Cases (caseref.dtx)

Charles Duan

May 3, 2023

```
\def\hi@case@def#1{%
    \hi@case@setparties{#1}%
    \hi@case@updaterep
    \hi@case@history
    \hi@ifset\hi@kv@inline{%
        \@expand{\hi@short@register{#1}}\hi@kv@inline i{y}%
    }{}%
    \hi@case@proc
    \@expand{\hi@short@register{#1}}\hi@kv@inline i{n}%
    \hi@ifset\hi@kv@slip{\def\hi@kv@slip{slip op. }}{\def\hi@kv@slip{}}%
    \hi@newcite@form{fc}{#1}{%
        \hi@inline@only{\hi@citeguts{\hi@kv@name}}%
        \hi@inline@never{%
            \hi@citeguts{%
                \hi@name@only{\hi@kv@aname\hi@font@comma}%
                \hi@ifset\hi@kv@rep{%
                    \hi@kv@vol\space \hi@kv@rep\space
                    \@format@page@macro\hi@kv@page
                    \hi@maybepage{, }%
                }{%
                    \hi@kv@docket % Docket number
                    \hi@ifset{\hi@kv@dbid}{, \hi@kv@dbid}{}%
                    \hi@maybepage{, \hi@kv@slip\hi@page@atorsect}%
                }%
            }%
            \the\hi@param@parens
            \hi@case@dateparen{%
                \hi@ifeitherset\hi@kv@year\hi@kv@court{%
                    \hi@parens@add{\hi@paren@date}{%
                        \hi@param@optspc\hi@kv@court\hi@kv@year
                    }%
                }{}%
            }y%
        }%
    }%
    \hi@newcite@form{lc}{#1}{%
        \hi@citeguts{%
            \hi@name@only{\@capnext\hi@kv@lname\unkern,\hfill\break}%
            \hi@ifset\hi@kv@rep{%
                \hi@kv@vol\space\hi@kv@rep\space
                \@format@page@macro\hi@kv@page
                \hi@maybepage{, }%
            }{%
                \hi@kv@docket % Docket number
                \hi@ifset{\hi@kv@dbid}{, \hi@kv@dbid}{}%
                \hi@maybepage{, \hi@kv@slip\hi@page@atorsect}%
            }%
        }%
        \hi@case@dateparen{%
            \hi@parens@add{\hi@paren@date}{%
                \hi@param@optspc\hi@kv@court\hi@kv@year
            }%
        }y%
        \the\hi@param@parens
    }%
    \hi@newcite@form{sc}{#1}{%
        \hi@short@use{#1}%
        \hi@citeguts{%
            \hi@inline@only{\hi@citeguts{\hi@kv@inline}}%
            \hi@inline@never{%
                \hi@name@only{%
                    \hi@kv@ainline
                    \hi@font@comma
                }%
```

```
                \hi@ifset\hi@kv@rep{%
                    \hi@kv@vol\space\hi@kv@rep\space %
                    \hi@ifpage{\hi@maybepage{\hi@page@atorsect}}{%
                        \@format@page@macro\hi@kv@page
                    }%
                }{%
                    \hi@ifset{\hi@kv@dbid}{\hi@kv@dbid}{\hi@kv@docket}%
                    \hi@maybepage{, \hi@kv@slip\hi@page@atorsect}%
                }%
            }%
        }%
        \hi@case@dateparen{}n%
    }%
}
```

1 Parties and Case Titles

The name of a case is generally defined by the following input parameters:

- p: The party before the "versus," generally the plaintiff.
- d: The party after the "versus," generally the defendant.
- name: A non-versus name for the case, such as "*In re* ⟨*party*⟩." This will be called the "common name" in the discussion below.
- inline: Optionally, the short form name.

It is possible for cases to use both the versus-party parameters and a common name, for example in multidistrict litigation that both has lead parties and a topical name.

These inputs are used to produce five formatted names for the case:

- The full title, used in full inline citations.
- The abbreviated full title, used in full non-inline citations.
- The abbreviated full title to be used in the Table of Authorities. This differs in spacing (an optional line break is placed before the versus) and in text order (phrases like *In re* are placed at the end rather than the beginning).
- The short form for inline citations.
- The abbreviated short form for non-inline citations.

