The tokgroupmark Package
Version 1.0

Alceu Frigeri*

February 2026

Abstract

This package aims at solving the problem of how to preserve some tokens beyond a variable number of nested groups.

Contents

1 Introduction

When it's necessary to preserve the value of some tokens beyond a local group, in simple cases, it's enough to use `\group_insert_after:N` or `\aftergroup`.

But, sometimes you don't have (or can't use) this information, see [1], for instance. For those cases, this package deploys one of the strategies presented and compared in [2], based on `\aftergroup`.

2 Custom Group Mark Commands

`\groupmark_new:n` `\groupmark_new:n` {⟨mark-prefix⟩}

This will globally create two commands (below), named after ⟨mark-prefix⟩, to "mark" and "restore" a token up to that "mark".

Note: Internally, an unique integer will be created to track the group mark.

Note: An error will be raised if ⟨mark-prefix⟩ is already used.

`\<mark-prefix>_groupmark:` `\<mark-prefix>_groupmark:`
`\<mark-prefix>_aftergroup:N` `\<mark-prefix>_aftergroup:N` {⟨token⟩}

The `\<mark-prefix>_groupmark:` will save, local assignment, the current group level, so that, when using `\<mark-prefix>_aftergroup:N` the token will be restored once the same group level is reached once again (similar to `\aftergroup`, but going beyond a single group).

Note: It is possible to have multiple, nested marks, associated with the same set, see example of use.

Note: If `\<mark-prefix>_aftergroup:N` is called at the same level (or above) of `\<mark-prefix>_groupmark:`, it will behave as a simple `\group_insert_after:N`

*https://github.com/alceu-frigeri/tokgroupmark

3 Examples of Use

In the following examples, two sets of mark commands will be used, ⟨myMarkA⟩ and ⟨myMarkB⟩.

```
\ExplSyntaxOn
  % Just a set of booleans for testing
  \bool_new:N \l__mytest_tmpa_bool
  \bool_new:N \l__mytest_tmpb_bool
  \bool_new:N \l__mytest_tmpc_bool
  \cs_new:Npn \mytest_show_bools:n #1
    { \underline{#1:}\par
      \bool_if:NTF \l__mytest_tmpa_bool {{\color{red}a~true}}{a~false}  ~~~
      \bool_if:NTF \l__mytest_tmpb_bool {{\color{red}b~true}}{b~false}  ~~~
      \bool_if:NTF \l__mytest_tmpc_bool {{\color{red}c~true}}{c~false}  \par
    }

  \groupmark_new:n {myMarkA}
  \groupmark_new:n {myMarkB}
\ExplSyntaxOff
```

3.1 Trivial Case, Single Mark

Note that, despite the group level at which the booleans were saved, they are all restored at the same group level, *T3* in the example below.

```
\ExplSyntaxOn
\group_begin:
  \myMarkA_groupmark:
  {{  \myMarkA_aftergroup:N \bool_set_true:N
      \myMarkA_aftergroup:N \l__mytest_tmpa_bool
    {
      \myMarkB_groupmark:  %dummy, not used...
     {{ \myMarkA_aftergroup:N \bool_set_true:N
        \myMarkA_aftergroup:N \l__mytest_tmpb_bool
        \mytest_show_bools:n {T1}
     }}
      \mytest_show_bools:n {T2}
    }
  }}
  \mytest_show_bools:n {T3}
\group_end:
\ExplSyntaxOff
```

T1:
a false - b false - c false
T2:
a false - b false - c false
T3:
a true - b true - c false

3.2 Multiple Marks, Single Command Set

Note that `\myMarkA_groupmark:` is called twice in the example below, effectively defining two, nested, restoring points.

```
\ExplSyntaxOn
\group_begin:
  \myMarkA_groupmark:
  {{  \myMarkA_aftergroup:N \bool_set_true:N
      \myMarkA_aftergroup:N \l__mytest_tmpa_bool
    {
      \myMarkA_groupmark:  %second call !
     {{ \myMarkA_aftergroup:N \bool_set_true:N
        \myMarkA_aftergroup:N \l__mytest_tmpb_bool
        \mytest_show_bools:n {T1}
     }}
      \mytest_show_bools:n {T2}
    }
  }}
  \mytest_show_bools:n {T3}
\group_end:
\ExplSyntaxOff
```

T1:
a false - b false - c false
T2:
a false - b true - c false
T3:
a true - b false - c false

3.3 Multiple Command Sets

This variant is less prone to errors than the previous one, when setting various, nested, restoring points.

```
\ExplSyntaxOn
\group_begin:
  \myMarkA_groupmark:
  {{  \myMarkA_aftergroup:N \bool_set_true:N
      \myMarkA_aftergroup:N \l__mytest_tmpa_bool
    {
      \myMarkB_groupmark:
     {{ \myMarkB_aftergroup:N \bool_set_true:N
        \myMarkB_aftergroup:N \l__mytest_tmpb_bool
        \mytest_show_bools:n {T1}
     }}
      \mytest_show_bools:n {T2}
    }
  }}
  \mytest_show_bools:n {T3}
\group_end:
\ExplSyntaxOff
```

T1:
a false - b false - c false
T2:
a false - b true - c false
T3:
a true - b false - c false

References

[1] David Carlisle. *Stackexchange about grouping*. 2026. URL: https://tex.stackexchange.com/questions/757755/coffins-scope-groups#comment1889872_757755 (visited on 01/01/2026).

[2] Alceu Frigeri. *The xstacks package*. 2026. URL: https://ctan.org/pkg/xstacks (visited on 02/18/2026).