The \texttt{ltxcmds} package

Heiko Oberdiek*

2020-05-10 v1.25

Abstract

The package \texttt{ltxcmds} exports some utility macros from the L\LaTeX\ kernel into a separate namespace and also provides them for other formats such as plain-T\TeX.

Contents

1 Documentation 3
  1.1 Introduction ........................................ 3
  1.2 Numbers ................................................. 3
  1.3 Scratch registers ........................................ 3
  1.4 Argument killers ......................................... 3
  1.5 Argument grabbers ....................................... 4
  1.6 List helpers ............................................. 5
  1.7 Tail recursion .......................................... 5
  1.8 Empty macro ............................................ 6
  1.9 Characters ............................................... 6
  1.10 Boolean switch ......................................... 6
  1.11 Command definitions .................................... 6
  1.12 Stripping ............................................... 7
  1.13 File management ........................................ 7
    1.13.1 File extensions ................................... 7
    1.13.2 Load check .......................................... 7
    1.13.3 Version date check .................................. 8
  1.14 Macro additions ......................................... 8
  1.15 Next character detection ................................. 8
  1.16 \texttt{\textbackslash ltx@leavevmode, \textbackslash ltx@mbox} .............. 9
  1.17 Expandable test for emptiness ............................ 9
  1.18 Stripping spaces ....................................... 9
  1.19 Check for emptiness of boxes ............................ 10

2 Implementation 10
  2.1 Identification ........................................... 10
  2.2 Numbers .................................................. 12
  2.3 Scratch registers ....................................... 12
  2.4 Argument killers ....................................... 14
  2.5 Argument grabbers ..................................... 15
  2.6 List helpers ............................................. 15
  2.7 Tail recursion .......................................... 17

*Please report any issues at \url{https://github.com/ho-tex/ltxcmds/issues}
1 Documentation

1.1 Introduction

Many of my packages also support other formats such as plain-\TeX. Because I am rather familiar with the utility macros from LaTeX's kernel (e.g. \@gobble, \@firstoftwo), I found myself rewriting them again and again, because they are lacking in plain-\TeX.

Therefore this package provides often used macros and similar ones with the name prefix \ltx@. This avoids also faulty redefinitions. I remember an example where a package redefined \@firstoftwo with forgetting \long.

1.2 Numbers

\begin{center}
\begin{tabular}{ll}
\verb|\ltx@zero| & \rightarrow 0 \\
\verb|\ltx@one| & \rightarrow 1 \\
\verb|\ltx@two| & \rightarrow 2 \\
\verb|\ltx@cclv| & \rightarrow 255 \\
\verb|\ltx@minusone| & \rightarrow -1
\end{tabular}
\end{center}

These commands are numbers 0, 1, 2, 255 and -1. They are not digits and a space is not gobbled afterwards. Macro \ltx@minusone is available since version 2010/12/12 v1.15.

1.3 Scratch registers

Following the conventions of plain \TeX and \LaTeX the first ten registers are free to use. Even numbered registers are for local, odd numbered for global use.

\begin{center}
\verb|\ltx@(Loc,Glob)(Toks,Dimen,Skip)(A,B,C,D,E)|
\end{center}

The name consists of the prefix \ltx@, then Loc or Glob for local or global usage follows. The register type is given by Toks for token register, Dimen for dimen register and Skip for skip register. As last part the registers are numbered from A to E. Example: \ltx@LocToksA.

Since 2011/04/14 v1.19.
1.4 Argument killers

\ltx@gobble \{1\} \rightarrow
\ltx@gobbletwo \{1\} \{2\} \rightarrow
\ltx@gobblethree \{1\} \{2\} \{3\} \rightarrow
\ltx@gobblefour \{1\} \{2\} \{3\} \{4\} \rightarrow

\ltx@GobbleNum \{\langle num \rangle \} \{\langle 1 \rangle \} \{\langle 2 \rangle \} \ldots \{\langle \langle num \rangle \rangle \} \rightarrow

The first argument \langle num \rangle of macro \ltx@GobbleNum specifies, how many following arguments are eaten. Macro \ltx@GobbleNum is expandable in exact two expansion steps.

1.5 Argument grabbers

\ltx@firstofone \{1\} \rightarrow \langle 1 \rangle
\ltx@firstoftwo \{1\} \{2\} \rightarrow \langle 1 \rangle
\ltx@secondoftwo \{1\} \{2\} \rightarrow \langle 2 \rangle
\ltx@firstofthree \{1\} \{2\} \{3\} \rightarrow \langle 1 \rangle
\ltx@secondofthree \{1\} \{2\} \{3\} \rightarrow \langle 2 \rangle
\ltx@thirdofthree \{1\} \{2\} \{3\} \rightarrow \langle 3 \rangle
\ltx@firstoffour \{1\} \{2\} \{3\} \{4\} \rightarrow \langle 1 \rangle
\ltx@secondoffour \{1\} \{2\} \{3\} \{4\} \rightarrow \langle 2 \rangle
\ltx@thirdoffour \{1\} \{2\} \{3\} \{4\} \rightarrow \langle 3 \rangle
\ltx@fourthoffour \{1\} \{2\} \{3\} \{4\} \rightarrow \langle 4 \rangle

Macros \ltx@firstofthree, \ltx@secondofthree and \ltx@thirdofthree were added in version 2010/11/12 v1.11. Macros \ltx@firstoffour, \ldots, \ltx@fourthoffour were added in version 2011/02/04 v1.16.
1.6 List helpers

\ltx@car \{⟨1⟩\} \ldots \@nil \quad \rightarrow \quad ⟨1⟩
\ltx@cdr \{⟨1⟩\} \ldots \@nil \quad \rightarrow \quad ⟨1⟩

\ltx@cartwo \{⟨1⟩\} \{⟨2⟩\} \ldots \@nil \quad \rightarrow \quad ⟨1⟩⟨2⟩
\ltx@carsecond \{⟨1⟩\} \{⟨2⟩\} \ldots \@nil \quad \rightarrow \quad ⟨2⟩
\ltx@cdrtwo \{⟨1⟩\} \{⟨2⟩\} \ldots \@nil \quad \rightarrow \quad ...

