibility

A theme changes visual presentation but must not replace the document structure created by `ltx-talk`. Mark frame titles with `\frametitle`, subtitles with `\framesubtitle`, and structural levels with `\section` and `\subsection`. Merely formatting text visually cannot replace these semantic commands.

5.1. Readable design

Custom foreground and background colours must provide sufficient contrast. Colour should never be the only means of conveying meaning. Small type, very thin rules, and densely filled

margin areas are particularly problematic on projectors and at high magnification. Test not only the title slide but also a content-heavy standard slide.

Text and essential graphics should remain within the available areas. Changes to margin-width, header-height, or footer-height must therefore be tested with long titles, multi-line metadata, and different aspect ratios.

5.2. Graphics and logos

Content graphics use the facilities of `ltx-talk` and the loaded graphics packages; provide meaningful alternative text there. Repeated logos, colour fields, and rules are usually decorative and must not burden the reading order with duplicate information. The slot role `artifact` marks decorative elements. For meaningful graphics in a slot, provide alternative text with `alt-text`; see Section 8.16.

5.3. Tagging and testing

For tagged PDF output, begin the presentation with `\DocumentMetadata`. At present, `ltx-talk` performs most tagging of frame content and headings. `LTXTALK-THEME` preserves that connection by typesetting frame titles in the content area through the class's public commands. The slot interface additionally controls the role and alternative text of theme elements in tagged PDF (Section 8.16).

Testing must therefore cover both visual appearance and structure: use long and empty metadata, multiple section levels, continuation frames, and a PDF checking tool. The bundle provides `l3build` tests that, among other things, check the heading structure generated by `ltx-talk`.

Part II.

Developing custom themes

6. The theme model

A theme consists of areas, slots, and styling. The slot type determines content; area and position determine location; slot styling determines font, colour, and frame. Use existing types for titles and metadata, and custom for custom content. The class remains responsible for frames, overlays, and the semantic structure of frame content. Specify the semantic role of theme elements through the accessibility options (Section 8.16).

6.1. Areas

A theme assigns page elements to the five areas `header`, `left`, `content`, `right`, and `footer`. The `ltxtheme` page style outputs the header, margins, and footer. The content area remains part of

the normal frame content, preserving line breaking, overlays, and semantic classification by [ltx-talk](#).

header: titles and navigation		
left	content: frame content	right
footer: metadata and frame number		

The diagram shows only the spatial nesting. A theme may leave any area empty, and several slots may occupy the same area. Overlaps are not prevented automatically.

6.2. Slots

A slot represents an element with a purpose, not a hard-wired location. Its assignment to an area, position, and rendering template form the concrete layout. Several slots of the same type are allowed; a frame title may occur in the header, the content area, or both.

6.3. Slot types

Predefined types provide the document title, author, institute, section, subsection, frame title, frame subtitle, and frame number, among other data. The `custom` type obtains its content from `\setltxtalkslotcontent`; it can therefore implement breadcrumb or tree navigation and freely designed logos.

6.4. Layout

Every slot has horizontal and vertical anchor positions, a width, and a height. Symbolic positions such as `left`, `center`, `right`, `top`, and `bottom` can be combined with absolute dimensions.

6.5. Styling

Every slot has an instance of the `ltxtheme-slot` template. Themes provide suitable defaults, and users can edit the instances through the public styling command. Font, colour, background, alignment, padding, and rules thus belong to the theme rather than the page style.

6.6. Area decoration

Area decorations are independent of slots. In particular, the `ltxtheme-area` template draws backgrounds and separator rules. A rule above the footer is therefore a property of the footer area, not an artificial footer slot.

The page style forms the stable connection to [LTX-TALK](#). Sockets additionally split its output into background, the four peripheral areas, and foreground. A complete theme need not touch these sockets; they are extension points for replaceable layout phases, not a replacement for the additive slot interface.

7. Developing a theme step by step

The most reliable development order is: first size the required areas with `\ltxtalksetup`; then define and position slots using test content; next style slots and areas; finally coordinate inner elements, title page, and colours. This keeps geometric and purely visual errors distinguishable.