Initially, macros `\case@`⟨*ref*⟩`@`⟨*party*⟩ are set to match the case names given in parameters p, d, and name. These must be saved for comparison purposes against other cases (e.g., for subsequent history).

```
\def\hi@case@setparties#1{%
    \hi@ifset\hi@kv@p{%
        \global\cslet{case@#1@p}\hi@kv@p
    }{%
        \global\cslet{case@#1@p}\relax
    }%
    \hi@ifset\hi@kv@d{%
        \global\cslet{case@#1@d}\hi@kv@d
    }{%
        \global\cslet{case@#1@d}\relax
    }%
    \hi@ifset\hi@kv@name{%
        \global\cslet{case@#1@name}\hi@kv@name
```

```
    }{%
        \global\cslet{case@#1@name}\relax
    }%
}
```

This macro actually assembles the names.

```
\def\hi@case@proc{%
    \begingroup
        % For finding the best party for inline citations
        \let\hi@tmp@allnames\@empty
        % Scores will always be positive other than for impossible party names,
        % so setting this to -1 guarantees a party name will be selected
        \let\hi@tmp@bestscore\m@ne
        \hi@ifset\hi@kv@name{\@expand\hi@case@procname\hi@kv@name i}{}%
        \hi@ifset\hi@kv@p{\hi@case@procparties}{}%
        % Make the full-cite names, both inline and abbreviated
        \hi@case@proc@fullnames
        %
        % Set up the short-cite names
        \hi@case@proc@shortnames
        %
        % Now we need to get all the defined names out of the group. We do so by
        % creating |\reserved@a| to end the group and redefine all the name
        % macros.
        %
        \def\reserved@a{\endgroup}%
        \hi@ifset\hi@kv@name{\hi@redef@macro\reserved@a\hi@kv@name}{}%
        \hi@ifset\hi@kv@lname{\hi@redef@macro\reserved@a\hi@kv@lname}{}%
        \hi@ifset\hi@kv@aname{\hi@redef@macro\reserved@a\hi@kv@aname}{}%
        \hi@ifset\hi@kv@ainline{\hi@redef@macro\reserved@a\hi@kv@ainline}{}%
        \hi@ifset\hi@kv@inline{\hi@redef@macro\reserved@a\hi@kv@inline}{}%
    \reserved@a
}
\hi@undefine\hi@kv@lname
\def\hi@redef@macro#1#2{%
    \add@macro@to@macro#1{\expandafter\def\expandafter#2\expandafter{#2}}%
}
```

1(a) Party Name Syntax Party names for p and d have the form:

⟨*party*⟩ := ⟨*text*⟩ [`ex rel.` ⟨*text*⟩]

Examples:

```
Doe
State ex rel. Adams
```

The “base” name and the name after “ex rel.” are treated as two separate names, each a candidate for being the case’s short name.

Receives a party name and a baseline score for the party type. This macro (1) separates into a base name and *ex rel.* part, (2) scores both parts as short names, (3) assembles the formatted name and abbreviates it, and (4) passes the resulting names to a callback. #1 is the party name as text, #2 is the baseline score, #3 the callback that should take two arguments for the unabbreviated and abbreviated name.

```
\def\hi@case@procparty#1#2#3{%
    \hi@abbrev@inlcase{#1}{\hi@case@procparty@}{#2}{#3}%
}
\def\hi@case@procparty@#1#2#3{%
    \find@in{ ex rel. }{#1}{\hi@case@procparty@parts}{%
        \hi@case@procparty@parts{#1}{}%
    }{#2}{#3}%
}
\make@find@in{ ex rel. }
```

#1 is the base name, #2 any ex-rel part, #3 the baseline score, #4 the callback.

```
\def\hi@case@procparty@parts#1#2#3#4{%
    \hi@case@cmpname{#1}{#3+10}%
    \ifblank{#2}{%
        % No ex-rel. Just run the callback
        \hi@abbrev@name{#1}{#4{#1}}%
    }{%
```

```
        % With an ex-rel.
        \hi@case@cmpname{#2}{#3}%
        \hi@abbrev@name{#1\hi@case@exrel#2}{#4{#1\hi@case@exrel#2}}%
    }%
}
%
% Draw the word "ex rel.", with fonts.
\DeclareRobustCommand\hi@case@exrel{ \hi@fn@caseproc{ex rel.} }
```

If the case has party names, we must form the full titles and then pick the party for the short form.

```
\def\hi@case@procparties{%
    \hi@ifset\hi@kv@inlinedefendant{%
        \@expandarg\hi@case@procparty\hi@kv@p{-20000}{%
            \@twodef\hi@tmp@p\hi@tmp@pa
        }%
    }{%
        \@expandarg\hi@case@procparty\hi@kv@p{5}{\@twodef\hi@tmp@p\hi@tmp@pa}%
    }%
    \@expandarg\hi@case@procparty\hi@kv@d{3}{\@twodef\hi@tmp@d\hi@tmp@da}%
}
```

For common case names with the name parameter, there are more possibilities because prefixes *In re* and *Ex parte* can be used:

⟨*name*⟩ := [`In re` ‖ `Ex parte`] ⟨*text*⟩ [`ex rel.` ⟨*text*⟩]

Examples:

```
In re Comiskey
Ex parte New York ex rel. Jones
```

The latter example is procedurally implausible but nevertheless syntactially acceptable.