\ltx@carthree \{⟨1⟩\} \{⟨2⟩\} \{⟨3⟩\} \ldots \@nil \quad \rightarrow \quad ⟨1⟩⟨2⟩⟨3⟩
\ltx@carthird \{⟨1⟩\} \{⟨2⟩\} \{⟨3⟩\} \ldots \@nil \quad \rightarrow \quad ⟨3⟩
\ltx@cdrthree \{⟨1⟩\} \{⟨2⟩\} \{⟨3⟩\} \ldots \@nil \quad \rightarrow \quad ...

\ltx@carfour \{⟨1⟩\} \{⟨2⟩\} \{⟨3⟩\} \{⟨4⟩\} \ldots \@nil \quad \rightarrow \quad ⟨1⟩⟨2⟩⟨3⟩⟨4⟩
\ltx@carfourth \{⟨1⟩\} \{⟨2⟩\} \{⟨3⟩\} \{⟨4⟩\} \ldots \@nil \quad \rightarrow \quad ⟨4⟩
\ltx@cdrfour \{⟨1⟩\} \{⟨2⟩\} \{⟨3⟩\} \{⟨4⟩\} \ldots \@nil \quad \rightarrow \quad ...

\ltx@CarNum \{⟨num⟩\} \{⟨1⟩\} \ldots \{⟨((num))⟩\} \{⟨((num)+1))⟩\} \ldots \@nil
\quad \rightarrow \quad \{⟨1⟩\} \ldots \{⟨((num))⟩\} \ldots
\ltx@CarNumth \{⟨num⟩\} \{⟨1⟩\} \ldots \{⟨((num))⟩\} \{⟨((num)+1))⟩\} \ldots \@nil
\quad \rightarrow \quad \{⟨((num))⟩\} \ldots
\ltx@CdrNum \{⟨num⟩\} \{⟨1⟩\} \ldots \{⟨((num))⟩\} \{⟨((num)+1))⟩\} \ldots \@nil
\quad \rightarrow \quad \{⟨((num)+1))⟩\} \ldots

Macros with uppercase letters are expandable in two expansion steps. Changes in version 2020-05-10 v1.25:

- Macros \ltx@carsecond, \ltx@carthird, \ltx@carfourth, \ltx@CarNumth added.

- Macros \ltx@cdr, \ltx@cdrtwo, \ltx@cdrthree, \ltx@cdrfour, \ltx@CdrNum are expandable in two expansion steps and retain spaces and braces after the first gobbled arguments.

1.7 Tail recursion

\ltx@ReturnAfterFi \{⟨1⟩\} \fi \quad \rightarrow \quad \fi \(⟨1⟩\)
\ltx@ReturnAfterElseFi \{⟨1⟩\} \else \{⟨2⟩\} \fi \quad \rightarrow \quad \fi \(⟨1⟩\)
1.8 Empty macro

\ltx@empty \rightarrow

1.9 Characters

| \ltx@space          | → \space  |
| \ltx@percentchar   | → %       |
| \ltx@backslashchar | → \      |
| \ltx@hashchar      | → #       (since v1.7) |
| \ltx@leftbracechar | → \{     (since v1.8) |
| \ltx@rightbracechar| → \}     (since v1.8) |

1.10 Boolean switch

\ltx@newif \{⟨cmd⟩\}

\ltx@newif defines a new boolean switch ⟨cmd⟩ like \newif. Unlike plain \TeX’s \newif, \ltx@newif is not \outer. The command ⟨cmd⟩ must start with the two characters if.

\ltx@newglobalif \{⟨cmd⟩\}

\ltx@newglobalif defines a new boolean switch ⟨cmd⟩ like \ltx@newif. However the switch setting commands, ⟨cmd⟩ without the prefix if and followed by true or false are acting globally.

1.11 Command definitions

\ltx@ifundefined \{⟨cmd⟩\} \{⟨yes⟩\} \{⟨no⟩\}

If ε-\TeX{} is available, \ifcsname is used that does not have the side effect of defining undefined commands with meaning of \relax. This command is always expandable. Change in version 1.1: Also the meaning \relax is always considered “undefined”.

\ltx@IfUndefined \{⟨cmd⟩\} \{⟨yes⟩\} \{⟨no⟩\}

If ε-\TeX{} is available, \ifcsname is used that does not have the side effect of defining undefined commands with meaning of \relax. Also it always checks for the meaning of \relax and considers this as undefined. This macro is not expandable without ε-\TeX{}.

\ltx@LocalExpandAfter

It expands the token after the next token but in a local context. That is the difference to \expandafter. The local context discards the side effect of \csname and let the command undefined after the expansion step.
1.12 Stripping

\texttt{\LaTeX@RemovePrefix} \hfill \texttt{\LaTeX@StripPrefix}

All tokens up to and including the next available character ‘>’ are thrown away.
Usually it is used to strip the first part of the output of the commands \texttt{\meaning} or \texttt{\pdflastmatch}. Macro \texttt{\LaTeX@RemovePrefix} has the same meaning as \LaTeX’s \texttt{\strip@prefix}, whereas macro \texttt{\LaTeX@StripPrefix} expands the next token once before stripping the prefix.

\texttt{\LaTeX@onelevel@sanitize \{\langle macro\rangle\}}

Macro \texttt{\LaTeX@onelevel@sanitize} provides \LaTeX’s \texttt{\@onelevel@sanitize}. The macro is expanded once and the contents is converted to characters with catcode 12 (other) and space tokens with catcode 10 (space). Then then sanitized contents is stored into the macro again. Since version 1.12.

1.13 File management

All macros in this section are expandable like the counterparts of the \LaTeX{} kernel. Also they can be used after the preamble.

1.13.1 File extensions

\texttt{\LaTeX@clsextension} \hfill \texttt{\LaTeX@pkgextension}

Macros \texttt{\LaTeX@clsextension} and \texttt{\LaTeX@pkgextension} stores the strings \texttt{cls} and \texttt{sty}. In opposite to \LaTeX{}’s \texttt{\@clsextension} and \texttt{\@styextension} they can also be used after \texttt{\begin{document}}.

