

# The l3flag package: expandable flags\*

The L<sup>A</sup>T<sub>E</sub>X3 Project<sup>†</sup>

Released 2015/06/15

Flags are the only data-type on which T<sub>E</sub>X can perform assignments in expansion-only contexts. This module is meant mostly for kernel use: in almost all cases, booleans or integers should be preferred to flags, because they are faster.

A flag can hold any non-negative value, which we call its *height*. In expansion-only contexts, a flag can only be “raised”: this normally increases the *height* by 1, but can be configured by defining specific traps. The *height* can also be queried expandably. However, decreasing it, or setting it to zero requires non-expandable assignments.

Flag variables are always local. They are referenced by a *name* of the form *package\_flag name*, for instance, `str_missing`.

## 1 Setting up flags

---

<code>\flag_new:n</code>	<code>\flag_new:n {&lt;flag name&gt;}</code>
--------------------------	--

Creates a new *flag* with a name given by *flag name*, or raises an error if the name is already taken. The *flag name* must consist of character tokens only. The declaration is global, but flags are always local variables. The *flag* will initially have zero height.

---

<code>\flag_clear:n</code>	<code>\flag_clear:n {&lt;flag name&gt;}</code>
----------------------------	--

The *flag*’s height is set to zero. The assignment is local.

---

<code>\flag_clear_new:n</code>	<code>\flag_clear_new:n {&lt;flag name&gt;}</code>
--------------------------------	--

Ensures that the *flag* exists globally by applying `\flag_new:n` if necessary, then applies `\flag_zero:n`, setting the height to zero locally.

---

<code>\flag_set_trap:nn</code>	<code>\flag_set_trap:nn {&lt;flag name&gt;} {&lt;inline function&gt;}</code>
--------------------------------	--

Changes the action that is taken when the *flag* is raised using `\flag_raise:n`. Instead of the default action which is to increase the *flag*’s height by 1, the *inline function* will be called, receiving the current flag’s height as `#1`. The *inline function* should expand to nothing; *e.g.*, it could call `\msg_expandable_error:n`. This function is very experimental.

---

\*This file describes v5630, last revised 2015/06/15.

<sup>†</sup>E-mail: [latex-team@latex-project.org](mailto:latex-team@latex-project.org)

## 2 Expandable flag commands

<hr/> <code>\flag_if_exist_p:n</code> ★	<code>\flag_if_exist:n {⟨flag name⟩}</code>
<code>\flag_if_exist:nTF</code> ★	This function returns <b>true</b> if the $\langle flag \text{ name} \rangle$ references a flag that has been defined previously, and <b>false</b> otherwise.
<hr/> <code>\flag_if_raised_p:n</code> ★	<code>\flag_if_raised:n {⟨flag name⟩}</code>
<code>\flag_if_raised:nTF</code> ★	This function returns <b>true</b> if the $\langle flag \rangle$ has non-zero height, and <b>false</b> if the $\langle flag \rangle$ has zero height.
<hr/> <code>\flag_height:n</code> ★	<code>\flag_height:n {⟨flag name⟩}</code>
	Expands to the height of the $\langle flag \rangle$ as an integer denotation.
<hr/> <code>\flag_raise:n</code> ★	<code>\flag_raise:n {⟨flag name⟩}</code>
	The $\langle flag \rangle$ 's trap is performed, taking the current height as its argument. The default behaviour is to increase the $\langle flag \rangle$ 's height by 1 locally. This function is expandable, as long as the trap is expandable (the default trap is expandable, despite being an assignment).

## 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.

<b>F</b>		<code>\flag_if_raised:nTF</code> ..... 2
flag commands:		<code>\flag_if_raised_p:n</code> ..... 2
<code>\flag_clear:n</code> ..... 1, 1		<code>\flag_new:n</code> ..... 1, 1, 1
<code>\flag_clear_new:n</code> ..... 1, 1		<code>\flag_raise:n</code> ..... 1, 2, 2
<code>\flag_height:n</code> ..... 2, 2		<code>\flag_set_trap:nn</code> ..... 1, 1
<code>\flag_if_exist:n</code> ..... 2		<code>\flag_zero:n</code> ..... 1
<code>\flag_if_exist:nTF</code> ..... 2		
<code>\flag_if_exist_p:n</code> ..... 2		<b>M</b>
<code>\flag_if_raised:n</code> ..... 2	msg commands:	<code>\msg_expandable_error:n</code> ..... 1