Processes a common name provided for a case. Besides setting up the standard variables, this macro also splits off initial texts like procedural phrases and the word “The.”

```
\def\hi@case@procname#1{%
    \let\hi@tmp@npre\@empty
    \@expand{\find@try\find@start{%
        {In re }{\def\hi@tmp@npre{\hi@fn@caseproc{In re} }\def\hi@kv@name}%
        {Ex parte }{%
            \def\hi@tmp@npre{\hi@fn@caseproc{Ex parte} }\def\hi@kv@name
        }%
    }}\hi@kv@name i{}%
    %
    % We use |\hi@inline@the| only if there is no other prefix. In such cases,
    % the word ``The'' seems more integrally part of the party name (In re The
    % Party), as opposed to when there is no other prefix and ``The'' is part of
    % a description of the litigation (The Slaughter-House Cases).
    %
    \ifx\hi@tmp@npre\@empty
        \hi@replacethe\hi@kv@name{\def\hi@kv@name}{}%
    \fi
    \@expandarg\hi@case@procparty\hi@kv@name{7}{\@twodef\hi@tmp@n\hi@tmp@na}%
    % The abbreviated full name always takes ``The'' if present
    \@expandarg\hi@replaceinlinethe\hi@tmp@na{The }{\def\hi@tmp@na}{}%
}
\make@find@start{In re }
\make@find@start{Ex parte }
```

1(b) Full Name Formatting There are three full names required, for inline, non-inline, and Table of Authorities citations. How each appears depends on which parameters are given:

- If party names are given alone, then the full names are “⟨*p*⟩ v. ⟨*d*⟩” appropriately abbreviated.

- If party names and a common name are given, then the full names are "⟨*p*⟩ v. ⟨*d*⟩ (⟨*name*⟩)" appropriately abbreviated.
- If only the name is given, this is the most complex case. If the name begins with "The," then the inline form uses `\hi@inline@the`, and the word "The" is placed at the end of the Table of Authorities form. If the name begins with a prefix like *In re*, then the prefix is placed at the end of the Table of Authorities form. Otherwise, the name is assembled with "The" and the prefix attached, appropriately abbreviated.

Additionally, the fonts need to be set up in assembling the full name forms. This is because the name can use multiple fonts internally, for different parts.

```
\def\hi@case@proc@fullnames{%
    \hi@ifset\hi@kv@p{\hi@case@proc@fullnames@parties}{%
        \hi@case@proc@fullnames@nameonly
    }%
}
\def\hi@case@proc@fullnames@parties{%
    %
    % Create the parenthetical if a name was given. We have to do this up here
    % because we overwrite |\hi@kv@name| later.
    %
    \hi@ifset\hi@kv@name{%
        % Unabbreviated name will always use ``The''
        \@expandarg\hi@replaceinlinethe\hi@tmp@n{The }{\def\hi@tmp@n}{}%
        \edef\hi@kv@name{ (\hi@case@namefmt\hi@tmp@n)}%
        \edef\hi@kv@aname{ (\hi@case@namefmt\hi@tmp@na)}%
    }{%
        % If no name was given, then there's a possibility we need to do dot
        % management if |\hi@tmp@d| ends with a dot.
        \edef\hi@kv@name{\hi@case@testdot\hi@tmp@d}%
        \let\hi@kv@aname\@empty
    }%
    \epreto\hi@kv@name{%
        \noexpand\hi@fn@casefc{%
            \expandonce\hi@tmp@p \noexpand\hi@versus \expandonce\hi@tmp@d
        }%
        \hi@ifset\hi@kv@name{}{\hi@case@testdot\hi@tmp@d}%
    }%
    \epreto\hi@kv@aname{%
        \noexpand\hi@fn@casefc{%
            \expandonce\hi@tmp@pa \noexpand\hi@versus \expandonce\hi@tmp@da
        }%
    }%
    \let\hi@kv@lname\hi@kv@name
}
\def\hi@case@proc@fullnames@nameonly{%
    \hi@ifset\hi@kv@name{}{%
        \PackageError\hi@pkgname{%
            Case citation requires either parties or a common name%
        }{Please use the p, d, or name parameters}%
    }%
    % The TOA name should put ``The'' and any prefix at the end, so we have to
    % search for a leading |\hi@inline@the|.
    \@expandarg\hi@replaceinlinethe\hi@tmp@n{}{%
        \hi@case@proc@toaname{The}%
    }{\hi@case@proc@toaname{}{\expandonce\hi@tmp@n}}%
    % For the full inline name, ``The'' will always be used if there is a prefix
    % like In re.
    \ifx\hi@tmp@npre\@empty\else
        \@expandarg\hi@replaceinlinethe\hi@tmp@n{The }{\def\hi@tmp@n}{}%
    \fi
    \edef\hi@kv@name{\hi@case@namefmt\hi@tmp@n \hi@case@testdot\hi@tmp@n}%
    \edef\hi@kv@aname{\hi@case@namefmt\hi@tmp@na}%
}
%
% Sets |\hi@kv@lname| when only a name is given. Primarily, the prefixes need to
% be revised and placed at the end.
% \#1 is ``The'' or empty, \#2 the rest of the name to include in the TOA name.
\def\hi@case@proc@toaname#1#2{%
    \edef\hi@kv@lname{%
        \noexpand\hi@fn@casefc{#2}%
        \@expandarg\ifblank{\hi@tmp@npre#1}{}{%
            ,\hi@case@proc@toaname@npre
```

```
            \ifblank{#1}{}{ \noexpand\hi@fn@casefc{#1}}%
        }%
    }%
}
%
% If |\hi@tmp@npre| is not empty, use it but chop off any space at the end.
%
\def\hi@case@proc@toaname@npre{%
    \ifx\hi@tmp@npre\@empty\else
        \space
        \@expand{\find@end{ }}\hi@tmp@npre i{\unexpanded}{\meaning\hi@tmp@npre}%
    \fi
}
\make@find@end{ }
```