1.13.2 Load check

\texttt{\LaTeX@ifclassloaded \{\langle class\rangle\} \{\langle yes\rangle\} \{\langle no\rangle\}} \hfill \texttt{\LaTeX@ifpackageloaded \{\langle package\rangle\} \{\langle yes\rangle\} \{\langle no\rangle\}}

Macros \texttt{\LaTeX@ifclassloaded/\LaTeX@ifpackageloader} execute \langle yes\rangle, if the \langle class\rangle or \langle package\rangle is loaded, otherwise \langle no\rangle is called. Both \langle class\rangle and \langle package\rangle are specified without extension. The macros can also be used after \texttt{\begin{document}}.

\texttt{\LaTeX@iffileloaded \{\langle file\rangle\} \{\langle yes\rangle\} \{\langle no\rangle\}}

If \LaTeX{}’s \texttt{\ProvidesFile} macro was called before using \langle file\rangle as argument, then \texttt{\LaTeX@iffileloaded} calls \langle yes\rangle, otherwise \langle no\rangle. Therefore it is possible that the \langle file\rangle is loaded, but \langle no\rangle is executed because of a missing \texttt{\ProvidesFile}. The \LaTeX{} kernel does not have a counterpart of \texttt{\LaTeX@iffileloaded}.

Note that the file name used in \texttt{\ProvidesFile} and \texttt{\LaTeX@iffileloaded} must match. For example, if \TeX{}’s default extension .\texttt{tex} was given in the first command, then it must also specified in the latter command and vice versa.
1.13.3 Version date check

\ltx@ifclasslater {{\texttt{⟨class⟩}}} {{\texttt{⟨date⟩}}} {{\texttt{⟨yes⟩}}} {{\texttt{⟨no⟩}}} \\ltx@ifpackagelater {{\texttt{⟨package⟩}}} {{\texttt{⟨date⟩}}} {{\texttt{⟨yes⟩}}} {{\texttt{⟨no⟩}}} \\ltx@iffilelater {{\texttt{⟨file⟩}}} {{\texttt{⟨date⟩}}} {{\texttt{⟨yes⟩}}} {{\texttt{⟨no⟩}}}

If a \texttt{ProvidesClass}/\texttt{ProvidesPackage}/\texttt{ProvidesFile} command with exact the same class/package/file was executed before with an optional argument that starts with a \LaTeX{} version date, then this version date is compared with the argument \texttt{⟨date⟩}. If they are equal or if the version date is the later date, then \texttt{⟨yes⟩} is called. In all other cases \texttt{⟨no⟩} is executed.

A \LaTeX{} date has the format \texttt{YYYY/MM/DD} with \texttt{YYYY} as year with four digits, \texttt{MM} as month with two digits and \texttt{DD} as day with two digits. If pdf\TeX{}'s \texttt{\pdfmatch} is available, then it is used to detect the version date, to reject invalid date formats and to reject some invalid dates. Dates before \texttt{1994/01/01} are always invalid, because version dates are introduced with \LaTeX{} 2ε in 1994.

1.14 Macro additions

\ltx@GlobalAppendToMacro {{\texttt{⟨addition⟩}}} \\ltx@LocalAppendToMacro {{\texttt{⟨addition⟩}}}

The \texttt{⟨addition⟩} is appended to the parameterless macro \texttt{⟨cmd⟩}. If \texttt{⟨cmd⟩} is undefined or has the meaning \texttt{\relax}, then it will be initialized as empty macro beforehand. Due to a bug \texttt{⟨addition⟩} must not contain \texttt{\par} before version 2010/10/25 v1.9.

\ltx@GlobalPrependToMacro {{\texttt{⟨addition⟩}}} \\ltx@LocalPrependToMacro {{\texttt{⟨addition⟩}}}

The \texttt{⟨addition⟩} is prepended to the parameterless macro \texttt{⟨cmd⟩}. If \texttt{⟨cmd⟩} is undefined or has the meaning \texttt{\relax}, then it will be initialized as empty macro beforehand. The macros were added in version 2011/08/22 v1.21.

1.15 Next character detection

\ltx@ifnextchar {{\texttt{⟨char⟩}}} {{\texttt{⟨yes⟩}}} {{\texttt{⟨no⟩}}}

If next character is \texttt{⟨char⟩} then \texttt{⟨yes⟩} is called, otherwise \texttt{⟨no⟩}. The character is not removed. Spaces are silently removed when looking for \texttt{⟨char⟩} as \LaTeX{}'s version \texttt{\kernel@ifnextchar} does. But there are also small differences:

- The space can be used as \texttt{⟨char⟩}. In this case optional spaces before \texttt{⟨char⟩} are not supported of course.

- If the optional space is a command that is a character (defined by \texttt{\let} or \texttt{\futurelet}), then \texttt{\kernel@ifnextchar} breaks with a \TeX{} error. \texttt{\ltx@ifnextchar} silently removes this token as optional space.

Since 2010/03/01 v1.3.
\ltx@ifnextchar@nospace \langle \langle \text{char} \rangle \rangle \{ \langle \text{yes} \rangle \} \{ \langle \text{no} \rangle \}

Macro \ltx@ifnextchar@nospace behaves like macro \ltx@ifnextchar with the exception that optional spaces are not supported before \langle \text{char} \rangle. Since 2011/04/14 v1.19.

1.16 \ltx@leavevmode, \ltx@mbox

\ltx@leavevmode

Macro \ltx@leavevmode calls pdf\TeX{}'s \texttt{\textbackslash quitvmode}. Otherwise \leavevmode is used and defined if it is necessary.

\ltx@mbox

Macro \ltx@mbox reimplements \mbox with two changes. Instead of \leavevmode it uses \ltx@leavevmode and stops right after \texttt{\hbox}. Especially it does not grab the argument and allows the extended syntax of \texttt{\hbox}.

1.17 Expandable test for emptiness

\ltx@ifempty \langle \langle \text{stuff} \rangle \rangle \{ \langle \text{yes} \rangle \} \{ \langle \text{no} \rangle \}

Macro \ltx@ifempty checks in exact two expansion steps whether \langle \text{stuff} \rangle is empty or contains token. Depending on the result \langle \text{yes} \rangle or \langle \text{no} \rangle is executed. The token in \langle \text{stuff} \rangle may contain \texttt{\textbackslash par} and unmatched conditionals (\texttt{\textbackslash if}, \texttt{\textbackslash else}, \texttt{\textbackslash fi}, \ldots). Since version 2010/11/12 v1.11.

