

The `morewrites` package: Always room for a new `\write`

Bruno Le Floch

2017/04/20

Contents

1	morewrites documentation	2
1.1	Commands defined or altered by <code>morewrites</code>	2
1.2	Known deficiencies and open questions	3
2	morewrites implementation	3
2.1	Overview of relevant <code>TEX</code> facts	3
2.2	Preliminaries	5
2.2.1	Copying some commands	5
2.2.2	Variants	5
2.2.3	Variables	6
2.2.4	Helpers for auxiliary file	8
2.2.5	Parsing and other helpers	9
2.3	Writing	11
2.3.1	Redefining <code>\immediate</code>	11
2.3.2	Immediate actions	12
2.3.3	Delayed actions	15
2.3.4	Shipout business	16
2.3.5	Hook at the very end	19
2.4	Redefining commands	19
2.4.1	Modified <code>\newwrite</code>	19
2.5	User commands and keys	20
	Index	22

1 morewrites documentation

This L^AT_EX package is a solution for the error “no room for a new `\write`”, which occurs when a document reserves too many streams to write data to various auxiliary files. It is in principle possible to rewrite other packages so that they are less greedy on resources, but that is often unpractical for the end-user. Instead, `morewrites` hooks at the lowest level (T_EX primitives).

Simply add the line `\usepackage{morewrites}` near the beginning of your L^AT_EX file’s preamble: the “no room for a new `\write`” error should vanish. If it does not, please contact me so that I can correct the problem. This can be done by posting a question on the tex.stackexchange.com question and answers website, logging an issue on GitHub (<https://github.com/blefloch/latex-morewrites>), or emailing me a minimal file showing the problem.

Notes.

- This package loads the `expl3` package, hence the `l3kernel` bundle needs to be up to date.
- This package uses an auxiliary file, `\jobname.mw`, which can safely be deleted. Versions from 2015 and later will only use the auxiliary file if it is originally empty, to avoid destroying data (such as `.mw` files used by Maple). This means that `.mw` files generated by versions before 2015 should be deleted by hand.
- LuaT_EX allows 128 `\write` streams, so this package does nothing (with a warning) when used with LuaT_EX.

1.1 Commands defined or altered by `morewrites`

<hr/> <code>\morewritessetup</code> <hr/> <small>New: 2014-07-26</small>	<code>\morewritessetup {<key-value list>}</code> Sets the options described by the <code><key-value list></code> .
<hr/> <code>allocate</code> <hr/> <small>New: 2017-04-10</small>	<code>\morewritessetup { allocate = <integer> }</code> Sets to (at least) <code><integer></code> the number of <code>\write</code> streams allocated to the inner workings of <code>morewrites</code> . By default this is zero but increasing this value to 10 (or so) may speed up <code>morewrites</code> .
<hr/> <code>file</code> <hr/> <small>New: 2014-07-26 Updated: 2015-08-01</small>	<code>\morewritessetup { file = <file name> }</code> Sets (globally) the name of the file which will be used by internal processes of <code>morewrites</code> . The file name is <code>\jobname.mw</code> by default (technically, <code>\c_sys_jobname_str.mw</code>). Contrarily to earlier versions of <code>morewrites</code> non-empty files will not be overwritten; this design choice may lead to unwanted <code>.mw</code> files remaining.
<hr/> <code>\newwrite</code> <hr/> <small>Updated: 2015-08-01</small>	This macro is redefined by <code>morewrites</code> . Since <code>morewrites</code> allows more than 16 write streams, it removes the corresponding restrictions in <code>\newwrite</code> .

T_EXhackers note: The revised `\newwrite` allocate stream numbers starting at 19. This might break some code that expects stream numbers to be less than 16.

`\immediate`

Updated: 2015-08-01

`\openout`
`\write`
`\closeout`

Updated: 2017-04-20

This primitive is altered by `morewrites`, to detect a following `\write` or `\openout` or `\closeout` and perform the appropriate action.

These three primitives are altered by `morewrites` so that they accept stream numbers outside the normal range $[0, 15]$ and open/write/close files as appropriate.

TeXhackers note: System calls using `\write18` are detected and forwarded to the engine.

`\shipout`

This primitive is altered by `morewrites` to ensure that delayed `\openout`, `\write` and `\closeout` commands are performed at `\shipout` time, and in the correct order.

1.2 Known deficiencies and open questions

See the bug tracker <https://github.com/blefloch/latex-morewrites/issues/> for a list of issues with `morewrites`.

The package code is not good `expl3` code. *Do not take this package as an example of how to code with `expl3`; go and see Joseph Wright's `siunitx` instead.* It uses `\...:D` primitives directly (the `:D` stands for “do not use”). This is unavoidable in order to hook into the primitives `\immediate`, `\write`, *etc.* and to keep a very strong control on what every command does.

2 morewrites implementation

```
<*package>
1 \RequirePackage {expl3} [2017/03/18]
2 \RequirePackage {primargs} [2017/04/10]
3 \ProvidesExplPackage
4   {morewrites} {2017/04/20} {} {Always room for a new write}
   Quit early under LuaTeX.
5 \sys_if_engine_luatex:T
6   {
7     \cs_new_protected:Npn \morewritessetup #1 { }
8     \msg_new:nnn { morewrites } { luatex }
9     { The~morewrites~package~is~unnecessary~in~LuaTeX. }
10    \msg_warning:nn { morewrites } { luatex }
11    \tex_endinput:D
12  }%
13 <@@=morewrites>
```

2.1 Overview of relevant TeX facts

The aim of the `morewrites` package is to lift TeX's restriction of only having 16 files open for writing at the same time. This requires patching the primitives `\immediate`, `\openout`, `\write`, `\closeout`, and `\shipout`, and the macro `\newwrite` present in plain TeX and L^ATeX 2_ε.

Note that doing the same for `\read` streams is impossible due to the `\ifeof` primitive: that primitive cannot be replaced by a macro without breaking nesting of conditionals.

The `morewrites` package should be loaded as early as possible, so that any package loaded later uses the redefined macros instead of the primitives. However, the format (plain `TEX` or `LATEX 2ε`) and the `expl3` programming language are always loaded before `morewrites`, and their interaction must be carefully monitored.

Henceforth, “`TEX` stream” will refer to stream numbers in the range $[0, 15]$ provided to `TEX`’s write primitives, while “user stream” will denote stream numbers in $[0, 15] \cup [19, \infty)$ manipulated by the redefined `\openout`, `\write`, `\closeout`, and `\newwrite`. A user stream in $[0, 15]$ (reserved by `LATEX 2ε` or allocated by `expl3`) is mapped to the same `TEX` stream number, while a user stream in $[19, \infty)$ is mapped to a `TEX` stream according to the property list (with integer keys and values) `\l_morewrites_write_prop`. Stream numbers 16, 17 and 18 are unused because `\write16` is often used to write to the terminal, and `\write18` sends its argument to a shell.

The primitives `\openout`, `\write`, and `\closeout` expect to be followed by an *integer*, normally in the range $[0, 15]$, then some further arguments.

```

\openout <integer> <equals> <file name>
\write <integer> <filler> <general text>
\closeout <integer>

```

All of the primitives above perform full expansion of all tokens when looking for their operands.

- *integer* denotes an integer in any form that `TEX` accepts as the right-hand side of a primitive integer assignment of the form `\count0=<integer>`;
- *equals* is an arbitrary (optional) number of explicit or implicit space characters, an optional explicit equal sign of category other, and further (optional) explicit or implicit space characters;
- *file name* is an arbitrary sequence of explicit or implicit characters with arbitrary category codes (except active characters, which are expanded before reaching `TEX`’s mouth), ending either with a space character (character code 32, arbitrary non-active category code, explicit or implicit), which is removed, or with a non-expandable token, with some care needed for the case of a `\notexpanded`: expandable token;
- *filler* is an arbitrary combination of tokens whose meaning is `\relax` or whose category code is 10;
- *general text* is formed of braced tokens, starting with an explicit or implicit begin-group character, and ending with the matching explicit end-group character (both with any character code), with an equal number of explicit begin-group and end-group characters in between: this is precisely the right-hand side of an assignment of the form `\toks0=<general text>`.

The `morewrites` package redefines these three control sequences to expect a user stream number rather than a `TEX` stream number as the *integer*, then map such a user stream to a `TEX` stream to call the primitive with the appropriate argument. The primitive `\immediate` must also be redefined to detect `\openout`, `\write`, and `\closeout`

and make them immediate, while still working with other primitives that can be made immediate. Finally, `\newwrite` must be patched to allocate stream numbers beyond 15.

