Citation Forms and Commands (`iface.dtx`)

Charles Duan

May 2, 2023

Depending on grammatical context, a legal citation in a work will take on one of several forms. *Hereinafter* supports these different grammatical forms of citation by offering several citation commands in this section.[1] This section describes each citation form along with the command used to access the form.

These variables communicate what type of citation is being produced within a citation command.

Conditional for whether or not we are doing an inline citation. This is used to determine, for instance, whether to abbreviate party names.

```
\newif\if@hi@inline \@hi@inlinefalse
```

Conditional that is true if we're writing a sentence, rather than a clause.

```
\newif\if@hi@sent
```

Conditional that is false if we're doing an inline followup cite (so don't show the case name).

```
\newif\if@hi@name \@hi@nametrue
```

Conditional for if this citation is an adjective form (no leading "the")

```
\newif\if@hi@adjective
```

1 Sentences

A *sentence*-form citation is set off from the text as a separate sentence:

> Example: `Segregation is unconstitutional.`
> `\sentence{see brown at 495}.`
> Output: Segregation is unconstitutional. *See Brown v. Bd. of Educ.*, 347 U.S. 483 (1954).

`\sentence` As seen above, `\sentence` produces this style of citation. The trailing period is mandatory; it avoids mistakes and assists word processors' detection of missed capitalization. In legal memoranda, the command may appear anywhere in a document. In law review articles, `\sentence` must appear in a footnote, and if it appears in body text, it will be automatically converted to `\note`. However, a sentence-style citation at the beginning of a paragraph in body text (typically used to cite a block quotation) cannot be automatically converted, so a warning is generated.

---

[1] *Hereinafter* also produces a distinct citation form for tables of authorities, but writers do not use this citation form directly.

Law reviews and legal memoranda treat sentence citations in body text slightly differently, so this code defines \sentence depending on the document type.

```
\appto\hi@hooks@review{
    \def\sentence#1{%
        \@test\if@hi@note\fi{%
            \hi@sentence{#1}%
        }{%
            \ifhmode\else
                \hi@sentence@review@warn
                \leavevmode
            \fi
            \unskip
            \footnote{\sentence{#1}.}%
            \@gobble@dot
        }%
    }
    \def\hi@sentence@review@warn{%
        \PackageWarning\hi@pkgname{%
            A sentence citation was used in text, at the beginning\MessageBreak
            of a paragraph; it needs to be moved to the end of the\MessageBreak
            previous paragraph.\MessageBreak
            This occurred after footnote \the\c@footnote%
        }%
    }
}
\appto\hi@hooks@memo{
    \def\sentence{\hi@sentence}%
}
```

For sentence-style citations, flags are set indicating the style and that the first letter should be capitalized. The only trailing punctuation check is whether the citation had a dot at the end. If not, then an additional period to end the sentence must be added. In any event, any deferred footnote is placed after the trailing period.

```
\def\hi@sentence#1{%
    \hi@pse@parse{#1}{\hi@draw@citation{\@hi@captrue\@hi@senttrue}{}}%
    \futurelet\@let@token\hi@end@sentence
}
\def\hi@end@sentence{%
    \ifdim\lastkern>\z@ \unkern \fi
    \@test \ifx\@let@token.\fi{%
        \if@hi@dot
            \@re@dot@space
        \else
            .%
        \fi
        \hi@deferred@note@endstring
        \@gobble
    }{%
        \PackageWarning\hi@pkgname{%
            Sentence citation did not end with period.\MessageBreak
            This occurred%
        }%
        % Add the period
        .\hi@deferred@note@endstring\space
        \expandafter\ignorespaces
    }%
}
```

2 Clauses, Instructions, and Nested Citations

A *clause*-form citation falls within a sentence, preceded by a comma and followed by a comma unless there is other punctuation. Typically clause-style citations are used when different citations are needed for different parts of a single long sentence.

> Example: Education is important, \clause{see brown at 493}, and segregation has a ``detrimental effect,''
> \clause{brown at 494}.
> Output: Education is important, *see Brown*, 347 U.S. at 493, and segregation has a "detrimental effect," *id.* at 494.

