The framecite package

Tobias J. Bauer
https://github.com/Hirnmoder/framecite

Version 0.8, released 2026/11/20

Abstract

framecite shows the sources cited on a beamer slide as a small block at the bottom of that same slide. Each entry renders the citation label, the author or editor, and the quoted title. The block sits between real LaTeX footnotes and the theme footline. It is populated automatically from ordinary citation commands and can be extended manually.

Contents

1 Introduction 1
2 Requirements and loading 2
3 Package options 2
4 Automatic collection 3
5 Manual registration 3
6 Layout and styling 4
7 Overlay behavior 4
8 A complete example 4
9 Known limitations 8
10 License 9
11 Implementation 9

1 Introduction

Short-form citations such as [2] keep slide text readable, but the audience can no longer see where a claim comes from. This package restores that link: on every frame where a citation appears, the source itself is listed at the bottom of that frame. Each entry has the form

```
<label> <author>:  \mkbibquote{<title>}
```

where *label* is whatever the active biblatex style prints (`[2]` for `numeric`, `[XYZ+23]` for `alphabetic`), and *title* is the `shorttitle` field if present, else the `title` field. Entries for one frame appear in order of first citation, and a work cited more than once on a frame is listed only once.

The source block is written into beamer's footnote insert at frame end, above the theme footline and below any real footnotes. It uses a blank footnote mark and its own beamer font/color element, so it reads as a separate 'sources' block and never disturbs the theme.

Bugs and feature requests can be reported in the package repository at `https://github.com/Hirnmoder/framecite/issues`.

2 Requirements and loading

The package requires the LaTeX beamer class; load framecite only in a beamer document. It further requires biblatex, which must be loaded *before* framecite (the wrapping of the cite commands depends on it). etoolbox and xspace are loaded automatically. Titles in the source block are wrapped in biblatex's `\mkbibquote`, so the quotation marks follow your setup: if you load csquotes — with any options, anywhere in the preamble — they come from its `\enquote`; otherwise biblatex's quotation fallback applies, as for entry titles in the bibliography. A minimal preamble:

```
\documentclass{beamer}
\usepackage[backend=biber, style=numeric]{biblatex}
\addbibresource{references.bib}
\usepackage{framecite}
```

Both requirements are enforced: loading without beamer or before biblatex terminates with a package error that names the fix.

3 Package options

- `patchcite` (default) intercepts the short-form cite commands so that their keys feed the per-frame source list. `nopatchcite` leaves those commands untouched; register sources manually with `\fcite` or `\fsource` instead.
- `label=margin` (default) places the cite label in the left margin, so the title (and its continuation lines) stays flush at the left edge, like a normal footnote. `label=inline` sets label and title in one flow: the entry starts with the label. In `margin` mode the label juts into the gutter with its natural width.
- `rule=auto/always/never` (default `auto`) controls the rule above the footnote region. `auto` draws the rule only when a real `\footnote` was called on the frame; source lines do not count. `always` draws it whenever any footnote content is present (beamer's default). `never` suppresses it entirely.

4 Automatic collection

With `patchcite` (the default) the short-form commands that print a citation label are intercepted: `\cite`, `\parencite`, and `\textcite`. Their optional `[pre][post]` arguments are forwarded untouched. All entries of a multi-key cite are listed separately; repeated cites of one key on a frame are listed once.

Starred variants are captured as well: the star is parsed and forwarded to the original command, so `\cite*{key}` still feeds the frame list.

Four families stay unordered *on purpose*:

- the field extractors `\citeauthor`, `\citeyear`, and `\citetitle` print only the name, the year, or the title — no citation label — so the block has no marker on the slide to explain. Register such keys with `\fsource` if you want the source line anyway.
- `\footcite` and `\footcitetext` already place the full citation into the footnote; an additional source line would just duplicate it. Use `\fsource{key}` if you want the key in the frame list anyway.
- `\fullcite` and `\footfullcite` already show the full entry on the slide.
- the multi-set commands `\cites`, `\textcites`, ... and the invisible `\nocite` are not captured.