\ltx@ifblank \langle \langle \text{stuff} \rangle \rangle \{ \langle \text{yes} \rangle \} \{ \langle \text{no} \rangle \}

Macro \ltx@ifblank tests in exact two expansion steps if \langle \text{stuff} \rangle is empty or contain only blank spaces. In this case argument \langle \text{yes} \rangle is called. If \langle \text{stuff} \rangle contains other tokens than spaces then \langle \text{no} \rangle is executed. Since version 2010/12/04 v1.13.

1.18 Stripping spaces

\ltx@zapspace \langle \langle \text{stuff} \rangle \rangle

Macro \ltx@zapspace strips spaces from \langle \text{stuff} \rangle that are not hidden inside curly braces. Like \LaTeX{}'s \texttt{\textbackslash zap@space} it is expandable. Differences:

- Syntax: \texttt{\textbackslash zap@space} also expects a space token and \texttt{\textbackslash empty} after \langle \text{stuff} \rangle.
- Macro \ltx@zapspace is expandable in exact two expansion steps.
- Macro \ltx@zapspace always retains curly braces.
- Macro \texttt{\textbackslash zap@space} has a bug. It stops stripping spaces after a token group in curly braces if the first two tokens inside the group are equal.
- Macro \ltx@zapspace also works with \texttt{\textbackslash par} and conditionals \texttt{\textbackslash if}, \texttt{\textbackslash else}, \texttt{\textbackslash fi}, \ldots).

Macro \ltx@zapspace is available since version 2010/12/07 v1.14.
1.19 Check for emptiness of boxes

\texttt{\ltx@IfBoxEmpty \{\langle \text{box register number} \rangle\} \{(\text{yes})\} \{(\text{no})\}}

Macro \texttt{\ltx@IfBoxEmpty} calls \langle yes \rangle if the box exists (\texttt{\ifvoid} returns false) and the box does not contain any content. Otherwise if the box is void or contains something, then \langle no \rangle is executed. Thus being empty means that the box exists and is either an \texttt{\hbox} or a \texttt{\vbox} and may even have dimensions other than 0.0 pt, but the box does not contain anything. Macro \texttt{\ltx@IfBoxEmpty} is available since 2010/02/04 v1.16.

\texttt{\ltx@IfBoxVoidOrEmpty \{\langle \text{box register number} \rangle\} \{(\text{yes})\} \{(\text{no})\}}

Macro \texttt{\ltx@IfBoxVoidOrEmpty} calls \langle yes \rangle if the box is either void or does not contain any content. Otherwise \langle no \rangle is executed. Macro \texttt{\ltx@IfBoxVoidOrEmpty} is available since 2010/02/04 v1.16.

2 Implementation

2.1 Identification

1 (*package)

Reload check, especially if the package is not used with \TeX.

2 \begin{group}
3 \catcode61\catcode48\catcode32=10\relax%
4 \endlinechar=13 %
5 \catcode13=5 % ^^M
6 \catcode35=6 % #
7 \catcode39=12 % '
8 \catcode44=12 % ,
9 \catcode45=12 % -
10 \catcode46=12 % .
11 \catcode58=12 % :
12 \catcode64=11 % @
13 \catcode123=1 % {
14 \catcode125=2 % }
15 \expandafter\let\expandafter\x\csname ver@ltxcmds.sty\endcsname
16 \iffalse\relax % plain-\TeX, first loading
17 \else
18 \def\empty\{}
19 \fi
20 \fi
21 \expandafter\let\expandafter\x\csname PackageInfo\endcsname\relax
22 \def\x#1#2{\immediate\write1{\PackageInfo{#1}{#2, stopped}}}%
23 \else
24 \expandafter\let\expandafter\x\csname PackageInfo\endcsname\relax
25 \immediate\write1{\PackageInfo{\PackageInfo}{\PackageInfo}}%
26 \else
27 \def\x#1#2{\PackageInfo{\PackageInfo}{\PackageInfo}}%
28 \fi
29 \fi
30 \fi
31 \fi
32 \endgroup%
\ltx@zero \chardef\ltx@zero=0 \\
\ltx@one \chardef\ltx@one=1 \\
\ltx@two \chardef\ltx@two=2 \\
\ltx@active \chardef\ltx@active=13 \\
\ltx@cclv \chardef\ltx@cclv=255 \\
\ltx@minusone \def\ltx@minusone{\-\ltx@one} \\
\ltx@LocToksA \toksdef\ltx@LocToksA=0 \\
\ltx@LocToksB \toksdef\ltx@LocToksB=2 

2.2 Numbers

2.3 Scratch registers
\ltx@LocToksC
126 \toksdef\ltx@LocToksC=4 \%
\ltx@LocToksD
127 \toksdef\ltx@LocToksD=6 \%
\ltx@LocToksE
128 \toksdef\ltx@LocToksE=8 \%
\ltx@GlobToksA
129 \toksdef\ltx@GlobToksA=1 \%
\ltx@GlobToksB
130 \toksdef\ltx@GlobToksB=3 \%
\ltx@GlobToksC
131 \toksdef\ltx@GlobToksC=5 \%
\ltx@GlobToksD
132 \toksdef\ltx@GlobToksD=7 \%
\ltx@GlobToksE
133 \toksdef\ltx@GlobToksE=9 \%
\ltx@LocDimenA
134 \dimendef\ltx@LocDimenA=0 \%
\ltx@LocDimenB
135 \dimendef\ltx@LocDimenB=2 \%
\ltx@LocDimenC
136 \dimendef\ltx@LocDimenC=4 \%
\ltx@LocDimenD
137 \dimendef\ltx@LocDimenD=6 \%
\ltx@LocDimenE
138 \dimendef\ltx@LocDimenE=8 \%
\ltx@GlobDimenA
139 \dimendef\ltx@GlobDimenA=1 \%
\ltx@GlobDimenB
140 \dimendef\ltx@GlobDimenB=3 \%
\ltx@GlobDimenC
141 \dimendef\ltx@GlobDimenC=5 \%
\ltx@GlobDimenD
142 \dimendef\ltx@GlobDimenD=7 \%
\ltx@GlobDimenE
143 \dimendef\ltx@GlobDimenE=9 \%
\ltx@LocSkipA
144 \skipdef\ltx@LocSkipA=0 \%
2.4 Argument killers