A few comments on the behaviour of primitives concerning the $\langle integer \rangle$ (TeX stream). The `\openout` primitive trigger errors if the $\langle integer \rangle$ is not in $[0, 15]$. The primitive `\write` outputs to the log if the $\langle integer \rangle$ is negative, and to the terminal if the TeX stream is closed or greater than 15, with the exception of `\write18` which runs code in a shell. The `\closeout` primitive triggers an error if the $\langle integer \rangle$ is not in $[0, 15]$ and silently do nothing if the TeX stream is not open, with the exception of `\closeout18` which causes a segfault at least in some versions.

By default, `\openout`, `\write` and `\closeout` are recorded in a whatsit node in the current list, and will be performed when the box containing the whatsit node is sent to the final pdf, *i.e.*, at “shipout” time. In particular, the $\langle general\ text \rangle$ for the `\write` primitive is expanded at shipout time. This behaviour may be modified by putting `\immediate` before any of these three primitives to force TeX to perform the action immediately instead of recording it in a whatsit node.

Since the `\openout`, `\write`, and `\closeout` primitives operate at `\shipout` time, we will have to hook into this primitive too. It expects to be followed by a box specification, for instance `\box\langle integer \rangle` or `\hbox{\langle material to typeset \rangle}`.

Finally, the `\newwrite` macro expects one token as its argument, and defines this token (with `\chardef`) to be an integer corresponding to the first available (TeX) write stream. This must be extended to allocate higher (user) streams.

2.2 Preliminaries

2.2.1 Copying some commands

```
\_morewrites_tex_immediate:w Aliases for the write-related primitives, to avoid having :D throughout the code.
```

```
\_morewrites_tex_openout:w 14 \cs_new_eq:NN \_morewrites_tex_immediate:w \tex_immediate:D
\_morewrites_tex_write:w    15 \cs_new_eq:NN \_morewrites_tex_openout:w \tex_openout:D
\_morewrites_tex_closeout:w 16 \cs_new_eq:NN \_morewrites_tex_write:w \tex_write:D
                             17 \cs_new_eq:NN \_morewrites_tex_closeout:w \tex_closeout:D
```

(End definition for `_morewrites_tex_immediate:w` and others.)

```
\_morewrites_tex_newwrite:N Copy \newwrite but making sure that it is not \outer. This copy will not be affected by redefinitions of \newwrite later on.
```

```
18 \exp_args:Nnf \cs_new_protected:Npn \_morewrites_tex_newwrite:N
19   { \exp_args:NNc \exp_after:wN \exp_stop_f: { newwrite } }
```

(End definition for `_morewrites_tex_newwrite:N`.)

2.2.2 Variants

```
\prop_gpop:NVNT We need these variants.
```

```
\prop_gput:NVx 20 \cs_generate_variant:Nn \prop_gpop:NnNT { NV }
\tl_gput_right:Nv 21 \cs_generate_variant:Nn \prop_gput:Nnn { NVx }
                22 \cs_generate_variant:Nn \tl_gput_right:Nn { Nv }
```

(End definition for `\prop_gpop:NVNT`, `\prop_gput:NVx`, and `\tl_gput_right:Nv`.)

2.2.3 Variables

- `\l__morewrites_internal_tl` Used for temporary scratch purposes.
23 `\tl_new:N \l__morewrites_internal_tl`
(End definition for `\l__morewrites_internal_tl`.)
- `__morewrites_tmp:w` Used for temporary definitions.
24 `\cs_new_eq:NN __morewrites_tmp:w ?`
(End definition for `__morewrites_tmp:w`.)
- `\g__morewrites_later_int` The integer `\g__morewrites_later_int` labels the various non-immediate operations in the order in which they appear in the source. We can never reuse a number because there is no way to know if a whatsit was recorded in a box register, which could be reused in a shipped-out box:

```
\vbox_set:Nn \l_my_box
{ \iow_shipout_x:Nn \c_term_iow {<text>} } \shipout \copy \l_my_box
\shipout \copy \l_my_box
```

will print `<text>` to the terminal twice.
25 `\int_new:N \g__morewrites_later_int`
(End definition for `\g__morewrites_later_int`.)
- `\g__morewrites_write_seq` Keep track of T_EX stream numbers managed by `morewrites` that are currently not in use as user streams.
26 `\seq_new:N \g__morewrites_write_seq`
(End definition for `\g__morewrites_write_seq`.)
- `\g__morewrites_write_prop` Map user streams to T_EX streams.
27 `\prop_new:N \g__morewrites_write_prop`
(End definition for `\g__morewrites_write_prop`.)
- `\g__morewrites_write_file_prop` Map user streams with no associated T_EX streams to file names.
28 `\prop_new:N \g__morewrites_write_file_prop`
(End definition for `\g__morewrites_write_file_prop`.)
- `\l__morewrites_code_tl` Stores the code to run after finding a user stream, in `__morewrites_get_user:n`.
29 `\tl_new:N \l__morewrites_code_tl`
(End definition for `\l__morewrites_code_tl`.)
- `\l__morewrites_user_int` The user stream number following redefined primitives is stored in `\l__morewrites_user_int` (see `__morewrites_get_user:N`). The corresponding T_EX stream number is eventually stored in `\l__morewrites_tstr_tl` (a token list).
30 `\int_new:N \l__morewrites_user_int`
31 `\tl_new:N \l__morewrites_tstr_tl`
(End definition for `\l__morewrites_user_int` and `\l__morewrites_tstr_tl`.)

`\l__morewrites_tstr_token` This token is given as an argument to `__morewrites_tex_newwrite:N`.

```
32 \cs_new_eq:NN \l__morewrites_tstr_token ?
```

(End definition for `\l__morewrites_tstr_token`.)

`\s__morewrites` A recognizable version of `\scan_stop:`. This is inspired by¹ scan marks (see the `l3quark` module of L^AT_EX3), but `__scan_new:N` is not used directly, since it is currently internal to L^AT_EX3.

```
33 \cs_new_eq:NN \s__morewrites \scan_stop:
```

(End definition for `\s__morewrites`.)

`\g__morewrites_iow` The expansion that `\write` performs is impossible to emulate (in X_YL^AT_EX at least) with anything else than `\write`. We will write on the stream `\g__morewrites_iow` to the file `\g__morewrites_tmp_file_tl` and read back from it in the stream `\g__morewrites_ior` for things to work properly. Unfortunately, this means that the file is repeatedly opened and closed, leaving a trace of that in the log.

```
34 \newwrite \g__morewrites_iow
```

```
35 \newread \g__morewrites_ior
```

(End definition for `\g__morewrites_iow` and `\g__morewrites_ior`.)

`\g__morewrites_tmp_file_tl` Temporary file used to do the correct expansion for each `\write`. Boolean indicating whether we have already checked that the file can be used by `morewrites:` before using a file, the `morewrites` package now checks it is empty, so as to avoid clobbering user data.

```
36 \tl_new:N \g__morewrites_tmp_file_tl
```

```
37 \bool_new:N \g__morewrites_tmp_file_bool
```

```
38 \bool_gset_false:N \g__morewrites_tmp_file_bool
```

(End definition for `\g__morewrites_tmp_file_tl` and `\g__morewrites_tmp_file_bool`.)

`\g__morewrites_group_level_int` The group level when `\shipout` is called: this is used to distinguish between explicit boxes and box registers.

```
39 \int_new:N \g__morewrites_group_level_int
```

(End definition for `\g__morewrites_group_level_int`.)

`\g__morewrites_shipout_box` The page to be shipped out.

```
40 \box_new:N \g__morewrites_shipout_box
```

(End definition for `\g__morewrites_shipout_box`.)

¹Historically, this might have happened the other way around, since the author of this package is also on the L^AT_EX3 Team.

2.2.4 Helpers for auxiliary file

`_morewrites_set_file:n` Sets `\g__morewrites_tmp_file_tl` to the given value (initially `\c_sys_jobname_str.mw`). Mark that the file has not been checked.

```

41 \cs_new_protected:Npn \_morewrites_set_file:n #1
42 {
43   \bool_gset_false:N \g__morewrites_tmp_file_bool
44   \tl_gset:Nn \g__morewrites_tmp_file_tl {#1}
45 }

```

(End definition for `_morewrites_set_file:n`.)