`\clause`   The `\clause` command makes a clause-form citation. It should be surrounded by appropriate punctuation, and in particular there must be punctuation immediately following the command. In law review articles, `\clause` must appear in a footnote, and if it appears in body text, it will be automatically converted to `\note`. However, a clause citation followed by punctuation that is not a comma cannot be automatically converted, so a warning is generated.

2(a) Instructional Citations   Citations may also appear in instructional sentences that end with a `\clause` citation. Typically these will be imperative sentences with an unitalicized spelled-out signal:

> Example: `On scientific evidence, see \clause{brown at 494 note 11}`.
> Output: On scientific evidence, see *Brown*, 347 U.S. at 494 n.11.

Although this context differs grammatically from other `\clause` citations, the formatting is largely identical so only one command is provided.[2]

2(b) Nested Citations   Citations may also appear nested in parentheticals within other citations. This can be accomplished by using `\clause` inside the parenthetical:

> Example: `\sentence{see brown at 494-495 (overruling \clause{plessy})}`.
> Output: *See Brown*, 347 U.S. at 494–95 (overruling *Plessy v. Ferguson*, 163 U.S. 537 (1896)).

More on nested citations, including a more convenient shortcut form, may be found in `parens.dtx`.

```
\appto\hi@hooks@review{
    \def\clause#1{%
        \@test\if@hi@note\fi{%
            \hi@clause{#1}%
        }{%
            \unskip
            \footnote{\sentence{#1}.}%
            \futurelet\@let@token\hi@clause@review@end
        }%
    }
    \def\hi@clause@review@end{%
        \ifx\@let@token,
            \expandafter\@gobble
        \else
            \hi@clause@review@warn
        \fi
    }%
    \def\hi@clause@review@warn{%
        \PackageWarning\hi@pkgname{%
            A clause citation was used in text, but its ending\MessageBreak
            punctuation could not be reconciled automatically.\MessageBreak
            This occurred at footnote \the\c@footnote%
```

---

[2]Some style guides use a slightly different format in which citation items are separated with commas and the word "and" rather than semicolons. This is a somewhat rare usage that could be accomplished by issuing separate `\clause` commands for each citation item, so there seems to be no particular need to implement a separate command for this use case.

```
            }%
        }
    }
}
\appto\hi@hooks@memo{
    \def\clause{\hi@clause}
}
%
% For clauses, no capitalization at the beginning of the sentence is requested.
% Trailing punctuation handling is especially complicated in view of deferred
% footnotes, because footnotes are to be placed after some punctuation but
% before others. As a result, the end of the clause citation tests many
% different types of punctuation for proper handling.
%
%    \begin{macrocode}
\def\hi@clause#1{%
    \hi@pse@parse{#1}{\hi@draw@citation{\@hi@capfalse\@hi@sentfalse}{}}%
    \futurelet\@let@token\hi@end@clause
}
```

This macro is not intended for execution, but is placed at the end of parentheticals. That way, when \clause is used as the last element of a parenthetical, we can give it special treatment to deal with trailing punctuation.

```
\def\hi@clause@endflag{\iffalse\hi@clause@endflag\fi}
```

This is how a clause deals with ending punctuation.

```
\def\hi@end@clause{%
```

If the clause ended with a dot and it is the terminator of a parenthetical, then leave \@hi@dottrue on. Otherwise reset to \@hi@dotfalse (deferred if the clause is immediately followed by a dot). This is necessary for dot management in comma-type parentheticals; see \hi@paren@show.

```
    \if@hi@dot
        \ifx\@let@token\hi@clause@endflag \else
            \ifx\@let@token.\else
                \@hi@dotfalse
            \fi
        \fi
    \fi
```

