

L^AT_EX and Friends

User-Defined Commands and Environments

M. R. C. van Dongen

ucc

January 18, 2012

Advantages of Automation

SE Tedious tasks can be automated.

reusability Define once, use many times.

simplicity Easier to use. Avoids errors.

refinement Allows stepwise refinement.

maintainability Make local changes with global effect.

consistency Guarantees consistent typesetting.

computing Tasks and results are controlled by document options.

style control Different output style for different option.

content control Commands may result in different output.

typeset results Do basic arithmetic and branching, and

- typeset *results* of computations.

Disadvantages of L^AT_EX Commands

namespace No local identifiers.

parameters Two problems related to the parameters of the commands.

- No more than 9 parameters.
- Formal parameters are numbers.

Defining Commands

```
\newcommand⟨cmd⟩{⟨subst⟩}
```

- Defines ⟨cmd⟩.
- Using ⟨cmd⟩ results in ⟨subst⟩.

```
\renewcommand⟨cmd⟩{⟨subst⟩}
```

- Redefines command.

Example

L^AT_EX Usage

```
\documentclass{article}
\newcommand\CTAN{Comprehensive \TeX{} Archive Network}
\begin{document}
  I always download my packages from the \CTAN.
  The \CTAN{} is the place to be.
\end{document}
```

L^AT_EX Output

I always download my packages from the Comprehensive T_EX
Archive Network. The Comprehensive T_EX Archive Network is
the place to be.

[Commands/Environments](#)[Some Terminology](#)[Advantages and Disadvantages](#)[User-defined Commands](#)[Commands and Parameters](#)[Defining Commands with T_EX](#)[Tweaking Existing Commands](#)[Using More than Nine Parameters](#)[Using Environments](#)[Acronyms & Abbreviations](#)[About this Document](#)

How did that Work in Math?

$$f: \mathbb{N} \rightarrow \mathbb{N}$$
$$x \mapsto x^2 + 2x.$$

- The x is the *formal* parameter of f .
- The expression $x^2 + 2x$ defines the computation.
 - In computer science terminology, this is called the *substitution text*.
- You write $f(\langle \text{expr} \rangle)$ to apply f to the *actual parameter* $\langle \text{expr} \rangle$.
- To evaluate $f(\langle \text{expr} \rangle)$ you substitute the actual parameter $\langle \text{expr} \rangle$ for each occurrence of the formal parameter x in the substitution text of f .

Commands with Parameters (No Options)

```
\newcommand<cmd> [<digit>] {<subst>}
```

- ▣ Defines `<cmd>`.
 - ▣ Command takes `<digit>` parameters (1–9).
 - ▣ The *i*th formal parameter is denoted `#i` in `<subst>`.
 - ▣ Actual parameter `#i` is substituted for `#i` in `<subst>`.

```
\renewcommand<cmd> [<digit>] {<subst>}
```

- ▣ Redefines `<cmd>`.

Commands with Parameters (Example)

L^AT_EX Usage

```
\newcommand\opening[1]{%  
  Dear #1,%  
}  
  
\begin{document}  
  \opening{Mum},\!\! [2\baselineskip]  
  {\LaTeX} is going great in~2012.  
  We're studying user-defined macros now.  
\end{document}
```

L^AT_EX Output

Dear Mum,

L^AT_EX is going great in 2012. We're studying user-defined macros now.

Commands with Parameters and Options

```
\newcommand<cmd> [<digit>] [<default>] {<subst>}
```

- Defines `<cmd>`.
 - Command takes `<digit>` parameters.
 - One parameter is optional.
 - Optional parameter is enclosed in square brackets.
 - Without optional parameter #1 is assigned `<default>`.

```
\renewcommand<cmd> [<digit>] [<default>] {<subst>}
```

- Redefines existing command.

Commands with Parameters and Options (Example)

L^AT_EX Usage

```
\newcommand\congratulations[2][a teddy bear]{%  
  Congratulations #2. You've won #1.  
}  
  
\begin{document}  
  \congratulations{John}  
  
  \congratulations[a train set]{Luke}  
\end{document}
```

L^AT_EX Output

Congratulations John. You've won a teddy bear.
Congratulations Luke. You've won a train set.