`_morewrites_empty_file:n` Empties the file `\g__morewrites_tmp_file_tl` by opening it and closing it right away. This is used when performing `\immediate \openout`. It is also used to ensure the file used by `morewrites` is left empty. We do this every time the auxiliary file is used, in case that run ends with an error mid-document.

```

46 \cs_new_protected:Npn \_morewrites_empty_file:n #1
47 {
48   \_morewrites_tex_immediate:w \_morewrites_tex_openout:w
49   \g__morewrites_iow = #1 \scan_stop:
50   \_morewrites_tex_immediate:w \_morewrites_tex_closeout:w
51   \g__morewrites_iow
52 }

```

(End definition for `_morewrites_empty_file:n`.)

`_morewrites_if_file_trivial:nTF` True if the file does not exist, or if it is empty. Only the TF variant is defined. We set `_morewrites_tmp:w` to `\prg_return_true:` or `\prg_return_false:` within the group and use it after cleaning up. The first `eof` test is `true` if the file does not exist. Then we read one line, the second `eof` test is `true` if the file was empty (it is `false` if the file contained anything, even a single space).

```

53 \prg_new_conditional:Npnn \_morewrites_if_file_trivial:n #1 { TF }
54 {
55   \group_begin:
56   \tex_openin:D \g__morewrites_iow = #1 \scan_stop:
57   \if_eof:w \g__morewrites_iow
58     \cs_gset_eq:NN \_morewrites_tmp:w \prg_return_true:
59   \else:
60     \int_set:Nn \tex_endlinechar:D { -1 }
61     \etex_readline:D \g__morewrites_iow to \l__morewrites_internal_tl
62     \if_eof:w \g__morewrites_iow
63       \cs_gset_eq:NN \_morewrites_tmp:w \prg_return_true:
64     \else:
65       \cs_gset_eq:NN \_morewrites_tmp:w \prg_return_false:
66     \fi:
67   \fi:
68   \tex_closein:D \g__morewrites_iow
69   \group_end:
70   \_morewrites_tmp:w
71 }

```

(End definition for `_morewrites_if_file_trivial:nTF`.)

`__morewrites_chk_file:` Check that the file `\g__morewrites_tmp_file_tl` does not exist or is blank. If not, try the file name obtained by adding `.mw`. This avoids clobbering files that the user would not want to lose.

```

72 \cs_new_protected:Npn \__morewrites_chk_file:
73 {
74   \__morewrites_if_file_trivial:nTF { \g__morewrites_tmp_file_tl }
75   { \bool_gset_true:N \g__morewrites_tmp_file_bool }
76   {
77     \msg_warning:nxxx { morewrites } { file-exists }
78     { \g__morewrites_tmp_file_tl }
79     { \g__morewrites_tmp_file_tl .mw }
80     \tl_gput_right:Nn \g__morewrites_tmp_file_tl { .mw }
81     \__morewrites_chk_file:
82   }
83 }
84 \msg_new:nnnn { morewrites } { file-exists }
85 { File~'#1'~exists,~using~'#2'~instead. }
86 {
87   The~file~'#1'~exists~and~was~not~created~by~this~version~of~the~
88   'morewrites'~package.~Please~move~or~delete~that~file,~or~provide~
89   another~file~name~by~adding
90   \\ \\
91   \iow_indent:n { \iow_char:N\morewritessetup~{~file~~~other-name~} }
92   \\ \\
93   to~your~source~file.~In~the~meantime,~the~file~'#2'~will~be~used.
94 }

```

(End definition for `__morewrites_chk_file:`.)

2.2.5 Parsing and other helpers

`__morewrites_equals_file:N` Most of the parsing for primitive arguments is done using `primargs`, except for one case we care about: after its $\langle number \rangle$ argument, the `\openout` primitive expects an $\langle equals \rangle$ (optional spaces and =) and a $\langle file name \rangle$.

```

95 \cs_new_protected:Npn \__morewrites_equals_file:N #1
96 {
97   \group_begin:
98   \tex_aftergroup:D \primargs_get_file_name:N
99   \tex_aftergroup:D #1
100   \primargs_remove_equals:N \group_end:
101 }

```

(End definition for `__morewrites_equals_file:N`.)

`__morewrites_get_user:n` `primargs` commands only take N-type arguments, but we often need to find an integer, save it in `\l__morewrites_user_int`, and run some code #1. This is analogous to `\primargs_get_number:N`.

```

102 \cs_new_protected:Npn \__morewrites_get_user:n #1
103 {
104   \tl_set:Nn \l__morewrites_code_tl {#1}
105   \tex_afterassignment:D \l__morewrites_code_tl
106   \l__morewrites_user_int =
107 }

```

(End definition for `_morewrites_get_user:n`.)

`_morewrites_user_to_tstr:NTF` The goal is to go from a user stream `\l__morewrites_user_int` to a TeX stream `\l__morewrites_tstr_tl` (it defaults to the user stream). Streams less than 19 are not managed by `morewrites`: actual TeX streams in $[0, 15]$; negative for writing to `log`; 16, 17 for writing to terminal; 18 for shell escape. Larger stream numbers are looked up in the property list `#1`, namely `\g__morewrites_write_prop`. If present, use the corresponding value as the TeX stream, otherwise run the `false` branch.

```

108 \cs_new_protected:Npn \_morewrites_user_to_tstr:NTF #1
109   {
110     \tl_set:NV \l__morewrites_tstr_tl \l__morewrites_user_int
111     \int_compare:nNnTF { \l__morewrites_user_int } < { 19 }
112       { \use_i:nn }
113       { \prop_get:NVNTF #1 \l__morewrites_user_int \l__morewrites_tstr_tl }
114   }

```

(End definition for `_morewrites_user_to_tstr:NTF`.)

`\l__morewrites_collect_next_int`
`_morewrites_collect:x`
`_morewrites_collect_aux:Nn`
`_morewrites_collect_aux:cf`
`_morewrites_collect_gput_right:N`
`_morewrites_collect_gput_right:c`

When encountering very large `\write` statements we may need to collect many lines. This can easily become an $O(n^2)$ task, and here we make sure that it remains around $O(n \log n)$, with a large constant unfortunately. Each of the token lists `\l__morewrites_#k$tl` is empty or contains 2^k lines. As lines accumulate, they move to token lists with larger values of k , and eventually all are combined. The integer `\l__morewrites_collect_next_int` is (one plus) the maximal k among non-empty token lists.

```

115 \int_new:N \l__morewrites_collect_next_int
116 \cs_new_protected:Npn \_morewrites_collect:x #1
117   {
118     \tl_set:Nx \l__morewrites_internal_tl {#1}
119     \_morewrites_collect_aux:cf { l__morewrites_0_tl } { 1 }
120   }
121 \cs_new_protected:Npn \_morewrites_collect_aux:Nn #1#2
122   {
123     \int_compare:nNnTF {#2} > \l__morewrites_collect_next_int
124       {
125         \tl_clear_new:N #1
126         \int_set:Nn \l__morewrites_collect_next_int {#2}
127       }
128     \tl_if_empty:NTF #1
129       { \tl_set_eq:NN #1 \l__morewrites_internal_tl }
130       {
131         \tl_put_left:No \l__morewrites_internal_tl {#1}
132         \tl_clear:N #1
133         \_morewrites_collect_aux:cf { l__morewrites_#2_tl }
134           { \int_eval:n { #2 + 1 } }
135       }
136   }
137 \cs_generate_variant:Nn \_morewrites_collect_aux:Nn { cf }
138 \cs_new_protected:Npn \_morewrites_collect_gput_right:N #1
139   {
140     \int_compare:nNnF \l__morewrites_collect_next_int = 0
141       {
142         \int_decr:N \l__morewrites_collect_next_int

```

```

143     \tl_gput_right:Nv #1
144     {
145       l__morewrites_
146       \int_use:N \l__morewrites_collect_next_int
147       _tl
148     }
149     \__morewrites_collect_gput_right:N #1
150   }
151 }
152 \cs_generate_variant:Nn \__morewrites_collect_gput_right:N { c }

```

(End definition for `\l__morewrites_collect_next_int` and others.)

`__morewrites_user_tl_name:n` The name of a global token list variable holding the text of a given user stream.

```

153 \cs_new:Npn \__morewrites_user_tl_name:n #1
154 { g__morewrites_iow_ \int_eval:n {#1} _tl }

```

(End definition for `__morewrites_user_tl_name:n`.)

2.3 Writing

We can hold on to material while a file is being written and only write it in one go once the file closes, to avoid using a stream throughout.