Now we test each type of trailing punctuation, considering whether to put any deferred footnote before or after it.

```
    \@testcase
    \ifx\@let@token,\fi{%
        \ifdim\lastkern>\z@ \unkern \fi
        ,\hi@deferred@note@endstring\@gobble
    }%
    \expandafter\ifx\space\@let@token\fi{%
        \PackageWarning\hi@pkgname{%
            \string\clause\space not terminated with punctuation%
        }%
        \hi@deferred@note@endstring
    }%
    \ifx\@let@token.\fi{%
        \ifdim\lastkern>\z@ \unkern \fi
        \if@hi@dot
            \@hi@dotfalse % For parenthetical endings
            \@re@dot@space
        \else
            .%
        \fi
        \hi@deferred@note@endstring\@gobble
    }%
    \ifx\@let@token\hi@clause@endflag\fi{\hi@deferred@note@endstring\@gobble}%
    \ifx\@let@token;\fi{\@let@token\hi@deferred@note@endstring\@gobble}%
    \ifx\@let@token:\fi{\@let@token\hi@deferred@note@endstring\@gobble}%
    \ifx\@let@token!\fi{\@let@token\hi@deferred@note@endstring\@gobble}%
    \ifx\@let@token?\fi{\@let@token\hi@deferred@note@endstring\@gobble}%
    \ifx\@let@token-\fi{\hi@deferred@note@endstring}%
    \ifx\@let@token)\fi{\hi@deferred@note@endstring}%
    \default{%
        \PackageWarning\hi@pkgname{%
            \string\clause\space not terminated with punctuation%
        }%
        \hi@deferred@note@endstring
    }%
}
```

3 Inline Citations

An *inline*-form citation acts as a part of speech within a sentence, typically a noun or an adjective. Naming a case in text uses an inline citation:

> Example: \Inline{brown} followed \inline{sweatt}.
> Output: *Brown* followed *Sweatt v. Painter*.

The following commands produce inline citations. Any of them may be used in text or footnotes, regardless of the document type.

<div style="margin-left:2em">\inline</div>

- \inline makes a typical inline citation.

<div>\Inline</div>

- \Inline capitalizes the first letter of the citation.

<div>\adjective</div>

- \adjective removes leading "the" from the citation text, making it amenable to use as an adjective in a sentence.

<div>\Adjective</div>

- \Adjective is like \adjective but capitalizes the first letter.

Hereinafter's support for inline citations is the most comprehensive for cases and statutes, where there are reasonably well-defined conventions for how to form such citations. For other reference types, such as books and articles, the package uses author names to the extent given. Support for more complete coverage of inline citations is a potential future improvement to the package.

Using inline citation commands is preferable to manually entering inline citations in text for several reasons. First, inline citations often have long and short forms, so programmatic formatting of inline citations ensures that the right one is chosen even if paragraphs or sections of text are moved around. Furthermore, inline citations can affect the surrounding non-inline citations. When citing a case for the first time, it is typical to omit the case name from a sentence citation when the case name was used inline:

> Example: \Inline{brown} involved class actions.
> \sentence{see brown at 495}.
> Output: *Brown* involved class actions. *See* 347 U.S. at 495.

This package detects such situations and removes duplicative names automatically.

3(a)   Inline Citations and *Id.* One of the most nuanced considerations for inline citations is the effect on *id.* short forms. Consider the following:

> Example: \sentence{brown at 496}. The Court overruled
> \inline{plessy}. \sentence{brown at 494-495}.
> Output: *Brown*, 347 U.S. at 496. The Court overruled *Plessy*. ???

Absent the inline citation to *Plessy*, the final citation in the above sentence should be "*Id.* at 494–95." But with the inline citation, it is ambiguous which case the *Id.* refers to. The better option, which *Hereinafter* follows, is to use the named short form.

If this is undesirable, one can cause the *Id.* form to be used by hiding the inline citation with braces:

> Example: \sentence{brown at 496}. The Court overruled
> {\inline{plessy}}. \sentence{brown at 494-495}.
> Output: *Brown*, 347 U.S. at 496. The Court overruled *Plessy*. *Id.* at
> 494–95.

By contrast, consider:

> Example: \sentence{brown at 496}. The Court applied
> \inline{14th-amend}. \sentence{brown at 494}. Output:
> *Brown*, 347 U.S. at 496. The Court applied the Fourteenth
> Amendment. *Id.* at 494.