7.1. Registering the theme and choosing its geometry

A theme package first loads `ltxtalk-theme`, defines a setup command, and registers it under a unique name. The setup command is not executed when the package is loaded but only by `\useltxtalktheme`. Several theme packages can therefore be loaded and one of them activated later.

```

1 \RequirePackage{ltxtalk-theme}
2 \ExplSyntaxOn
3 \cs_new_protected:Npn \__my_theme_setup:
4 {
5   \ltxtalksetup{header=true,header-height=1.5cm}
6   % Slots, Positionen und Gestaltung
7 }
8 \RegisterLTXTalkTheme{my-theme}{\__my_theme_setup:}
9 \ExplSyntaxOff

```

The package logic does not know the name of the theme. On activation it clears the previous theme state, executes the registered setup command, adjusts the text area, and selects the `lththeme` page style. The theme therefore describes only areas, slots, and their appearance.

7.2. Defining areas and slots

A slot is defined by name, area, type, and accessibility data. Its name must be unique within the active theme. The standard areas are header, left, content, right, and footer.

```

1 \defineltxtalkslot{my-section}{header}{section}{
2   role = navigation
3 }
4 \defineltxtalkslot{my-path}{right}{custom}{
5   role = navigation
6 }
7 \setltxtalkslotcontent{my-path}{
8   \thesection\quad\thesubsection
9 }

```

The predefined types `title`, `author`, `date`, `institute`, `section`, `subsection`, `frametitle`, `framesubtitle`, and `pagenumber` obtain their content from the class. The custom type and unknown custom types use content stored with `\setltxtalkslotcontent`. Logos are conveniently supplied in the same way using explicit graphics code.

7.3. Positioning slots

After definition, every visible slot receives a position and a size:

```
1 \positionltxtalkslot
2   {my-section}{left}{center}{0.55\paperwidth}{0.8cm}
3 \positionltxtalkslot
4   {my-path}{center}{1cm}{2cm}{4cm}
```

The second and third arguments specify horizontal and vertical positions within the area. Horizontally, `left`, `center`, and `right` are accepted; vertically, `top`, `center`, and `bottom`. Alternatively, a dimension is measured from the left or top edge. Width and height are dimensions; slots are not automatically protected from overlap.

7.4. Styling slots

`\styleltxtalkslot` edits the `ltheme-slot` template instance created for a slot. Content, position, and styling therefore remain separate.

```
1 \styleltxtalkslot{my-section}{
2   font = \large\bfseries,
3   color = white,
4   align = left,
5   padding = 2mm,
6   frame = box,
7   rule-color = ltxtalk-accent,
8   rule-thick = 0.8pt
9 }
```

Besides `font`, `color`, `align`, and `padding`, the keys `background`, `frame`, `rule-color`, `rule-thick`, and `space-after` are available. `frame` accepts `none`, `box`, `top`, `bottom`, `left`, and `right`. The interface also contains `rotate`, `opacity`, `shadow`, `uppercase`, and `lowercase`; these currently have no visible effect. Individual slot frame edges are not drawn either; use `box` or decorate the entire area instead.

7.5. Decorating areas

Backgrounds and separator rules affecting an entire area do not belong in a slot. Configure them on the `lttheme-area` template instance with `\decorationltxtalkarea`:

```

1 \decorationltxtalkarea{footer}{
2   background = white,
3   frame = top,
4   rule-color = ltxtalk-primary,
5   rule-thickness = 0.5pt,
6   artifact-type = decorative
7 }
```

For `frame`, `topbottom` and the usual individual sides are also accepted. The decoration is drawn before the area's slots. The keys `background-opacity`, `shadow`, `shadow-color`, `shadow-offset`, `margin-top`, and `margin-bottom` are already part of the interface but are not yet rendered completely.

7.6. Extending existing themes

A registered theme may contain only additional slots or a single decoration. After activating the base theme, `\addltxtalktheme{<name>}` adds such a partial theme without deleting existing slots. The `example-switching.tex` example adds breadcrumb navigation in this way. Activating a full theme again deliberately starts from an empty theme state.

```

1 \useltxtalktheme{minimal}
2 \addltxtalktheme{course-navigation}
```

A partial theme should define only the properties it actually adds. In particular, it must neither clear the theme state nor select the page style itself. Prefix slot names with a package-specific identifier to avoid collisions with the base theme.

7.7. Coordinating inner elements

A complete theme should not stop at the header, margins, and footer. `\styleltxtalkblock` coordinates normal, alert, and example blocks with the semantic colours, while `\setltxtalkitemize` selects matching list markers. Both commands change only the appearance of structures provided by `ltx-talk`.