`\ocite` Independently of `patchcite`, the untouched original cite command is always available as `\ocite`: it takes the same arguments as `\cite` — star, pre-/postnote, key lists — prints the style label, and never feeds the per-frame source list. Use it where a source line would only repeat what the slide already shows, and in frame titles, section titles, and other headings: beamer expands these late and theme-dependently, so a patched cite there can miss its frame or leak into the next one.

`\oparencite` `\otextcite` `\oparencite` and `\otextcite` are the matching originals of the other two wrappable commands, with the same rules.

5 Manual registration

`\fcite` `\fcite` feeds the per-frame source list and prints the style-correct label in the text, like an ordinary citation:

```
\fcite[pre]{key}[post]
```

The optional *pre* and *post* texts are shown in the footer line only: *pre* before the label, *post* after the title. They are intended for page numbers or remarks ('see', 'ch. 3'). Use `\fcite` when `patchcite` is off, or to attach extra information to a source line.

`\fsource` `\fsource` is the *blind* variant. It registers the key (and optional *pre*/*post* texts) exactly like `\fcite`, but typesets nothing in the text. It is safe between words: no spurious space remains. Use it for implicit references (images, background claims) where a visible label would distract.

Both commands accept the same starred form. Unstarred calls insert an `\xspace` after *pre* and before *post* automatically; the starred `\fcite*`/`\fsource*` stores the texts verbatim, so you control the spacing exactly. A missing optional

argument is treated as empty. If a key is registered more than once on a frame, the list keeps one entry; the last call determines its *pre*/*post* texts.

6 Layout and styling

The bottom of a frame is stacked, top to bottom: slide body, real `\footnote` region (drawn by the theme), the framecite source block, then the theme footline. The source block occupies its own vertical space (like a real footnote region, it reduces the slide body height), so it never overlaps content and never gets clipped. Real footnotes always stay above the source lines.

Each entry prints as `<label> <names>: \mkbibquote{<title>}`. The names are the author as biblatex prints them; if the entry has no author, the editor is named instead; an entry with neither prints the bare quoted title (never a stray colon).

The block is styled through the beamer element `framecite`; override it as usual:

```
\setbeamerfont{framecite}{size=\tiny}
\setbeamercolor{framecite}{fg=normal text.fg!70}
```

The defaults set `size=\tiny` and `fg=normal text.fg!70`. Entries in the block render with tight line spacing; the blank footnote mark leaves the visual room to the label, which the entry itself repeats in the style of the running bibliography.

7 Overlay behavior

A frame with overlays is typeset once per slide. All collection state is global, so each slide shows the source block exactly once. What is listed follows where the cite command *executes*, not where it becomes visible: uncover-style hiding (`\pause`, `\onslide`, `\visible`, `<n>` item specifications) still typesets in every pass, so such sources are listed from the first slide. `\only`-gated cites execute on their slides only and are listed from there onward.

