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

Bruno Le Floch

2024/02/02

# Contents

1	morewrites documentation	1
	1.1 Commands defined or altered by morewrites	2
	1.2 Known deficiencies and open questions	3
2	morewrites implementation	3
	2.1 Overview of relevant T <sub>E</sub> X 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 Verbosity	8
	2.2.5 Helpers for auxiliary file	8
	2.2.6 Parsing and other helpers	10
	2.3 Writing	11
	2.3.1 Redefining \immediate	12
	2.3.2 Immediate actions	12
	2.3.3 Delayed actions	16
	2.3.4 Shipout business	18
	$2.3.5$ Hook at the very end $\ldots$	21
	2.4 Redefining commands	21
	2.4.1 Modified \newwrite	21
	2.5 User commands and keys	22

# Index

 $\mathbf{24}$ 

# 1 morewrites documentation

This  $IAT_EX$  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 LATEX 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,  $\langle job name \rangle.mw$ , which can safely be deleted. The package only overwrites this auxiliary file if it is empty, and otherwise uses a modified file name, obtained by adding an integer to the name (previously it was obtained by appending additional copies of .mw to the name). Such files can be safely deleted. Be careful though to not delete a Maple worksheet by accident when cleaning up your files.
- LuaTEX allows 128 \write streams, so this package does nothing (with a warning) when used with LuaTEX.

## 1.1 Commands defined or altered by morewrites

\morewritessetup	$morewritessetup { key-value list }$
New: 2014-07-26	Sets the options described by the $\langle key-value \ list \rangle$ .
allocate	\morewritessetup { allocate = $\langle integer \rangle$ }
New: 2017-04-10	Sets to (at least) $\langle integer \rangle$ the number of \write streams allocated to the inner workings of morewrites. By default this is zero but increasing this value to 10 (or so) may speed up morewrites.
file	\morewritessetup { file = $\langle file name \rangle$ }
	Sets (globally) the name of the file which will be used by internal processes of morewrites. The file name is \jobname.mw by default (technically, \c_sys_jobname_str.mw). Contrarily to earlier versions of morewrites non-empty files will not be overwritten; this design choice may lead to unwanted .mw files remaining.
verbose	<pre>\morewritessetup { verbose }</pre>
New: 2024-01-05	This boolean option (false by default) makes the package write to the terminal all of the operations that it performs. This can render morewrites useful for debugging some file-writing operations.

\newwrite	This macro is redefined by morewrites. Since morewrites allows more than 16 write
Updated: 2015-08-01	streams, it removes the corresponding restrictions in \newwrite.

 $T_EX$  hackers note: The revised \newwrite allocate stream numbers starting at 129. This might break some code that expects stream numbers to be less than 16.

\immediate Updated: 2015-08-01

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

\openout 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.

Updated: 2017-04-20

 $\mathbf{T}_{\!E\!}\mathbf{X}\mathbf{hackers}$  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 {primargs} [2024/01/05]
2 \ProvidesExplPackage
    {morewrites} {2024/02/02} {} {Always room for a new write}
   Quit early under LuaT<sub>E</sub>X.
  \sys_if_engine_luatex:T
4
    {
5
      \cs_new_protected:Npn \morewritessetup #1 { }
      \msg_new:nnn { morewrites } { luatex }
         { The~morewrites~package~is~unnecessary~in~LuaTeX. }
8
      \msg_warning:nn { morewrites } { luatex }
9
10
      \tex_endinput:D
    }%
12 (@@=morewrites)
```

## 2.1 Overview of relevant T<sub>E</sub>X facts

The aim of the morewrites package is to lift  $T_EX$ '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  $T_EX$  and  $L^AT_EX 2_{\varepsilon}$ .

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 T<sub>E</sub>X or IAT<sub>E</sub>X  $2_{\varepsilon}$ ) 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  $[0, 15] \cup$  $[129, \infty)$  manipulated by the redefined **\openout**, **\write**, **\closeout**, and **\newwrite**. A user stream in [0, 15] (reserved by LATEX  $2_{\varepsilon}$  or allocated by expl3) is mapped to the same TEX stream number, while a user stream in  $[129, \infty)$  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 stream number 128 is also often used like 16 to avoid distinguishing LuaTEX. Rather than special-casing it we skip directly to larger stream numbers.

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

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

- (*integer*) denotes an integer in any form that T<sub>E</sub>X accepts as the right-hand side of a primitive integer assignment of the form \countO=(*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;
- $\langle file \ name \rangle$  is an arbitrary sequence of explicit or implicit characters with arbitrary category codes (except active characters, which are expanded before reaching  $T_EX$ '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 <code>\notexpanded:</code> 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 begingroup character, and ending with the matching explicit end-group character (both

with any character code), with an equal number of explicit begin-group and endgroup characters in between: this is precisely the right-hand side of an assignment of the form  $\toks0=\langle general\ text \rangle$ .

The morewrites package redefines these three control sequences to expect a user stream number rather than a  $T_EX$  stream number as the  $\langle integer \rangle$ , then map such a user stream to a  $T_EX$  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  $T_EX$  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  $\box(integer)$  or  $\box(material to typeset)$ .

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 ( $T_EX$ ) write stream. This must be extended to allocate higher (user) streams.

### 2.2 Preliminaries

#### 2.2.1 Copying some commands

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

13 \cs\_new\_eq:NN \\_\_morewrites\_tex\_immediate:w \tex\_immediate:D
14 \cs\_new\_eq:NN \\_\_morewrites\_tex\_openout:w \tex\_openout:D
15 \cs\_new\_eq:NN \\_\_morewrites\_tex\_write:w \tex\_write:D
16 \cs\_new\_eq:NN \\_\_morewrites\_tex\_closeout:w \tex\_closeout:D

(End of definition for \\_\_morewrites\_tex\_immediate:w and others.)

\_\_morewrites\_tex\_newwrite:N

\_\_morewrites\_tex\_openout:w

\\_\_morewrites\_tex\_write:w

\\_\_morewrites\_tex\_closeout:w

\\_\_morewrites\_tex\_immediate:w

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

17 \exp\_args:NNf \cs\_new\_protected:Npn \\_\_morewrites\_tex\_newwrite:N
18 { \exp\_args:NNc \exp\_after:wN \exp\_stop\_f: { newwrite } }

(End of definition for \\_\_morewrites\_tex\_newwrite:N.)

## 2.2.2 Variants

We need these variants.

	<pre>19 \cs_generate_variant:Nn \prop_gpop:NnNT { NV } 20 \cs_generate_variant:Nn \prop_gput:Nnn { NVx } 21 \cs_generate_variant:Nn \tl_gput_right:Nn { Nv }</pre>
	2.2.3 Variables
<pre>\lmorewrites_internal_tl \lmorewrites_internal_seq</pre>	Used for temporary scratch purposes. 22 \tl_new:N \lmorewrites_internal_tl 23 \seq_new:N \lmorewrites_internal_seq
	$(End of definition for \l\_morewrites\_internal\_tl and \l\_morewrites\_internal\_seq.)$
\morewrites_tmp:w	Used for temporary definitions. 24 \cs_new_eq:NN \morewrites_tmp:w ?
	(End of definition for \morewrites_tmp:w.)
\lmorewrites_verbose_bool	When this boolean is set true, morewrites will print to the terminal all of the operations that it does. <sup>25</sup> \bool_new:N \lmorewrites_verbose_bool
	(End of definition for \1morewrites_verbose_bool.)
\gmorewrites_later_int	The integer \gmorewrites_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:
	<pre>\vbox_set:Nn \l_my_box { \iow_shipout_x:Nn \c_term_iow {{\text}} } \shipout \copy \l_my_box \shipout \copy \l_my_box</pre>
	<pre>will print (text) to the terminal twice. 26 \int_new:N \gmorewrites_later_int</pre>
	(End of definition for \gmorewrites_later_int.)
\gmorewrites_write_seq	Keep track of TEX stream numbers managed by morewrites that are currently not in use as user streams. 27 \seq_new:N \gmorewrites_write_seq
	(End of definition for \gmorewrites_write_seq.)
\gmorewrites_write_prop	Map user streams to T <sub>E</sub> X streams. 28 \prop_new:N \gmorewrites_write_prop
	(End of definition for \gmorewrites_write_prop.)
\gmorewrites_write_file_prop	Map user streams with no associated TeX streams to file names. $_{29} \text{prop_new:N } g\_morewrites_write_file_prop}$
	(End of definition for \gmorewrites_write_file_prop.)

\lmorewrites_code_tl	Stores the code to run after finding a user stream, in \morewrites_get_user:n. 30 \tl_new:N \lmorewrites_code_tl
	(End of definition for \lmorewrites_code_tl.)
<pre>\lmorewrites_user_int    \lmorewrites_tstr_tl</pre>	The user stream number following redefined primitives is stored in \lmorewrites user_int (see \morewrites_get_user:N). The corresponding T <sub>E</sub> X stream number is eventually stored in \lmorewrites_tstr_tl (a token list). <sup>31</sup> \int_new:N \lmorewrites_user_int <sup>32</sup> \tl_new:N \lmorewrites_tstr_tl
	(End of definition for \lmorewrites_user_int and \lmorewrites_tstr_tl.)
\lmorewrites_tstr_token	This token is given as an argument to \morewrites_tex_newwrite:N. 33 \cs_new_eq:NN \lmorewrites_tstr_token ?
	(End of definition for \lmorewrites_tstr_token.)
\smorewrites	A recognizable version of $scan_stop:$ . This is inspired by <sup>1</sup> scan marks (see the l3quark module of IAT <sub>E</sub> X3), but $scan_new:N$ is not used directly, since it is has been made available in IAT <sub>E</sub> X3 too recently.
	34 \cs_new_eq:NN \smorewrites \scan_stop:
	(End of definition for \smorewrites.)
\gmorewrites_iow \gmorewrites_ior	The expansion that \write performs is impossible to emulate (in X <sub>fTE</sub> X at least) with anything else than \write. We will write on the stream \gmorewrites_iow to the file \gmorewrites_tmp_file_tl and read back from it in the stream \gmorewrites 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.
	35 \newwrite \g_morewrites_iow 36 \newread \g_morewrites_ior
	(End of definition for $g\_morewrites_iow$ and $g\_morewrites_ior$ .)
\gmorewrites_tmp_file_tl \gmorewrites_tmp_file_bool	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. <sup>37</sup> \tl_new:N \gmorewrites_tmp_file_tl <sup>38</sup> \bool_new:N \gmorewrites_tmp_file_bool <sup>39</sup> \bool_gset_false:N \gmorewrites_tmp_file_bool
	$(End of definition for \verb+g_morewrites_tmp_file_tl and \verb+g_morewrites_tmp_file_bool.)$
\gmorewrites_group_level_int	The group level when \shipout is called: this is used to distinguish between explicit boxes and box registers. 40 \int_new:N \gmorewrites_group_level_int
	(End of definition for \gmorewrites_group_level_int.)
\gmorewrites_shipout_box	The page to be shipped out.
19-7more#rices_purhour_poy	41 \box_new:N \gmorewrites_shipout_box
	(End of definition for \g_morewrites_shipout_box.)
	<sup>1</sup> Historically, this might have happened the other way around, since the author of this package is also

ight have happened the other way around, since the author of this package is also <sup> $^{1}$ </sup>Historically, this m on the  $L^{AT}EX3$  Team.

#### 2.2.4 Verbosity

\\_\_morewrites\_verbose:n Messages to put in the terminal if the verbose option is selected.