7.8. Complete example

This document combines the steps into a small theme. It places the frame title in the content area and the slide number in the footer. The setup registration can later move unchanged into a custom package, which loads the core package with `\RequirePackage`.

```

1 \DocumentMetadata{}
2 \documentclass{ltx-talk}
3 \usepackage{ltxtalk-theme}
4 \RegisterLTXTalkTheme{my-theme}{
5   \ltxtalksetup{footer=true, footer-height=0.7cm}
6   \setltxtalkcolors{primary={model=HTML,value=005B50}}
7   \defineltxtalkslot{my-title}{content}{frametitle}{}
8   \positionltxtalkslot{my-title}{0pt}{0pt}{\textwidth}{1cm}
9   \styleltxtalkslot{my-title}{
10     font-slot=heading, font=\Large\bfseries,
11     color=ltxtalk-primary, align=left
12   }
13   \defineltxtalkslot{my-page}{footer}{pagenumber}{}
14   \DeclareLTXTalkAreaGrid{footer}{*,2cm}{1}
15   \PositionLTXTalkSlotInGrid{my-page}{2}{1}{1}{1}
16   \decorateltxtalkarea{footer}{
17     frame=top, rule-color=ltxtalk-primary
18   }
19 }
20 \uselxtalktheme{my-theme}
21 \begin{document}
22 \begin{frame}
23   \frametitle{A custom theme}
24   Content belongs to the presentation.
25 \end{frame}
26 \end{document}

```

7.9. Recommendations

Prefer semantic slot types to custom types. Mark decorative elements as artifacts, navigation as navigation, and genuine headings with the appropriate level. Repeated page elements must not duplicate the actual frame content or its reading order.

Themes should cover short and long content, missing metadata, multi-line headers, both margin areas, and frames with and without subtitles. An `l3build` visual test is recommended for every theme; structural tests should also check the tagged heading hierarchy. Theme-specific logic remains in the theme package, while `LTXALK-THEME` provides only the general areas, slots, templates, and page style.

8. Reference

8.1. Package options

Specify core package options before loading the first theme package:

```

1 \usepackage[block-hairlines=true]{ltxtalk-theme}
2 \usepackage[shadow=true]{ltxtalk-theme-magpie}
3 \useltxtalktheme{magpie}

```

outer-only = `true|false` Default: `false`
 Disables theme font-palette management. Text roles use existing LaTeX font families. Activating a font palette then only issues a warning. Available at load time only; document-class requirements still apply.

block-hairlines = `true|false` Default: `false`
 Adds thin block rules on the next theme activation. Also settable through `\ltxtalksetup` before `\useltxtalktheme`.

shadow = `true|false` Default: `false`
 An option of `ltxtalk-theme-magpie` only: enables shadows beneath blocks.

8.2. Theme management

`\useltxtalktheme{⟨name⟩}`
 Activates the registered theme `⟨name⟩` and completely clears the previous theme state. default restores the class page style without lththeme slots.

`\addltxtalktheme{⟨name⟩}`
 Executes the setup command of a registered theme in addition to the active theme without deleting existing slots. This command primarily adds partial themes.

`\RegisterLTXTalkTheme{⟨name⟩}{⟨setup code⟩}`
 Registers `⟨Setup code⟩` under `⟨name⟩`. This is normally a protected internal command of the theme package. An existing registration with the same name is replaced.

8.3. Layout configuration

`\ltxtalksetup{⟨options⟩}`
 Changes only the specified properties and adjusts the text area. The defaults below apply before theme activation; themes set their own values.

`\DeclareLTXTalkLayout{⟨name⟩}{⟨options⟩}[⟨setup code⟩]`

8. Reference

`\UseLTXTalkLayout{⟨name⟩}[⟨Overrides⟩]`

`\DeclareLTXTalkLayout` registers the layout configuration.

`\UseLTXTalkLayout` activates it. *⟨options⟩* are keys from `ltxtalk / layout`; the optional *⟨setup code⟩* may define slots, decorate areas, or bind a navigation. `\UseLTXTalkLayout` applies the stored options first, then the *⟨Overrides⟩* in the same key space, then the setup code – so the code sees the final values (such as `\LTXTalkHeaderRows`). Keys not named keep their value, so layouts are additive. A theme may declare several layouts and switch between them per frame; `\UseLTXTalkLayout` may be used immediately before a frame or as its first command. If the setup code contains unbalanced square brackets, enclose it additionally in braces.

`\LTXTalkBaseLeftMargin`

`\LTXTalkBaseRightMargin`

`\LTXTalkBaseTopMargin`

`\LTXTalkBaseBottomMargin`

`\LTXTalkCurrentLeftMargin`

`\LTXTalkCurrentRightMargin`

`\LTXTalkCurrentTopMargin`

`\LTXTalkCurrentBottomMargin`

Expandable lengths for the original and currently effective page margins. Theme code can use them to align content without accessing internals.

