

Package: devforge (via r-universe)

October 7, 2026

Title Development tooling for the bixverse family of R packages

Version 0.0.7

Description Dev-time tooling, not a runtime dependency. Generates the parameter wrapper constructors and their paired checkmate extensions from a declarative spec, and lints the things the house style asks for but cannot enforce: roxygen that has drifted away from the formals it documents, function inputs that never reach a checkmate assertion, and rextendr bindings without an R wrapper.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

Imports checkmate, purrr, utils

Suggests knitr, quarto, tinytest

VignetteBuilder quarto

Depends R (>= 4.2)

Config/roxygen2/version 8.1.0

URL <https://github.com/GregorLueg/devforge>

Repository <https://gregorlueg.r-universe.dev>

Date/Publication 2026-10-06 20:37:53 UTC

RemoteUrl <https://github.com/GregorLueg/devforge>

RemoteRef main

RemoteSha 2d3758e4beaa8d79d99423c6b5b21de591da122c

Contents

forge_params	2
load_specs	2
p_choice	3
p_chr	3

p_dbl	4
p_free	5
p_int	5
p_lgl	6
p_merge	7
param_defaults	8
param_spec	9
params_up_to_date	10
use_drift_ci	11
Index	12

forge_params	<i>Generate the parameter wrappers and their checkmate extensions</i>
--------------	---

Description

Reads every spec under inst/params/, writes the three generated files under R/ and formats them with air. The generated files are meant to be committed.

Usage

```
forge_params(pkg = ".", .verbose = TRUE)
```

Arguments

- pkg String. Path to the package root. Defaults to ".".
- .verbose Boolean. Report what was written. Defaults to TRUE.

Value

Character vector of the paths written, invisibly.

load_specs	<i>Collect the specs defined in a package's spec directory</i>
------------	--

Description

Sources every .R file under inst/params/ in a fresh environment and returns the devforge_spec objects it finds, in file then definition order.

Usage

```
load_specs(pkg = ".")
```

Arguments

pkg String. Path to the package root. Defaults to ".".

Value

A named list of devforge_spec objects, named by their name.

p_choice	<i>Choice field</i>
----------	---------------------

Description

Emits `match.arg()` plus `checkmate::assertChoice()` in the constructor and a `apply_choice_rules()` entry in the checker, which is what the packages already do by hand. `default` is moved to the front of choices so the generated formal reads `method = c("default", ...)`.

Usage

```
p_choice(default, choices, doc = NULL)
```

Arguments

default String. The default value. Must be one of choices.
choices Character vector. The allowed values.
doc String or NULL. Roxygen prose for this field.

Value

A devforge_field.

p_chr	<i>String field</i>
-------	---------------------

Description

String field

Usage

```
p_chr(default, null_ok = FALSE, len = 1L, doc = NULL)
```

Arguments

default	String or NULL. The default value. Leave it out for a formal the caller must supply.
null_ok	Boolean. Whether NULL is permitted. Defaults to FALSE.
len	Integer or string. 1L for a scalar, "+" for one or more. Defaults to 1L.
doc	String or NULL. Roxygen prose for this field.

Value

A devforge_field.

p_dbl	<i>Numeric field</i>
-------	----------------------

Description

Numeric field

Usage

```
p_dbl(  
  default,  
  range = NULL,  
  null_ok = FALSE,  
  len = 1L,  
  strict = FALSE,  
  doc = NULL  
)
```

Arguments

default	Numeric or NULL. The default value. Leave it out for a formal the caller must supply.
range	String or NULL. Checkmate range suffix, for example "(0, 1]".
null_ok	Boolean. Whether NULL is permitted. Defaults to FALSE.
len	Integer or string. 1L for a scalar, "+" for one or more. Defaults to 1L.
strict	Boolean. TRUE demands a double ("R") rather than any number ("N"). Defaults to FALSE.
doc	String or NULL. Roxygen prose for this field.

Value

A devforge_field.

p_free	<i>Unvalidated field</i>
--------	--------------------------

Description

For anything a qassert pattern cannot express. The field contributes a formal, a default and its roxygen, but no assertion and no rule table entry. Pair it with extra_ctor / extra_check on the spec.

Usage

```
p_free(default, doc = NULL)
```

Arguments

default	Any. The default value. Leave it out for a formal the caller must supply.
doc	String or NULL. Roxygen prose for this field.