Fragile and Robust Commands

- *Moving parameters* are saved to be reread later on.
- Examples: parameters that are written to auxiliary files.
- Moving parameters are expanded before they are saved.
- Sometimes expansion leads to invalid T_EX.
- Command is *robust* if it expands to valid T_EX.
- Otherwise it's called *fragile*.
- The command `\protect` protects commands against expansion.
- Saving `\protect\cmd` saves `\cmd` without expanding.
- Protects fragile commands in moving arguments.

Defining Robust Commands

```
\DeclareRobustCommand⟨cmd⟩{⟨subst⟩}
```

```
\DeclareRobustCommand⟨cmd⟩[⟨digit⟩]{⟨subst⟩}
```

```
\DeclareRobustCommand⟨cmd⟩[⟨digit⟩][⟨default⟩]{⟨subst⟩}
```

```
\MakeRobustCommand⟨cmd⟩
```

Kinds of Tokens

Commands/Environments

Some Terminology

Advantages and Disadvantages

User-defined Commands

Commands and Parameters

Defining Commands with T_EX

Tweaking Existing Commands

Using More than Nine Parameters

Using Environments

Acronyms & Abbreviations

About this Document

character token Corresponds to a single character (not `\`).

control sequence token Correspond to a command.

Kinds of Parameters

primitive parameter Single character or control sequence token.

- Opening and closing brace token not allowed.

compound parameter Brace-delimited group.

- To evaluate an n -parameter macro, L^AT_EX does the following.
 - It removes the macro and the n actual parameters from the token stream.
 - It carries out parameter substitution for i from 1 to n .
 - Substitutes i th actual parameter for i th formal parameter in substitution text.
 - For compound parameter, the outermost brace pair is removed.
 - It puts the resulting expression in front of the token stream.

Example

L^AT_EX Usage

```
\newcommand\swop[2]{#2#1}
```

```
\newcommand\SWOP[2]{#2#1}
```

□ `\swop2\SWOP31`

Example

L^AT_EX Usage

```
\newcommand\swop[2]{#2#1}
```

```
\newcommand\SWOP[2]{#2#1}
```

□ $\swop{2}{\SWOP{31}} \mapsto \SWOP{231}$

Example

L^AT_EX Usage

```
\newcommand\swop[2]{#2#1}
```

```
\newcommand\SWOP[2]{#2#1}
```

□ $\backslash\text{swop}2\backslash\text{SWOP}31 \mapsto \backslash\text{SWOP}231 \mapsto 321$

L^AT_EX Input

```
\documentclass{article}

\newcommand\K[2]{#1}
\newcommand\S[3]{#1#3{#2#3}}
\newcommand\I{\S\K\K}
\newcommand\X{\S{\K{\S\I}}{\S{\K\K}\I}}

\begin{document}
  \X abc
\end{document}
```

Combinatory Logic (Continued)

