The environmentoptional package*

Sergio Martinez-Losa

February 14, 2026

1 Introduction

The `environmentoptional` package provides an alternative to LaTeX's standard `\newenvironment` command, allowing you to define environments with a variable number of optional and mandatory arguments.

2 Usage

`\newenvironmentoptionals` The primary command is `\newenvironmentoptionals`:

> `\newenvironmentoptionals{`⟨*name*⟩`}[`⟨*n_ opt*⟩`][`⟨*n_ mand*⟩`]{`⟨*begin code*⟩`}{`⟨*end code*⟩`}`

where ⟨*n_ opt*⟩ is the number of optional arguments and ⟨*n_ mand*⟩ the number of mandatory arguments.

`\beginoptionals` To use such an environment, use:

> `\beginoptionals{`⟨*name*⟩`}[`⟨*opt1*⟩`]...{`⟨*mand1*⟩`}...`

`\endoptionals` And close it with:

> `\endoptionals{`⟨*name*⟩`}`

3 Implementation

```
\RequirePackage{xparse}

\ExplSyntaxOn

% --- Global Metadata Storage ---
\prop_new:N \g__envopt_metadata_prop

% --- Variables ---
```

*This document corresponds to environmentoptional v1.14, dated 2026/02/13.

```
\int_new:N \l__envopt_opt_count_int
\int_new:N \l__envopt_mand_count_int
\int_new:N \l__envopt_total_count_int
\tl_new:N \l__envopt_name_tl
\tl_new:N \l__envopt_arg_spec_tl
\tl_new:N \l__envopt_m_spec_tl
\tl_new:N \l__envopt_exec_tl
\tl_new:N \l__envopt_begin_code_tl
\tl_new:N \l__envopt_end_code_tl

\seq_new:N \l__envopt_def_vals_seq
\seq_new:N \l__envopt_values_seq

% --- Error messages ---
\msg_new:nnn { environmentoptional } { name-mismatch }
{
  Environment~name~mismatch.~Expected~'#1',~but~got~'#2'~in~\token_to_str:N \endoptionals.
}

\msg_new:nnn { environmentoptional } { too-many-args }
{
  The~sum~of~optional~(#1)~and~mandatory~(#2)~arguments~must~be~maximum~9.
}

% --- \newenvironmentoptionals Implementation ---

\NewDocumentCommand{\newenvironmentoptionals}{ m O{0} O{0} }
{
  \tl_set:Nn \l__envopt_name_tl { #1 }
  \int_set:Nn \l__envopt_opt_count_int { #2 }
  \int_set:Nn \l__envopt_mand_count_int { #3 }
  \int_set:Nn \l__envopt_total_count_int { #2 + #3 }

  \int_compare:nNnTF { \l__envopt_total_count_int } > { 9 }
  {
    \msg_error:nnxx { environmentoptional } { too-many-args }
    { \int_use:N \l__envopt_opt_count_int }
    { \int_use:N \l__envopt_mand_count_int }
  }
  {
    \prop_gput:Nnn \g__envopt_metadata_prop { #1 } { #2 : #3 }
    \seq_clear:N \l__envopt_def_vals_seq
    \__envopt_scan_opt_defaults:n { #2 }
  }
}

\cs_new_protected_nopar:Npn \__envopt_scan_opt_defaults:n #1
{
  \int_compare:nNnTF { #1 } > { 0 }
  {
```

```
    \peek_meaning_ignore_spaces:NTF [
      { \__envopt_grab_opt_default:n { #1 } }
      {
        \seq_put_right:Nn \l__envopt_def_vals_seq { }
        \__envopt_scan_opt_defaults:n { \int_eval:n { #1 - 1 } }
      }
  }
  {