Value

A devforge_field.

p_int	<i>Integer field</i>
-------	----------------------

Description

Integer field

Usage

```
p_int(  
  default,  
  range = NULL,  
  null_ok = FALSE,  
  len = 1L,  
  integerish = FALSE,  
  check_as = NULL,  
  doc = NULL  
)
```

Arguments

default	Integer or NULL. The default value. Leave it out for a formal the caller must supply.
range	String or NULL. Checkmate range suffix, for example "[1,)"
null_ok	Boolean. Whether NULL is permitted. Defaults to FALSE.
len	Integer or string. 1L for a scalar, "+" for one or more. Defaults to 1L.
integerish	Boolean. Accept whole doubles such as 1000 as well ("X" rather than "I"). Pair it with an as.integer() in extra_ctor when downstream code needs a real integer. Defaults to FALSE.
check_as	Character vector or NULL. Overrides the checker's pattern, for a field the constructor resolves before returning it.
doc	String or NULL. Roxygen prose for this field.

Value

A devforge_field.

<i>p_lgl</i>	<i>Boolean field</i>
--------------	----------------------

Description

Boolean field

Usage

```
p_lgl(default, null_ok = FALSE, doc = NULL)
```

Arguments

default	Boolean or NULL. The default value. Leave it out for a formal the caller must supply.
null_ok	Boolean. Whether NULL is permitted. Defaults to FALSE.
doc	String or NULL. Roxygen prose for this field.

Value

A devforge_field.

p_merge

*Merged sub-list field***Description**

For the constructors that take a list of overrides for a shared defaults block, e.g. `knn = list(k = 25L)` on top of `params_knn_defaults()`. The constructor merges the caller's list into the base with `modifyList(keep.null = TRUE)` and splices the result flat into the returned list at this field's position. When `from` names another spec, the checker requires the spliced elements and validates them with that spec's rules.

Usage

```
p_merge(
  from,
  default = list(),
  overrides = NULL,
  drop = NULL,
  strict = FALSE,
  doc = NULL
)
```

Arguments

from	String or language object. The name of a spec in the same package (" <code>knn_defaults</code> " merges into <code>params_knn_defaults()</code>), or an expression evaluated inside the constructor, for a base that depends on the other arguments. A language <code>from</code> contributes nothing to the checker.
default	List or language object. The formal default. Defaults to <code>list()</code> .
overrides	List, language object or <code>NULL</code> . Constructor-specific defaults applied over <code>from</code> before the caller's list. A language object may reference the other formals, e.g. <code>quote(list(k = neighbours_within_batch * 2L))</code> . Defaults to <code>NULL</code> .
drop	Character vector or <code>NULL</code> . Elements removed from the base before the merge, for a constructor that sets them itself or has no use for them. They drop out of the checker too. Defaults to <code>NULL</code> .
strict	Boolean. The constructor rejects caller elements that are not in the base (after <code>drop</code>) instead of splicing them through unchecked. Defaults to <code>FALSE</code> .
doc	String or <code>NULL</code> . Roxygen prose for this field.

Value

A `devforge_field`.

param_defaults	<i>Defaults block specification</i>
----------------	-------------------------------------

Description

A zero-argument `params_*_defaults()` helper. Same fields as a `param_spec()`, but the generated function takes no arguments, runs no assertions and simply returns the defaults. These exist to be embedded in other constructors.

Usage

```
param_defaults(
  name,
  title,
  fields,
  description = NULL,
  details = NULL,
  references = NULL,
  checker = NULL,
  label = NULL,
  hint = NULL,
  export = TRUE
)
```

Arguments

name	String. Snake case stem. "knn_defaults" gives <code>params_knn_defaults()</code> .
title	String. The roxygen title line.
fields	Named list of <code>p_int()</code> and friends.
description	String or NULL. Roxygen @description prose.
details	String or NULL. Roxygen @details prose.
references	String or NULL. Roxygen @references prose.
checker	String or NULL. PascalCase stem for a checker over this block, for the defaults blocks that are validated on their own. Defaults to NULL.
label	String or NULL. Label used in the checker's error messages.
hint	String or NULL. Appended to the checker's error messages.
export	Boolean. Whether the helper gets @export. Defaults to TRUE.

Value

A list of class `devforge_spec`.

param_spec

Parameter wrapper specification

Description

The declarative description of one `params_*`() constructor and its paired checkmate extension. Everything the generator emits comes from here.