8 A complete example

The demo file framecite_demo.tex is generated from this source by `framecite.ins`, so the listing below and the generated file cannot drift apart. It is fully self-contained: the `.bib` data is embedded via `filecontents`. Build it with `latexmk -pdf` (or `pdflatex`, `biber`, `pdflatex`, `pdflatex`) in a directory that contains framecite.sty, e.g. the one generated by `framecite.ins`.

```
⟨*demo⟩
% framecite: per-frame mini references. Load AFTER biblatex.
\documentclass[aspectratio=169, 10pt]{beamer}

\usetheme[progressbar=frametitle, numbering=fraction]{metropolis}

\usepackage[backend=biber, style=numeric, sortcites]{biblatex}
% To see alphabetic labels like [Knu84] instead, switch to:
%   \usepackage[backend=biber, style=alphabetic]{biblatex}
```

```
10
11 \begin{filecontents*}[overwrite]{\jobname.bib}
12 @article{knuth1984,
13   author    = {Donald E. Knuth},
14   title     = {Literate Programming},
15   shorttitle= {Literate Programming},
16   journal   = {The Computer Journal},
17   volume    = {27},
18   number    = {2},
19   pages     = {97--111},
20   year      = {1984},
21 }
22 @book{lamport1994,
23   author    = {Leslie Lamport},
24   title     = {{LaTeX: A Document Preparation System}},
25   shorttitle= {{LaTeX}},
26   publisher = {Addison-Wesley},
27   year      = {1994},
28 }
29 @inproceedings{mittelbach2004,
30   author    = {Frank Mittelbach and Michel Goossens and Johannes Braams
31                and David Carlisle and Chris Rowley},
32   title     = {The {\LaTeX} Companion},
33   shorttitle= {{LaTeX} Companion},
34   publisher = {Addison-Wesley},
35   year      = {2004},
36 }
37 @proceedings{nips2017,
38   editor    = {Isabelle Guyon and Ulrike von Luxburg and Samy Bengio
39                and Hanna Wallach and Rob Fergus and S. V. N. Vishwanathan
40                and Roman Garnett},
41   title     = {Advances in Neural Information Processing Systems 30},
42   shorttitle= {NeurIPS 30},
43   publisher = {Curran Associates},
44   year      = {2017},
45 }
46 \end{filecontents*}
47 \addbibresource{\jobname.bib}
48
49 \usepackage[rule=auto, label=margin]{framecite}
50
51 \title{framecite}
52 \subtitle{Per-frame mini references for beamer + biblatex}
53 \author{Tobias J. Bauer}
54 \date{v0.8 (2026-11-20)}
55
56 \begin{document}
57
58 \begin{frame}
59   \titlepage
60 \end{frame}
61
62 \begin{frame}{The idea}
63   Short-form citations stay readable in the text, but the audience still
```

```
  sees the source on the very slide: the literate-programming idea goes
  back to~\cite{knuth1984}, and the standard reference
  is~\cite{lamport1994}. \pause

  \medskip
  Each cited work appears once below, in first-citation order --- even if
  you cite it repeatedly~\cite{knuth1984,knuth1984}.
\end{frame}

\begin{frame}{Automatic collection}
  With \texttt{patchcite} (the default) the short-form commands that
  print a citation label are intercepted: \texttt{\string\cite},
  \texttt{\string\parencite}, and \texttt{\string\textcite}.

  \medskip
  The field extractors \texttt{\string\citeauthor},
  \texttt{\string\citeyear} and \texttt{\string\citetitle} print no
  label, so they leave the block alone; \texttt{\string\fsource}
  registers such keys manually.

  \medskip
  \parencite{mittelbach2004} extends the basics, while
  \textcite{knuth1984} coined the term. Proceedings with editors but no
  authors are handled too~\cite{nips2017}: the source line names the
  editor.
\end{frame}

\begin{frame}{What is \emph{not} intercepted}
  \texttt{\string\footcite} and \texttt{\string\footfullcite} already
  place the full citation in the footnote, so they are left alone --- an
  extra source line would only duplicate it.

  \medskip
  For comparison, here is a full footnote
  citation~\footfullcite{lamport1994}, with no additional source line
  below.

  \medskip
  The untouched originals are always available as
  \texttt{\string\ocite}, \texttt{\string\oparencite} and
  \texttt{\string\otextcite}:~\ocite{knuth1984} prints the style label
  and feeds nothing below --- with or without \texttt{patchcite}. Use
  them in frame titles and section headings, where a patched cite can
  misfire.
\end{frame}

\begin{frame}{Manual registration}
