# TOKENS

## tokens as I see them

context 2020 meeting

## About tokens

- Like nodes, it’s a common term used in programming.
- In TEX The Program tokens and nodes are therefore omni-present.
- For most users they are irrelevant concepts.
- But we will explain them anyway.
- Let’s try to avoid the snobbish token-speak sometimes heard in the community.
- So . . . I won’t correct you as long as you don’t correct me.
- Let’s now enter the world of tokens in the naïve way.

## What are tokens

- It is an internal data structure, effectively a (32 bit) integer.
- This integer encodes a command (opcode) and an char code (operand).
- But often it's not a character but more a sub command.
- Input is converted into tokens.
- Tokens are either expanded (interpreted) or stored.
- When they are stored they are part of a larger data structure, a memory word.
- Token memory is an array of such memory words.
- The token memory 'word' has two integers: a token value and an index into token memory.
- That way TEX can have forward linked lists of tokens.
- A hash table maps control sequences onto indices into token memory.

## Some implementation details

- Sometimes there is special head token at the start.
- A head token makes for easier appending of extra tokens.
- Shared lists use the head node for a reference count.
- Original TEX uses global temporary lists.
- This is needed when we expand (nested) and need to report issues.
- This is not needed when we just serialize (which we do a lot in LuaTEX).
- So, this is all optimized for performance and memory consumption.
- Freed tokens are collected in a cache so tokens can get scattered.
- In LuaMetaTEX we stay as close to original TEX as possible.
- But the Lua interfaces force us to occasionally divert.

# A schematic view of tokens

A token value:

| | |
|---|---|
| cmd | chr |

Token memory:

| | | |
|---|---|---|
| 1 | info | link |
| 2 | info | link |
| 3 | info | link |
| n | info | link |

## Looking up control sequences

- A very visible to-be-token is a `\controlsequence`.
- When read, the name will be looked up in the hash table.
- When found its value will point to the table of equivalents.
- That table keeps track of:
  - the type (cmd)
  - the current level (grouping)
  - the current meaning (token list)

## The (big) table of equivalents (simplified)

| | |
|---|---|
| main hash | null control sequence |
| | 128K hash entries |
| | frozen control sequences |
| | special sequences (undefined) |
| registers | 17 internal & 64K user glues |
| | 4 internal & 64K user mu glues |
| | 12 internal & 64K user tokens |
| | 2 internal & 64K user boxes |
| | 116 internal & 64K user integers |
| | 0 internal & 64K user attribute |
| | 22 internal & 64K user dimensions |
| specifications | 5 internal & 0 user |
| extra hash | additional entries (grows dynamic) |

## The hash table (simplified)

The hash table runs parallel to the main hash. On the todo list is is to move the registers to its own tables and make them dynamic.

| | | |
|---|---|---|
| 1 | string index | equivalents or (next > n) index |
| 2 | string index | equivalents or (next > n) index |
| n | string index | equivalents or (next > n) index |
| n + 1 | string index | equivalents or (next > n) index |
| n + 2 | string index | equivalents or (next > n) index |
| n + m | string index | equivalents or (next > n) index |

Equivalents (registers direct, macros indirect i.e. token lists):

| | | | |
|---|---|---|---|
| 1 | level | type | value |
| 2 | level | type | value |
| 3 | level | type | value |
| n | level | type | value |

## Other data management

- Grouping is handles by a nesting stack.
- Nested conditionals (`\if...`) have their own stack.
- The values before assignments are saved ion the save stack.
- Also other local changes (housekeeping) ends up in the save stack.
- Token lists and macro aliases have references pointers (reuse).
- Attributes, being linked node lists, have their own management.

## Example 1: in the input

```
\luatokentable{1 \bf{2} 3\what {!}}
```

**given token list:**

| | | | | | | |
|---|---|---|---|---|---|---|
| 644687 | 12 | 49 | other char | 1 | U+00031 | |
| 648283 | 10 | 32 | spacer | | | |
| 648454 | 126 | 0 | protected call | | | bf |
| 648164 | 1 | 123 | left brace | | | |
| 648330 | 12 | 50 | other char | 2 | U+00032 | |
| 186739 | 2 | 125 | right brace | | | |
| 648908 | 10 | 32 | spacer | | | |
| 648219 | 12 | 51 | other char | 3 | U+00033 | |
| 648355 | 113 | 0 | undefined cs | | | what |
| 648440 | 1 | 123 | left brace | | | |
| 648165 | 12 | 33 | other char | ! | U+00021 | |
| 644722 | 2 | 125 | right brace | | | |