At any given time, each user stream may point to an open T_EX stream, given in `\g__morewrites_write_prop`, or may point to a token list that will eventually be written to a file whose file name is stored in `\g__morewrites_write_file_prop`, or may be closed.

When a user stream points to a token list rather than a T_EX stream, any material to be written must be written to our temporary file and read back in to apply the same expansion as `\write` does.

Another difficulty is that users may mix immediate and non-immediate operations. The biggest difficulty comes from the possibility of copying boxes containing delayed actions. If we ever produced a whatsit `\write(number){text}` then the T_EX stream `<number>` would have to be reserved forever, as as copies of the box containing this delayed actions may be shipped out at any later point in the document.

Each delayed action is thus saved in a separate numbered token list and `\write\g__morewrites_iow{number}` is inserted instead of the delayed action. At each `\shipout`, the stream `\g__morewrites_iow` is opened, to catch the `<number>` of each action that should be performed at this `\shipout`.

2.3.1 Redefining `\immediate`

To accomodate the `\immediate` primitive, our versions of `\openout`, `\write` and `\closeout` will take the form

```

\__morewrites \use_i:nn {<code for delayed action>}
{<code for immediate action>}
<further code>

```

The leading `\s__morewrites` allows the redefined `\immediate` to detect these redefined primitives, and to run the `<code for immediate action>` instead of the `<code for delayed action>` which is run by default. In both cases, any `<further code>` is run.

`__morewrites_immediate:w` T_EX's `\immediate` primitive raises a flag which is cancelled after T_EX sees a non-expandable token. We use `\primargs_read_x_token:N` to find the next non-expandable token then test for `\openout`, `\write`, and `\closeout`. More precisely we test for the marker `\s__morewrites` and run the appropriate code as described above. Otherwise we call the primitive, for cases where the next token is `\pdfobj` or similar. This code performs too much expansion for some nonsensical uses of `\noexpand` after `\immediate`.

```

155 \cs_new_protected:Npn \__morewrites_immediate:w
156   { \primargs_read_x_token:N \__morewrites_immediate_auxii: }
157 \cs_new_protected:Npn \__morewrites_immediate_auxii:
158   {
159     \token_if_eq_meaning:NNTF \g_primargs_token \s__morewrites
160     { \__morewrites_immediate_auxiii:N }
161     { \__morewrites_tex_immediate:w }
162   }
163 \cs_new_protected:Npn \__morewrites_immediate_auxiii:N #1
164   { \str_if_eq:nnTF { #1 } { \s__morewrites } { \use_iii:nnn } { #1 } }

```

(End definition for `__morewrites_immediate:w`, `__morewrites_immediate_auxii:`, and `__morewrites_immediate_auxiii:N`.)

2.3.2 Immediate actions

The `\openout`, `\write`, and `\closeout` primitive can be either delayed or immediate. In all cases they begin by looking for a user stream. Here, we implement the immediate versions only.

`__morewrites_closeout:w` In the immediate case `__morewrites_closeout_now:`, there are three cases. The stream may point to a T_EX stream, in which case it is closed, removed from `\g__morewrites_write_prop`, and put back in the list of usable streams. The stream may point to a token list, in which case that token list should be written to the appropriate file. The stream may be closed, in which case nothing happens. The auxiliary `__morewrites_closeout_now:nn` writes the material collected so far for a given user stream #1 to the file #2. This uses the T_EX stream `\g__morewrites_iow`. The token list consists of multiple `\immediate \write \g__morewrites_iow {{text}}` statements because that is the only safe way to obtain new lines. We do not remove the stream/file pair from `\g__morewrites_write_file_prop`.

```

165 \cs_new_protected:Npn \__morewrites_closeout:w
166   {
167     \s__morewrites
168     \use_i:nn
169     { \__morewrites_get_user:n { \__morewrites_closeout_later: } }
170     { \__morewrites_get_user:n { \__morewrites_closeout_now: } }
171   }
172 \cs_new_protected:Npn \__morewrites_closeout_now:
173   {
174     \__morewrites_user_to_tstr:NTF \g__morewrites_write_prop
175     {
176       \__morewrites_tex_immediate:w \__morewrites_tex_closeout:w \l__morewrites_tstr_tl \ex
177       \int_compare:nNnF { \l__morewrites_tstr_tl } = { \l__morewrites_user_int }
178       {
179         \prop_gremove:NV \g__morewrites_write_prop \l__morewrites_user_int
180         \seq_gput_left:NV \g__morewrites_write_seq \l__morewrites_tstr_tl
181       }
182     }

```

```

182     }
183     {
184         \prop_gpop:NVNT \g__morewrites_write_file_prop \l__morewrites_user_int \l__morewrites_
185         { \__morewrites_closeout_now:nn { \l__morewrites_user_int } { \l__morewrites_intern
186     }
187 }
188 \cs_new_protected:Npn \__morewrites_closeout_now:nn #1#2
189 {
190     \__morewrites_tex_immediate:w \__morewrites_tex_openout:w \g__morewrites_iow = #2 \scan_s
191     \group_begin:
192         \int_set:Nn \tex_newlinechar:D { -1 }
193         \tl_use:c { \__morewrites_user_tl_name:n {#1} }
194         \tl_gclear:c { \__morewrites_user_tl_name:n {#1} }
195     \group_end:
196     \__morewrites_tex_immediate:w \__morewrites_tex_closeout:w \g__morewrites_iow
197 }

```

(End definition for `__morewrites_closeout:w`, `__morewrites_closeout_now:`, and `__morewrites_closeout_now:nn`.)

`__morewrites_openout:w`
`__morewrites_openout_now:n`

In the immediate case find a file name, then allocate a TeX stream if possible, and otherwise point the user stream to a token list. In all cases, close the stream to avoid losing any material that TeX would have written, and empty the file by opening and closing it (actually that's done automatically by the primitive).

```

198 \cs_new_protected:Npn \__morewrites_openout:w
199 {
200     \s__morewrites
201     \use_i:nn
202     { \__morewrites_get_user:n { \__morewrites_openout_later:w } }
203     { \__morewrites_get_user:n { \__morewrites_equals_file:N \__morewrites_openout_now:n } }
204 }
205 \cs_new_protected:Npn \__morewrites_openout_now:n #1
206 {
207     \__morewrites_closeout_now:
208     \int_compare:nNnTF { \l__morewrites_user_int } < { 19 }
209     {
210         \__morewrites_tex_immediate:w \__morewrites_tex_openout:w \l__morewrites_user_int
211         = \tl_to_str:n {#1} \scan_stop:
212     }
213     {
214         \seq_gpop:NNTF \g__morewrites_write_seq \l__morewrites_tstr_tl
215         {
216             \prop_gput:NVV \g__morewrites_write_prop \l__morewrites_user_int \l__morewrites_t
217             \__morewrites_tex_immediate:w \__morewrites_tex_openout:w \l__morewrites_tstr_tl
218             = \tl_to_str:n {#1} \scan_stop:
219         }
220         {
221             \__morewrites_empty_file:n {#1}
222             \prop_gput:NVx \g__morewrites_write_file_prop \l__morewrites_user_int
223             { \tl_to_str:n {#1} }
224             \tl_gclear_new:c { \__morewrites_user_tl_name:n { \l__morewrites_user_int } }
225         }
226     }
227 }

```

(End definition for `__morewrites_openout:w` and `__morewrites_openout_now:n`.)

`__morewrites_write:w` In the immediate case we use `__morewrites_write_now_open:n` if the stream points to a token list, and otherwise use the primitive, with the dummy stream 16 if closed (the text is then written to the terminal).

```

228 \cs_new_protected:Npn \__morewrites_write:w
229   {
230     \s__morewrites
231     \use_i:nn
232     { \__morewrites_get_user:n { \__morewrites_write_later:w } }
233     { \__morewrites_get_user:n { \__morewrites_write_now:w } }
234   }
235 \cs_new_protected:Npn \__morewrites_write_now:w
236   {
237     \__morewrites_user_to_tstr:NTF \g__morewrites_write_prop
238     { \__morewrites_tex_immediate:w \__morewrites_tex_write:w \l__morewrites_tstr_tl \exp_s
239     { \primargs_get_general_text:N \__morewrites_write_now:n }
240   }
241 \cs_new_protected:Npn \__morewrites_write_now:n
242   {
243     \prop_get:NVNTF \g__morewrites_write_file_prop \l__morewrites_user_int \l__morewrites_int
244     { \__morewrites_write_now_open:n }
245     { \__morewrites_tex_immediate:w \__morewrites_tex_write:w 16 }
246   }

```

(End definition for `__morewrites_write:w`, `__morewrites_write_now:w`, and `__morewrites_write_now:n`.)