\ltx@gobble
\long\def\ltx@gobble#1{}

\ltx@gobbletwo
\long\def\ltx@gobbletwo#1#2{}

\ltx@gobblethree
\long\def\ltx@gobblethree#1#2#3{}

\ltx@gobblefour
\long\def\ltx@gobblefour#1#2#3#4{}

\ltx@GobbleNum
\def\ltx@GobbleNum#1{% 
  \romannumeral
  \csname ltx@zero%
  \expandafter\LTXcmds@GobbleNum
  \romannumeral\LTXcmds@num{#1}000{m\endcsname}%
}

\LTXcmds@GobbleNum
\def\LTXcmds@GobbleNum#1{% 
  \endcsname
}

\LTXcmds@Gm
\long\def\LTXcmds@Gm#1{% 
  \endcsname
}
2.5 Argument grabbers

\ltx@firstofone
170 \long\def\ltx@firstofone#1{#1}

\ltx@firstoftwo
171 \long\def\ltx@firstoftwo#1#2{#1}

\ltx@secondoftwo
172 \long\def\ltx@secondoftwo#1#2{#2}

\ltx@firstofthree
173 \long\def\ltx@firstofthree#1#2#3{#1}

\ltx@secondofthree
174 \long\def\ltx@secondofthree#1#2#3{#2}

\ltx@thirdofthree
175 \long\def\ltx@thirdofthree#1#2#3{#3}

\ltx@firstoffour
176 \long\def\ltx@firstoffour#1#2#3#4{#1}

\ltx@secondoffour
177 \long\def\ltx@secondoffour#1#2#3#4{#2}

\ltx@thirddoffour
178 \long\def\ltx@thirddoffour#1#2#3#4{#3}

\ltx@fourthoffour
179 \long\def\ltx@fourthoffour#1#2#3#4{#4}

2.6 List helpers

\ltx@carzero
180 \long\def\ltx@carzero#1\@nil{}\%

\LTXcmds@cdrzero
181 \long\def\LTXcmds@cdrzero#1\@nil{#1}

\ltx@cdrzero
182 \def\ltx@cdrzero{%
183 \romannumeral\LTXcmds@cdrzero\ltx@zero
184 }

\ltx@car
185 \long\def\ltx@car#1#2\@nil{#1}

\ltx@cdr
186 \long\def\ltx@cdr#1{%
187 \romannumeral\LTXcmds@cdrzero\ltx@zero
188 }

\ltx@cartwo
189 \long\def\ltx@cartwo#1#2#3\@nil{#1#2}
\ltx@carsecond
190 \long\def\ltx@carsecond#1#2#3\@nil{#2}

\ltx@cdrtwo
191 \long\def\ltx@cdrtwo#1#2{%
192 \romannumeral\LTXcmds@cdrzero\ltx@zero
193}

\ltx@carthree
194 \long\def\ltx@carthree#1#2#3#4\@nil{#1#2#3}

\ltx@carthird
195 \long\def\ltx@carthird#1#2#3#4\@nil{#3}

\ltx@cdrthree
196 \long\def\ltx@cdrthree#1#2#3{%
197 \romannumeral\LTXcmds@cdrzero\ltx@zero
198}

\ltx@carfour
199 \long\def\ltx@carfour#1#2#3#4#5\@nil{#1#2#3#4}

\ltx@carfourth
200 \long\def\ltx@carfourth#1#2#3#4#5\@nil{#4}

\ltx@cdrfour
201 \long\def\ltx@cdrfour#1#2#3#4{%
202 \romannumeral\LTXcmds@cdrzero\ltx@zero
203}

\ltx@CarNum
204 \def\ltx@CarNum#1{%
205 \romannumeral
206 \csname LTXcmds@CarNumFinish\endcsname{#1}000\{x\endcsname}{%%
207 \expandafter\LTXcmds@CarNum
208 \romannumeral\LTXcmds@num{#1}000\{x\endcsname}{%}
209}

\LTXcmds@CarNum
210 \def\LTXcmds@CarNum#1{%
211 \csname LTXcmds@C#1\endcsname{#1}LTXcmds@CarNum
212}

\LTXcmds@Cm
213 \long\def\LTXcmds@Cm#1#2{%
214 \endcsname{#1#2}%
215}

\LTXcmds@Cx
216 \def\LTXcmds@Cx#1{%
217 \endcsname{}%
218}

\LTXcmds@CarNumFinish
219 \long\def\LTXcmds@CarNumFinish#1#2\@nil{%
220 \ltx@zero
221 #1%
222}
\ltx@CarNumth
223 \def\ltx@CarNumth#1{\%
224 \romannumeral
225 \expandafter\expandafter\expandafter
226 \LTXcmds@CarNumth
227 \ltx@GobbleNum{#1}\%
228 }
\LTXcmds@CarNumth
229 \long\def\LTXcmds@CarNumth#1#2\@nil{\%
230 \ltx@zero
231 #1\%
232 }
\ltx@CdrNum
233 \def\ltx@CdrNum#1{\%
234 \romannumeral\%
235 \expandafter\expandafter\expandafter\ltx@cdrzero
236 \expandafter\expandafter\expandafter\ltx@zero
237 \ltx@GobbleNum{#1}\%
238 }
2.7 Tail recursion
\ltx@ReturnAfterFi
239 \long\def\ltx@ReturnAfterFi#1\fi{\fi#1}
\ltx@ReturnAfterElseFi
240 \long\def\ltx@ReturnAfterElseFi#1\else#2\fi{\fi#1}
2.8 Empty macro
\ltx@empty
241 \def\ltx@empty{}
2.9 Characters
\ltx@space
242 \def\ltx@space{ }
\ltx@percentchar
243 \begingroup
244 \lccode'0='\%\relax
245 \lowercase{\endgroup
246 \def\ltx@percentchar{0}\%
247 }
\ltx@backslashchar
248 \begingroup
249 \lccode'0='\relax
250 \lowercase{\endgroup
251 \def\ltx@backslashchar{0}\%
252 }
2.10 Boolean switch