`header = true | false`

Enables or disables the header.

Default: false

`left-margin = true | false`

Enables or disables the left sidebar.

Default: false

`right-margin = true | false`

Enables or disables the right sidebar.

Default: false

`footer = true | false`

Enables or disables the footer.

Default: false

8. Reference

<code>header-height = {⟨length⟩}</code>	Default: 1.5cm
Height of the header.	
<code>footer-height = {⟨length⟩}</code>	Default: 1cm
Height of the footer.	
<code>margin-width = {⟨length⟩}</code>	Default: 2.5cm
Sets both sidebar widths and both content insets together. Subsequent individual keys override these values.	
<code>left-area-width = {⟨length⟩}</code>	Default: 2.5cm
Width of the left sidebar.	
<code>right-area-width = {⟨length⟩}</code>	Default: 2.5cm
Width of the right sidebar.	
<code>content-inset-left = {⟨length⟩}</code>	Default: 2.5cm
Left content inset when the sidebar is enabled.	
<code>content-inset-right = {⟨length⟩}</code>	Default: 2.5cm
Right content inset when the sidebar is enabled.	
<code>header-rows = {⟨integer⟩}</code>	Default: 1
Row count accessible through <code>\LTXTalkHeaderRows</code> for layout setup code. Does not change an area grid by itself.	
<code>header-columns = {⟨list⟩}</code>	Default: {0.2,0.6,0.2}
Legacy interface; no effect on freely positioned slots.	
<code>footer-columns = {⟨list⟩}</code>	Default: {0.33,0.34,0.33}
Legacy interface; use area grids for new layouts.	
<code>block-hairlines = true false</code>	Default: false
Global preference for the next theme activation, not a geometry setting.	

8.4. Metadata

`\LTXTalkMetadata{⟨field⟩}`

`\UseLTXTalkMetadata[⟨options⟩]{⟨field⟩}`

Provides presentation metadata through a public interface. Available fields are title, short-title, subtitle, short-subtitle, author, short-author, institute, short-institute, date, short-date, section, subsection, frametitle, framesubtitle, framenumbers, and totalframes. The second form accepts `lines=1` to replace forced line breaks with `linebreak`, which defaults to a space.

`\LTXTalkMetadata` is expandable. Use `\UseLTXTalkMetadata` for output with the following options:

<code>lines = all 1</code>	Default: all
Retains line breaks or replaces them for single-line output.	
<code>linebreak = {⟨replacement⟩}</code>	Default: space
Replacement for forced breaks with <code>lines=1</code> .	

```
1 \UseLTXTalkMetadata[lines=1,linebreak={ -- }]{short-title}
```

8.5. Navigation

`\LTXTalkNavigation{⟨name⟩}`

`\DeclareLTXTalkNavigation{⟨name⟩}{⟨Code⟩}`

`\LTXTalkNavigation` typesets a named navigation renderer – intended as slot content in the setup code of a layout or theme. `\DeclareLTXTalkNavigation` registers a custom renderer; `⟨Code⟩` runs on every call and can use `\LTXTalkMetadata`, for example.

Bundled:

- `current-section`, `current-subsection` – title of the current section or subsection; one pass suffices.
- `section-list` – every section in document order, side by side.
- `subsection-list` – the current section’s subsections, side by side.
- `toc` – sections one per line, the current section’s subsections indented below it.
- `section-dots` – one dot per section (current filled, past dimmed, upcoming open).
- `section-progress` – “`⟨position⟩ / ⟨count⟩`”.

The current entry is set in `ltxtalk-primary`, the rest dimmed.

Navigation uses the section structure from the preceding LaTeX run. Every renderer except `current-section` and `current-subsection` therefore needs two passes; if the structure changes, `lththeme` asks for another run. Section titles with markup appear as plain text, being a moving argument.

8.6. Content elements

`\InsertLTXTalkElement{⟨name⟩}`

Typesets a named instance of the `ltxtalk-element` template type – intended for single, formattable content values such as page numbers or section headings, as opposed to `\LTXTalkNavigation`’s renderers for the *whole* section sequence. The tagged role comes, like for any other slot content, from the surrounding `\defineltxtalkslot`’s fourth argument (Section 8.16 covers the slots’ tagging options accordingly).