`__morewrites_write_now_open:n` Only `\write` itself can emulate how `\write` expands tokens, because `#` don't have to be doubled, and because the `\newlinechar` has to be changed to new lines. Hence, we start by writing `#1` to a file (after making sure we are allowed to alter it), yielding some lines. The lines are then read one at a time using ϵ -TeX's `\readline` with `\endlinechar` set to `-1` to avoid spurious characters. Each line becomes a `\immediate \write` statement added to a token list whose name is constructed using `__morewrites_user_tl_name:n`. This token list will be called when it is time to actually write to the file. At that time, `\newlinechar` will be `-1`, so that writing each line will produce no extra line.

```

247 \cs_new_protected:Npn \__morewrites_write_now_open:n #1
248   {
249     \bool_if:NF \g__morewrites_tmp_file_bool { \__morewrites_chk_file: }
250     \__morewrites_tex_immediate:w \__morewrites_tex_openout:w
251     \g__morewrites_iow = \g__morewrites_tmp_file_tl \scan_stop:
252     \__morewrites_tex_immediate:w \__morewrites_tex_write:w
253     \g__morewrites_iow {#1}
254     \__morewrites_tex_immediate:w \__morewrites_tex_closeout:w
255     \g__morewrites_iow
256     \group_begin:
257     \int_set:Nn \tex_endlinechar:D { -1 }
258     \tex_openin:D \g__morewrites_ior = \g__morewrites_tmp_file_tl \scan_stop:
259     \__morewrites_write_now_loop:
260     \tex_closein:D \g__morewrites_ior
261     \__morewrites_collect_gput_right:c
262     { \__morewrites_user_tl_name:n { \l__morewrites_user_int } }
263     \group_end:

```

```

264   \__morewrites_empty_file:n { \g__morewrites_tmp_file_tl }
265 }
266 \cs_new_protected:Npn \__morewrites_write_now_loop:
267 {
268   \etex_readline:D \g__morewrites_ior to \l__morewrites_internal_tl
269   \ior_if_eof:NF \g__morewrites_ior
270   {
271     \__morewrites_collect:x
272     {
273       \__morewrites_tex_immediate:w \__morewrites_tex_write:w
274       \g__morewrites_iow { \l__morewrites_internal_tl }
275     }
276     \__morewrites_write_now_loop:
277   }
278 }

```

(End definition for `__morewrites_write_now_open:n` and `__morewrites_write_now_loop:.`)

2.3.3 Delayed actions

`__morewrites_later:n` Store the action to be done at shipout in a token list, and non-immediately write the label `\g__morewrites_later_int` of the output operation to the temporary file.

```

279 \cs_new_protected:Npn \__morewrites_later:n #1
280 {
281   \int_gincr:N \g__morewrites_later_int
282   \tl_const:cx
283   {
284     c__morewrites_later_
285     \int_use:N \g__morewrites_later_int
286     _tl
287   }
288   {
289     \int_set:Nn \exp_not:N \l__morewrites_user_int
290     { \exp_not:V \l__morewrites_user_int }
291     \exp_not:n {#1}
292   }
293   \exp_args:NNx \__morewrites_tex_write:w \g__morewrites_iow
294   { '( \int_use:N \g__morewrites_later_int ) }
295 }
296 \cs_new_protected:Npn \__morewrites_later_do:n #1
297 { \tl_use:c { c__morewrites_later_ \int_eval:n {#1} _tl } }

```

(End definition for `__morewrites_later:n` and `__morewrites_later_do:n.`)

`__morewrites_closeout_later:` If the user stream is a TeX stream, use the primitive, otherwise save `__morewrites_closeout_now:` for later.

```

298 \cs_new_protected:Npn \__morewrites_closeout_later:
299 {
300   \int_compare:nNnTF \l__morewrites_user_int < { 19 }
301   { \__morewrites_tex_closeout:w \l__morewrites_user_int }
302   { \__morewrites_later:n { \__morewrites_closeout_now: } }
303 }

```

(End definition for `__morewrites_closeout_later:.`)

`_morewrites_openout_later:w` If the user stream is a T_EX stream use the primitive, otherwise find a file name and call
`_morewrites_openout_later:n` `_morewrites_openout_now:n` later.

```

304 \cs_new_protected:Npn \_morewrites_openout_later:w
305   {
306     \int_compare:nNnTF \l__morewrites_user_int < { 19 }
307       { \_morewrites_tex_openout:w \l__morewrites_user_int }
308       { \_morewrites_equals_file:N \_morewrites_openout_later:n }
309   }
310 \cs_new_protected:Npn \_morewrites_openout_later:n #1
311   { \_morewrites_later:n { \_morewrites_openout_now:n {#1} } }

```

(End definition for `_morewrites_openout_later:w` and `_morewrites_openout_later:n`.)

`_morewrites_write_later:w` For T_EX streams use the primitive, otherwise find a general text and save it for later; the
`_morewrites_write_later:n` auxiliary is very similar to `_morewrites_write_now:w`.

```

312 \cs_new_protected:Npn \_morewrites_write_later:w
313   {
314     \int_compare:nNnTF \l__morewrites_user_int < { 19 }
315       { \_morewrites_tex_write:w \l__morewrites_user_int }
316       { \primargs_get_general_text:N \_morewrites_write_later:n }
317   }
318 \cs_new_protected:Npn \_morewrites_write_later:n #1
319   { \_morewrites_later:n { \_morewrites_write_later_aux:n {#1} } }
320 \cs_new_protected:Npn \_morewrites_write_later_aux:n
321   {
322     \_morewrites_user_to_tstr:NTF \g__morewrites_write_prop
323       { \_morewrites_tex_immediate:w \_morewrites_tex_write:w \l__morewrites_tstr_tl \exp_s
324       { \_morewrites_write_now:n }
325   }

```

(End definition for `_morewrites_write_later:w`, `_morewrites_write_later:n`, and `_morewrites_write_later_aux:n`.)

2.3.4 Shipout business

In this section, we hook into the `\shipout` primitive, and redefine it to first build a box with the material to ship out, then perform

```

\_morewrites_before_shipout:
  <primitive shipout> <collected box>
\_morewrites_after_shipout:

```

Each delayed output operation has been replaced by `\write \g__morewrites_iow {‘(‘<operation number>)}’}`. The delimiters we chose to put around numbers must be at least two distinct characters on the left (then `\tex_newlinechar:D` cannot be equal to the delimiter), and at least one non-digit character on the right.

`_morewrites_before_shipout:` Immediately before the shipout, we must open the writing stream `\g__morewrites_iow` (after making sure we are allowed to alter the auxiliary file).

```

326 \cs_new_protected:Npn \_morewrites_before_shipout:
327   {
328     \bool_if:NF \g__morewrites_tmp_file_bool { \_morewrites_chk_file: }
329     \_morewrites_tex_immediate:w \_morewrites_tex_openout:w
330     \g__morewrites_iow = \g__morewrites_tmp_file_tl \scan_stop:
331   }

```


(End definition for `_morewrites_before_shipout:`.)

```

\_morewrites_after_shipout: Immediately after all the \writes are performed, close the file, then read the file with
\_morewrites_after_shipout_loop:ww \endlinechar set to \newlinechar2 to get exactly the original characters that have
been written, possibly with extra characters between ‘( . . . )’ groups. The file is then read
with all the appropriate category codes set up (no other character can appear in the file).
The looping auxiliary \_morewrites_after_shipout_loop:ww extracts the <operation>
numbers from the file, and makes a token list out of those. This token list is then used
in a mapping function to perform the appropriate \write operations. Note that those
operations may reuse the file, so we have to fully parse the file before moving on.
332 \cs_new_protected:Npn \_morewrites_after_shipout:
333   {
334     \_morewrites_tex_immediate:w \_morewrites_tex_closeout:w
335     \g_morewrites_iow
336     \group_begin:
337       \int_set_eq:NN \tex_endlinechar:D \tex_newlinechar:D
338       \char_set_catcode_other:n { \tex_endlinechar:D }
339       \tl_map_inline:nn { ‘(0123456789) }
340         { \char_set_catcode_other:n {‘##1} }
341       \etex_everyeof:D { ‘() \exp_not:N }
342       \tl_set:Nx \l__morewrites_internal_tl
343         {
344           \exp_after:wN \_morewrites_after_shipout_loop:ww
345           \tex_input:D \g__morewrites_tmp_file_tl \c_space_tl
346         }
347       \_morewrites_empty_file:n { \g__morewrites_tmp_file_tl }
348       \exp_args:NNo
349     \group_end:
350     \tl_map_function:nN { \l__morewrites_internal_tl } \_morewrites_later_do:n
351   }
352 \cs_new:Npn \_morewrites_after_shipout_loop:ww #1 ‘( #2 )
353   {
354     \tl_if_empty:nF {#2}
355     {
356       {#2}
357       \_morewrites_after_shipout_loop:ww
358     }
359   }

```

(End definition for `_morewrites_after_shipout:` and `_morewrites_after_shipout_loop:ww`.)

```

\_morewrites_shipout:w Grab the shipped out box using \setbox and regain control using \afterassignment.
\_morewrites_shipout_i: There are two cases: either the box is given as \box or \copy followed by a number, in
\_morewrites_shipout_ii: which case \_morewrites_shipout_i: is inserted afterwards at the same group level, or
the box is given as \hbox (or \vtop and so on) and an additional \aftergroup is needed
to reach a point where we can use the box saved in \g__morewrites_shipout_box.