```
42 \cs_new_protected:Npn \__morewrites_verbose:n #1
```

```
43 { \bool_if:NT \l__morewrites_verbose_bool { \iow_term:e { morewrites:~#1 } } }
```

(End of definition for \\_\_morewrites\_verbose:n.)

#### 2.2.5 Helpers for auxiliary file

\\_\_morewrites\_set\_file:n Sets \g\_\_morewrites\_tmp\_file\_tl to the given value (initially \c\_sys\_jobname\_str.mw). We do not yet expand, delaying that to the time where we start opening/closing
the file, in case #1 contains something that has not yet been fixed. Mark that the file has
not been checked.

```
44 \cs_new_protected:Npn \__morewrites_set_file:n #1
45 {
46      \bool_gset_false:N \g__morewrites_tmp_file_bool
47      \tl_gset:Nn \g__morewrites_tmp_file_tl {#1}
48   }
```

```
(End of definition for \__morewrites_set_file:n.)
```

\\_\_morewrites\_empty\_file:n

Empties a file 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.

```
(End of definition for \__morewrites_empty_file:n.)
```

\\_\_morewrites\_if\_file\_trivial:n<u>TF</u>

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).

56	<pre>\prg_new_conditional:Npnn \morewrites_if_file_trivial:n #1 { TF }</pre>
57	{
58	\group_begin:
59	<pre>\tex_openin:D \gmorewrites_ior = {#1}</pre>
60	<pre>\if_eof:w \gmorewrites_ior</pre>
61	<pre>\cs_gset_eq:NN \morewrites_tmp:w \prg_return_true:</pre>
62	\else:
63	<pre>\int_set:Nn \tex_endlinechar:D { -1 }</pre>
64	<pre>\tex_readline:D \gmorewrites_ior to \lmorewrites_internal_tl</pre>
65	\if_eof:w \gmorewrites_ior
66	<pre>\cs_gset_eq:NN \morewrites_tmp:w \prg_return_true:</pre>
67	\else:
68	<pre>\cs_gset_eq:NN \morewrites_tmp:w \prg_return_false:</pre>

```
69 \fi:
70 \fi:
71 \tex_closein:D \g__morewrites_ior
72 \group_end:
73 \__morewrites_tmp:w
74 }
(End of definition for \ morewrites if file trivial:nTF.)
```

```
\__morewrites_chk_file:
```

Expand the file name in \g\_\_morewrites\_tmp\_file\_tl once and for all. Check that the file does not exist or is blank. If not, try another file name obtained as follows: if it ends with .mw pick up any number that lies just before .mw and increment it, and otherwise just add .mw at the end of the file. This avoids clobbering files that the user would not want to lose.

```
75 \cs_new_protected:Npn \__morewrites_chk_file:
76
    {
       \tl_gset:Ne \g__morewrites_tmp_file_tl
77
         { \tl_to_str:e { \g_morewrites_tmp_file_tl } }
78
       \__morewrites_if_file_trivial:nTF { \g__morewrites_tmp_file_tl }
79
         { \bool_gset_true:N \g_morewrites_tmp_file_bool }
80
         {
81
           \__morewrites_chk_file_aux:
82
           \msg_warning:nnxx { morewrites } { file-exists }
83
             { \g_morewrites_tmp_file_tl } { \l_morewrites_internal_tl }
84
           \tl_gset_eq:NN \g__morewrites_tmp_file_tl \l__morewrites_internal_tl
85
           \__morewrites_chk_file:
86
         }
87
    }
88
89 \cs_new_protected:Npn \__morewrites_chk_file_aux:
     ſ
90
       \regex_extract_once:nVNTF
91
         { \A (\D*) (\d*) .mw \Z } \g__morewrites_tmp_file_tl \l__morewrites_internal_seq
92
         ł
93
           \tl_set:Ne \l__morewrites_internal_tl
94
             {
95
                \seq_item:Nn \l__morewrites_internal_seq { 2 }
96
97
                \int_eval:n { \seq_item:Nn \l__morewrites_internal_seq { 3 } + 1 }
98
                .mw
             }
99
         }
100
         { \tl_set:Ne \l__morewrites_internal_tl { \g__morewrites_tmp_file_tl .mw } }
101
    }
102
   \msg_new:nnnn { morewrites } { file-exists }
103
     { File~'#1'~exists,~using~'#2'~instead. }
104
     ſ
105
       The~file~'#1'~exists~and~was~not~created~by~this~version~of~the~
106
       'morewrites' ~ package. ~ Please ~ move ~ or ~ delete ~ that ~ file, ~ or ~ provide ~
       another~file~name~by~adding
108
       11 11
109
       \iow_indent:n { \iow_char:N\\morewritessetup~{~file~=~other-name~} }
       // //
       to~your~source~file.~In~the~meantime,~the~file~'#2'~will~be~used.
     }
113
```