  \texttt{\string\fcite\,[pre]\{key\}[post]} prints the label and
  registers the source; the optional pre/post texts appear in the footer
  line only --- before the label and after the title.

  \medskip
  The idea goes back to~\fcite[cf.]{knuth1984}[ch.~2], and the standard
  reference is~\fcite{lamport1994}.\pause
```

```
118
119   \medskip
120   \texttt{\string\fsource} is the \emph{blind} variant: it registers the
121   source without printing a marker.\fsource{mittelbach2004} Nothing
122   visible here, yet the \LaTeX{} Companion appears below.
123 \end{frame}
124
125 \begin{frame}{Overlay behavior}
126   A source is listed where the cite \emph{executes}, not where it becomes
127   visible.
128   \medskip
129   \begin{itemize}
130     \item \cite{knuth1984} sits in the open: listed on every slide.
131     \item<2-> \only<2>{\fcite{lamport1994}} runs on slide~2 only, so Lamport
132       appears from slide~2.
133   \end{itemize}
134   \medskip
135   Uncover-style hiding (\texttt{\string\pause},
136   \texttt{\string\onslide}, \texttt{\string\visible}) still typesets every
137   pass; \texttt{\string\only} discards its content on other slides.
138 \end{frame}
139
140 \begin{frame}{Configuration}
141   Three option groups, set at load time:
142   \begin{description}
143     \item[\texttt{patchcite}] intercept the standard cite commands
144       (default). \texttt{nopatchcite} leaves them untouched; register
145       sources with \texttt{\string\fcite} or \texttt{\string\fsource}.
146     \item[\texttt{label=margin}] place the label in the left margin
147       (default); \texttt{label=inline} sets label and title in one flow.
148     \item[\texttt{rule=auto}] draw the footnote rule only when a real
149       \texttt{\string\footnote} was called (default). \texttt{always}
150       draws it for any footnote content; \texttt{never} suppresses it.
151   \end{description}
152   \medskip
153   This demo loads \texttt{[rule=auto, label=margin]}.
154 \end{frame}
155
156 \begin{frame}{Known limitations}
157   \begin{itemize}
158     \item In \texttt{\string\frametitle}, \texttt{\string\framesubtitle},
159       and section titles, cite with \texttt{\string\ocite}: titles
160       expand late and theme-dependently, so a patched cite there can
161       miss its slide, attach to the next frame, or work in one theme
162       and fail in another. Add \texttt{\string\fsource} in the body
163       if the source should still be listed.
164     \item The source order is first appearance, not bibliography order.
165     \item The multi-set commands \texttt{\string\cites},
166       \texttt{\string\textcites}, \ldots\ and \texttt{\string\nocite} are
167       not captured.
168     \item In \texttt{rule=auto} mode, a \texttt{\string\footnote} inside a
169       \texttt{minipage} (without the \texttt{[frame]} option) falsely
170       draws the rule above the source block: the wrapper cannot tell a
171       minipage-scoped footnote from a frame-scoped one (next slide).
```

```
  \end{itemize}
\end{frame}

\begin{frame}{Limitation: \texttt{rule=auto} and minipage footnotes}
  A \texttt{\string\footnote} inside a \texttt{minipage} (without the
  beamer \texttt{[frame]} option) is set at the minipage bottom --- standard
  beamer behavior, not a bug. The subtle part is the interaction with
  \texttt{rule=auto}: the rule above the source block should appear only
  when a real frame-level footnote sits above it. The package wraps
  \texttt{\string\footnote} to flag ``real footnote'', but cannot tell a
  minipage-scoped call from a frame-scoped one, so a minipage footnote
  still raises the flag.\vspace{2ex}

  \begin{minipage}{0.46\linewidth}
    A footnote inside a minipage.\footnote{Placed at the minipage bottom,
    yet it draws the rule above the source lines.}
  \end{minipage}\hfill%
  \begin{minipage}{0.46\linewidth}
    The cite here registers normally~\cite{mittelbach2004}, and a rule
    appears above it though no frame footnote is present.
  \end{minipage}\vspace{2ex}

  Workaround: \texttt{\string\footnote[frame]\{...\}} for a genuine frame
  footnote, or \texttt{rule=never} / \texttt{rule=always}.
\end{frame}

\begin{frame}[allowframebreaks]{References}
  \printbibliography
\end{frame}

\end{document}
⟨/demo⟩
```