Emits the text to be added in standard uses of a name in a case. #1 is the name macro (`\hi@tmp@n` or `\hi@tmp@na`). It is assumed that any leading "the" has already been taken care of before here.

This macro is completely expandable and will produce text amenable to an `\edef`.

```
\def\hi@case@namefmt#1{%
    \expandonce\hi@tmp@npre \noexpand\hi@fn@casefc{\expandonce#1}%
}
```

If the text of macro #1 ends with a dot, then appends `\@hi@dottrue` to the end of the text of #2. This can be placed inside an `\edef`.

```
\def\hi@case@testdot#1{%
    \@expandarg\if@endswithdot{#1}{\unexpanded{\protect\@hi@dottrue}}%
}
```

Appearance of the "v." in party names.

```
\DeclareRobustCommand\hi@versus{%
    \if@hi@in@toa
        \ifnum\hi@citelevel=\@ne
            \hfill\penalty\m@ne\space\hskip\z@\@plus-1fill\relax
        \else
            ~%
        \fi
    \else
        ~%
    \fi
    v.\@\space
}
```

1(c) Selecting the Short Name In theory, there are up to six possible candidates for the short name of a case citation: the two parties, the common name, and the *ex rel.* components of each of those. To decide which of them is the best, this package employs a scoring system applied to each candidate name. The following factors are considered:

- Names that are impermissible party names receive the lowest scores; they will be used only if there is no other option.
- Names with capitalized first words or the word "of" are less likely to be good short names so they receive lower scores.
- *Ex rel.* names are disfavored over base party names to the extent that none of the above factors apply.
- Common names are preferred over topside parties, which are preferred over bottomside parties, to the extent that none of the above factors apply.

Compares the name against the reigning best name, stored in `\hi@tmp@bestname`, and the best score in `\hi@tmp@bestscore`. If the provided name (#1) has a better score, then it replaces the macros for best name. #2 is a baseline score based on the nature of the name being compared (e.g., topside parties are preferable to bottomside parties).

```
\def\hi@case@cmpname#1#2{%
    \appto\hi@tmp@allnames{#1;}%
    \hi@case@scorename{#1}{\hi@case@cmpname@{#1}{#2}}%
}
\def\hi@case@cmpname@#1#2#3{%
    \ifnumcomp{#2+#3}>{\hi@tmp@bestscore}{%
        \def\hi@tmp@bestname{#1}%
        \def\hi@tmp@bestscore{#2+#3}%
    }{}%
}
```

Scores a name #1, running #2 as a callback.

```
\def\hi@case@scorename#1#2{%
    % Empty name
    \ifblank{#1}{#2{-10000}}{%
        % Name in the nocite list
        \ifcsdef{nocite@\detokenize{#1}}{#2{1000}}{%
            % Capitalized first word
            \hi@iffirstwordcaps{#1}{#2{2000}}{%
            % Name with the word ``of''
                \find@in{ of }{#1}{\@firstofthree{#2{3000}}}{%
                    #2{4000}%
                }%
            }%
        }%
    }%
}
\make@find@in{ of }
```