# Example 1: in the input

```
\luatokentable{a \the\scratchcounter b \the\parindent \hbox to 10pt{x}}
```

**given token list:**

| | | | | | | |
|---|---|---|---|---|---|---|
| 648496 | 11 | 97 | letter | a | U+00061 | |
| 648386 | 10 | 32 | spacer | | | |
| 649615 | 123 | 0 | the | | | the |
| 648482 | 80 | 257 | register int | | | scratchcounter |
| 644629 | 11 | 98 | letter | b | U+00062 | |
| 647985 | 10 | 32 | spacer | | | |
| 648857 | 123 | 0 | the | | | the |
| 648273 | 83 | 0 | internal dimen | | | parindent |
| 648469 | 21 | 9 | make box | | | hbox |
| 648759 | 11 | 116 | letter | t | U+00074 | |
| 648307 | 11 | 111 | letter | o | U+0006F | |
| 644693 | 10 | 32 | spacer | | | |
| 649264 | 12 | 49 | other char | 1 | U+00031 | |
| 648391 | 12 | 48 | other char | 0 | U+00030 | |
| 273707 | 11 | 112 | letter | p | U+00070 | |
| 647938 | 11 | 116 | letter | t | U+00074 | |
| 648032 | 1 | 123 | left brace | | | |
| 648302 | 11 | 120 | letter | x | U+00078 | |
| 648102 | 2 | 125 | right brace | | | |

# Example 2: user registers

```
\scratchtoks{foo \framed{\red 123}456}

\luatokentable\scratchtoks
```

**token register: scratchtoks**

| | | | | | | |
|---|---|---|---|---|---|---|
| 648408 | 11 | 102 | letter | f | U+00066 | |
| 648511 | 11 | 111 | letter | o | U+0006F | |
| 186746 | 11 | 111 | letter | o | U+0006F | |
| 648280 | 10 | 32 | spacer | | | |
| 648592 | 126 | 0 | protected call | | | framed |
| 648704 | 1 | 123 | left brace | | | |
| 649436 | 126 | 0 | protected call | | | red |
| 648650 | 12 | 49 | other char | 1 | U+00031 | |
| 649178 | 12 | 50 | other char | 2 | U+00032 | |
| 649390 | 12 | 51 | other char | 3 | U+00033 | |
| 649797 | 2 | 125 | right brace | | | |
| 648533 | 12 | 52 | other char | 4 | U+00034 | |
| 596463 | 12 | 53 | other char | 5 | U+00035 | |
| 596479 | 12 | 54 | other char | 6 | U+00036 | |

## Example 3: internal variables

```
\luatokentable\everypar
```

**internal token variable: everypar**

| | | | | |
|---|---|---|---|---|
| 648283 | 0 | 1114112 | relax | dotagsetparcounter |
| 648454 | 126 | 0 | protected call | page_otr_command_synchronize_side_floats |
| 648164 | 125 | 0 | call | checkindentation |
| 648330 | 0 | 1114112 | relax | showparagraphnumber |
| 186739 | 0 | 1114112 | relax | restoreinterlinepenalty |
| 648908 | 0 | 1114112 | relax | flushnotes |
| 648219 | 0 | 1114112 | relax | synchronizenotes |
| 648355 | 126 | 0 | protected call | registerparoptions |
| 648440 | 0 | 1114112 | relax | flushpostponednodedata |
| 648165 | 0 | 1114112 | relax | typo_delimited_repeat |
| 644722 | 0 | 1114112 | relax | insertparagraphintro |
| 648873 | 0 | 1114112 | relax | typo_initial_handle |
| 589549 | 0 | 1114112 | relax | typo_firstline_handle |
| 648496 | 0 | 1114112 | relax | spac_paragraph_wrap |
| 648386 | 126 | 0 | protected call | spac_paragraph_freeze |

# Example 4: macro definitions

```
\protected\def\whatever#1[#2](#3)\relax{oeps #1 and #2 & #3 done ## error}

\luatokentable\whatever
```

```
protected control sequence: whatever
650652 13       1  argument
650684 12      91  other char  [  U+0005B
648475 13       2  argument
650683 12      93  other char  ]  U+0005D
650695 12      40  other char  (  U+00028
650636 13       3  argument
650713 12      41  other char  )  U+00029
648666  0 1114112  relax                     relax
650716 14       0  end match
649490 11     111  letter      o  U+0006F
650626 11     101  letter      e  U+00065
650661 11     112  letter      p  U+00070
650627 11     115  letter      s  U+00073
650024 10      32  spacer
649973  5       1  parameter
648210 10      32  spacer
649978 11      97  letter      a  U+00061
648824 11     110  letter      n  U+0006E
649968 11     100  letter      d  U+00064
649956 10      32  spacer
650587  5       2  parameter
649519 10      32  spacer
648966 12      38  other char  &  U+00026
650719 10      32  spacer
650634  5       3  parameter
650638 10      32  spacer
650602 11     100  letter      d  U+00064
650699 11     111  letter      o  U+0006F
650643 11     110  letter      n  U+0006E
650644 11     101  letter      e  U+00065
650679 10      32  spacer
650692  6      35  mac param
649987 10      32  spacer
650708 11     101  letter      e  U+00065
650078 11     114  letter      r  U+00072
650017 11     114  letter      r  U+00072
650051 11     111  letter      o  U+0006F
649902 11     114  letter      r  U+00072
```