9 Known limitations

- In `\frametitle`, `\framesubtitle`, and section titles, cite with the untouched `\ocite`, `\oparencite`, or `\otextcite`. Beamer expands such titles late and theme-dependently, typically after the frame-end source flush: a patched cite there can miss its own slide, attach to the next frame, or seem to work until the theme changes. To have the source listed anyway, register it with `\fsource` in the frame body.
- The *order* of the source block is first appearance on the frame, not bibliography order. A bibliography-ordered block would need per-frame `refsection`s; this is out of scope.
- The multi-set commands `\cites` and friends, and `\nocite`, are not captured.
- The package targets the `numeric` and `alphabetic` styles. With `verbose`/`authoryear` the source block still renders, but calling the original cite command inside the block may influence biblatex's cite tracker (first/short-form logic). If you rely on verbose shortening, test carefully.

- In `rule=auto` mode, a `\footnote` called inside a `minipage` (without beamer's `[frame]` option) falsely raises the "real footnote" flag and draws the rule above the source block, even though no frame-level footnote sits above it. The package's `\footnote` wrapper cannot distinguish a minipage-scoped call from a frame-scoped one. Use `\footnote[frame]{...}` for a genuine frame footnote, or `rule=never` / `rule=always`.

10 License

The files of this work are distributed under the LaTeX Project Public License, version 1.3c or later; see the file header for details. The current maintainer is Tobias J. Bauer.

11 Implementation

The implementation order follows the file: options, guards, the per-frame collection, the rendering commands, the frame-end flush, the cite-command interception, and the styling hooks.

The package needs e-TeX (etoolbox keys off it), the beamer class (we patch `\endbeamer@frameslide` and hook the `frame` environment), and biblatex (`\DeclareCiteCommand`).

```
⟨*package⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{framecite}[2026/11/20 v0.8 Beamer per-frame mini references]
```

First the options. `patchcite`/`nopatchcite` select whether the cite commands are wrapped at all.

```
\newif\iframecite@patchcite
\framecite@patchcitetrue
\DeclareOption{patchcite}{\framecite@patchcitetrue}
\DeclareOption{nopatchcite}{\framecite@patchcitefalse}
```

The label placement modes. The public option is `label=margin/inline`; internally the mode lives in the `\iframecite@hang` flag: `margin` means the label is hung into the margin (hanging label), `inline` means label and title flow in one line.

```
\newif\iframecite@hang
\framecite@hangtrue
\DeclareOption{label=margin}{\framecite@hangtrue}
\DeclareOption{label=inline}{\framecite@hangfalse}
```

The rule visibility mode. The chosen mode string is stored verbatim and compared later against the three saved literals with `\ifx`.

```
\def\framecite@rulemode{auto}
\def\framecite@rule@always{always}%
\def\framecite@rule@auto{auto}%
\def\framecite@rule@never{never}%
\DeclareOption{rule=always}{\def\framecite@rulemode{always}}
\DeclareOption{rule=auto}{\def\framecite@rulemode{auto}}
\DeclareOption{rule=never}{\def\framecite@rulemode{never}}
```

```
\DeclareOption*{\PackageWarning{framecite}{Unknown option `\CurrentOption'}}
\ProcessOptions\relax
```

etoolbox drives the option handling below (lists, csname tools, patching). beamer must already be the document class: we patch `\endbeamer@frameslide` and hook `\AtBeginEnvironment{frame}`.

```
\RequirePackage{etoolbox}
```

Guard: the beamer class.

```
\@ifclassloaded{beamer}{}%
  {\PackageError{framecite}{Load framecite only under the beamer class.}%
   {Use \string\documentclass{beamer}\space BEFORE framecite.}}
```