Commands/Environments

Some Terminology

Advantages and Disadvantages

User-defined Commands

Commands and Parameters

Defining Commands with \TeX

Tweaking Existing Commands

Using More than Nine Parameters

Using Environments

Acronyms & Abbreviations

About this Document

1		$\backslash X_1 a_1 b_1 c_1$
2		$\backslash S_1 \{ \{ \backslash K_2 \{ \{ \backslash S_3 \backslash I_3 \} \}_2 \}_1 \{ \{ \backslash S_2 \{ \{ \backslash K_3 \backslash K_3 \} \}_2 \} \backslash I_2 \} \}_1 a_1 b_1 c_1$
3		$\backslash K_1 \{ \{ \backslash S_2 \backslash I_2 \} \}_1 a_1 \{ \{ \backslash S_2 \{ \{ \backslash K_3 \backslash K_3 \} \}_2 \} \backslash I_2 a_2 \} \}_1 b_1 c_1$
4		$\backslash S_1 \backslash I_1 \{ \{ \backslash S_2 \{ \{ \backslash K_3 \backslash K_3 \} \}_2 \} \backslash I_2 a_2 \} \}_1 b_1 c_1$
5		$\backslash I_1 b_1 \{ \{ \backslash S_2 \{ \{ \backslash K_3 \backslash K_3 \} \}_2 \} \backslash I_2 a_2 b_2 \} \}_1 c_1$
6		$\backslash S_1 \backslash K_1 \backslash K_1 b_1 \{ \{ \backslash S_2 \{ \{ \backslash K_3 \backslash K_3 \} \}_2 \} \backslash I_2 a_2 b_2 \} \}_1 c_1$
7		$\backslash K_1 b_1 \{ \{ \backslash K_2 b_2 \} \}_1 \{ \{ \backslash S_2 \{ \{ \backslash K_3 \backslash K_3 \} \}_2 \} \backslash I_2 a_2 b_2 \} \}_1 c_1$
8		$b_1 \{ \{ \backslash S_2 \{ \{ \backslash K_3 \backslash K_3 \} \}_2 \} \backslash I_2 a_2 b_2 \} \}_1 c_1$
9	b	$\{ \{ \backslash S_2 \{ \{ \backslash K_3 \backslash K_3 \} \}_2 \} \backslash I_2 a_2 b_2 \} \}_1 c_1$
10	b	$\backslash S_2 \{ \{ \backslash K_3 \backslash K_3 \} \}_2 \backslash I_2 a_2 b_2 \} \}_1 c_1$
11	b	$\backslash K_2 \backslash K_2 a_2 \{ \{ \backslash I_3 a_3 \} \}_2 b_2 \} \}_1 c_1$
12	b	$\backslash K_2 \{ \{ \backslash I_3 a_3 \} \}_2 b_2 \} \}_1 c_1$
13	b	$\backslash I_2 a_2 \} \}_1 c_1$
14	b	$\backslash S_2 \backslash K_2 \backslash K_2 a_2 \} \}_1 c_1$
15	b	$\backslash K_2 a_2 \{ \{ \backslash K_3 a_3 \} \}_2 \} \}_1 c_1$
16	b	$a_2 \} \}_1 c_1$
17	ba	$\} \}_1 c_1$
18	ba	c_1

```
\def⟨cmd⟩#1#2...#n{⟨subst⟩}
```

- Defines command `⟨cmd⟩` with n parameters.
- The substitution text of `⟨cmd⟩` is `⟨subst⟩`.

```
\edef⟨cmd⟩#1#2...#n{⟨subst⟩}
```

- Defines command `⟨cmd⟩` with n parameters.
- The substitution text of `⟨cmd⟩` is the *expansion* of `⟨subst⟩`.

T_EX Commands Without Delimiters (Example)

L^AT_EX Input

```
\def\hi{hi}

\def\hello{\hi}
\edef\ehello{\hi}

\def\hi{HI}

\ehello. \hello.
```

L^AT_EX Output

hi. HI.

Commands/Environments

Some Terminology

Advantages and Disadvantages

User-defined Commands

Commands and Parameters

Defining Commands with T_EX

Tweaking Existing Commands

Using More than Nine Parameters

Using Environments

Acronyms & Abbreviations

About this Document

Local Macro Definitions

- For each level of local T_EX macro definitions you double the number of #s.

L^AT_EX Usage

```

\def\topLevelCommand#1{%
  \def\lowLevelCommand##1{%
    Top level: #1. Low level: ##1.%
  }
  \lowLevelCommand{LOW}%
}
\topLevelCommand{HIGH}

```

L^AT_EX Output

Top level: HIGH. Low level: LOW.

Low-level T_EX Commands

`\csname_<tokens>\endcsname`

Expands `<tokens>` and turns it into a control sequence.

`\noexpand<token>`

Returns `<token>` without expanding it.

`\expandafter<token><tokens>`

Expands first token in `<tokens>` (once).

Low-level T_EX Command (Example)

L^AT_EX Usage

```

\def\property#1{%
  % ‘‘\def\#1##1{##1 is #1}’’
  \expandafter\def\csname#1\endcsname##1{%
    ##1\ is #1%
  }%
}
\property{brilliant}
\property{excellent}
...
\excellent{\TeX} and
\brilliant{\LaTeX}.

```

L^AT_EX Output

T_EX is excellent and L^AT_EX is brilliant.

Definitions with Delimiters

L^AT_EX Usage

```
\def\command|#1|#2|{...}
```

Example

L^AT_EX Usage

```
% allow @ in macro names
\makeatletter%
\def\cmd#1{%
  \@ifnextchar [%
    % use the given option
    {\cmd@relay{#1}}%
    % use the default option
    {\cmd@relay{#1}[dflt]}%
  }
\def\cmd@relay#1[#2]{...}
% disallow @ in macro names
\makeatother
```

L^AT_EX Usage

```
\makeatletter
\def\cmd#1{%
  \def\cmd@relay##1[##2]{...}
  \@ifnextchar [%
    {\cmd@relay{#1}}%
    {\cmd@relay{#1}[dflt]}%
  }
}
```