Bundled:

- `pagenumber`, `totalframes` – current frame number and total count.
- `pagenumber-of-total` – “ $\langle frame\ number \rangle / \langle total\ count \rangle$ ”.
- `section-pagenumber` – “ $\langle section\ title \rangle - \langle frame\ number \rangle$ ”.
- `sectionheading`, `subsectionheading` – numbered section or subsection title.

Every built-in instance uses one of three templates and can be adjusted with `\EditInstance{ltxtalk-element}{name}{keys}`:

- `counter`: key source (the value to output).
- `fraction`: keys left, right, separator.
- `heading`: keys numbered (Boolean), level, number, text, separator. With level (section, subsection, or subsubsection) the number is omitted while the current heading of that level is unnumbered (`\section*`).

New instances – even reusing one of these three templates with just different values – are created like for any other template type, via `\DeclareInstance{ltxtalk-element}{name}{template}{keys}`.

```

1 \defineltxttalkslot{my-page}{footer}{custom}{role=navigation}
2 \setltxtalkslotcontent{my-page}{\InsertLXTalkElement{pagenumber-of-total}}
3 \EditInstance{ltxtalk-element}{pagenumber-of-total}{separator = { of }}
```

8.7. Colour configuration

`\setltxtalkcolors{<colours>}`

Defines the semantic colours primary, secondary, accent, text, background, structure, alert, and example. Each value has the form `model=...`, `value=...`. Their colour names use the `ltxtalk-` prefix, with `ltxtalk-bg` for the background. The final three are also mapped to the class colour names `structure`, `alert`, and `example`.

8.8. Font configuration

`\DeclareLXTalkFontPalette{<name>}{<fonts>}`

`\UseLXTalkFontPalette{<name>}`

Declares or activates a font palette. Its portable slots are `main`, `companion`, `display`, `accent`, `mono`, `math`, and `symbol`. A resource has the form `font={...}`, optionally with features; alternatively, `alias=...` refers to another slot. `main` is required. The bundled `ltxtalk-default` palette uses New Computer Modern Sans with Latin Modern Mono.

`\UseLTXTalkFont{<role>}`

Selects a text face by semantic role. The roles are body, support, heading, emphasis, code, and symbol. The technical palette names main, companion, display, accent, and mono are accepted as well. math names a resource only and cannot be selected as a text face.

Roles are fixed names for palette entries:

Role	Palette entry
body	main
support	companion
heading	display
emphasis	accent
code	mono
symbol	symbol

```

1 \DeclareLTXTalkFontPalette{my-fonts}{
2   main = {font={Latin Modern Sans}},
3   companion = {alias=main},
4   display = {alias=companion},
5   accent = {alias=display},
6   mono = {font={Latin Modern Mono}},
7   symbol = {alias=main}
8 }
9 \UseLTXTalkFontPalette{my-fonts}

```

Font names must be available in the TeX installation. features contains [fontspec](#) font options. See the [osgstyler](#) font-palette reference for omitted entries, alias resolution, and mathematical fonts; the two palette commands use that interface. Palette selection is disabled with the load option `outer-only=true`.

8.9. Inner theme

`\styleltxtalkblock{<type>}{<options>}`

Edits the block, alertblock, or exampleblock instance of the `osgstyler-block` template type provided by [osgstyler](#). The standard interface includes background, rule-color, rule-thickness, padding, corner-radius, shadow, shadow-offset, before, and after. Ready-made looks are available as [osgstyler](#) traits: `card` (a lightly framed, faintly tinted box with a bold title), `titlebar` (beamer's Madrid: a full colour bar over a tinted body area with a bottom edge), `heading` (a coloured title only, flowing), `soft` (a title with thin rules, flowing), `plain` (frameless), and `framed/filled` (a frame without / with a tint). `heading`, `soft` and `plain` leave the class's tagging untouched; the rest are real boxes and add exactly one Div per block in tagged PDF. Without a call `lththeme` uses the `card` trait, so a block is always recognisable; `magpie`

and bluebird use titlebar, hawk, sparrow and goose use heading – each in the theme colour. The load option `block-hairlines` (or `\ltxtalksetup`) adds thin dividing rules for every theme (a border on the box, head/foot rules on the plain title). A block style is reset to the card default on a theme switch.

`\setltxtalkitemize{⟨options⟩}`

Sets the item marker for each itemize level with level-1 through level-4. Only the visual marker changes; the class’s list and list-item tags remain unchanged.