Guard: biblatex must precede this package, because we wrap its cite commands and call `\DeclareCiteCommand`.

```
\@ifpackageloaded{biblatex}%
  {}%
  {\PackageError{framecite}{Load biblatex BEFORE framecite.}%
   {E.g. \string\usepackage[style=numeric]{biblatex}\space
    \string\addbibresource{refs.bib}\space\string\usepackage{framecite}}}
\RequirePackage{xspace}% \xspace
```

The heart of the per-frame bookkeeping is a *generation counter*. It bumps at every frame start; the seen flag `\framecite@seen@<gen>@<key>` dedups within one frame.

Invariant: all state writes in this package are global (`\csgdef`, `\xdef`, `\g@addto@macro`). beamer typesets a frame once per overlay slide, with the body inside the frame's `vbox` group: group-local writes die at the end of each pass, so every bookkeeping write must survive the group.

`\ifframecite@realfootnote` is raised by the `\footnote` wrapper in `auto` mode and gates the footnote rule. The flush writes `\footnotetext` directly, bypassing that wrapper, so source lines never count as real footnotes.

```
\newif\ifframecite@realfootnote
\newcommand*{\framecite@framekeys}{}
\gdef\framecite@framekeys{}
\gdef\framecite@gen{0}%
\AtBeginEnvironment{frame}{\gdef\framecite@framekeys{}%
  \global\framecite@realfootnotefalse
  \xdef\framecite@gen{\the\numexpr\framecite@gen+1\relax}}%
```

Registering is a two-step affair: split a key list, then record each unseen key.

```
\newcommand*{\framecite@addkeys}[1]{\forcsvlist\framecite@addone{#1}}
```

`\framecite@framekeys` is a token list of `\framecite@emit{`⟨*key*⟩`}` calls: a small *program* list. `\g@addto@macro` appends without re-expanding the old list, and the flush later simply executes the whole macro. Deduplication stays in `\framecite@addone`.

```
\newcommand*{\framecite@addone}[1]{%
  \ifcsdef{framecite@seen@\framecite@gen @#1}{}%
    {\csgdef{framecite@seen@\framecite@gen @#1}{}%
     \g@addto@macro\framecite@framekeys{\framecite@emit{#1}}}%
}
```

The pre/post texts. `\framecite@addprepost` receives the star flag, the key list, and the two optional texts; each key of the list is processed by `\framecite@addonepp`, with the arguments baked in via the partial-application form of `\forcsvlist`.

Starred calls store the texts verbatim; missing texts are stored empty. Unstarred calls wrap *pre* as `<pre>\xspace` and *post* as `\xspace<post>`, and touch nothing when the argument is missing. The storage key is generation-stamped, so pre/post state is per frame like the key list. Note that unlike the list, the texts are not deduplicated: the last registration for a key on a frame wins.

```
\NewDocumentCommand{\framecite@addprepost}{m m m m}{\forcsvlist{\framecite@addonepp{#1}{#3}{#
\NewDocumentCommand{\framecite@addonepp}{m m m m}{%
  \IfBooleanTF{#1}{% star
    \IfValueTF{#2}{\csgdef{framecite@pre@\framecite@gen @#4}{#2}}{\csgdef{framecite@pre@\fram
    \IfValueTF{#3}{\csgdef{framecite@post@\framecite@gen @#4}{#3}}{\csgdef{framecite@post@\fr
  }{% no star
    \IfValueTF{#2}{\csgdef{framecite@pre@\framecite@gen @#4}{#2\xspace}}{}%
    \IfValueTF{#3}{\csgdef{framecite@post@\framecite@gen @#4}{\xspace#3}}{}%
  }%
}
```