Commands/Environments

Some Terminology

Advantages and Disadvantages

User-defined Commands

Commands and Parameters

Defining Commands with T_EX

Tweaking Existing Commands

Using More than Nine Parameters

Using Environments

Acronyms & Abbreviations

About this Document



The `\let` Command

L^AT_EX Usage

```
\makeatletter
% Save meaning of old \section command.
\let\old@section=\section
\def\section#1#2{%
  % Define section using old \section command.
  \old@section{#2}
  % Define label for the section.
  \label{#1}
}
\makeatother
```

More than Nine Parameters

L^AT_EX Usage

```
\makeatletter
\def\cmd#1#2#3#4#5#6#7#8#9{%
  \def\cmd@arg@A{#1}%
  \def\cmd@arg@B{#2}%
  :
  :
  \def\cmd@arg@I{#9}%
  \relay%
}
\def\relay#1{%
  Parameters: \cmd@arg@A, \cmd@arg@B, ..., and #1.%
}
\makeatother
```

[Commands/Environments](#)[Some Terminology](#)[Advantages and Disadvantages](#)[User-defined Commands](#)[Commands and Parameters](#)[Defining Commands with T_EX](#)[Tweaking Existing Commands](#)[Using More than Nine Parameters](#)[Using Environments](#)[Acronyms & Abbreviations](#)[About this Document](#)

More than Nine Parameters (Continued)

[Commands/Environments](#)[Some Terminology](#)[Advantages and Disadvantages](#)[User-defined Commands](#)[Commands and Parameters](#)[Defining Commands with T_EX](#)[Tweaking Existing Commands](#)[Using More than Nine Parameters](#)[Using Environments](#)[Acronyms & Abbreviations](#)[About this Document](#)

L^AT_EX Usage

```
\def\cmd#1#2#3#4#5#6#7#8#9{%  
  \def\relay##1{Parameters: #1, #2, ..., and ##1.}%  
  \relay%  
}
```

Advantages of Environments

less ambiguity If nested, makes it easier to read.

higher efficiency Reduces need for extra stack space.

User-defined Environments

L^AT_EX Usage

```

\newenvironment{SectionalUnit}[2][section]
    {\csname#1\endcsname{#2}%
     \begin{refsection}}
    {\printbibliography%
     \end{refsection}}

\begin{document}
  \begin{SectionalUnit}[chapter]{Introduction}
    \begin{SectionalUnit}{Conventions}
      ...
    \end{SectionalUnit}
    \begin{SectionalUnit}{Notation}
      ...
    \end{SectionalUnit}
  \end{SectionalUnit}
  :
\end{document}

```

Commands/Environments

Some Terminology

Advantages and Disadvantages

User-defined Commands

Commands and Parameters

Defining Commands with \TeX

Tweaking Existing Commands

Using More than Nine Parameters

Using Environments

Acronyms & Abbreviations

About this Document



Environment Definitions

Commands/Environments

[Some Terminology](#)[Advantages and Disadvantages](#)[User-defined Commands](#)[Commands and Parameters](#)[Defining Commands with T_EX](#)[Tweaking Existing Commands](#)[Using More than Nine Parameters](#)[Using Environments](#)

Acronyms & Abbreviations

[About this Document](#)

```
\newenvironment{<name>}{<begin subst>}{<end subst>}  
\newenvironment{<name>}[<digit>]{<begin subst>}{<end subst>}  
\newenvironment{<name>}[<digit>][<default>]{<begin subst>}{<end subst>}
```

Bibliography

Commands/Environments

Some Terminology

Advantages and Disadvantages

User-defined Commands

Commands and Parameters

Defining Commands with \TeX

Tweaking Existing Commands

Using More than Nine Parameters

Using Environments

Acronyms & Abbreviations

About this Document

Acronyms and Abbreviations

AMS American Mathematical Society

API Application Programming Interface

APL A Programming Language

CTAN Comprehensive T_EX Archive Network

CD Compact Disk

FAQ Frequently Asked Question

GUI Graphical User Interface

IDE Integrated Development Environment

ISBN International Standard Book Number

SI Système International d'Unités/International System of Units

OS Operating System

TUG T_EX Users Group

URL Uniform Resource Locator

WYSIWYG What You See is What You Get

About this Document

- This document was created with `pdflatex`.
- The LaTeX document class is `beamer`.
- The main font is *TeX Gyre Heros Condensed*.
 - You may obtain the font from <http://www.gust.org.pl>.