Here, it is unambiguous that the *Id.* does not refer to the constitution. *Hereinafter* implements this distinction by designating the inline forms of statutes, constitutions, and a few other references as "invisible" to the *id.* citation algorithm. This behavior can be overridden by using the \noid command.

User commands for accessing inline citations.

```
\def\inline#1{\hi@inline{#1}{\@hi@capfalse\@hi@sentfalse}}
\def\Inline#1{\hi@inline{#1}{\@hi@captrue\@hi@senttrue}}
\def\adjective#1{\hi@inline{#1}{\@hi@capfalse\@hi@sentfalse\@hi@adjectivetrue}}
\def\Adjective#1{\hi@inline{#1}{\@hi@captrue\@hi@senttrue\@hi@adjectivetrue}}
```

This command may be used in reference definition macros to render inlines invisible as described above.

```
\newif\if@hi@invis@inline
\DeclareRobustCommand\hi@invis@inline{\if@hi@inline \@hi@invis@inlinetrue \fi}
```

Inline citations require a significant amount of state management because of how they affect surrounding citations. The overall strategy is to save the state to the extent possible, produce the citation, and then selectively restore state with appropriate changes (unless the citation was invisible, in which case all state is restored).

```
\def\hi@inline#1#2{%
    \let\hi@inline@lastcase\@last@case
    \hi@state@save\hi@inline@restorestate
    \@hi@invis@inlinefalse
    \@hi@inlinetrue
    \hi@pse@parse{#1}{\hi@draw@citation{#2}{}}%
    \@hi@inlinefalse
    \@hi@adjectivefalse
    \if@hi@invis@inline
        \hi@inline@restorestate
    \else
        \def\hi@inline@setlast{\def\@last@inline}%
        \expandafter\addto@macro\expandafter\hi@inline@setlast\expandafter{%
            \expandafter{\@last@case}%
        }%
        \hi@inline@post
        \hi@inline@setlast
    \fi
    \futurelet\@let@token\hi@end@inline
}
```

In a law review article, non-invisible inline citations inside a footnote restore state only when the inline citation is identical to the last citation. Otherwise the state is reset. Outside a footnote, state is always restored because anything written in text does not affect the footnotes.

For legal memoranda, state is restored only when the inline citation is identical to the last citation.

```
\appto\hi@hooks@review{
    \def\hi@inline@post{%
        \if@hi@note
            \ifx\hi@inline@lastcase\@last@case
                \hi@inline@restorestate
            \else
                \hi@state@reset
            \fi
        \else
            \hi@inline@restorestate
        \fi
    }
}
\appto\hi@hooks@memo{
    \def\hi@inline@post{%
        \ifx\hi@inline@lastcase\@last@case
            \hi@inline@restorestate
        \else
            \hi@state@reset
        \fi
    }
}
```

Dealing with trailing punctuation.

```
\def\hi@end@inline{%
    \@testcase
    \ifx\@let@token.\fi{%
        \unkern
        %
        % Set \@hi@dotfalse only after determining whether to gobble the dot.
        %
        \expandafter\@hi@dotfalse
        \if@hi@dot
            \@re@dot@space\expandafter\@gobble
        \fi
    }%
    \ifx\@let@token,\fi{\@hi@dotfalse\unkern}%
    \ifx\@let@token\hi@clause@endflag\fi{}% Leave \@hi@dottrue alone
    \default{\@hi@dotfalse}%
}
```

Decides whether case names should be omitted after an inline citation to a case. The rule followed here is that the name is omitted after a matching inline cite except when the case is the first citation in a footnote.

```
\def\hi@noname@sometimes{
    \def\hi@noname{%
        \if@hi@note
            \ifnum\hi@note@cites>\z@ \@hi@namefalse \fi
        \else
            \@hi@namefalse
        \fi
    }
}
```

Alternatively, the name can always be suppressed when the appropriate option is set.

```
\def\hi@noname@always{
    \def\hi@noname{\@hi@namefalse}
}
```