```
(End \ of \ definition \ for \ \verb+\_morewrites\_chk\_file:.)
```

#### 2.2.6 Parsing and other helpers

\\_\_morewrites\_equals\_file:N

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

```
114 \cs_new_protected:Npn \__morewrites_equals_file:N #1
115 {
116 \group_begin:
117 \tex_aftergroup:D \primargs_get_input_file_name:N
118 \tex_aftergroup:D #1
119 \primargs_remove_equals:N \group_end:
120 }
```

(End of 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.

(End of 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 T<sub>E</sub>X stream  $\l_\_morewrites\_tstr\_tl$  (it defaults to the user stream). Streams less than 129 are not managed by morewrites: actual T<sub>E</sub>X streams in [0, 15]; negative for writing to log; 16, 17, 128 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 T<sub>E</sub>X stream, otherwise run the false branch.

```
127 \cs_new_protected:Npn \__morewrites_user_to_tstr:NTF #1
128 {
129 \tl_set:NV \l__morewrites_tstr_tl \l__morewrites_user_int
130 \int_compare:nNnTF { \l__morewrites_user_int } < { 129 }
131 { \use_i:nn }
132 { \prop_get:NVNTF #1 \l__morewrites_user_int \l__morewrites_tstr_tl }
133 }</pre>
```

(End of 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  $\mbox{write}$  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  $\1\_$ morewrites\_\$k\$\_t1 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  $\1\_$ morewrites\_collect\_next\_int is (one plus) the maximal k among non-empty token lists.

```
134 \int_new:N \l__morewrites_collect_next_int
135 \cs_new_protected:Npn \__morewrites_collect:x #1
136 {
```

```
\tl_set:Nx \l__morewrites_internal_tl {#1}
         __morewrites_collect_aux:cf { l__morewrites_0_tl } { 1 }
138
     }
139
   \cs_new_protected:Npn \__morewrites_collect_aux:Nn #1#2
140
     ł
141
       \int_compare:nNnT {#2} > \1__morewrites_collect_next_int
142
         ł
143
            \tl_clear_new:N #1
144
            \int_set:Nn \l__morewrites_collect_next_int {#2}
145
         7
146
147
       \tl_if_empty:NTF #1
         { \tl_set_eq:NN #1 \l__morewrites_internal_tl }
148
149
         ł
            \tl_put_left:No \l__morewrites_internal_tl {#1}
150
            \tl_clear:N #1
151
            \__morewrites_collect_aux:cf { l__morewrites_#2_tl }
152
              { \int_eval:n { #2 + 1 } }
         }
154
     }
155
   \cs_generate_variant:Nn \__morewrites_collect_aux:Nn { cf }
156
157
   \cs_new_protected:Npn \__morewrites_collect_gput_right:N #1
     {
158
       \int_compare:nNnF \l__morewrites_collect_next_int = 0
159
160
         {
            \int_decr:N \l__morewrites_collect_next_int
161
            \tl_gput_right:Nv #1
162
163
              ſ
164
                l__morewrites_
                \int_use:N \l__morewrites_collect_next_int
165
166
                _tl
              7
167
168
              _morewrites_collect_gput_right:N #1
         }
169
     }
170
171 \cs_generate_variant:Nn \__morewrites_collect_gput_right:N { c }
```

(End of definition for \1\_\_morewrites\_collect\_next\_int and others.)

\\_\_morewrites\_user\_tl\_name:n

172 \cs\_new:Npn \\_\_morewrites\_user\_tl\_name:n #1
173 { g\_morewrites\_iow\_ \int\_eval:n {#1} \_tl }

(End of 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.

The name of a global token list variable holding the text of a given user stream.

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  $\langle write \langle number \rangle \{ \langle text \rangle \}$  then the T<sub>E</sub>X stream  $\langle number \rangle$  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  $\langle number \rangle$  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

\s\_\_morewrites \use\_i:nn {\code for delayed action\}
{\code for immediate action\}
\frac{further code\

The leading  $s\_morewrites$  allows the redefined immediate to detect these redefined primitives, and to run the  $\langle code \ for \ immediate \ action \rangle$  instead of the  $\langle code \ for \ delayed \ action \rangle$  which is run by default. In both cases, any  $\langle further \ code \rangle$  is run.

\_\_morewrites\_immediate:w
 \\_\_morewrites\_immediate\_auxii:
 \\_\_morewrites\_immediate\_auxiii:N

 $T_EX$ '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. In contrived situations involving nonsensical uses of \noexpand after \immediate, this code does not perfectly match how  $T_FX$  expands.

```
\cs_new_protected:Npn \__morewrites_immediate:w
174
     { \primargs_read_x_token:N \__morewrites_immediate_auxii: }
175
  \cs_new_protected:Npn \__morewrites_immediate_auxii:
176
177
    {
       \token_if_eq_meaning:NNTF \g_primargs_token \s__morewrites
178
         { \__morewrites_immediate_auxiii:N }
179
         Ł
180
             _morewrites_verbose:n
             { \tl to str:n { \immediate } \token to meaning:N \g primargs token }
           \__morewrites_tex_immediate:w
         7
184
    }
185
  \cs_new_protected:Npn \__morewrites_immediate_auxiii:N #1
186
    { \str_if_eq:nnTF { #1 } { \s__morewrites } { \use_iii:nnn } { #1 } }
187
```

```
(End \ of \ definition \ for \ \_morewrites\_immediate:w, \ \_morewrites\_immediate\_auxii:, \ and \ \_morewrites\_immediate\_auxii: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. In this subsubsection we implement the immediate versions only.

 In the immediate case \\_\_morewrites\_closeout\_now:, there are three cases. The stream may point to a T<sub>E</sub>X 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<sub>E</sub>X stream \g\_\_morewrites\_iow. The token list consists of multiple \immediate \write \g\_\_morewrites\_iow { $\langle text \rangle$ } 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.

```
188 \cs_new_protected:Npn \__morewrites_closeout:w
189
    {
       \s__morewrites
190
191
       \use_i:nn
         { \__morewrites_get_user:n { \__morewrites_closeout_later: } }
192
193
         { \__morewrites_get_user:n { \__morewrites_closeout_now: } }
    }
194
  \cs_new_protected:Npn \__morewrites_closeout_now:
195
    {
196
       \__morewrites_verbose:n { \tl_to_str:n { \immediate \closeout } \int_use:N \l__morewrites
197
       \__morewrites_closeout_now_silent:
198
    }
199
   \cs_new_protected:Npn \__morewrites_closeout_now_silent:
200
    {
201
         _morewrites_user_to_tstr:NTF \g__morewrites_write_prop
         {
203
           \__morewrites_tex_immediate:w \__morewrites_tex_closeout:w \l__morewrites_tstr_tl \ex
204
           \int_compare:nNnF { \1__morewrites_tstr_t1 } = { \1__morewrites_user_int }
205
             ł
206
               \prop_gremove:NV \g_morewrites_write_prop \l_morewrites_user_int
207
               \seq_gput_left:NV \g__morewrites_write_seq \l__morewrites_tstr_tl
208
             }
209
         }
         ł
           \prop_gpop:NVNT \g__morewrites_write_file_prop \l__morewrites_user_int \l__morewrites
             { \__morewrites_closeout_now:nn { \l__morewrites_user_int } { \l__morewrites_intern
         3
214
    }
  \cs_new_protected:Npn \__morewrites_closeout_now:nn #1#2
216
    ł
       \__morewrites_tex_immediate:w \__morewrites_tex_openout:w \g__morewrites_iow = {#2}
218
219
       \group_begin:
         \int_set:Nn \tex_newlinechar:D { -1 }
         \tl_use:c { \__morewrites_user_tl_name:n {#1} }
         \tl_gclear:c { \__morewrites_user_tl_name:n {#1} }
       \group_end:
223
         _morewrites_tex_immediate:w \__morewrites_tex_closeout:w \g__morewrites_iow
224
     }
```