```

```

360 \cs_new_protected:Npn \_morewrites_shipout:w
361   {
362     \int_gset_eq:NN \g__morewrites_group_level_int \etex_currentgrouplevel:D
363     \tex_afterassignment:D \_morewrites_shipout_i:

```

²Note that the `\newlinechar` used by `\writes` at `\shipout` time are those in effect when the page is shipped out, *i.e.*, just after the closing brace of the `\shipout` construction, which is exactly where we have added this hook.

```

364 \tex_global:D \tex_setbox:D \g__morewrites_shipout_box
365 }
366 \cs_new_protected:Npn \__morewrites_shipout_i:
367 {
368 \int_compare:nNnTF { \g__morewrites_group_level_int }
369 = { \etex_currentgrouplevel:D }
370 { \__morewrites_shipout_ii: }
371 { \tex_aftergroup:D \__morewrites_shipout_ii: }
372 }
373 \cs_new_protected:Npn \__morewrites_shipout_ii:
374 {
375 \__morewrites_before_shipout:
376 \__morewrites_tex_shipout:w \tex_box:D \g__morewrites_shipout_box
377 \__morewrites_after_shipout:
378 }

```

(End definition for `__morewrites_shipout:w`, `__morewrites_shipout_i:`, and `__morewrites_shipout_ii:`.)

`\shipout` The task is now to locate the shipout primitive, which may have been renamed and hooked into by many different packages loaded before `morewrites`. Any of those control sequences which are equal to the primitive are redefined to do `__morewrites_shipout:w` instead. If the primitive is not located at all, the fallback is to hook into the control sequence `\shipout`.

`__morewrites_tex_shipout:w`

```

379 \cs_gset_protected:Npn \__morewrites_tmp:w #1
380 {
381 \cs_if_exist:NF \__morewrites_tex_shipout:w
382 { \cs_new_eq:NN \__morewrites_tex_shipout:w #1 }
383 \cs_gset_eq:NN #1 \__morewrites_shipout:w
384 }
385 \tl_map_inline:nn
386 {
387 \xyrealshipout@
388 \org@shipout
389 \PDFSYNCship@ut@ld
390 \CROP@shipout
391 \@soORI
392 \tex_shipout:D
393 \zwpl@Hship
394 \o@shipout@TP
395 \LL@shipout
396 \Shipout
397 \GXTorg@shipout
398 \AtBegShi@OrgShipout
399 \AtBeginShipoutOriginalShipout
400 \minidocument@orig@shipout
401 \shipout
402 }
403 {
404 \str_if_eq_x:nnT
405 { \cs_meaning:N #1 }
406 { \token_to_str:N \shipout }
407 { \__morewrites_tmp:w #1 }
408 }

```

```

409 \cs_if_exist:NF \__morewrites_tex_shipout:w
410 {
411   \cs_new_eq:NN \__morewrites_tex_shipout:w \shipout
412   \cs_gset_eq:NN \shipout \__morewrites_shipout:w
413 }

```

(End definition for `\shipout` and `__morewrites_tex_shipout:w`.)

2.3.5 Hook at the very end

`__morewrites_close_all:` At the end of the document, close all the files.

```

414 \cs_new_protected:Npn \__morewrites_close_all:
415 {
416   \prop_map_inline:Nn \g__morewrites_write_prop
417   { \__morewrites_tex_immediate:w \__morewrites_tex_closeout:w ##2 \scan_stop: }
418   \prop_gclear:N \g__morewrites_write_prop
419   \prop_map_function:NN \g__morewrites_write_file_prop
420   \__morewrites_closeout_now:nn
421   \prop_gclear:N \g__morewrites_write_file_prop
422 }

```

(End definition for `__morewrites_close_all:.`)

`__morewrites_close_all_at_end:nw` At the end of the run, we try very hard to put some material at the `\@@end`, just in case some other very late code writes to files that are not yet closed. This is tried at most 5 times, to avoid infinite loops in case two packages compete for that last place. The four @ become two after `l3docstrip`.

```

423 \cs_new_protected:Npn \__morewrites_close_all_at_end:nw #1#2 \@@end
424 {
425   \int_compare:nNnTF {#1} > \c_zero
426   { #2 \__morewrites_close_all_at_end:nw { #1 - 1 } }
427   { \__morewrites_close_all: #2 }
428   \@@end
429 }
430 \AtEndDocument { \__morewrites_close_all_at_end:nw { 5 } }

```

(End definition for `__morewrites_close_all_at_end:nw`.)

2.4 Redefining commands

2.4.1 Modified `\newwrite`

`\g__morewrites_alloc_write_int` Counter to allocate user streams. Initialized to 18 so that the first user stream allocated by `morewrites` is 19. Indeed, 18 is reserved for shell commands and packages may expect 16 or 17 to write to the terminal.

```

431 \int_new:N \g__morewrites_alloc_write_int
432 \int_set:Nn \g__morewrites_alloc_write_int { 18 }

```

(End definition for `\g__morewrites_alloc_write_int`.)

`__morewrites_newwrite:N` Reimplementation of `\newwrite` but protected and using a counter `\g__morewrites_alloc_write_int` instead of what `TEX/LATEX 2ε` use.

```

433 \cs_new_protected:Npn \__morewrites_newwrite:N #1
434 {

```

```

435 \int_gincr:N \g__morewrites_alloc_write_int
436 \int_set_eq:NN \allocationnumber \g__morewrites_alloc_write_int
437 \cs_undefine:N #1
438 \int_const:Nn #1 { \allocationnumber }
439 \wlog
440 {
441   \token_to_str:N #1
442   = \token_to_str:N \write \int_use:N \allocationnumber
443 }
444 }

```

(End definition for `__morewrites_newwrite:N`.)

`__morewrites_allocate:n` Raise to #1 the number of `\write` streams allocated to morewrites.

```

445 \cs_new_protected:Npn \__morewrites_allocate:n #1
446 {
447   \prg_replicate:nn
448   {
449     \int_max:nn { 0 }
450     {
451       (#1) - \seq_count:N \g__morewrites_write_seq
452       - \prop_count:N \g__morewrites_write_prop
453     }
454   }
455   {
456     \__morewrites_tex_newwrite:N \l__morewrites_tstr_token
457     \seq_put_right:NV \g__morewrites_write_seq \l__morewrites_tstr_token
458   }
459 }

```

(End definition for `__morewrites_allocate:n`.)

2.5 User commands and keys

`\morewritessetup` Set whatever keys the user passes to `\morewritessetup`.

```

460 \cs_new_protected:Npn \morewritessetup #1
461 { \keys_set:nn { __morewrites } {#1} }

```

(End definition for `\morewritessetup`. This function is documented on page 2.)

file Because of our use of `.initial:n`, this code must appear after `__morewrites_set_file:n` is defined.

```

462 \keys_define:nn { __morewrites }
463 {
464   allocate .code:n = \__morewrites_allocate:n {#1} ,
465   file .code:n = \__morewrites_set_file:n {#1} ,
466   file .initial:n = \c_sys_jobname_str .mw
467 }

```

(End definition for `file`. This function is documented on page 2.)