4 Note

\note \note makes a sentence-style citation inside a footnote; it is essentially a synonym for:

$$\unskip\footnote\{\sentence\{\langle string\rangle\}.\}$$

It provides two conveniences for the author. First, the `\unskip` means that a space before `\note` is automatically removed, making TEX code more readable. Second, `\note` can take a second argument of text (which can include sentence and clause citations) appended after the citation:

```
\note{case}{For further information, see
\clause{case2}.}
```

The note command first checks if it is inside a footnote already; if so it converts to a sentence citation. It also performs a variety of trailing punctuation checks.

```
\def\note#1{%
    \if@hi@note
        \PackageWarning\hi@pkgname{%
            Cannot create footnote citation inside a footnote%
        }%
        \sentence{#1}.\space\expandafter\ignorespaces
    \else
        \unskip
        \def\hi@note@content{\sentence{#1}.}\expandafter\hi@note@post
    \fi
}%
\def\hi@note@post{\futurelet\hi@note@token\hi@note@post@}
\def\hi@note@post@switch{%
    {\hiendnotelabel}{}%
    {\hinotelabel}{}%
    {\egroup}{}%
    { }{}%
    {-}{}%
    {)}{}%
    {,}{\hi@note@post@warn \spacefactor\sfcode`\,\relax \@gobble}%
    {.}{\hi@note@post@warn \spacefactor\sfcode`\.\relax \@gobble}%
}
\def\hi@note@post@warn{%
    \PackageWarning\hi@pkgname{%
        Unexpected character after \string\note: \meaning\hi@note@token
    }%
}
\def\hi@note@post@{%
    \@test\ifx\hi@note@token\bgroup\fi{%
        \hi@note@post@more
    }{%
        \expandafter\footnote\expandafter{\hi@note@content}%
        \expandafter\hi@note@post@@\hi@note@post@switch\@stop
    }%
}
\def\hi@note@post@more#1{%
    \addto@macro\hi@note@content{ #1}%
    \hi@note@post
}
\def\hi@note@post@@#1#2#3\@stop{%
    \@test\ifx\hi@note@token#1\fi{%
        #2%
    }{%
        \ifstrempty{#3}{%
            \hi@note@post@warn
        }{%
            \hi@note@post@@#3\@stop
        }%
    }%
}
```

5   Optional Clause

\optclause The `\optclause` command is placed immediately after an `\inline` citation and takes no arguments. Its purpose is to serve a style of citation that some writers prefer, in which a clause-style citation (in text or in a footnote, depending on the type of document) immediately follows the first inline citation to a reference:

> Example: In `\inline{brown}\optclause`, the Court
> held...
> Output (case already cited): In *Brown*, the Court held...
> Output (case not already cited): In *Brown v. Board of Education*, 347
> U.S. 483 (1954), the Court held...

In other words, this is identical to `\inline{ref}`, `\clause{ref at 570}`, if the reference has not been cited so far, but produces nothing if it has been cited already.[3]

```
\appto\hi@hooks@review{
    \def\optclause{%
        \@test\if@hi@note\fi{%
            \hi@optclause@memo
        }{%
            \hi@record@choose@cite{\@last@inline}{%
                \futurelet\@let@token\hi@optclause@with
            }{%
                \futurelet\@let@token\hi@optclause@without
            }%
        }%
    }
    \def\hi@optclause@without{%
        \ifcat a\@let@token\space \fi
    }
    \def\hi@optclause@with{%
        \@test\ifx\ \@let@token\fi{%
            \footnote{\sentence{*}.}\ %
        }{%
            \@test\ifcat\space\@let@token\fi{%
                \footnote{\sentence{*}.}%
            }{%
                \@test\ifcat a\@let@token\fi{%
                    \footnote{\sentence{*}.}\space
                }{%
                    \hi@optclause@hop
                }%
            }%
        }%
    }
    \def\hi@optclause@hop#1{%
        #1\futurelet\@let@token\hi@optclause@with
    }
}
\appto\hi@hooks@memo{
    %
    % Adds a clause for *, with no pin cite, unless the case is being
    % short-cited.
    \def\optclause{\hi@optclause@memo}
}
\def\hi@optclause@memo{%
    \hi@record@choose@cite{\@last@inline}{%
        \unkern, \clause{*}\hi@clause@endflag
        \futurelet\@let@token\hi@optclause@endwith
    }{\futurelet\@let@token\hi@optclause@endwithout}%
}
\def\hi@optclause@endwith{%
    \ifx\ \@let@token,\else
        \ifcat a\@let@token,\space \fi
    \fi
}
\def\hi@optclause@endwithout{%
    \ifx\@let@token,\unkern\else
        \ifx\@let@token.\unkern\else
            \ifcat a\@let@token\space \fi
        \fi
    \fi
}
```