Usage

```
param_spec(
  name,
  title,
  fields,
  description = NULL,
  details = NULL,
  references = NULL,
  return_order = NULL,
  checker = to_pascal_case(name),
  checker_args = list(),
  extra_ctor = NULL,
  extra_check = NULL,
  class_tag = NULL,
  strict_names = FALSE,
  label = NULL,
  hint = NULL,
  test_fn = FALSE,
  export = TRUE
)
```

Arguments

name	String. Snake case stem. "ica_general" gives <code>params_ica_general()</code> .
title	String. The roxygen title line for the constructor.
fields	Named list of <code>p_int()</code> and friends. Order is the order of the formals and of the returned list.
description	String or NULL. Roxygen @description prose. Defaults to NULL.
details	String or NULL. Roxygen @details prose. Line breaks are kept so it can carry its own <code>\itemize{}</code> , and lines over the width are wrapped. Defaults to NULL.
references	String or NULL. Roxygen @references prose. Defaults to NULL.
return_order	Character vector or NULL. The order of the returned list, when it differs from the order of the formals. Must be a permutation of <code>names(fields)</code> . Defaults to NULL, meaning the order of the formals.

checker	String or NULL. PascalCase stem for the checkmate extension, giving <code>check<stem>Params()</code> and <code>assert<stem>Params()</code> . Two specs may share a stem, in which case the checker is emitted once and the field sets must agree. NULL emits no checker. Defaults to the PascalCase form of name.
checker_args	A list made with <code>base::alist()</code> . Extra formals for the checker beyond <code>x</code> , for the checkers that cross-reference an outside object. Defaults to an empty list.
extra_ctor	Language object or NULL. Emitted verbatim into the constructor after the per-field assertions. For cross-field rules.
extra_check	Language object or NULL. Emitted verbatim into the checker before its final return. Must return() an error string on failure.
class_tag	String or NULL. When set, the returned list is tagged <code>c(class_tag, "list")</code> . Only for the params objects that are dispatched on with <code>inherits()</code> . Defaults to NULL.
strict_names	Boolean. TRUE demands exactly the declared names, FALSE uses <code>must.include</code> and tolerates extras. Defaults to FALSE.
label	String or NULL. Human readable label used in the checker's error messages. Defaults to name with underscores replaced by spaces, followed by "params".
hint	String or NULL. Appended to the checker's error messages to describe the expected types and ranges.
test_fn	Boolean. Also emit <code>test<stem>Params()</code> via <code>checkmate::makeTestFunction()</code> . Defaults to FALSE.
export	Boolean. Whether the constructor gets <code>@export</code> . Defaults to TRUE.

Value

A list of class `devforge_spec`.

params_up_to_date	<i>Check whether the generated files match the specs</i>
-------------------	--

Description

Regenerates into a temporary directory and compares. Meant for CI, so that a hand-edit of a generated file or a spec change that was never regenerated fails the build.

Usage

```
params_up_to_date(pkg = ".")
```

Arguments

pkg String. Path to the package root. Defaults to ".".

Value

TRUE when everything matches, otherwise a character vector of the paths that are out of date.

use_drift_ci	<i>Add the params drift workflow to a package</i>
--------------	---

Description

Copies a GitHub Actions workflow into `.github/workflows/params-drift.yml` that installs devtools and fails the build when `params_up_to_date()` reports stale generated files. Also adds `^\.github$` to `.Rbuildignore` if it is not there yet.

Usage

```
use_drift_ci(  
  pkg = ".",  
  runner = "macos-latest",  
  overwrite = FALSE,  
  .verbose = TRUE  
)
```

Arguments

pkg	String. Path to the package root. Defaults to ".".
runner	String. The GitHub Actions runner. Has to match the platform the specs are forged on: macOS arm64 R has no extended long double, so literals like <code>1e-300</code> deparse differently than on Linux and every file reads as stale. Defaults to "macos-latest".
overwrite	Boolean. Replace an existing workflow file. Defaults to FALSE.
.verbose	Boolean. Report what was written. Defaults to TRUE.

Value

The path of the workflow file, invisibly.

Index

`base::alist()`, [10](#)

`checkmate::assertChoice()`, [3](#)

`checkmate::makeTestFunction()`, [10](#)

`forge_params`, [2](#)

`load_specs`, [2](#)

`p_choice`, [3](#)

`p_chr`, [3](#)

`p_dbl`, [4](#)

`p_free`, [5](#)

`p_int`, [5](#)

`p_int()`, [8](#), [9](#)

`p_lgl`, [6](#)

`p_merge`, [7](#)

`param_defaults`, [8](#)

`param_spec`, [9](#)

`param_spec()`, [8](#)

`params_up_to_date`, [10](#)

`params_up_to_date()`, [11](#)

`use_drift_ci`, [11](#)