\ltx@newif

\LTXcmds@newif

\ltx@newglobalif
\LTXcmds@newglobalif

\begingroup
\escapechar=-1 %
\expandafter\endgroup
\expandafter
\def\expandafter\LTXcmds@newglobalif\string\if#1\@nil{%
\expandafter\edef\csname#1true\endcsname{%
\global\let
\expandafter\noexpand\csname if#1\endcsname
\noexpand\iftrue
}
\expandafter\edef\csname#1false\endcsname{%
\global\let
\expandafter\noexpand\csname if#1\endcsname
\noexpand\iffalse
\csname#1false\endcsname
}
}

\ltx@LocalExpandAfter
\def\ltx@LocalExpandAfter{%
\begingroup
\expandafter\expandafter\expandafter\endgroup
\expandafter\expandafter\expandafter}
\ltx@LocalExpandAfter
\ifx\csname ifcsname\endcsname\relax
\ltx@ifundefined
\def\ltx@ifundefined#1{%
\expandafter\ifx\csname #1\endcsname\relax
\expandafter\ltx@firstoftwo
\else
\expandafter\ltx@secondoftwo
\fi
}

\ltx@IfUndefined
\def\ltx@IfUndefined#1{%
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname #1\endcsname\relax
\expandafter\ltx@firstoftwo
\else
\expandafter\ltx@secondoftwo
\fi
\expandafter\ltx@gobble
\else
\expandafter\ltx@firstofone
\fi
}

2.11 Command definitions

\ltx@ifundefined
\def\ltx@ifundefined#1{%
\expandafter\ifx\csname #1\endcsname\relax
\expandafter\ltx@firstoftwo
\else
\expandafter\ltx@secondoftwo
\fi
}

\ltx@IfUndefined
\def\ltx@IfUndefined#1{%
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname #1\endcsname\relax
\expandafter\ltx@firstoftwo
\else
\expandafter\ltx@secondoftwo
\fi
\expandafter\ltx@gobble
\else
\expandafter\ltx@firstofone
\fi
}

19
2.12 Stripping

2.13 File management

2.13.1 File extensions
2.13.3 Version date check
changed 2020-05-10 to adapt to dates with dashes (ISO) The core of the commands are copies from the latex commands.

2.14 Macro additions
2.15 Next character detection
\LTXcmds@ifnextchar  
\def\LTXcmds@ifnextchar{%  
\ifx\LTXcmds@LetToken\LTXcmds@CharToken  
\the\expandafter\ltx@LocToksA  
\else  
\expandafter  
\ifx\csname LTXcmds@LetToken\endcsname\LTXcmds@SpaceToken  
\expandafter\expandafter\expandafter\LTXcmds@@ifnextchar  
\else  
\the\expandafter\expandafter\expandafter\ltx@LocToksB  
\fi  
\fi  
\fi  
\def\LTXcmds@@ifnextchar{\futurelet\expandafter\LTXcmds@LetToken\expandafter\LTXcmds@ifnextchar}  
2.16  
\ltx@leavevmode, \ltx@mbox  
\ltx@leavevmode  
\ltx@ifnextchar@nospace  
\long\def\ltx@ifnextchar@nospace#1#2#3{%  
\begingroup  
\let\LTXcmds@CharToken= #1\relax  
\ltx@LocToksA{\endgroup#2}  
\ltx@LocToksB{\endgroup#3}  
\futurelet\LTXcmds@LetToken\LTXcmds@ifnextchar@nospace  
\ltx@ifnextchar@nospace  
\def\LTXcmds@ifnextchar@nospace{%  
\ifx\LTXcmds@LetToken\LTXcmds@CharToken  
\expandafter\ltx@LocToksA  
\else  
\expandafter\ltx@LocToksB  
\fi  
\fi  
\fi  
\fi  
\endgroup  
\setbox\ltx@zero=\hbox{}
2.17 Help macros

2.18 Expandable test for emptiness

The macro is based on `\ifempty` of Robert R. Schneck [1] and `\ifnull` of Ulrich Diez [2]. There are three cases to consider:
1. \#1 is empty,
2. \#1 is not empty and the first token is not a begingroup character,
3. \#1 starts with a begingroup character (catcode 1).

2.18.2 With \detokenize

Ahmed Musa provided \ifstrempty using \detokenize and \pdfstrcmp [3]. Ulrich Diez, GL, Heiko Oberdiek improved it further by removing \pdfstrcmp and taking three arguments [4, 5, 6, 7, 8].
2.18.3 \ltx@ifblank

\ltx@ifblank

2.19 \ltx@zapspace

2.20 \ltx@IfBoxEmpty

In case of \TeX{} the test for an empty box is done via \texttt{\lastnodetype} as suggested by David Kastrup [9].
Implementation using \texttt{\LaTeX}'s \texttt{lastnodetype}.

\begin{verbatim}
& \begingroup
& \setbox\ltx@zero=\ifhbox#1\hbox\else\vbox\fi{%
& \ifhmode\unhcopy\else\unvcopy\fi#1\relax
& \expandafter
& }% \\
& \expandafter\endgroup
& \ifnum\lastnodetype<\ltx@zero
& \expandafter\expandafter\expandafter\ltx@firstoftwo
& \else
& \expandafter\expandafter\expandafter\ltx@secondoftwo
& \fi
\end{verbatim}

Implementation without \texttt{\LaTeX} using a signature at the beginning of the test box.

\begin{verbatim}
& \begingroup
& \setbox\ltx@zero=\ifhbox#1\hbox\else\vbox\fi{%
& \penalty\ltx@one
& \ifhmode\unhcopy\else\unvcopy\fi#1\relax
& \expandafter
& }% \\
& \ifnum\lastpenalty=\ltx@one
Box 0 has been changed and is restored by closing the group.
& \endgroup
& \begingroup
& \setbox\ltx@zero=\ifhbox#1\hbox\else\vbox\fi{%
& \penalty\ltx@two
& \ifhmode\unhcopy\else\unvcopy\fi#1\relax
& \expandafter
& }% \\
& \ifnum\lastpenalty=\ltx@two
& \def\next{\endgroup\expandafter\ltx@firstoftwo}\
& \else
& \def\next{\endgroup\expandafter\ltx@secondoftwo}\
& \fi
& \else
& \def\next{\endgroup\expandafter\ltx@secondoftwo}\
& \fi
& \next
& \fi
\end{verbatim}

\texttt{\ltx@ifBoxVoidOrEmpty}

\begin{verbatim}
& \def\ltx@ifBoxVoidOrEmpty#1{%
& \ifvoid#1\relax
& \expandafter\ltx@thirdoffour
& \fi
\ltx@ifBoxEmpty{#1}\
& \fi
& \next
& \fi
\end{verbatim}

\texttt{\LTXcmds@AtEnd%}

\texttt{(/package)
3 Installation

3.1 Download

Package. This package is available on CTAN:\footnote{CTAN:pkg/ltxcmds}

\texttt{CTAN:macros/latex/contrib/ltxcmds/ltxcmds.dtx} The source file.
\texttt{CTAN:macros/latex/contrib/ltxcmds/ltxcmds.pdf} Documentation.