8.10. Slot definition

`\defineltxtalkslot{⟨name⟩}{⟨area⟩}{⟨type⟩}{⟨accessibility fields⟩}`

Defines a slot and appends it to the specified area. Redefinition replaces its assignment and moves it to the end of that area; it does not create a second slot with the same name. Standard areas are header, left, content, right, and footer. Predefined types are title, subtitle, author, date, institute, section, subsection, frametitle, framesubtitle, pagenumber, logo, and custom.

`\setltxtalkslotcontent{⟨name⟩}{⟨content⟩}`

Stores the executable content of a free slot. It is used for custom, logo, and non-predefined custom types.

Type	Description	Possible role
frametitle	Frame title	heading (H2)
framesubtitle	Frame subtitle	heading (H3)
title	Presentation title	heading (H1)
section	Current section	heading (H2)
subsection	Current subsection	heading (H3)
author	Author name	note
date	Date	note
institute	Institute name	note
pagenumber	Page number	navigation
logo	Logo	artifact
custom	User-defined	artifact

TABLE 1: Predefined slot types

The roles in the table are examples, not defaults imposed by the slot type. Without an explicit role, artifact applies; frametitle is an exception, with tagging provided by the class.

8.11. Slot positioning

`\positionltxtalkslot{⟨name⟩}{⟨x⟩}{⟨y⟩}{⟨width⟩}{⟨height⟩}`

Positions a slot relative to its area. For $\langle x \rangle$, left, center, right, or a dimension is accepted; for $\langle y \rangle$, use top, center, bottom, or a dimension. Positive dimensions are measured from the left or top edge.

`\DeclareLTXTalkAreaGrid{⟨area⟩}{⟨columns⟩}{⟨rows⟩}`

`\PositionLTXTalkSlotInGrid{⟨name⟩}{⟨column⟩}{⟨row⟩}{⟨column span⟩}{⟨row span⟩}`

`\SetLTXTalkAreaGridRows{⟨area⟩}{⟨rows⟩}`

Defines a grid for an area and positions slots in it. Columns are a comma-separated list of lengths; * distributes the remaining width equally among all star columns. Columns and rows are one-based. The final two positioning arguments are the numbers of columns and rows spanned, not dimensions. Grid declarations and row-count changes are global; a TeX group or the end of a frame does not undo them. Set the desired row count explicitly for the next slide.

8.12. Title pages

`\DeclareLTXTalkTitleLayout{⟨name⟩}{⟨layout options⟩}{⟨content⟩}`

`\UseLTXTalkTitleLayout{⟨name⟩}`

`\MakeLTXTalkTitle[⟨name⟩]`

`\DeclareLTXTalkTitleLayout` registers a title-page variant consisting of temporary layout options and its contents. `\UseLTXTalkTitleLayout` selects the default; `\MakeLTXTalkTitle` creates the frame and can optionally select a variant for that invocation. The previous geometry is restored after the title page. The bluebird theme ships one such variant as bluebird/band; ltx-talk's own `\maketitle` is unaffected.

`\LTXTalkHeaderRows`

`\IfLTXTalkTitlePageTF{⟨title-page code⟩}{⟨frame code⟩}`

Gives theme authors the effective number of header rows and distinguishes a title page created by `\MakeLTXTalkTitle` from an ordinary frame. Both interfaces remain valid when the base class defers construction of the running head until shipout.

8.13. Slot styling

`\styleltxtalkslot{⟨name⟩}{⟨options⟩}`

Changes the specified properties of the slot instance. Omitted properties keep their previous values. Defaults below apply to new instances; themes set their own values.

8. Reference

font-slot = {*<role>*} Default: companion
Font role: body, support, heading, emphasis, code, symbol, or the corresponding palette name. math is not a text font.

font = {*<commands>*} Default: empty
Additional font commands, such as `\Large\bfseries`, applied after the font role.

color = {*<colour>*} Default: ltxtalk-text
Text colour.

background = {*<colour>*} Default: none
Background colour or none. With `frame=box`, none gives a white background.

background-fill = `true`|`false` Default: `false`
With a background colour, fills the declared slot area instead of only the text area. Box frames use their own rendering.

align = `left`|`center`|`right` Default: `center`
Horizontal text alignment.

frame = `none`|`box`|`top`|`bottom`|`left`|`right` Default: `none`
box draws a frame. Individual edge values are accepted but do not draw a rule in the current slot renderer.

padding = {*<length>*} Default: 0pt
Inset for colour backgrounds and box frames.

rule-color = {*<colour>*} Default: ltxtalk-primary
Frame colour.

rule-thick = {*<length>*} Default: 0.4pt
Box-frame rule width. Unlike area decoration, this key is named `rule-thick`, not `rule-thickness`.

space-after = {*<length>*} Default: 0pt
Gap below a title slot in the content area. Positive values set the gap; otherwise it is `0.4\baselineskip` in the slot font. No effect on absolutely positioned slots.