---

[3]Personally, I do not like this style of citation, as it breaks the rule that a citation comes after the proposition being cited. But it is a common enough style that I provide it as an option.

6 Miscellaneous Commands

A few additional commands are defined here.

\textellipsis   Ellipses in legal documents are idiosyncratic, using full spaces between dots. The package redefines ellipses in LaTeX to conform.

```
\AtBeginDocument{%
    \def\textellipsis{%
        \ifhmode\unskip\@\hi@ellipsis@kern\fi
        .\hi@ellipsis@kern
        .\hi@ellipsis@kern
        .\hbox{\vrule \@width 0pt}% Looks better with microtype
        \futurelet\@let@token\hi@ellipsis@end
    }%
}
```

The `times` package used to give me problems—I don't know if it is still problematic.

```
\@ifpackageloaded{times}{%
    \def\hi@ellipsis@kern{\kern\fontdimen3\font}%
    \def\hi@ellipsis@space{\hskip\fontdimen3\font\relax}%
}{%
    \let\hi@ellipsis@kern~%
    \let\hi@ellipsis@space\space
}
```

At the end of an ellipsis, test if the next character is a letter, for purposes of adjusting the `\spacefactor`.

```
\def\hi@ellipsis@end{%
    \@test\ifx\@let@token\fi{%
        \futurelet\@let@token\hi@ellipsis@end
    }{%
        \@test\ifcat A\noexpand\@let@token\fi{%
            \@\hi@ellipsis@space
        }{%
            \hi@ellipsis@kern
            \spacefactor999\relax
        }%
    }%
}
```

\sic   The \sic! command inserts a properly formatted [*sic*] into text. The exclamation point deals with TeX spacing issues.

```
\def\sic!{[\emph{sic}]}
```

Several macros affect memory of citation state to change what forms of citations will be produced.

\noid
- The \noid command informs the package not to use *id.* for the next citation, by resetting memory of the last reference cited.

\fullcite
- Similarly, the \fullcite{⟨*ref*⟩} command informs the package to produce a full citation of ⟨*ref*⟩ by resetting memory of that reference's use.

\nofullcite
- By contrast, the \nofullcite{⟨*ref*⟩} effectively cites ⟨*ref*⟩ without displaying anything, so that the next citation to that reference is a short form.

\fulltitle
- The \fulltitle{⟨*ref*⟩} command resets memory of the last statutory title cited.

\addtotoa
- The \addtotoa{⟨*ref*⟩} command adds ⟨*ref*⟩ to the table of authorities without producing any citation text.

```
\def\hi@noid{%
    \let\@last@case\relax\let\@last@page\relax\let\@last@vol\relax
    \let\@last@title\relax
}
\let\noid\hi@noid
\def\fullcite#1{%
    \expandafter\let\csname dfc@#1\endcsname\relax
    \expandafter\let\csname dfi@#1\endcsname\relax
    \noid
}
\def\nofullcite#1{\hi@record@cite{#1}}
\def\fulltitle{\let\@this@title\relax}
\def\addtotoa#1#2{\@bsphack\hi@addtotoa{#1}{#2}\@esphack}
```

7 Shorter Commands

The following are synonyms to the citation commands thus described.

Command	Synonym
\sentence	\Ct
\clause	\ct
\Inline	\Ci
\inline	\ci
\note	\cn
\addtotoa	\cx