`\immediate`
`\openout`
`\write`
`\closeout`
`\newwrite`

```

468 \cs_gset_eq:NN \immediate \__morewrites_immediate:w
469 \cs_gset_eq:NN \openout \__morewrites_openout:w

```

```
470 \cs_gset_eq:NN \write      \__morewrites_write:w  
471 \cs_gset_eq:NN \closeout   \__morewrites_closeout:w  
472 \cs_gset_eq:NN \newwrite   \__morewrites_newwrite:N
```

(End definition for \immediate and others. These functions are documented on page 3.)

```
</package>
```

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Symbols	
<code>\</code>	90, 91, 92
A	
<code>allocate</code>	2
<code>\allocationnumber</code>	436, 438, 442
<code>\AtBeginShipoutOriginalShipout</code> ...	399
<code>\AtEndDocument</code>	430
B	
bool commands:	
<code>\bool_gset_false:N</code>	38, 43
<code>\bool_gset_true:N</code>	75
<code>\bool_if:NTF</code>	249, 328
<code>\bool_new:N</code>	37
box commands:	
<code>\box_new:N</code>	40
<code>\l_my_box</code>	6, 6, 6
C	
char commands:	
<code>\char_set_catcode_other:n</code> ..	338, 340
<code>\closeout</code>	3, 468
cs commands:	
<code>\cs_generate_variant:Nn</code>	
.....	20, 21, 22, 137, 152
<code>\cs_gset_eq:NN</code>	58,
63, 65, 383, 412, 468, 469, 470, 471, 472	
<code>\cs_gset_protected:Npn</code>	379
<code>\cs_if_exist:NTF</code>	381, 409
<code>\cs_meaning:N</code>	405
<code>\cs_new:Npn</code>	153, 352
<code>\cs_new_eq:NN</code>	
... 14, 15, 16, 17, 24, 32, 33, 382, 411	
<code>\cs_new_protected:Npn</code>	
..... 7, 18, 41, 46, 72, 95,	
102, 108, 116, 121, 138, 155, 157,	
163, 165, 172, 188, 198, 205, 228,	
235, 241, 247, 266, 279, 296, 298,	
304, 310, 312, 318, 320, 326, 332,	
360, 366, 373, 414, 423, 433, 445, 460	
<code>\cs_undefine:N</code>	437
E	
else commands:	
<code>\else:</code>	59, 64
etex commands:	
<code>\etex_currentgrouplevel:D</code> ..	362, 369
<code>\etex_everyeof:D</code>	341
<code>\etex_readline:D</code>	61, 268
exp commands:	
<code>\exp_after:wN</code>	19, 344
<code>\exp_args:NNc</code>	19
<code>\exp_args:NNf</code>	18
<code>\exp_args:NNo</code>	348
<code>\exp_args:NNx</code>	293
<code>\exp_not:N</code>	289, 341
<code>\exp_not:n</code>	290, 291
<code>\exp_stop_f:</code> ...	19, 176, 217, 238, 323
F	
fi commands:	
<code>\fi:</code>	66, 67
file	2, 462
G	
group commands:	
<code>\group_begin:</code> ...	55, 97, 191, 256, 336
<code>\group_end:</code> ...	69, 100, 195, 263, 349
I	
if commands:	
<code>\if_eof:w</code>	57, 62
<code>\immediate</code>	3, 468
int commands:	
<code>\int_compare:nNnTF</code>	111, 123,
140, 177, 208, 300, 306, 314, 368, 425	
<code>\int_const:Nn</code>	438
<code>\int_decr:N</code>	142
<code>\int_eval:n</code>	134, 154, 297
<code>\int_gincr:N</code>	281, 435
<code>\int_gset_eq:NN</code>	362
<code>\int_max:nn</code>	449
<code>\int_new:N</code>	25, 30, 39, 115, 431
<code>\int_set:Nn</code> 60, 126, 192, 257, 289, 432	
<code>\int_set_eq:NN</code>	337, 436
<code>\int_use:N</code>	146, 285, 294, 442
<code>\c_zero</code>	425
ior commands:	
<code>\ior_if_eof:NTF</code>	269
ior internal commands:	
<code>\g_morewrites_ior</code>	7,
34, 56, 57, 61, 62, 68, 258, 260, 268, 269	
iow commands:	
<code>\iow_char:N</code>	91
<code>\iow_indent:n</code>	91
<code>\iow_shipout_x:Nn</code>	6
<code>\c_term_iow</code>	6

low internal commands:

- `\g_morewrites_iow` 7, 11, 11, 12, 12, 16, 16, 34, 49, 51, 190, 196, 251, 253, 255, 274, 293, 330, 335

K

keys commands:

- `\keys_define:nn` 462
- `\keys_set:nn` 461

M

morewrites internal commands:

- `__morewrites_after_shipout:` 16, 332, 377
- `__morewrites_after_shipout_loop:ww` 17, 332
- `\g_morewrites_alloc_write_int` 19, 431, 435, 436
- `__morewrites_allocate:n` 445, 464
- `__morewrites_before_shipout:` 16, 326, 375
- `__morewrites_chk_file:` 72, 249, 328
- `__morewrites_close_all:` 414, 427
- `__morewrites_close_all_at_end:nw` 423
- `__morewrites_closeout:w` 165, 471
- `__morewrites_closeout_later:` 169, 298
- `__morewrites_closeout_now:` 12, 15, 165, 207, 302
- `__morewrites_closeout_now:nn` 12, 165, 420
- `\l_morewrites_code_tl` 29, 104, 105
- `__morewrites_collect:n` 115, 271
- `__morewrites_collect_aux:Nn` 115
- `__morewrites_collect_gput_right:N` 115, 261
- `\l_morewrites_collect_next_int` 10, 115
- `__morewrites_empty_file:n` 46, 221, 264, 347
- `__morewrites_equals_file:N` 95, 203, 308
- `__morewrites_get_user:N` 6
- `__morewrites_get_user:n` 6, 102, 169, 170, 202, 203, 232, 233
- `\g_morewrites_group_level_int` 39, 362, 368
- `__morewrites_if_file_trivial:nTF` 53, 74
- `__morewrites_immediate:w` 155, 468
- `__morewrites_immediate_auxii:` 155
- `__morewrites_immediate_auxiii:N` 155
- `\l_morewrites_internal_tl` 23, 61, 118, 129, 131, 184, 185, 243, 268, 274, 342, 350
- `__morewrites_later:n` 279, 302, 311, 319
- `__morewrites_later_do:n` 279, 350
- `\g_morewrites_later_int` 6, 15, 25, 281, 285, 294
- `__morewrites_newwrite:N` 433, 472
- `__morewrites_openout:w` 198, 469
- `__morewrites_openout_later:n` 304
- `__morewrites_openout_later:w` 202, 304
- `__morewrites_openout_now:n` 16, 198, 311
- `__morewrites_set_file:n` 20, 41, 465
- `__morewrites_shipout:w` 18, 360, 383, 412
- `\g_morewrites_shipout_box` 17, 40, 364, 376
- `__morewrites_shipout_i:` 17, 360
- `__morewrites_shipout_ii:` 360
- `__morewrites_tex_closeout:w` 14, 50, 176, 196, 254, 301, 334, 417
- `__morewrites_tex_immediate:w` 14, 48, 50, 161, 176, 190, 196, 210, 217, 238, 245, 250, 252, 254, 273, 323, 329, 334, 417
- `__morewrites_tex_newwrite:N` 7, 18, 456
- `__morewrites_tex_openout:w` 14, 48, 190, 210, 217, 250, 307, 329
- `__morewrites_tex_shipout:w` 376, 379
- `__morewrites_tex_write:w` 14, 238, 245, 252, 273, 293, 315, 323
- `__morewrites_tmp:w` 8, 24, 58, 63, 65, 70, 379, 407
- `\g_morewrites_tmp_file_bool` 36, 43, 75, 249, 328
- `\g_morewrites_tmp_file_tl` 7, 8, 8, 9, 36, 44, 74, 78, 79, 80, 251, 258, 264, 330, 345, 347
- `\l_morewrites_tstr_tl` 6, 10, 30, 110, 113, 176, 177, 180, 214, 216, 217, 238, 323
- `\l_morewrites_tstr_token` 32, 456, 457
- `\l_morewrites_user_int` 6, 9, 10, 30, 106, 110, 111, 113, 177, 179, 184, 185, 208, 210, 216, 222, 224, 243, 262, 289, 290, 300, 301, 306, 307, 314, 315
- `__morewrites_user_tl_name:n` 14, 153, 193, 194, 224, 262