Also accepted, but currently without visible effect: `rotate` (default 0), `opacity` (1), `shadow`, `uppercase`, and `lowercase` (each `false`). Do not rely on these keys for your layout.

8.14. Area decoration

`\decorationltxtalkarea`{*<area>*}{*<options>*}
Changes the specified properties of the area instance, retaining the others.

background = {*<colour>*} Default: none
Solid background or no fill.

8. Reference

background-split = { $\langle$ colour list $\rangle$ } Default: empty
Adjacent colour fields; colour*2 gives double weight. Use valid colour names such as `ltxtalk-primary`.

background-gradient = { $\langle$ colours $\rangle$ } Default: empty
Two colours {top,bottom} for a vertical gradient.

frame = none|box|top|bottom|left|right|topbottom Default: none
top, bottom, and topbottom draw horizontal rules. left and right draw the corresponding side. box currently draws only the top and bottom rules.

rule-color = { $\langle$ colour $\rangle$ } Default: `ltxtalk-primary`
Area-rule colour.

rule-thickness = { $\langle$ length $\rangle$ } Default: 0.4pt
Area-rule width.

padding = { $\langle$ length $\rangle$ } Default: 0pt
Spacing after a left area rule; does not move slots.

artifact-type = { $\langle$ name $\rangle$ } Default: layout
Descriptive field; standard output does not currently use its value.

The keys `background-opacity` (default 1), `shadow` (false), `shadow-color` (black!30), `shadow-offset` (3pt), `margin-top`, and `margin-bottom` (each 0pt) are accepted but currently have no visible effect. Use `\ltxtalksetup` for geometry.

8.15. Title placement

`\setltxtalktitleplacement`{ $\langle$ options $\rangle$ }
Declares the intended title placement with `frametitle=content|header|both` and `section=header|none`. The active theme must define slots in the selected areas.

`\setltxtalkheadingbehavior`{ $\langle$ options $\rangle$ }
Configures, with the Boolean key `auto-section-title` (default: false), whether the first `\frametitle` frame after `\section`/`\subsection` takes over its numbered heading as the title. Text given at that same frame becomes the subtitle; a separately set `\framesubtitle` is dropped.
Known issue: With `tagging=on`, this option can cause a “TeX capacity exceeded” error with some frame content. Compile the complete document and disable `auto-section-title` if that error occurs.

`\contframetitle`[$\langle$ subtitle $\rangle$]
Repeats the numbered heading most recently set by `\section` or `\subsection`, adding a continuation marker. Use it inside a new frame when a topic spans several separate frames. It is independent of `auto-section-title` and does not repeat the last manually set frame title. The optional argument supplies a subtitle; there is no mandatory argument. As with `\frametitle`, an overlay specification in angle brackets may precede the optional argument. For overlays within the same frame, `\frametitle` suffices.

8.16. Accessibility options

The fourth argument of `\defineltxtalkslot` specifies the semantic role and, optionally, alternative text. These settings take effect when tagging is enabled. Empty slots create no structure element. The class handles tagging for the `frametitle` type; the following options do not change its structure.

`role = {\langle role \rangle}` Default: artifact
 heading marks a heading at the level selected by `heading-level`; `heading-1`, `heading-2`, and `heading-3` select that level directly. `note` marks a note. `artifact` and `navigation` exclude decorative or recurring navigation elements from the reading order. `content` uses automatic paragraph tagging for ordinary content.

`heading-level = {\langle integer \rangle}` Default: 1
 Heading level for `role=heading`, limited to 1 through 6.

`alt-text = {\langle text \rangle}` Default: empty
 Alternative text for the slot content. A non-empty value preserves heading and note roles; for every other role, the slot becomes a figure (Figure) with alternative text. This includes it in the document structure even with `artifact` or `navigation`. Leave the value empty for purely decorative elements.

`continuation-text = {\langle text \rangle}`
 Intended continuation text; currently without output effect.

`continuation-mode = explicit|implicit`
 Intended continuation mode; currently without output effect.

Split and gradient area backgrounds are always explicitly emitted as artifacts, independently of the above.

8.17. Templates and sockets

Every slot has an instance of the `lttheme-slot` template type and every area an instance of `lttheme-area`. Use the previously documented slot styling and area decoration commands to edit them.