Test if the first word of the name is all caps.

```
\def\hi@iffirstwordcaps#1{%
    \find@in{ }{#1}{\hi@iffirstwordcaps@}{\hi@iffirstwordcaps@{#1}{}}%
}
\make@find@in{ }
\def\hi@iffirstwordcaps@#1#2{%
    \uppercase{\def\reserved@a{#1}}%
    \ifdefstring\reserved@a{#1}%
}
```

1(d) Formatting the Short Name Short names may be user-provided or automatically determined based on the selection algorithm above. To format the short name, several tasks must be performed:

- For user-given short names, the word “the” at the beginning needs to be processed for `\hi@inline@the`.
- For automatically determined short names, some conventional words are removed from the name as described below.
- The short name must have appropriate fonts set.
- If the short name is not obviously connected to the full name, then the full name inline name requires a parenthetical, and the full citation name (a) requires a parenthetical the first time it is ever used and (2) should actually use the short name on subsequent uses, as described in `state.dtx`.

```
\def\hi@case@proc@shortnames{%
    % If the user gave an inline name, deal with ``the'' at the beginning.
    % Otherwise, it is assumed that leading ``the'' was dealt with, so apply the
    % short name simplification algorithm.
    \hi@ifset\hi@kv@inline{\hi@replacethe\hi@kv@inline{\def\hi@kv@inline}}{%
```

```
        \@expandarg\hi@case@shortfmt\hi@tmp@bestname{\def\hi@kv@inline}%
    }%
    % Abbreviate the short name
    \@expandarg\hi@abbrev@name\hi@kv@inline{\def\hi@kv@ainline}%
    % Determine if the inline name chosen requires a parenthetical
    \edef\reserved@a{%
        \noexpand\in@{\expandafter\strip@prefix\meaning\hi@kv@inline}%
            {\expandafter\strip@prefix\meaning\hi@tmp@allnames}%
    }\reserved@a
    \hi@ifset\hi@kv@inlineparen{\in@false}{}%
    % Add formatting to the names. Leading ``the'' should have already been
    % converted to |\hi@inline@the|.
    \edef\hi@kv@inline{%
        \noexpand\hi@fn@casesc{\expandonce\hi@kv@inline}%
        \hi@case@testdot\hi@kv@inline
    }
    \edef\hi@kv@ainline{%
        \noexpand\hi@fn@casesc{\expandonce\hi@kv@ainline}%
    }
    % Add the parenthetical to the full names if needed. This must be done down
    % here, because the fonts for the short names need to be set first.
    \ifin@\else
        \eappto\hi@kv@name{%
            \noexpand\hi@short@maybeshow{\@this@case}{%
                \space(``\expandonce\hi@kv@inline'')%
                \unexpanded{\protect\@hi@dotfalse}%
            }%
        }%
        \edef\hi@kv@aname{%
            \noexpand\hi@case@choosefullorainline{%
                \expandonce\hi@kv@aname
            }{\expandonce\hi@kv@ainline}%
        }%
    \fi
}
```

Chooses the form of the full citation title for cases, where there is a short form. If it is the first full citation of the case ever, then it will be ”Full Case Name (“Short Name”).” If it is a repeat full citation, then it will be ”Short Name.”

```
\DeclareRobustCommand\hi@case@choosefullorainline[2]{%
    \hi@record@firstfullcite\@this@case{%
        #1\hi@short@maybeshow{\@this@case}{%
            \space(``#2'')\protect\@hi@dotfalse
        }%
    }{%
        #2%
    }%
}
```

Regarding the removal of conventional words, these are typically business or geographical designations that are uninformative for the short name. A list of ignored words follows:

Short case name adjustments: generally we want to drop common phrases from the ends of short case names. #1 is the name to abbreviate, #2 a callback.

The way this macro works is somewhat obscure. `\hi@case@shortfmt@words` expands to a `\find@try\find@end` list pairing truncatable end-words with `\hi@case@shortfmt`. The callback #2 is placed at the end. If a match is found, then `\find@end` executes `\hi@case@shortfmt`, which picks up as its arguments the string prefix (passed by `\find@end`) and the callback (left on the stack). As a result, `\hi@case@shortfmt` is recursively called with the correct arguments. If no match is found, then `\hi@case@shortfmt@run` is executed, which again picks up the correct arguments.

```
\def\hi@case@shortfmt#1#2{%
    \@expand{\find@try\find@end}\hi@case@shortfmt@words i{#1}{%
        \hi@case@shortfmt@run{#1}%
    }{#2}%
}
\def\hi@case@shortfmt@run#1#2{#2{#1}}
\def\hi@case@shortfmt@words{}
\def\hi@case@shortfmt@add#1{%
    \addto@macro\hi@case@shortfmt@words{%
        {#1}{\hi@case@shortfmt}%
    }%
    \make@find@end{#1}%
}
```