3.2 Package installation

The package is at best installed with the package manager of the \TeX{} system. Manual installation is possible too:

Unpacking. The \texttt{.dtx} file is a self-extracting \texttt{docstrip} archive. The files are extracted by running the \texttt{.dtx} through plain \TeX{}:

\begin{verbatim}
tex ltxcmds.dtx
\end{verbatim}

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as \texttt{texmf} tree):

\begin{verbatim}
ltxcmds.sty \rightarrow tex/generic/ltxcmds/ltxcmds.sty
ltxcmds.pdf \rightarrow doc/latex/ltxcmds/ltxcmds.pdf
ltxcmds.dtx \rightarrow source/latex/ltxcmds/ltxcmds.dtx
\end{verbatim}

If you have a \texttt{docstrip.cfg} that configures and enables \texttt{docstrip}'s TDS installing feature, then some files can already be in the right place, see the documentation of \texttt{docstrip}.

3.3 Refresh file name databases

If your \TeX{} distribution (\TeX{} Live, MiK\TeX{}, \ldots) relies on file name databases, you must refresh these. For example, \TeX{} Live users run \texttt{texhash} or \texttt{mktexlsr}.

3.4 Some details for the interested

Unpacking with \LaTeX{}. The \texttt{.dtx} chooses its action depending on the format:

plain \TeX{}: Run \texttt{docstrip} and extract the files.

\LaTeX{}: Generate the documentation.

If you insist on using \LaTeX{} for \texttt{docstrip} (really, \texttt{docstrip} does not need \LaTeX{}), then inform the autodetect routine about your intention:

\begin{verbatim}
latex \let\install=y\input{ltxcmds.dtx}
\end{verbatim}

Do not forget to quote the argument according to the demands of your shell.
Generating the documentation. You can use both the .dtx or the .drv to generate the documentation. The process can be configured by the configuration file \texttt{ltxdoc.cfg}. For instance, put this line into this file, if you want to have A4 as paper format:

\PassOptionsToClass{a4paper}{article}

An example follows how to generate the documentation with \TeX{}:

\begin{verbatim}
 pdflatex ltxcmds.dtx
 makeindex -s gind.ist ltxcmds.idx
 pdflatex ltxcmds.dtx
 makeindex -s gind.ist ltxcmds.idx
 pdflatex ltxcmds.dtx
\end{verbatim}

4 References

[1] Robert R. Schneck: \textit{Re: \texttt{\texttt{\textbackslash ifempty} solution (was Macro puzzle: maximally general \texttt{\texttt{\textbackslash ifempty}}); newsgroup comp.text.tex, news:3eflida_6@corp.newsgroups.com}, 2003-06-17.
https://groups.google.com/group/comp.text.tex/msg/be03a159ec374895

[2] Ulrich Diez: \textit{Re: TeX refuses to strip outer braces in argument}; newsgroup comp.text.tex, news:ibk3t8$ee7$1@news.albasani.net, 2010-11-12.
https://groups.google.com/group/comp.text.tex/msg/803bd57221a04996

https://groups.google.com/group/comp.text.tex/msg/fbf7d61a0c3a807d

[4] Ulrich Diez: \textit{Re: TeX refuses to strip outer braces in argument}; newsgroup comp.text.tex, news:idbo94$uka$1@four.albasani.net, 2010-12-03.
https://groups.google.com/group/comp.text.tex/msg/0c230ee479487962

[5] Ulrich Diez: \textit{Re: TeX refuses to strip outer braces in argument}; newsgroup comp.text.tex, news:idbpu4$cg1$1@news.albasani.net, 2010-12-03.
https://groups.google.com/group/comp.text.tex/msg/bbef4263390d647b

https://groups.google.com/group/comp.text.tex/msg/00dfd1ec103cd272

[7] GL: \textit{Re: TeX refuses to strip outer braces in argument}; newsgroup comp.text.tex, news:4cfa2e27$0$7389$426a74cc@news.free.fr, 2010-12-04.
https://groups.google.com/group/comp.text.tex/msg/d3a75995c1cf267e

https://groups.google.com/group/comp.text.tex/msg/5f7a23e3ab7e347

https://groups.google.com/group/comp.text.tex/msg/8d3cb89496a4d86d
5 History

[2009/08/05 v1.0]
- First version.

[2009/12/12 v1.1]
- Short title shortened.
- \ltx@ifUndefined added.

[2010/01/28 v1.2]
- \ltx@RemovePrefix and \ltx@StripPrefix added.
- \ltx@ifclassloaded, \ltx@ifpackageloaded, \ltx@iffileloaded, \ltx@ifclasslater, \ltx@ifpackagelater, \ltx@iffilelater, \ltx@clsextension, \ltx@pkgextension added.
- \ltx@GlobalAppendToMacro, \ltx@LocalAppendToMacro added.

[2010/03/01 v1.3]
- \ltx@newif added.
- \ltx@ifnextchar added.
- Numbers \ltx@zero, \ltx@one, \ltx@two, \ltx@cclv added.

[2010/03/09 v1.4]
- \ltx@pkgextension and \ltx@clsextension are hardcoded to avoid trouble with \@onlypreamble.

[2010/04/08 v1.5]
- \ltx@cartwo, \ltx@cdrtwo, \ltx@carthree, \ltx@cdrthree, \ltx@carfour, \ltx@cdrfour added.
- \ltx@ReturnAfterFi and \ltx@ReturnAfterElseFi fixed.

[2010/04/16 v1.6]
- \ltx@leavevmode, \ltx@mbox added.

[2010/04/26 v1.7]
- \ltx@GobbleNum, \ltx@CdrNum, \ltx@CarNum added.
- \ltx@carzero, \ltx@cdrzero added.
- \ltx@hashchar added.