(End of definition for \\_\_morewrites\_closeout:w and others.)

In the immediate case find a file name, then allocate a  $T_{\rm E}X$  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  $T_{E}X$  would have written, and empty the file by opening and closing it (actually that's done automatically by the primitive).

```
226 \cs_new_protected:Npn \__morewrites_openout:w
227
     ł
228
       \s__morewrites
       \use_i:nn
229
         { \__morewrites_get_user:n { \__morewrites_openout_later:w } }
230
         { \__morewrites_get_user:n { \__morewrites_equals_file:N \__morewrites_openout_now:n }
231
    }
   \cs_new_protected:Npn \__morewrites_openout_now:n #1
233
     {
234
235
       \__morewrites_verbose:n
236
237
           \tl_to_str:n { \immediate\openout }
           \int_use:N \l__morewrites_user_int
238
           \c_space_tl = ~ {#1}
239
         7
240
         _morewrites_openout_now_silent:n {#1}
241
    }
242
   \cs_new_protected:Npn \__morewrites_openout_now_silent:n #1
243
     {
244
       \__morewrites_closeout_now_silent:
245
       \int_compare:nNnTF { \l__morewrites_user_int } < { 129 }</pre>
246
         {
247
248
           \__morewrites_tex_immediate:w \__morewrites_tex_openout:w \l__morewrites_user_int
249
             = { \tl_to_str:n {#1} }
         }
250
         ł
251
           \seq_gpop:NNTF \g__morewrites_write_seq \l__morewrites_tstr_tl
252
253
             {
                \prop_gput:NVV \g__morewrites_write_prop \l__morewrites_user_int \l__morewrites_t
254
                \__morewrites_tex_immediate:w \__morewrites_tex_openout:w \l__morewrites_tstr_tl
255
                  = { \tl_to_str:n {#1} }
256
             }
257
             ł
258
                \__morewrites_empty_file:n {#1}
259
                \prop_gput:NVx \g__morewrites_write_file_prop \l__morewrites_user_int
260
                  { \tl_to_str:n {#1} }
261
                \tl_gclear_new:c { \__morewrites_user_tl_name:n { \l__morewrites_user_int } }
262
             }
263
         }
264
     }
265
   \sys_if_engine_xetex:T
266
267
       \cs_new_eq:NN \__morewrites_openout_now_silent_aux:n \__morewrites_openout_now_silent:n
268
       \cs_gset_protected:Npn \__morewrites_openout_now_silent:n #1
269
         ł
           \tl_set:Nn \l__morewrites_internal_tl {#1}
           \tl_remove_all:Nn \l_morewrites_internal_tl { " } % { " }
            \exp_args:No \__morewrites_openout_now_silent_aux:n \l__morewrites_internal_tl
         }
274
     }
275
```

 $(\mathit{End of definition for \_morewrites_openout_now:n, and \_m$ 

#### openout\_now\_silent:n.)

\\_\_morewrites\_write:w
\\_\_morewrites\_write\_now:w
\\_\_morewrites\_write\_now:n

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 w:n text is then written to the terminal).

```
276 \cs_new_protected:Npn \__morewrites_write:w
     ł
277
       \s__morewrites
278
       \use_i:nn
279
         { \__morewrites_get_user:n { \__morewrites_write_later:w } }
280
          { \__morewrites_get_user:n { \__morewrites_write_now:w } }
281
282
     }
283
   \cs_new_protected:Npn \__morewrites_write_now:w
284
     {
       \__morewrites_user_to_tstr:NTF \g__morewrites_write_prop
285
         ł
286
           \int_compare:nNnT \l__morewrites_user_int = { 18 } { \use_iii:nnn }
287
           \int_compare:nT { -1 < \l__morewrites_user_int < 16 }</pre>
288
              {
289
                \__morewrites_verbose:n
290
                  ſ
291
                    \tl_to_str:n { \immediate \write }
292
                    \int_use:N \l__morewrites_user_int
293
                  }
294
              7
295
296
            \__morewrites_tex_immediate:w \__morewrites_tex_write:w \l__morewrites_tstr_tl \exp_s
         7
297
         { \primargs_get_general_text:N \__morewrites_write_now:n }
298
     }
299
   \cs_new_protected:Npn \__morewrites_write_now:n #1
300
     {
301
       \prop_get:NVNTF \g__morewrites_write_file_prop \l__morewrites_user_int \l__morewrites_int
302
303
         {
            \__morewrites_verbose:n
304
305
                \tl_to_str:n { \immediate \write }
306
                \int_use:N \l__morewrites_user_int
307
                \tl_to_str:n { ~ {#1} }
308
              7
309
            \__morewrites_write_now_open:n {#1}
310
         }
311
         ł
312
            \__morewrites_verbose:n
313
314
              {
                \tl_to_str:n { \immediate \write }
315
                \int_use:N \l__morewrites_user_int
316
                \tl_to_str:n { ~ (closed) ~ {#1} }
317
              3
318
            \__morewrites_tex_immediate:w \__morewrites_tex_write:w 16 {#1}
319
         }
320
     }
321
```

(End of definition for \\_\_morewrites\_write:w, \\_\_morewrites\_write\_now:w, and \\_\_morewrites\_write\_now:n.)

\\_\_morewrites\_write\_now\_open:n
\ morewrites write now loop:

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  $\varepsilon$ -T<sub>E</sub>X'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.

```
\cs_new_protected:Npn \__morewrites_write_now_open:n #1
322
    {
323
       \bool_if:NF \g__morewrites_tmp_file_bool { \__morewrites_chk_file: }
324
       \__morewrites_tex_immediate:w \__morewrites_tex_openout:w
         \g__morewrites_iow = { \g__morewrites_tmp_file_tl }
326
       \__morewrites_tex_immediate:w \__morewrites_tex_write:w
327
         \g__morewrites_iow {#1}
328
       \__morewrites_tex_immediate:w \__morewrites_tex_closeout:w
329
         \g__morewrites_iow
330
       \group_begin:
         \int_set:Nn \tex_endlinechar:D { -1 }
         \tex_openin:D \g_morewrites_ior = { \g_morewrites_tmp_file_tl }
         \__morewrites_write_now_loop:
334
         \tex_closein:D \g__morewrites_ior
335
         \__morewrites_collect_gput_right:c
336
           { \__morewrites_user_tl_name:n { \l__morewrites_user_int } }
337
       \group_end:
338
       \__morewrites_empty_file:n { \g__morewrites_tmp_file_t1 }
339
    }
340
  \cs_new_protected:Npn \__morewrites_write_now_loop:
341
342
    ł
       \tex_readline:D \g__morewrites_ior to \l__morewrites_internal_tl
343
       \ior_if_eof:NF \g__morewrites_ior
344
         ſ
345
             _morewrites_collect:x
           \_
346
347
             ł
                 _morewrites_tex_immediate:w \__morewrites_tex_write:w
348
349
                 3
350
351
           \__morewrites_write_now_loop:
        }
352
    }
353
```

(End of definition for \\_\_morewrites\_write\_now\_open:n and \\_\_morewrites\_write\_now\_loop:.)