The page style provides sockets for the page background, the header, left, right, and footer areas, and the page foreground. Their names follow `lttheme/area/\langle Area \rangle` and `lttheme/page/\langle Layer \rangle`. The standard plug renders the corresponding area. Custom plugs are intended for fundamental replaceable output phases; additional page elements should normally be slots in a partial theme.

Index

A	
<code>\addltxtalktheme</code>	18
<code>align</code>	26
<code>alt-text</code>	28
<code>artifact-type</code>	27
B	
<code>background</code>	26
<code>background-fill</code>	26
<code>background-gradient</code>	27
<code>background-split</code>	27
<code>block-hairlines</code>	18, 20, 24
C	
<code>color</code>	26
<code>content-inset-left</code>	20
<code>content-inset-right</code>	20
<code>\contframetitle</code>	27
<code>continuation-mode</code>	28
<code>continuation-text</code>	28
D	
<code>\DeclareInstance</code>	22
<code>\DeclareLTXTalkAreaGrid</code>	25
<code>\DeclareLTXTalkFontPalette</code>	22
<code>\DeclareLTXTalkLayout</code>	18
<code>\DeclareLTXTalkNavigation</code>	21
<code>\DeclareLTXTalkTitleLayout</code>	25
<code>\decorationltxtalkarea</code>	26
<code>\defineltxtalkslot</code>	24
E	
<code>\EditInstance</code>	22
F	
<code>font</code>	26
<code>font-slot</code>	26
<code>footer</code>	19
<code>footer-columns</code>	20
<code>footer-height</code>	20
<code>frame (environment)</code>	27
<code>frame</code>	26 f.
<code>\framesubtitle</code>	10, 27
<code>\frametitle</code>	10, 27
H	
<code>header</code>	19
<code>header-columns</code>	20
<code>header-height</code>	20
<code>header-rows</code>	20
<code>heading-level</code>	28
I	
<code>\IfLTXTalkTitlePageTF</code>	25
<code>\InsertLTXTalkElement</code>	21
L	
<code>left-area-width</code>	20
<code>left-margin</code>	19
<code>linebreak</code>	21
<code>lines</code>	21
<code>ltx-talk (class)</code>	2, 4, 11 ff., 16, 25
<code>ltxtalk-theme (package)</code>	14
<code>\LTXTalkBaseBottomMargin</code>	19
<code>\LTXTalkBaseLeftMargin</code>	19
<code>\LTXTalkBaseRightMargin</code>	19
<code>\LTXTalkBaseTopMargin</code>	19
<code>\LTXTalkCurrentBottomMargin</code>	19
<code>\LTXTalkCurrentLeftMargin</code>	19
<code>\LTXTalkCurrentRightMargin</code>	19
<code>\LTXTalkCurrentTopMargin</code>	19
<code>\LTXTalkHeaderRows</code>	25
<code>\LTXTalkMetadata</code>	20
<code>\LTXTalkNavigation</code>	21
<code>\ltxtalksetup</code>	18
M	
<code>\MakeLTXTalkTitle</code>	25
<code>margin-width</code>	20
O	
<code>outer-only</code>	18
P	
<code>padding</code>	26 f.
<code>\positionltxtalkslot</code>	25
<code>\PositionLTXTalkSlotInGrid</code>	25
R	
<code>\RegisterLTXTalkTheme</code>	18
<code>right-area-width</code>	20
<code>right-margin</code>	19
<code>role</code>	28
<code>rule-color</code>	26 f.
<code>rule-thick</code>	26
<code>rule-thickness</code>	27
S	
<code>\section</code>	10, 27
<code>\SetLTXTalkAreaGridRows</code>	25
<code>\setltxtalkcolors</code>	22
<code>\setltxtalkheadingbehavior</code>	27
<code>\setltxtalkkitemize</code>	24

INDEX

<code>\setltxtalkslotcontent</code>	24	U	
<code>\setltxtalktitleplacement</code>	27	<code>\UseLTXTalkFont</code>	23
<code>shadow</code>	18	<code>\UseLTXTalkFontPalette</code>	22
<code>space-after</code>	26	<code>\UseLTXTalkLayout</code>	19
<code>\styleltxtalkblock</code>	23	<code>\UseLTXTalkMetadata</code>	20
<code>\styleltxtalkslot</code>	25	<code>\useltxtalktheme</code>	18
<code>\subsection</code>	10, 27	<code>\UseLTXTalkTitleLayout</code>	25