    \__envopt_read_env_code:w
  }
}

\NewDocumentCommand{ \__envopt_grab_opt_default:n } { m r[] }
{
  \seq_put_right:Nn \l__envopt_def_vals_seq { #2 }
  \__envopt_scan_opt_defaults:n { \int_eval:n { #1 - 1 } }
}

\NewDocumentCommand{ \__envopt_read_env_code:w } { +m +m }
{
  \tl_set:Nn \l__envopt_begin_code_tl { #1 }
  \tl_set:Nn \l__envopt_end_code_tl { #2 }
  \__envopt_define_all:
}

\cs_new_protected_nopar:Npn \__envopt_define_all:
{
  \tl_clear:N \l__envopt_arg_spec_tl
  \tl_clear:N \l__envopt_m_spec_tl

  \seq_map_inline:Nn \l__envopt_def_vals_seq
  {
    \tl_put_right:Nn \l__envopt_arg_spec_tl { O { ##1 } }
    \tl_put_right:Nn \l__envopt_m_spec_tl { m }
  }

  \int_step_inline:nn { \l__envopt_mand_count_int }
  {
    \tl_put_right:Nn \l__envopt_arg_spec_tl { m }
    \tl_put_right:Nn \l__envopt_m_spec_tl { m }
  }

  % 1. Define Backing Begin command (m...m)
  \use:x {
    \exp_not:N \DeclareDocumentCommand \exp_not:c { envopt@ \tl_use:N \l__envopt_name_tl @begi
    { \exp_not:V \l__envopt_m_spec_tl }
    { \exp_not:V \l__envopt_begin_code_tl }
  }

  % 2. Define Backing End command
```

```
  \use:x {
    \exp_not:N \DeclareDocumentCommand \exp_not:c { envopt@ \tl_use:N \l__envopt_name_tl @end }
    { }
    { \exp_not:V \l__envopt_end_code_tl }
  }

  % 3. Build Wrapper Body: \backing {#1}{#2}...
  \tl_set:Nn \l_tmpa_tl { \use:c { envopt@ \tl_use:N \l__envopt_name_tl @begin } }

  \int_compare:nNnTF { \l__envopt_total_count_int } > { 0 } { \tl_put_right:Nn \l_tmpa_tl { { #
  \int_compare:nNnTF { \l__envopt_total_count_int } > { 1 } { \tl_put_right:Nn \l_tmpa_tl { { #
  \int_compare:nNnTF { \l__envopt_total_count_int } > { 2 } { \tl_put_right:Nn \l_tmpa_tl { { #
  \int_compare:nNnTF { \l__envopt_total_count_int } > { 3 } { \tl_put_right:Nn \l_tmpa_tl { { #
  \int_compare:nNnTF { \l__envopt_total_count_int } > { 4 } { \tl_put_right:Nn \l_tmpa_tl { { #
  \int_compare:nNnTF { \l__envopt_total_count_int } > { 5 } { \tl_put_right:Nn \l_tmpa_tl { { #
  \int_compare:nNnTF { \l__envopt_total_count_int } > { 6 } { \tl_put_right:Nn \l_tmpa_tl { { #
  \int_compare:nNnTF { \l__envopt_total_count_int } > { 7 } { \tl_put_right:Nn \l_tmpa_tl { { #
  \int_compare:nNnTF { \l__envopt_total_count_int } > { 8 } { \tl_put_right:Nn \l_tmpa_tl { { #

  \use:x {
    \exp_not:N \DeclareDocumentEnvironment { \tl_use:N \l__envopt_name_tl } { \exp_not:V \l__en
    {
       \exp_not:V \l_tmpa_tl
    }
    {
       \exp_not:N \use:c { envopt@ \tl_use:N \l__envopt_name_tl @end }
    }
  }
}

% --- \beginoptionals Implementation ---

\NewDocumentCommand{\beginoptionals}{ m }
{
  \tl_set:Nn \l__envopt_name_tl { #1 }
  \prop_get:NnNTF \g__envopt_metadata_prop { #1 } \l_tmpa_tl
  {