#### 2.3.3 Delayed actions

\\_\_morewrites\_later:n
\\_\_morewrites\_later\_do: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.

```
354 \cs_new_protected:Npn \__morewrites_later:n #1
355 {
356 \int_gincr:N \g__morewrites_later_int
357 \tl_const:cx
358 {
359 c__morewrites_later_
```

```
\int_use:N \g__morewrites_later_int
                        360
                        361
                                    _tl
                                  }
                        362
                                  {
                        363
                                    \int_set:Nn \exp_not:N \l__morewrites_user_int
                        364
                                      { \exp_not:V \l__morewrites_user_int }
                        365
                                    \exp_not:n {#1}
                        366
                                  }
                        367
                                \exp_args:NNx \__morewrites_tex_write:w \g__morewrites_iow
                        368
                                  { '( \int_use:N \g_morewrites_later_int ) }
                        369
                             }
                        370
                        371 \cs_new_protected:Npn \__morewrites_later_do:n #1
                             { \tl_use:c { c__morewrites_later_ \int_eval:n {#1} _tl } }
                        372
                        (End of definition for \__morewrites_later:n and \__morewrites_later_do:n.)
                       If the user stream is a T_FX stream, use the primitive, otherwise save \_morewrites_-
\__morewrites_closeout_later:
                        closeout_now_silent: for later.
                           \cs_new_protected:Npn \__morewrites_closeout_later:
                        373
                             {
                        374
                                \__morewrites_verbose:n { \tl_to_str:n { \closeout (later) ~ } \int_use:N \l__morewrites_
                        375
                                \int_compare:nNnTF \l__morewrites_user_int < { 129 }</pre>
                        376
                                  { \__morewrites_tex_closeout:w \l__morewrites_user_int }
                        377
                                  { \__morewrites_later:n { \__morewrites_closeout_now_silent: } }
                        378
                             }
                        379
                        (End of definition for \__morewrites_closeout_later:.)
                       If the user stream is a T<sub>F</sub>X stream use the primitive, otherwise find a file name and call
\ morewrites openout later:w
\ morewrites openout later:n
                          _morewrites_openout_now_silent:n later.
                        \_
                        380 \cs_new_protected:Npn \__morewrites_openout_later:w
                             ſ
                        381
                                \int_compare:nNnTF \l__morewrites_user_int < { 129 }</pre>
                        382
                                  ł
                        383
                                       _morewrites_verbose:n { \tl_to_str:n { \openout (later) ~ } \int_use:N \l__morewrit
                        384
                                     \__morewrites_tex_openout:w \l__morewrites_user_int
                        385
                                  }
                        386
                                  { \_morewrites_equals_file: \mathbb{N} \_morewrites_openout_later:n }
                        387
                             }
                        388
                           \cs_new_protected:Npn \__morewrites_openout_later:n #1
                        389
                        390
                             ł
                                  _morewrites_verbose:n
                        391
                                  ł
                        392
                                    \tl_to_str:n { \openout (later)~ }
                        393
                                    \int_use:N \l__morewrites_user_int
                        394
                                    \c_space_tl = ~ {#1}
                        395
                                  }
                        396
                                \__morewrites_later:n { \__morewrites_openout_now_silent:n {#1} }
                        397
                             }
                        398
                        (End of definition for \__morewrites_openout_later:w and \__morewrites_openout_later:n.)
```

For  $T_{EX}$  streams use the primitive, otherwise find a general text and save it for later; the auxiliary is very similar to  $\_morewrites\_write\_now:w$ .

```
\cs_new_protected:Npn \__morewrites_write_later:w
300
400
     ł
       \int_compare:nNnTF \l__morewrites_user_int < { 129 }</pre>
401
         ł
402
              _morewrites_verbose:n { \tl_to_str:n { \write (later)~ } \int_use:N \l__morewrites
403
            \__morewrites_tex_write:w \l__morewrites_user_int
404
         }
405
           \primargs_get_general_text:N \__morewrites_write_later:n }
         ſ
406
     }
407
  \cs_new_protected:Npn \__morewrites_write_later:n #1
408
409
     {
410
         _morewrites_verbose:n
         ſ
411
           \tl_to_str:n { \write (later)~ }
412
           \int_use:N \l__morewrites_user_int
413
           \tl_to_str:n { ~ {#1} }
414
         7
415
       \__morewrites_later:n { \__morewrites_write_later_aux:n {#1} }
416
     }
417
   \cs_new_protected:Npn \__morewrites_write_later_aux:n
418
419
     {
         _morewrites_user_to_tstr:NTF \g__morewrites_write_prop
420
           \__morewrites_tex_immediate:w \__morewrites_tex_write:w \l__morewrites_tstr_tl \exp_s
421
         {
422
         ł
           \prop_get:NVNTF \g__morewrites_write_file_prop \l__morewrites_user_int \l__morewrites
423
             { \__morewrites_write_now_open:n }
424
425
             { \__morewrites_tex_immediate:w \__morewrites_tex_write:w 16 \exp_stop_f: }
         }
426
     }
427
```

(End of 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

 $\_morewrites_before_shipout: \langle primitive shipout \rangle \langle collected box \rangle \\ \_morewrites_after_shipout:$ 

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).

(End of definition for \\_\_morewrites\_before\_shipout:.)

Immediately after all the \writes are performed, close the file, then read the file with \endlinechar set to \newlinechar<sup>2</sup> 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.

```
434 \cs_new_protected:Npn \__morewrites_after_shipout:
435
     {
       \__morewrites_tex_immediate:w \__morewrites_tex_closeout:w
436
         \g__morewrites_iow
437
       \group_begin:
438
         \int_set_eq:NN \tex_endlinechar:D \tex_newlinechar:D
439
         \char_set_catcode_other:n { \tex_endlinechar:D }
440
         \tl_map_inline:nn { (0123456789) }
441
           { \char_set_catcode_other:n {'##1} }
442
         \tex_everyeof:D { () \exp_not:N }
443
         \tl_set:Nx \l__morewrites_internal_tl
444
           ł
445
             \exp_after:wN \__morewrites_after_shipout_loop:ww
446
             \tex_input:D { \g__morewrites_tmp_file_tl }
           }
448
         \__morewrites_empty_file:n { \g__morewrites_tmp_file_tl }
449
         \exp_args:NNo
450
       \group_end:
451
       \tl_map_function:nN { \l__morewrites_internal_tl } \__morewrites_later_do:n
452
     }
453
454
  \cs_new:Npn \__morewrites_after_shipout_loop:ww #1 '( #2 )
455
     {
456
       \tl_if_empty:nF {#2}
457
         {
           {#2}
458
              _morewrites_after_shipout_loop:ww
         3
460
     }
461
```

(End of definition for \\_\_morewrites\_after\_shipout: and \\_\_morewrites\_after\_shipout\_loop:ww.)