[2010/09/11 v1.8]
- \ltx@leftbracechar, \ltx@rightbracechar added.
[2010/10/25 v1.9]
  • \ltx@LocalAppendToMacro and \ltx@GlobalAppendToMacro are now \long.

[2010/10/31 v1.10]
  • \ltx@newglobalif added.

[2010/11/12 v1.11]
  • \ltx@ifempty added.
  • \ltx@firstofthree, \ltx@secondofthree, \ltx@thirdofthree added.

[2010/12/02 v1.12]
  • \ltx@onelevel@sanitize added.
  • \LTXcmds@num fixed for the case with \numexpr (bug found by GL).

[2010/12/04 v1.13]
  • \ltx@ifblank added.
  • Optimization for \ltx@ifempty.

[2010/12/07 v1.14]
  • \ltx@zapspace added.

[2010/12/12 v1.15]
  • \ltx@minusone added.

[2011/02/04 v1.16]
  • \ltx@IfBoxEmpty and \ltx@IfBoxVoidOrEmpty added.
  • \ltx@firstoffour, ..., \ltx@fourthoffour added.

[2011/02/05 v1.17]
  • \ltx@IfBoxEmpty: an empty box may have non-zero dimensions.

[2011/03/16 v1.18]
  • \ltx@ifclasslater fixed.

[2011/04/14 v1.19]
  • \ltx@ifnextchar: detection of optional spaces modified.
  • \ltx{(Loc,Glob)(Toks,Dimen,Skip)(A,B,C,D,E) added.

[2011/04/18 v1.20]
  • \ltx@ifnextchar with conditional support (thanks GL for bug report).
[2011/08/22 v1.21]
- \ltx@GlobalPrependToMacro, \ltx@LocalPrependToMacro added (feature request of Martin M"unch).

[2011/11/09 v1.22]
- \ltx@carsecond, \ltx@carthird, \ltx@carfourth, \ltx@CarNumth added.
- \ltx@cdrzero, \ltx@cdr, \ltx@cdrtwo, c\ltx@cdrthree, \ltx@cdrfour, \ltx@CdrNum modified to retain braces and spaces. They are expandable in two expansion steps.

[2016/05/16 v1.23]
- Documentation updates.

[2019/12/15 v1.24]
- Documentation updates.

[2020-05-10 v1.25]
- Changed the definitions of \ltx@iffilelater, \ltx@ifpackagelater and \ltx@ifclasslater to support dates in ISO format in same way as the LaTeX kernel does it since 2017. The commands now use the same test as the LaTeX kernel. \pdfmatch is no longer used with pdftex, and the tests for dates before 1994 have been removed

6 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; plain numbers refer to the code lines where the entry is used.

<table>
<thead>
<tr>
<th>Symbols</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>. . . . . . . . . . . . . . . . . . 254</td>
<td>\box</td>
<td>. . . . . . . . . . . . . . . 501, 512</td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>. . . . . . . . . . . . . . . . . . 610, 613</td>
<td>\catcode</td>
<td>. . . . . . . . . . . . . . . . . . 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 69, 70, 72, 73, 74, 78, 79, 80, 81, 82, 83, 84, 87, 88, 90, 91, 92, 93, 97, 99, 610, 611, 613, 614</td>
<td>\chardef</td>
</tr>
<tr>
<td>%</td>
<td>. . . . . . . . . . . . . . . . . . 244</td>
<td>\csname</td>
<td>. . . . . . . . . . . . . . . . . .</td>
<td></td>
</tr>
<tr>
<td>&amp;</td>
<td>. . . . . . . . . . . . . . . . . . 611, 614</td>
<td>\dertokenize</td>
<td>. . . . . . . . . . . . . . . . . .</td>
<td></td>
</tr>
<tr>
<td>|</td>
<td>. . . . . . . . . . . . . . . . . . 475</td>
<td>\edef</td>
<td>. . . . . . . . . . . . . . . . . .</td>
<td></td>
</tr>
<tr>
<td>@firstoftwo</td>
<td>. . . . . . . . . . . . . . . . . . 383</td>
<td>\expandafter</td>
<td>. . . . . . . . . . . . . . . . . .</td>
<td></td>
</tr>
<tr>
<td>@nil</td>
<td>. . . . . . . . . . . . . . . . . . 180, 181, 185, 189, 190, 194, 195, 199, 200, 219, 229, 272, 277, 294, 300, 379, 380, 386, 387, 389, 600, 602, 606</td>
<td>\expandafter</td>
<td>. . . . . . . . . . . . . . . . . .</td>
<td></td>
</tr>
<tr>
<td>@secondoftwo</td>
<td>. . . . . . . . . . . . . . . . . . 381</td>
<td>\expandafter</td>
<td>. . . . . . . . . . . . . . . . . .</td>
<td></td>
</tr>
<tr>
<td>@undefined</td>
<td>. . . . . . . . . . . . . . . . . . 58</td>
<td>\expandafter</td>
<td>. . . . . . . . . . . . . . . . . .</td>
<td></td>
</tr>
<tr>
<td>\</td>
<td>. . . . . . . . . . . . . . . . . . 249</td>
<td>\expandafter</td>
<td>. . . . . . . . . . . . . . . . . .</td>
<td></td>
</tr>
<tr>
<td>|</td>
<td>. . . . . . . . . . . . . . . . . . 259</td>
<td>\expandafter</td>
<td>. . . . . . . . . . . . . . . . . .</td>
<td></td>
</tr>
<tr>
<td>|</td>
<td>. . . . . . . . . . . . . . . . . . 264</td>
<td>\expandafter</td>
<td>. . . . . . . . . . . . . . . . . .</td>
<td></td>
</tr>
<tr>
<td>\aftergroup</td>
<td>. . . . . . . . . . . . . . . . . . 29</td>
<td>\string</td>
<td>. . . . . . . . . . . . . . . . . .</td>
<td></td>
</tr>
</tbody>
</table>

32
<table>
<thead>
<tr>
<th>V</th>
<th>\write</th>
<th>W</th>
<th>\vbox</th>
<th>621, 632, 640</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>\voidb@x</td>
<td>518</td>
<td>14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87</td>
</tr>
</tbody>
</table>