<code>_morewrites_user_to_tstr:N</code>	NTF	..	
			108 , 174 , 237 , 322
<code>_morewrites_write:w</code>			228 , 470
<code>\g_morewrites_write_file_prop</code>		..	
			11 , 12 , 28 , 184 , 222 , 243 , 419 , 421
<code>_morewrites_write_later:n</code>		..	312
<code>_morewrites_write_later:w</code>			232 , 312
<code>_morewrites_write_later_aux:n</code>			312
<code>_morewrites_write_now:n</code>		..	228 , 324
<code>_morewrites_write_now:w</code>		..	16 , 228
<code>_morewrites_write_now_loop:</code>		..	247
<code>_morewrites_write_now_open:n</code>		..	
			14 , 244 , 247
<code>\g_morewrites_write_prop</code>		..	
			10 , 11 , 12 , 27 , 174 , 179 , 216 , 237 , 322 , 416 , 418 , 452
<code>\l_morewrites_write_prop</code>		..	4
<code>\g_morewrites_write_seq</code>		..	
			26 , 180 , 214 , 451 , 457
<code>\morewritessetup</code>		..	2 , 2 , 2 , 2 , 7 , 20 , 460
msg commands:			
<code>\msg_new:nnn</code>		..	8
<code>\msg_new:nnnn</code>		..	84
<code>\msg_warning:nn</code>		..	10
<code>\msg_warning:nnnn</code>		..	77
N			
<code>\newread</code>		..	35
<code>\newwrite</code>		..	2 , 34 , 468
O			
<code>\openout</code>		..	3 , 468
P			
prg commands:			
<code>\prg_new_conditional:Npnn</code>		..	53
<code>\prg_replicate:nn</code>		..	447
<code>\prg_return_false:</code>		..	8 , 65
<code>\prg_return_true:</code>		..	8 , 58 , 63
primargs commands:			
<code>\primargs_get_file_name:N</code>		..	98
<code>\primargs_get_general_text:N</code>		..	239 , 316
<code>\primargs_get_number:N</code>		..	9
<code>\primargs_read_x_token:N</code>		..	12 , 156
<code>\primargs_remove_equals:N</code>		..	100
prop commands:			
<code>\prop_count:N</code>		..	452
<code>\prop_gclear:N</code>		..	418 , 421
<code>\prop_get:NnNTF</code>		..	113 , 243
<code>\prop_gpop:NnNTF</code>		..	20 , 20 , 184
<code>\prop_gput:Nnn</code>		..	20 , 21 , 216 , 222
<code>\prop_gremove:Nn</code>		..	179
<code>\prop_map_function:NN</code>		..	419
<code>\prop_map_inline:Nn</code>		..	416
<code>\prop_new:N</code>		..	27 , 28
<code>\ProvidesExplPackage</code>		..	3
Q			
quark internal commands:			
<code>\s_morewrites</code>		..	11 , 11 , 12 , 33 , 159 , 164 , 167 , 200 , 230
R			
<code>\RequirePackage</code>		..	1 , 2
S			
scan commands:			
<code>\scan_stop:</code>		..	7 , 33 , 49 , 56 , 190 , 211 , 218 , 251 , 258 , 330 , 417
scan internal commands:			
<code>_scan_new:N</code>		..	7
seq commands:			
<code>\seq_count:N</code>		..	451
<code>\seq_gpop:NNTF</code>		..	214
<code>\seq_gput_left:Nn</code>		..	180
<code>\seq_new:N</code>		..	26
<code>\seq_put_right:Nn</code>		..	457
<code>\Shipout</code>		..	396
<code>\shipout</code>		..	3 , 379
str commands:			
<code>\str_if_eq:nnTF</code>		..	164
<code>\str_if_eq_x:nnTF</code>		..	404
sys commands:			
<code>\sys_if_engine luatex:TF</code>		..	5
<code>\c_sys_jobname_str</code>		..	2 , 8 , 466
T			
TeX and L ^A T _E X 2 _ε commands:			
<code>\@@end</code>		..	19 , 423 , 428
<code>\@soORI</code>		..	391
<code>\afterassignment</code>		..	17
<code>\aftergroup</code>		..	17
<code>\AtBegShi@OrgShipout</code>		..	398
<code>\box</code>		..	5 , 17
<code>\chardef</code>		..	5
<code>\closeout</code>		..	3 , 3 , 3 , 4 , 4 , 4 , 5 , 5 , 5 , 5 , 11 , 12 , 12
<code>\copy</code>		..	6 , 6 , 17
<code>\count</code>		..	4
<code>\CROP@shipout</code>		..	390
<code>\endlinechar</code>		..	14 , 17
<code>\GXTorg@shipout</code>		..	397
<code>\hbox</code>		..	5 , 17
<code>\ifdef</code>		..	4
<code>\immediate</code>		..	1 , 3 , 3 , 4 , 5 , 8 , 11 , 11 , 11 , 12 , 12 , 12 , 14
<code>\jobname</code>		..	2
<code>\LL@shipout</code>		..	395
<code>\minidocument@orig@shipout</code>		..	400

<code>\newlinechar</code>	14, 14, 17, 17	<code>\tex_setbox:D</code>	364
<code>\newwrite</code> .	1, 2, 2, 3, 4, 5, 5, 5, 5, 19, 19	<code>\tex_shipout:D</code>	392
<code>\noexpand</code>	12	<code>\tex_write:D</code>	16
<code>\notexpanded:</code>	4	tl commands:	
<code>\o@shipout@TP</code>	394	<code>\c_space_tl</code>	345
<code>\openout</code>	3,	<code>\tl_clear:N</code>	132
	3, 3, 4, 4, 4, 4, 5, 5, 5, 8, 9, 11, 12, 12	<code>\tl_clear_new:N</code>	125
<code>\org@shipout</code>	388	<code>\tl_const:Nn</code>	282
<code>\outer</code>	5	<code>\tl_gclear:N</code>	194
<code>\pdfobj</code>	12	<code>\tl_gclear_new:N</code>	224
<code>\PDFSYNCSHIP@ut@ld</code>	389	<code>\tl_gput_right:Nn</code>	20, 22, 80, 143
<code>\read</code>	4	<code>\tl_gset:Nn</code>	44
<code>\readline</code>	14	<code>\tl_if_empty:NTF</code>	128
<code>\relax</code>	4	<code>\tl_if_empty:nTF</code>	354
<code>\setbox</code>	17	<code>\tl_map_function:nN</code>	350
<code>\shipout</code>	3,	<code>\tl_map_inline:nn</code>	339, 385
	3, 5, 6, 6, 7, 11, 11, 16, 17, 17, 18	<code>\tl_new:N</code>	23, 29, 31, 36
<code>\toks</code>	4	<code>\tl_put_left:Nn</code>	131
<code>\vtop</code>	17	<code>\tl_set:Nn</code>	104, 110, 118, 342
<code>\write</code>	1, 2, 2,	<code>\tl_set_eq:NN</code>	129
	2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 5,	<code>\tl_to_str:n</code>	211, 218, 223
	5, 5, 5, 5, 7, 7, 7, 10, 11, 11, 11, 11,	<code>\tl_use:N</code>	193, 297
	12, 12, 12, 14, 14, 14, 16, 17, 17, 17, 20	token commands:	
<code>\xyrealshipout@</code>	387	<code>\g_primargs_token</code>	159
<code>\zwpl@Hship</code>	393	<code>\token_if_eq_meaning:NNTF</code>	159
tex commands:		<code>\token_to_str:N</code>	406, 441, 442
<code>\tex_afterassignment:D</code>	105, 363		
<code>\tex_aftergroup:D</code>	98, 99, 371	U	
<code>\tex_box:D</code>	376	use commands:	
<code>\tex_closein:D</code>	68, 260	<code>\use_i:nn</code>	11, 112, 168, 201, 231
<code>\tex_closeout:D</code>	17	<code>\use_iii:nnn</code>	164
<code>\tex_endinput:D</code>	11	V	
<code>\tex_endlinechar:D</code> ..	60, 257, 337, 338	vbox commands:	
<code>\tex_global:D</code>	364	<code>\vbox_set:Nn</code>	6
<code>\tex_immediate:D</code>	14	W	
<code>\tex_input:D</code>	345	<code>\wlog</code>	439
<code>\tex_newlinechar:D</code>	16, 192, 337	<code>\write</code>	3, 442, 468
<code>\tex_openin:D</code>	56, 258		
<code>\tex_openout:D</code>	15		