Note that `\hi@case@shortfmt` is run on the unabbreviated short name. However, just in case the user enters a word abbreviated to force it to be abbreviated in the inline context, abbreviations are included below to ensure their removal.

```
 ,
Inc.
Co.
Corp.
LP
LLP
LLLP
P'ship
Partnership
LLC
Ltd.
America
Am.
USA
International
Int'l
of
```

2 Nominative Reporters

Generally, reporter names are not abbreviated or changed from what is given as input to the parameter rep. This is because reporters are conventionally known by their abbreviations, and there is no consistency in reporter abbreviations. However, some common reporters have both an official series name and nominative reporter names. Volume 21 of the *United States Reports*, for example, is also volume 8 of Wheaton's reports, and typically both are cited:

```
21 U.S. (8 Wheat.) 1
```

In most cases, the pagination of the series reporter and the nominative reporter is the same, and adding the nominative reporter is a simple matter of looking up the corresponding name in a table. As a result, this package handles these nominative reports automatically. For nominative reporters with different pagination, use the parallel parameter.

```
%
% Adds a parenthetical nominative reporter, based on a table.
%
\def\hi@case@updaterep{%
    \hi@ifset\hi@kv@vol{%
        \hi@ifset\hi@kv@rep{%
            \@ifundefined{hi@trep@\hi@kv@vol @\hi@kv@rep}{}{%
                \protected@edef\hi@kv@rep{%
                    \hi@kv@rep\space(%
                        \csname hi@trep@\hi@kv@vol @\hi@kv@rep\endcsname
                    )%
```

```
                }%
            }%
        }{}%
    }{}%
}
\def\hi@abbrev#1#2#3{%
    \@namedef{hi@trep@#1@#2}{#3}%
}
\input hireps
```

3 Date Parenthetical

Ordinarily the date parenthetical for cases contains the court and year parameters. A complication arises when the case is being cited as part of a casedoc reference. In that case two things need to happen:

- If this case has a reporter listed (rep is set), then a parenthetical for the docket number must be added.
- If the case document has its own date set, then that date overrides the date given for the case.[1]

#1 is the ordinary content for the date parenthetical. #2 is y or n depending on whether the court should be included in casedoc citations.

```
\def\hi@case@dateparen#1#2{%
    \noexpand\ifx\noexpand\hi@casedoc@docket\relax
        #1%
    \noexpand\else
        \hi@ifset\hi@kv@rep{%
            \hi@parens@add{\hi@paren@casedocdocket}{%
                \hi@ifset\hi@kv@docket{\hi@kv@docket}{%
                    \noexpand\hi@casedoc@docket
                }%
            }%
        }{}%
        \noexpand\ifx\noexpand\hi@casedoc@date\relax\noexpand\else
            \hi@parens@add{\hi@paren@date}{%
                \ifx#2y%
                    \hi@ifset\hi@kv@court{\hi@kv@court\space}{}%
                \fi
                \noexpand\hi@casedoc@date
            }%
        \noexpand\fi
    \noexpand\fi
}
```

4 Procedural History

Procedural history for a case is given in the prior and subsequent parameters, which are used to construct parentheticals. The syntax for those parameters is:

⟨*history*⟩ := ⟨*phrase*⟩ **:** ⟨*ref*⟩

Example:

```
reversed: prior-case
```

[1] This does not conform to some legal citation practices, which attach the date of the case decision rather than the date of the document. That practice seems questionable since the date of interest to the reader is more likely the case document's date. Nevertheless, it is simple enough to assign the case decision's date to the case document when defining the case document reference.

The ⟨*phrase*⟩ is an explanatory phrase that will be abbreviated and also checked against a table of valid explanatory phrases. If a desired phrase is not already listed, the macro `\ExplanatoryPhrase{`⟨*phrase[,]*⟩`}` is used. Note that the comma at the end of the argument is significant, as some explanatory phrases use a comma and others do not.

The main task to be accomplished in producing case history is reconciliation of names. If the case being defined and the case in its history parenthetical have the same-named parties, then the name can be omitted from the case in the history parenthetical. If they have different names, then the phrase "sub nom." must be added to explanatory phrases that end with a comma.