\\_\_morewrites\_shipout:w
\\_\_morewrites\_shipout\_i:
\\_\_morewrites\_shipout\_ii:

Grab the shipped out box using \setbox and regain control using \afterassignment. There are two cases: either the box is given as \box or \copy followed by a number, in 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.

```
462 \cs_new_protected:Npn \__morewrites_shipout:w
463 {
464 \int_gset_eq:NN \g__morewrites_group_level_int \tex_currentgrouplevel:D
465 \tex_afterassignment:D \__morewrites_shipout_i:
```

<sup>&</sup>lt;sup>2</sup>Note 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.

```
\tex_global:D \tex_setbox:D \g__morewrites_shipout_box
466
    }
467
  \cs_new_protected:Npn \__morewrites_shipout_i:
468
     ſ
469
       \int_compare:nNnTF { \g_morewrites_group_level_int }
470
                         = { \tex_currentgrouplevel:D }
471
         { \__morewrites_shipout_ii: }
472
         { \tex_aftergroup:D \__morewrites_shipout_ii: }
473
     }
474
   \cs_new_protected:Npn \__morewrites_shipout_ii:
475
476
     {
       \__morewrites_before_shipout:
477
       \__morewrites_tex_shipout:w \tex_box:D \g__morewrites_shipout_box
478
       \__morewrites_after_shipout:
479
     }
480
```

(End of definition for \\_\_morewrites\_shipout:w, \\_\_morewrites\_shipout\_i:, and \\_\_morewrites\_shipout\_i:.)

\shipout

\\_\_morewrites\_tex\_shipout:w

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.

```
481 \cs_gset_protected:Npn \__morewrites_tmp:w #1
     {
482
       \cs_if_exist:NF \__morewrites_tex_shipout:w
483
          { \cs_new_eq:NN \__morewrites_tex_shipout:w #1 }
484
       \cs_gset_eq:NN #1 \__morewrites_shipout:w
485
     }
486
   \tl_map_inline:nn
487
488
     {
       \xyrealshipout@
489
       \org@shipout
490
       \PDFSYNCship@ut@ld
491
       \CROP@shipout
492
       \@soORI
493
       \tex_shipout:D
494
       \zwpl@Hship
495
       \o@shipout@TP
496
       \LL@shipout
497
       \Shipout
498
       \GXTorg@shipout
499
       \AtBegShi@OrgShipout
500
       \AtBeginShipoutOriginalShipout
501
       \minidocument@orig@shipout
       \shipout
     }
504
     {
505
       \str_if_eq:eeT
506
         { \cs_meaning:N #1 }
507
          { \token_to_str:N \shipout }
508
509
         {
           \__morewrites_tmp:w #1 }
     }
510
```

```
511 \cs_if_exist:NF \__morewrites_tex_shipout:w
512 {
513 \cs_new_eq:NN \__morewrites_tex_shipout:w \shipout
514 \cs_gset_eq:NN \shipout \__morewrites_shipout:w
515 }
```

#### 2.3.5 Hook at the very end

\\_\_morewrites\_close\_all: At the end of the document, close all the files.

```
516 \cs_new_protected:Npn \__morewrites_close_all:
     {
517
       \prop_map_inline:Nn \g__morewrites_write_prop
518
         ł
519
           \__morewrites_verbose:n { \tl_to_str:n { \immediate \closeout } ##1 ~ (at~end) }
520
           \__morewrites_tex_immediate:w \__morewrites_tex_closeout:w ##2 \scan_stop:
         7
       \prop_gclear:N \g__morewrites_write_prop
       \prop_map_function:NN \g_morewrites_write_file_prop
524
         \__morewrites_closeout_now:nn
       \prop_gclear:N \g__morewrites_write_file_prop
526
     }
527
```

(End of 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.

```
528 \cs_new_protected:Npn \__morewrites_close_all_at_end:nw #1#2 \@@end
529 {
530 \int_compare:nNnTF {#1} > 0
531 { #2 \__morewrites_close_all_at_end:nw { #1 - 1 } }
532 { \__morewrites_close_all: #2 }
533 \@@end
534 }
535 \AtEndDocument { \__morewrites_close_all_at_end:nw { 5 } }
(End of 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. We used to initialize it to 18 so that the first user stream allocated by morewrites was 19. Indeed, 18 is reserved for shell commands and packages may expect 16 or 17 to write to the terminal. This is now changed to start allocation at 129, since some packages that do not want to distinguish LuaT<sub>E</sub>X from other engines simply use 128 as a never-open stream.

```
536 \int_new:N \g__morewrites_alloc_write_int
537 \int_gset:Nn \g__morewrites_alloc_write_int { 128 }
```

```
(End of 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 TFX/IATFX  $2_{\varepsilon}$  use.

```
538 \cs_new_protected:Npn \__morewrites_newwrite:N #1
539
     ſ
       \int_gincr:N \g__morewrites_alloc_write_int
540
       \int_set_eq:NN \allocationnumber \g_morewrites_alloc_write_int
541
       \cs_undefine:N #1
542
       \int_const:Nn #1 { \allocationnumber }
543
       \wlog
544
         ł
545
            \token_to_str:N #1
546
547
           = \token_to_str:N \write \int_use:N \allocationnumber
548
         }
     }
549
```

```
(End of definition for \__morewrites_newwrite:N.)
```

\\_\_morewrites\_allocate:n

:n Raise to #1 the number of <code>\write</code> streams allocated to morewrites.

```
550 \cs_new_protected:Npn \__morewrites_allocate:n #1
551
     {
552
       \prg_replicate:nn
553
         ſ
            \int_max:nn { 0 }
554
555
              {
                (#1) - \seq_count:N \g_morewrites_write_seq
556
                - \prop_count:N \g_morewrites_write_prop
557
              3
558
         }
559
         ł
560
            \__morewrites_tex_newwrite:N \l__morewrites_tstr_token
561
            \seq_gput_right:NV \g__morewrites_write_seq \l__morewrites_tstr_token
562
         }
563
     }
564
```

(End of definition for \\_\_morewrites\_allocate:n.)

## 2.5 User commands and keys

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

```
565 \cs_new_protected:Npn \morewritessetup #1
566 { \keys_set:nn { __morewrites } {#1} }
```

(End of 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.

```
567 \keys_define:nn { __morewrites }
568 {
569 allocate .code:n = \__morewrites_allocate:n {#1} ,
570 file .code:n = \__morewrites_set_file:n {#1} ,
571 file .initial:n = \c_sys_jobname_str .mw ,
572 verbose .bool_set:N = \l__morewrites_verbose_bool
573 }
```

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

\immediate	
\openout	574 \cs_gset_eq:NN \immediate \morewrites_immediate:w
\write	575 \cs_gset_eq:NN \openout \morewrites_openout:w
\closeout	576 \cs_gset_eq:NN \write \morewrites_write:w
\newwrite	577 \cs_gset_eq:NN \closeout \morewrites_closeout:w
	578 \cs_gset_eq:NN \newwrite \morewrites_newwrite:N
	(End of definition for \immediate and others. These functions are documented on page 3.)