    \exp_after:wN \__envopt_split_counts:w \l_tmpa_tl \q_stop
    \int_set:Nn \l__envopt_total_count_int { \l__envopt_opt_count_int + \l__envopt_mand_count_i
    \seq_clear:N \l__envopt_values_seq
    \__envopt_scan_args_opt:n { \l__envopt_opt_count_int }
  }
  {
    \__envopt_grab_counts:w
  }
}

\NewDocumentCommand{ \__envopt_grab_counts:w } { O{0} O{0} }
{
  \int_set:Nn \l__envopt_opt_count_int { #1 }
```

```
  \int_set:Nn \l__envopt_mand_count_int { #2 }
  \int_set:Nn \l__envopt_total_count_int { #1 + #2 }
  \seq_clear:N \l__envopt_values_seq
  \__envopt_scan_args_opt:n { \l__envopt_opt_count_int }
}

\cs_new:Npn \__envopt_split_counts:w #1 : #2 \q_stop
{
  \int_set:Nn \l__envopt_opt_count_int { #1 }
  \int_set:Nn \l__envopt_mand_count_int { #2 }
}

\cs_new_protected_nopar:Npn \__envopt_scan_args_opt:n #1
{
  \int_compare:nNnTF { #1 } > { 0 }
  {
    \__envopt_grab_arg_opt:n { #1 }
  }
  {
    \__envopt_scan_args_mand:n { \l__envopt_mand_count_int }
  }
}

\NewDocumentCommand{ \__envopt_grab_arg_opt:n } { m r[] }
{
  \seq_put_right:Nn \l__envopt_values_seq { #2 }
  \__envopt_scan_args_opt:n { \int_eval:n { #1 - 1 } }
}

\cs_new_protected_nopar:Npn \__envopt_scan_args_mand:n #1
{
  \int_compare:nNnTF { #1 } > { 0 }
  {
    \__envopt_grab_arg_mand:n { #1 }
  }
  {
    \__envopt_read_body_capture:w
  }
}

\NewDocumentCommand{ \__envopt_grab_arg_mand:n } { m m }
{
  \seq_put_right:Nn \l__envopt_values_seq { #2 }
  \__envopt_scan_args_mand:n { \int_eval:n { #1 - 1 } }
}

\cs_new_protected_nopar:Npn \__envopt_read_body_capture:w #1 \endoptionals #2
{
  \str_if_eq:VnTF \l__envopt_name_tl { #2 }
  {
```

```
    \__envopt_execute:n { #1 }
  }
  {
    \msg_error:nnxx { environmentoptional } { name-mismatch }
    { \tl_use:N \l__envopt_name_tl }
    { #2 }
  }
}

\cs_new_protected_nopar:Npn \__envopt_execute:n #1
{
  \cs_if_exist:cTF { envopt@\tl_use:N \l__envopt_name_tl @begin }
  {
    \tl_set:Nn \l__envopt_exec_tl { \use:c { envopt@\tl_use:N \l__envopt_name_tl @begin } }
    \seq_map_inline:Nn \l__envopt_values_seq
    {
       \tl_put_right:Nn \l__envopt_exec_tl { { ##1 } }
    }
    \l__envopt_exec_tl
    #1
    \use:c { envopt@\tl_use:N \l__envopt_name_tl @end }
  }
  {
    \tl_clear:N \l__envopt_arg_spec_tl
    \int_step_inline:nn { \l__envopt_total_count_int } { \tl_put_right:Nn \l__envopt_arg_spec_t

    \use:x {
       \exp_not:N \DeclareDocumentCommand \exp_not:N \__envopt_temp_cmd { \exp_not:V \l__envopt
       { \exp_not:n { #1 } }
    }

    \tl_set:Nn \l__envopt_exec_tl { \__envopt_temp_cmd }
    \seq_map_inline:Nn \l__envopt_values_seq
    {
       \tl_put_right:Nn \l__envopt_exec_tl { { ##1 } }
    }
    \l__envopt_exec_tl
  }
}

\NewDocumentCommand{\endoptionals}{ m }{}

\ExplSyntaxOff

```