4(a) Input for Case History

Parses case history and creates appropriate parentheticals.

```
\def\hi@case@history{%
    \hi@ifset\hi@kv@prior{%
        \@expandarg\hi@expl@parse\hi@kv@prior{}\hi@paren@prior
    }\relax
    \hi@ifset\hi@kv@subsequent{%
        \@expandarg\hi@expl@parse
            \hi@kv@subsequent
            { \hi@fn@latin{sub nom.}}%
            \hi@paren@subseq
    }\relax
}
```

History for things that are not cases.

```
\def\hi@cite@history{%
    \hi@ifset\hi@kv@prior{%
        \@expand\hi@expl@parse\hi@kv@prior{}\hi@paren@prior
    }\relax
    \hi@ifset\hi@kv@subsequent{%
        \@expand\hi@expl@parse\hi@kv@subsequent{}\hi@paren@subseq
    }\relax
}
```

Parses the explanatory phrase and citation reference syntax. This macro splits the phrase and the citation by colon, comma, or space, and passes to `\hi@expl@abbrev` to abbreviate the phrase and create the parenthetical. #1 is the text to be parsed. #2 is "sub nom." if that is to be used. #3 is the parenthetical class.

```
\def\hi@expl@parse#1#2#3{%
    \find@in{: }{#1}{\hi@expl@abbrev{}}{%
        \find@last{, }{#1}{\hi@expl@abbrev{,}}{%
            \find@last{ }{#1}{\hi@expl@abbrev{}}{%
                \hi@expl@err
                \hi@expl@abbrev{}{}{#1}% Set procedural phrase to nothing
            }%
        }%
    }{#2}{#3}% Parameters passed to |\hi@expl@abbrev| in all cases
}
\make@find@in{: }
\make@find@in{, }
\make@find@in{ }
```

Abbreviates the procedural phrase and passes control go `\hi@expl@addparen` to add the parenthetical. #1 trailing matter for the phrase, #2 the phrase, #3 the citation, #4 the sub-nom to be used, and #5 the parenthetical class.

```
\def\hi@expl@abbrev#1#2#3#4#5{%
    \hi@abbrev@expl{#2#1}{\hi@expl@addparen}{#3}{#4}{#5}%
}
```

Adds the parenthetical for a procedural history item. #1 is the explanatory phrase, #2 the reference name, #3 the "sub nom.", and #4 the parenthetical class. The parenthetical essentially just contains the above parameters and calls `\hi@expl` for formatting, except that no parenthetical is added in the Table of Authorities and to subsequent full cites.

```
\def\hi@expl@addparen#1#2#3#4{%
    \push@toks\hi@param@parens{%
        \if@hi@in@toa\else
```

```
            \hi@record@firstfullcite\@this@case{%
                \hi@parens@add#4{\hi@expl{#1}{#2}{#3}}%
            }{}%
        \fi
    }%
}
```

Error.

```
\def\hi@expl@err{%
    \PackageError\hi@pkgname{Explanatory phrase in history missing}{%
        You must include explanatory history phrase and citation.
    }%
}
```

4(b) Displaying Case History Shows an procedural phrase parenthetical. #1 is the phrase, #2 the reference name, #3 any sub-nom text. Two choices must be made: (1) whether to display the sub-nom text, and (2) whether to suppress the second case's name. The macro `\hi@expl@determinename` determines whether the two cases have the same name, and the outcomes are as follows:

Condition	Sub nom	Use name
Rehearing/certiorari denied phrase	No	No
No sub nom text	No	If different
No comma at end of phrase	No	If different
Both comma and sub nom, same name	No	No
Both comma and sub nom, different name	Yes	Yes

```
\def\hi@expl#1#2#3{%
    \find@end{,}{#1}{\hi@expl@{,}}{\hi@expl@{}{#1}}{#2}{#3}%
}
\make@find@end{,}
% \#1 is the phrase's comma, \#2 the phrase with no comma, \#3 the reference
% name, \#4 any sub nom text.
\def\hi@expl@#1#2#3#4{%
    \ifcsdef{hi@expl@noname@#2}{%
        \hi@fn@sig{#2}#1\space
        \def\@last@inline{#3}%
    }{%
        \ifstrempty{#4}{% No sub-nom text
            \hi@fn@sig{#2}#1\space
            \hi@expl@determinename{#3}{\def\@last@inline{#3}}{}%
        }{%
            \ifstrempty{#1}{% No comma
                \hi@fn@sig{#2}#1\space
                \hi@expl@determinename{#3}{\def\@last@inline{#3}}{}%
            }{%
                \hi@expl@determinename{#3}{%
                    \hi@fn@sig{#2}#1\space
                    \def\@last@inline{#3}%
                }{%
                    \hi@fn@sig{#2}#4\space
                }%
            }%
        }%
    }%
    \clause{#3}%
}
\@namedef{hi@expl@noname@cert. denied}{}
\@namedef{hi@expl@noname@and cert. denied}{}
\@namedef{hi@expl@noname@reh'g denied}{}
\@namedef{hi@expl@noname@and reh'g denied}{}
```