# 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	exp commands:
\\ 109, 110, 111	$\exp_{\text{after:wN}} \dots \dots \dots \dots \dots \dots 18, 440$
٨	$\exp_{args:NNc}$ 18
A	$\exp_{args:NNf} \dots 1'$
\A 92	$\exp_{args:NNo}$
allocate	\exp_args:NNx 368
\allocationnumber 541, 543, 547	\exp_args:No 273
\AtBeginShipoutOriginalShipout 501	\exp_not:N 364, 443
\AtEndDocument 535	\exp_not:n 365, 366 \exp_stop_f: 18, 204, 255, 296, 421, 423
В	(exp_500p_1. 10, 204, 200, 200, 421, 420
bool commands:	$\mathbf{F}$
\bool_gset_false:N 39, 46	fi commands:
\bool_gset_true:N 80	\fi: 69, 70
\bool_if:NTF 43, 324, 430	file 2, <u>56</u>
\bool_new:N 25, 38	111e · · · · · · · · · · · · · · · · · ·
box commands:	G
$box_new:N$ 41	group commands:
_	\group_begin: 58, 116, 219, 331, 438
С	\group_end: 72, 119, 223, 338, 45.
char commands:	(group_ond: 12, 115, 226, 666, 15
$\char_set_catcode_other:n 440, 442$	Ι
\closeout 3, 197, 375, 520, <u>574</u>	if commands:
cs commands:	\if_eof:w 60, 6
\cs_generate_variant:Nn	\immediate 3,
$\dots \dots \dots \dots \dots \dots \dots \dots \dots \dots 19, 20, 21, 156, 171$	182, 197, 237, 292, 306, 315, 520, <u>57</u>
$cs_gset_eq:NN \dots 61,$	int commands:
66,68,485,514,574,575,576,577,578	\int_compare:nNnTF . 130, 142, 159,
\cs_gset_protected:Npn 269, 481	205, 246, 287, 376, 382, 401, 470, 530
\cs_if_exist:NTF 483, 511	\int_compare:nTF 288
$cs_meaning:N \dots 507$	\int_const:Nn 543
\cs_new:Npn 172, 454	\int_decr:N 16
$cs_{new_eq:NN} \dots 13,$	_
14, 15, 16, 24, 33, 34, 268, 484, 513	\int_eval:n 97, 153, 173, 373
\cs_new_protected:Npn 6,	\int_gincr:N 356, 540
17, 42, 44, 49, 75, 89, 114, 121, 127,	\int_gset:Nn 53'
135, 140, 157, 174, 176, 186, 188,	\int_gset_eq:NN 464
195, 200, 216, 226, 233, 243, 276,	\int_max:nn 554
283, 300, 322, 341, 354, 371, 373,	\int_new:N 26, 31, 40, 134, 530
380, 389, 399, 408, 418, 428, 434,	\int_set:Nn 63, 145, 220, 332, 364
462, 468, 475, 516, 528, 538, 550, 565	\int_set_eq:NN 439, 54
\cs_undefine:N 542	\int_use:N
·	$\ldots$ 165, 197, 238, 293, 307, 316,
D	360, 369, 375, 384, 394, 403, 413, 54
\D 92	ior commands:
\d 92	\ior_if_eof:NTF 34-
$\mathbf{E}$	\iow_char:N 110
else commands:	\iow_indent:n 110
	,

\iow_term:n															43
$c_term_iow$	•	•		•	•		•	•	•	•	•		•	•	6

# K

17	
keys commands:	
\keys_define:nn	567
\keys_set:nn	566

# $\mathbf{M}$

141
morewrites internal commands:
\morewrites_after_shipout:
\morewrites_after_shipout
loop:ww 19, <u>434</u> , 446, 454, 459
\gmorewrites_alloc_write_int
22, 536, 540, 541
\morewrites_allocate:n <u>550</u> , 550, 569
<pre>\morewrites_before_shipout:</pre>
18, 428, 428, 477
\morewrites_chk_file:
$ \underline{75}, 75, 86, 324, 430$
$\_\_morewrites\_chk_file_aux:$ . 82, 89
$\_\_morewrites\_close\_all: 516, 516, 532$
\morewrites_close_all_at
end:nw $\dots \dots \dots 528, 528, 531, 535$
$\sum morewrites_closeout:w 188, 188, 577$
\morewrites_closeout_later:
$ 192, \frac{373}{373}, 373$
\morewrites_closeout_now:
\morewrites_closeout_now:nn
$13, \underline{188}, 213, 216, 525$
\morewrites_closeout_now
silent: 17, <u>188</u> , 198, 200, 245, 378
\1morewrites_code_t1 <u>30</u> , 123, 124
\morewrites_collect:n <u>134</u> , 135, 346
\morewrites_collect_aux:Nn
\morewrites_collect_gput
right:N <u>134</u> , 157, 168, 171, 336
$\begin{array}{c} \text{IIght: N} & \dots & \underline{104}, 107, 100, 171, 300 \\ \end{array}$
\lmorewrites_collect_next_int .
<i>10</i> , <u>134</u>
\morewrites_empty_file:n
\morewrites_equals_file:N
<u>114</u> , 114, 231, 387
$\sum$ _morewrites_get_user:N 7
$\sum$ _morewrites_get_user:n $6$ ,
$\underline{121}, 121, 192, 193, 230, 231, 280, 281$
\gmorewrites_group_level_int
40, 464, 470
$\sum$ _morewrites_if_file_trivial:n . 56
\morewrites_if_file_trivial:nTF
$\underbrace{}_{}$
/

\morewrites_immediate:w
174, 174, 574
\morewrites_immediate_auxii:
<u>174</u> , 175, 176
\morewrites_immediate_auxiii:N
<u></u> <u>174</u> , 179, 186
\lmorewrites_internal_seq
<u></u>
\1morewrites_internal_t1
22, 64, 84, 85, 94, 94, 94, 94, 94, 94, 94, 94, 94, 94
101, 137, 148, 150, 212, 213, 271,
$\begin{array}{c} 101, 137, 148, 150, 212, 213, 211, \\ 272, 273, 302, 343, 349, 423, 444, 452 \end{array}$
<pre>\gmorewrites_ior 7, 35, 59, 60, 64, 65, 71, 333, 335, 343, 344</pre>
\gmorewrites_iow
$\ldots$ 7, 12, 13, 18, <u>35</u> , 52, 54, 218,
224, 326, 328, 330, 349, 368, 432, 437
\morewrites_later:n
$\dots \dots \dots \dots \dots \underline{354}, 354, 378, 397, 416$
\morewrites_later_do:n $\underline{354},371,452$
\gmorewrites_later_int
$\dots \dots $
\morewrites_newwrite:N <u>538</u> , 538, 578
$\sum morewrites_openout:w 226, 226, 575$
\morewrites_openout_later:n
<u></u>
\morewrites_openout_later:w
<b>..</b>
\morewrites_openout_now:n
<u>226</u> , 231, 233
\morewrites_openout_now
silent:n
17, 226, 241, 243, 268, 269, 397
\morewrites_openout_now
silent_aux:n 268, 273
\morewrites_set_file:n
$22, \underline{44}, 44, 570$
<pre>\morewrites_shipout:w</pre>
$20, \underline{462}, 462, 485, 514$
\gmorewrites_shipout_box
$ 19, \underline{41}, 466, 478$
\morewrites_shipout_i:
$\dots \dots 19, \underline{462}, 465, 468$
\morewrites_shipout_ii:
462, 472, 473, 475
$\_morewrites_tex_closeout:w$ 13,
16, 53, 204, 224, 329, 377, 436, 521
<pre>\morewrites_tex_immediate:w</pre>
$\dots \dots 13, 13, 51, 53, 183, 204,$
218, 224, 248, 255, 296, 319, 325,
327, 329, 348, 421, 425, 431, 436, 521
\morewrites_tex_newwrite:N
/

```
\_\_morewrites\_tex\_openout:w . <u>13</u>,
   14, \ 51, \ 218, \ 248, \ 255, \ 325, \ 385, \ 431
\__morewrites_tex_shipout:w ....
   \dots \dots \dots 478, \underline{481}, 483, 484, 511, 513
\_morewrites\_tex\_write:w 13, 15,
   296,\ 319,\ 327,\ 348,\ 368,\ 404,\ 421,\ 425
\__morewrites_tmp:w .....
   \ldots 8, <u>24</u>, 24, 61, 66, 68, 73, 481, 509
\g__morewrites_tmp_file_bool ...
   \dots \dots \dots \dots \dots \dots \underline{37}, 46, 80, 324, 430
\g__morewrites_tmp_file_tl ....
   ..... 7–9, <u>37</u>, 47, 77, 78, 79, 84,
   85, 92, 101, 326, 333, 339, 432, 447, 449
\l__morewrites_tstr_tl .....
    ..... 7, 10, <u>31</u>, 129, 132,
   204, 205, 208, 252, 254, 255, 296, 421
l\_morewrites\_tstr\_token 33, 561, 562
\l__morewrites_user_int 7, 10, <u>31</u>,
   125, 129, 130, 132, 197, 205, 207,
   212, 213, 238, 246, 248, 254, 260,
   262, 287, 288, 293, 302, 307, 316,
   337, 364, 365, 375, 376, 377, 382,
   384, 385, 394, 401, 403, 404, 413, 423
\__morewrites_user_tl_name:n ...
   ..... 16, <u>172</u>, 172, 221, 222, 262, 337
\__morewrites_user_to_tstr:NTF ...
   ..... <u>127</u>, 127, 202, 285, 420
\__morewrites_verbose:n .....
    ..... <u>42</u>, 42, 181, 197, 235, 290,
   304, 313, 375, 384, 391, 403, 410, 520
\l__morewrites_verbose_bool ....
   \dots \underline{25}, 43, 572
\__morewrites_write:w . <u>276</u>, 276, 576
\g__morewrites_write_file_prop ..
    11, 13, <u>29</u>, 212, 260, 302, 423, 524, 526
\__morewrites_write_later:n ....
   .... 399, 406, 408
\__morewrites_write_later:w ....
   \__morewrites_write_later_aux:n .
   \__morewrites_write_now:n .....
   .... <u>276</u>, 298, 300
\__morewrites_write_now:w .....
   ..... 17, <u>276</u>, 281, 283
\__morewrites_write_now_loop: ...
   ..... <u>322</u>, 334, 341, 351
\__morewrites_write_now_open:n ...
   ..... 15, 310, <u>322</u>, 322, 424
\g__morewrites_write_prop .....
   .... 10, 11, 13, 28,
   202, 207, 254, 285, 420, 518, 523, 557
\l__morewrites_write_prop ..... 4
```