Rendering: `\framecite@namesprefix` prints the author names (or, if there is no author, the editor names) with a trailing colon and tie; an entry with neither name prints nothing, so the entry falls back to the bare quoted title and no stray colon remains.

```
\newcommand*{\framecite@namesprefix}{%
  \ifnameundef{author}%
    {\ifnameundef{editor}{}{\printnames[][1-]{editor}:~}}%
    {\printnames[][1-]{author}:~}}
\DeclareCiteCommand{\framecite@cite}
  {}
  {\framecite@namesprefix\mkbibquote{\iffieldundef{shorttitle}{\printfield{title}}{\printfiel
  {}
  {}
```

One entry, two shapes, selected by the placement mode.

```
\newcommand*{\framecite@entrytext}[1]{%
  \iframecite@hang
```

In `margin` mode the pre text (if any), the label, and a small gap are pushed left into the margin with `\llap`. The title stays flush at the left edge like a normal footnote; the total width of the `\llap` box is zero, its content juts leftward into the gutter, and `\enspace` is the gap.

```
    \llap{\csuse{framecite@pre@\framecite@gen @#1}\framecite@orig@cite{#1}\enspace}%
    \framecite@cite{#1}\csuse{framecite@post@\framecite@gen @#1}%
  \else
    \csuse{framecite@pre@\framecite@gen @#1}\framecite@orig@cite{#1}\enspace\framecite@cite{#
  \fi
}
```

`\fcite` prints the style-correct label in the text and delegates the registration to `\fsource`; the star is passed through, so the starred form stores the texts verbatim.

```
\NewDocumentCommand{\fcite}{s o m !o}{%
  \framecite@orig@cite{#3}%
  \IfBooleanTF{#1}{\fsource*[#2]{#3}[#4]}{\fsource[#2]{#3}[#4]}%
}
```

`\fsource` registers the key and the texts but typesets nothing itself. `\@bsphack`/`\@esphack` swallow the surrounding space, so no spurious white remains wherever it is used.

```
\NewDocumentCommand{\fsource}{s o m !o}{\@bsphack\framecite@addkeys{#3}\framecite@addprepost{
```

The flush runs at the end of `\endbeamer@frameslide`, that is, after the body but before the footnote output. Real `\footnote`s therefore land above the flushed source lines. (beamer ignores `\AtEndEnvironment` here, hence the patch made below.) In a `.sty` file `@` is already a letter, so no `\makeatletter` is needed.

```
\newcommand*{\framecite@flush}{%
  \ifx\framecite@framekeys\@empty\else
    \begingroup
      \renewcommand{\@makefnmark}{}% blank mark: the source line itself carries the cite labe
```

The key list is executed directly. `\framecite@emit` is bound here, locally, so every stored `\framecite@emit{`⟨*key*⟩`}` emits one `\footnotetext` with the framecite font and color and a tighter line spread for the tiny source lines.

```
      \def\framecite@emit##1{\footnotetext{\usebeamerfont{framecite}\usebeamercolor[fg]{frame
          \linespread{0}\selectfont% tighter line spacing for the tiny source lines
          \raggedright\framecite@entrytext{##1}}}%
      \framecite@framekeys% executes \framecite@emit{k1}\framecite@emit{k2}\ldots
    \endgroup
  \fi
}
```

The interception itself. For each command we save the current meaning under `\framecite@orig@<cmd>` with `\csletcs` (a plain `\let` cannot handle csnames built at runtime), then redefine the public command with an initial `\expandafter` so that `\RenewDocumentCommand` sees the resolved csname. The wrapper spec `s o o m` parses star and both optional arguments, forwards every combination to the saved original, and registers the keys along the way; starred variants keep their star.

```
\newcommand*{\framecite@patch}[1]{%
  \@ifundefined{#1}%
    {\PackageWarning{framecite}{Cannot patch \string#1: not defined.}}%
    {\csletcs{framecite@orig@#1}{#1}%
     \expandafter\RenewDocumentCommand\csname #1\endcsname{s o o m}{%
       \framecite@addkeys{##4}%
       \IfBooleanTF{##1}%
         {\IfValueTF{##2}%
            {\IfValueTF{##3}%
               {\csname framecite@orig@#1\endcsname*[##2][##3]{##4}}%
               {\csname framecite@orig@#1\endcsname*[##2]{##4}}}%
            {\csname framecite@orig@#1\endcsname*{##4}}}%
         {\IfValueTF{##2}%
            {\IfValueTF{##3}%
               {\csname framecite@orig@#1\endcsname[##2][##3]{##4}}%
               {\csname framecite@orig@#1\endcsname[##2]{##4}}}%
            {\csname framecite@orig@#1\endcsname{##4}}}}}%
}
```