Tests whether the given reference has the same case name as the current `\@this@case` such that no sub-nom should be used. #1 is the reference name to test, #2 the callback if the references are considered to have the same name, and #3 the callback if the references have different names.

```
\def\hi@expl@determinename#1#2#3{%
    \edef\reserved@a{%
        \noexpand\hi@expl@determinename@
            \expandonce{\csname case@#1@p\endcsname}%
            \expandonce{\csname case@#1@d\endcsname}%
            \expandonce{\csname case@#1@name\endcsname}%
            \expandonce{\csname case@\@this@case @p\endcsname}%
            \expandonce{\csname case@\@this@case @d\endcsname}%
```

```
            \expandonce{\csname case@\@this@case @name\endcsname}%
        }\reserved@a
        \@test\if@hi@name\fi{#3}{\@hi@nametrue#2}%
    }
```

Sets `\if@hi@name` to false if there is a name match, and true if no match. In other words, `\if@hi@name` indicates whether the reference name ought to be included.

```
    \def\hi@expl@determinename@#1#2#3#4#5#6{%
        \@hi@nametrue
        \ifx#1\relax\else % Don't do this if it's not a case
            \ifx#1#4\ifx#2#5\@hi@namefalse\fi\fi % (i)
            \ifx#1#5\ifx#2#4\@hi@namefalse\fi\fi
            \hi@ifset#1{}{% (iii)
                \ifx#3#4\@hi@namefalse\fi
                \ifx#3#5\@hi@namefalse\fi
                \ifx#3#6\@hi@namefalse\fi
            }%
            \hi@ifset#4{}{%
                \ifx#6#1\@hi@namefalse\fi
                \ifx#6#2\@hi@namefalse\fi
                \ifx#6#3\@hi@namefalse\fi
            }%
        \fi
    }
```

case: A case citation

Key Parameters:

p, **d**: Parties named in the case.

name: A common or single name for the case. Required unless parties are given.

vol, **rep**, **page**, **cite**: Citation to the reporter containing the case.

docket: The docket number of the case. Required if cite parameters are not given, or may optionally be given if casedoc references will use this case.

court: The court deciding the case, which may be omitted if the reporter uniquely identifies the court.

year: The date of decision. For citing cases not yet decided, a qualifier word should be included to explain the procedural posture.

inline: A user-defined short name for this case.

Optional Parameters:

inlinedefendant: Don't use the first party as the short name.

dbid: A database identifier for unreported cases. This is optional since the docket number and court are sufficient to identify an unreported case.

slip: Whether this is a slip opinion; this adds a note to the page numbers.

prior, **subsequent**: Procedural history of the case, see `caseref.dtx`.

enbanc, **mem**, **percuriam**: Parentheticals denoting the nature of the opinion.

admincase: An administrative decision

This is the same as a case with citetype set to `admin`. Parameters are identical to case with the following addition:

Key Parameters:

agency: Same as court but abbreviates the name.

casedoc: Document used in a case

Key Parameters:

name: The name of the document.

citation: The reference name of the case in which this document was filed.

docket: The docket number of the case, if it was not already given in the case.

year: The date of the document itself.

number: The document number or docket list entry. This is added as a parenthetical, preceded by "Doc. No."

```
\let\hi@casedoc@docket\relax
\let\hi@casedoc@date\relax
```

courtdoc: A document in the docket of current litigation

It is intended for use in a legal brief or filing, to cite to other documents in the docket of the same case in which the document is being filed. For example, a brief in opposition to a motion may use this reference type to cite arguments in the motion papers. To cite briefs or documents filed in another case, use casedoc.

This package does not use parentheses around citations to court documents. This is because it makes the citation form incompatible with other citations (e.g., if one cites a court document and a statute in one string cite). Furthermore, most court documents filed today do not appear to use parentheses, and courts do not appear to require them. If they are desired, commands such as `\sentence` or `\clause` can be surrounded with parentheses.

In most appellate work and particularly before the Supreme Court, this reference type is rarely used. The more common practice now is to name the document to be cited in text and then include "(at ⟨*page*⟩)" within the sentence:

> Petitioner's brief argues (at 8) that the sky is red, but that is wrong.

There is not really a need for automation of these kinds of citations.

Key Parameters:

name, **inline**: Full and short names for the document. They will be abbreviated according to the CDOC scheme.

number: The document number of the document being cited, if one is present. The phrase "Doc. No." will be prepended to it, and the number will be made a parenthetical to the citation. If inline is not given, then the document number will be used as the short name. (Otherwise the short form will be the same as the full name.)

year: The date the document was filed.