\gmorewrites_write_seq
$\dots \dots \dots \dots \dots \underline{27}, 208, 252, 556, 562$
\morewritessetup 2, 22, 6, <u>565</u>
msg commands:
\msg_new:nnn 7
\msg_new:nnnn 103
\msg_warning:nn 9
\msg_warning:nnnn 83
my commands:
\l_my_box 6
Ν
\newread 36
\newwrite 2, 35, <u>574</u>
0
\openout <i>3</i> , 237, 384, 393, <u>574</u>
7

#### Ρ

prg commands:
\prg_new_conditional:Npnn 56
\prg_replicate:nn 552
\prg_return_false:
\prg_return_true: 8, 61, 66
primargs commands:
\primargs_get_general_text:N 298, 406
\primargs_get_input_file_name:N 117
\primargs_get_number:N 10
$\primargs_read_x_token: N \dots 12, 175$
$\primargs_remove_equals:N \dots 119$
\g_primargs_token 178, 182
prop commands:
\prop_count:N 557
\prop_gclear:N 523, 526
\prop_get:NnNTF 132, 302, 423
\prop_gpop:NnNTF 19, 212
\prop_gput:Nnn 20, 254, 260
\prop_gremove:Nn 207
$prop_map_function:NN \dots 524$
\prop_map_inline:Nn 518
\prop_new:N 28, 29
\ProvidesExplPackage 2

## R

regex commands:	
<pre>\regex_extract_once:nnNTF</pre>	91
\RequirePackage	1

# $\mathbf{S}$

scan commands:
$scan_new:N \dots 7$
\scan_stop: 7, 34, 52, 521
scan internal commands:
\smorewrites
$\dots$ 12, <u>34</u> , 178, 187, 190, 228, 278

seq	commands:
-----	-----------

$seq_count:N \dots 556$
$seq_gpop:NNTF \dots 252$
\seq_gput_left:Nn 208
\seq_gput_right:Nn 562
\seq_item:Nn 96, 97
$seq_new:N \dots 23, 27$
\Shipout 498
\shipout 3, <u>481</u>
str commands:
\str_if_eq:nnTF 187, 506
sys commands:
\sys_if_engine_luatex:TF 4
\sys_if_engine_xetex:TF 266
\c_sys_jobname_str 2, 8, 571

# т

$\mathbf{T}$
TEX and LATEX $2\varepsilon$ commands:
\@@end $21, 528, 533$
\@soORI 493
$afterassignment \dots 19$
\aftergroup 19
\AtBegShi@OrgShipout 500
\box $5, 19$
\chardef $\dots \dots 5$
\closeout
\copy <i>6</i> , <i>19</i>
$count \dots 4$
CROP@shipout
\endlinechar 16, 19
\GXTorg@shipout $\dots \dots \dots \dots 499$
\hbox 5, 19
\ifeof 4
\immediate 3, 5, 8, 12, 13, 16
\jobname $2$
$LL@shipout \dots 497$
$\min document@orig@shipout \dots 502$
\newlinechar 16, 19
\newwrite 2-5, 21, 22
\noexpand 12
$\notexpanded: \dots 4$
o@shipout@TP 496
\openout 3-5, 8, 10, 12
$\operatorname{Org}$
\outer $5$
\pdfobj 12
\PDFSYNCship@ut@ld $\dots \dots 491$
\read $\ldots$ 4
\readline 16
\relax 4
\setbox 19
\shipout 3, 5-7, 12, 18-20
\toks 4
\vtop 19

\write 1-5, 7, 10-13, 16, 18, 19, 22
xyrealshipout@ 489
\zwpl@Hship 495
tex commands:
$tex_afterassignment:D \dots 124, 465$
\tex_aftergroup:D 117, 118, 473
\tex_box:D 478
\tex_closein:D
\tex_closeout:D 16
\tex_currentgrouplevel:D 464, 471
\tex_endinput:D 10
\tex_endlinechar:D 63, 332, 439, 440
\tex_everyeof:D 443
\tex_global:D 466
\tex_immediate:D 13
\tex_input:D 447
\tex_newlinechar:D 18, 220, 439
\tex_openin:D
\tex_openout:D 14
\tex_readline:D 64, 343
\tex_setbox:D 466
\tex_shipout:D 494
\tex_write:D 15
tl commands:
\c_space_tl 239, 395
\tl_clear:N 151 \tl clear new:N 144
\tl_const:Nn 357
\tl_gclear:N 222
\tl_gclear_new:N 262
\tl_gput_right:Nn 21, 162
\tl_gset:Nn 47,77
\tl_gset_eq:NN 85
\tl_if_empty:NTF 147
\tl_if_empty:nTF 456
\t1_map_function:nN 452
\tl_map_inline:nn 441, 487
\tl_new:N 22, 30, 32, 37
\tl_put_left:Nn 150
\tl_remove_all:Nn 272
\tl_set:Nn
94, 101, 123, 129, 137, 271, 444
\tl_set_eq:NN 148
\tl_to_str:n 78, 182, 197, 237,
249, 256, 261, 292, 306, 308, 315, 217, 275, 284, 202, 402, 412, 414, 520
317, 375, 384, 393, 403, 412, 414, 520
\tl_use:N 221, 372
token commands:
\token_if_eq_meaning:NNTF 178
\token_to_meaning:N 182
$token_to_{str:N} \dots 508, 546, 547$

$\mathbf{U}$	verbose
use commands:	
\use_i:nn <i>12</i> , 131, 191, 229, 279	$\mathbf{W}$
\use_iii:nnn 187, 287	\wlog 544
	\write . 3, 292, 306, 315, 403, 412, 547, <u>574</u>
$\mathbf{V}$	
vbox commands:	$\mathbf{Z}$
$vbox_set:Nn \dots 6$	\Z 92