At `\AtBeginDocument` three things happen.

First, the original `\cite` is saved unconditionally: `\fcite` prints it, the flush renders the labels with it, and neither may recurse into the wrapped `\cite`. For each wrappable command, `\ocite`, `\oparencite`, and `\otextcite` snapshot the untouched original here, before the patch list runs. This works whether or not `patchcite` is on. Second, the frame-end hook is installed: one `\patchcmd` call adds the flush after the `\par` that closes the body of `\endbeamer@frameslide`.

```
\AtBeginDocument{%
  \csletcs{framecite@orig@cite}{cite}%
  \csletcs{ocite}{cite}%
  \csletcs{oparencite}{parencite}%
  \csletcs{otextcite}{textcite}%
  \patchcmd{\endbeamer@frameslide}{\par}{\par\framecite@flush}{}%
    {\PackageWarning{framecite}{Could not patch \string\endbeamer@frameslide}}%
```

Third, the patch list: only `\cite`, `\parencite`, and `\textcite` are wrapped — the short-form commands that print a citation label. The field extractors `\citeauthor`, `\citeyear`, and `\citetitle` print no label, so they leave the source block alone. `\footcite` and `\footcitetext` are out as well: they already place the full citation in the footnote. `\fsource` feeds any of these keys manually if ever wanted.

```
  \iframecite@patchcite
    \framecite@patch{cite}\framecite@patch{parencite}\framecite@patch{textcite}%
  \fi
```

The rule modes. beamer draws `\footnoterule` once per frame (in `\beamer@autobreakframebox`) when the footnote insert is non-empty. We keep a copy of the original and redefine the rule per mode: `never` drops it, `auto` draws it only when the real-footnote flag is up, `always` leaves beamer's default intact.

```
  \let\framecite@orig@footnoterule\footnoterule
  \ifx\framecite@rulemode\framecite@rule@never
    \renewcommand{\footnoterule}{}%
  \else\ifx\framecite@rulemode\framecite@rule@auto
    \renewcommand{\footnoterule}{\iframecite@realfootnote\framecite@orig@footnoterule\fi}%
  \else % always: leave beamer's default \footnoterule intact
  \fi\fi
```

In `auto` mode we wrap `\footnote` so that real footnotes raise the rule flag. The wrapper parses the optional `<overlay>` with `\@ifnextchar` and forwards everything to the saved original; calling the saved token avoids recursion. The flag assignment is deliberately global, because `\footnoterule` runs later inside the output routine.

```
  \ifx\framecite@rulemode\framecite@rule@auto
    \csletcs{framecite@orig@footnote}{footnote}%
    \def\footnote{\@ifnextchar<\framecite@footnote@ov\framecite@footnote@noov}%
    \def\framecite@footnote@ov<#1>{\global\framecite@realfootnotetrue\framecite@orig@footnote
    \def\framecite@footnote@noov{\global\framecite@realfootnotetrue\framecite@orig@footnote}%
  \fi
}
```

Finally the styling: the beamer element `framecite` controls font and color of the source lines only, never of real footnotes.

```
\setbeamercolor{framecite}{fg=normal text.fg!70}
\setbeamerfont{framecite}{size=\tiny}

\endinput
⟨/package⟩
```

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

F

\fcite *3*, *11*
\fsource *3*, *12*

O

\ocite *3*
\oparencite *3*
\otextcite *3*