# Example 5: commands

```
\luatokentable\startitemize
```

**protected control sequence: startitemize**

| | | | | | | |
|---|---|---|---|---|---|---|
| 650979 | 14 | 0 | end match | | | |
| 650976 | 126 | 0 | protected call | | | startitemgroup |
| 650593 | 12 | 91 | other char | [ | U+0005B | |
| 634308 | 11 | 105 | letter | i | U+00069 | |
| 649896 | 11 | 116 | letter | t | U+00074 | |
| 650735 | 11 | 101 | letter | e | U+00065 | |
| 650737 | 11 | 109 | letter | m | U+0006D | |
| 648712 | 11 | 105 | letter | i | U+00069 | |
| 651309 | 11 | 122 | letter | z | U+0007A | |
| 650628 | 11 | 101 | letter | e | U+00065 | |
| 644614 | 12 | 93 | other char | ] | U+0005D | |

# Example 6: commands

```
\luatokentable\doifelse
```

**protected control sequence: doifelse**

| | | | | |
|---|---|---|---|---|
| 651688 | 13 | 1 | argument | |
| 651700 | 13 | 2 | argument | |
| 651667 | 14 | 0 | end match | |
| 650720 | 120 | 21 | if test | iftok |
| 644716 | 1 | 123 | left brace | |
| 649929 | 5 | 1 | parameter | |
| 651673 | 2 | 125 | right brace | |
| 650614 | 1 | 123 | left brace | |
| 644705 | 5 | 2 | parameter | |
| 648373 | 2 | 125 | right brace | |
| 648338 | 114 | 0 | expand after | expandafter |
| 651711 | 125 | 0 | call | firstoftwoarguments |
| 651654 | 120 | 3 | if test | else |
| 651653 | 114 | 0 | expand after | expandafter |
| 651650 | 125 | 0 | call | secondoftwoarguments |
| 651658 | 120 | 2 | if test | fi |

## Example 7: nothing

```
\luatokentable\relax
```

| control sequence: relax |
|---|
| <no tokens> |

## Example 8: Hashes

```
\edef\foo#1#2{(#1)(\letterhash)(#2)}  \luatokentable\foo
```

**control sequence: foo**

| | | | | | |
|---|---|---|---|---|---|
| 651327 | 13 | 1 | argument | | |
| 648347 | 13 | 2 | argument | | |
| 651826 | 14 | 0 | end match | | |
| 651844 | 12 | 40 | other char | ( | U+00028 |
| 651839 | 5 | 1 | parameter | | |
| 651862 | 12 | 41 | other char | ) | U+00029 |
| 651837 | 12 | 40 | other char | ( | U+00028 |
| 650556 | 12 | 35 | other char | # | U+00023 |
| 651279 | 12 | 41 | other char | ) | U+00029 |
| 650404 | 12 | 40 | other char | ( | U+00028 |
| 650948 | 5 | 2 | parameter | | |
| 649394 | 12 | 41 | other char | ) | U+00029 |

# Example 9: Nesting

```
\def\foo#1{\def\foo##1{(#1)(##1)}}   \luatokentable\foo
```

**control sequence: foo**

| | | | | | | |
|---|---|---|---|---|---|---|
| 652157 | 13 | 1 | argument | | | |
| 648413 | 14 | 0 | end match | | | |
| 651407 | 109 | 0 | def | | | def |
| 651747 | 125 | 0 | call | | | foo |
| 651388 | 6 | 35 | mac param | | | |
| 651387 | 12 | 49 | other char | 1 | U+00031 | |
| 651414 | 1 | 123 | left brace | | | |
| 644654 | 12 | 40 | other char | ( | U+00028 | |
| 651849 | 5 | 1 | parameter | | | |
| 651389 | 12 | 41 | other char | ) | U+00029 | |
| 596523 | 12 | 40 | other char | ( | U+00028 | |
| 648286 | 6 | 35 | mac param | | | |
| 651510 | 12 | 49 | other char | 1 | U+00031 | |
| 652057 | 12 | 41 | other char | ) | U+00029 | |
| 647947 | 2 | 125 | right brace | | | |