

Package: xlamisc (via r-universe)

October 8, 2026

Title Helper Functions for 'r-xla'

Version 0.5.1

Description Frequently used generics and helper functions used in 'anvl' and its companion packages 'stablehlo' and 'pjrt'. Comes with S3 generics to query arrays (shape, data type, device) and to convert them to R arrays and raw vectors, as well as a class for data types.

License MIT + file LICENSE

URL <https://r-xla.github.io/xlamisc/>, <https://github.com/r-xla/xlamisc>

BugReports <https://github.com/r-xla/xlamisc/issues>

Encoding UTF-8

Depends R (>= 4.3.0)

Imports checkmate, cli, rlang, utils

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Sebastian Fischer [aut, cre] (ORCID: [<https://orcid.org/0000-0002-9609-3197>](https://orcid.org/0000-0002-9609-3197)), Daniel Falbel [aut] (ORCID: [<https://orcid.org/0009-0006-0143-2392>](https://orcid.org/0009-0006-0143-2392))

Maintainer Sebastian Fischer <seb.fischer@tutamail.com>

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

Date/Publication 2026-10-08 22:13:54 UTC

RemoteUrl <https://github.com/cran/xlamisc>

RemoteRef HEAD

RemoteSha 795a2aeae087353b34c0c7a0fa979736f0c2cffe

Contents

as_array	2
as_dtype	3
as_raw	3
assert_dtype	4
DataType	4
device	5
dtype	5
dtype_category	6
dtype_width	6
is_dtype	7
is_dtype_float	7
naxes	8
nelts	8
shape	9
Index	10

as_array	<i>As Array</i>
----------	-----------------

Description

Convert a value (usually a tensor from some framework) to an R array.

Usage

```
as_array(x, ...)
```

Arguments

- x (any)
Object.
- ... (any)
Additional arguments.

Value

(any)

as_dtype	<i>Convert to DataType</i>
----------	----------------------------

Description

Convert to [DataType](#).

Usage

as_dtype(x)

Arguments

x	(any) A dtype string (e.g. "f32", "i64", "bool"; the aliases "pred" and "i1" map to "bool") or a DataType object.
---	--

Value

DataType

as_raw	<i>As Raw</i>
--------	---------------

Description

Convert a value (usually a tensor from some framework) to a raw() vector.

Usage

as_raw(x, ...)

Arguments

x	(any) Object.
...	(any) Additional arguments.

Value

(any)

assert_dtype	<i>Assert DataType</i>
--------------	------------------------

Description

Assert that an object is a [DataType](#).

Usage

```
assert_dtype(x, arg = rlang::caller_arg(x))
```

Arguments

x	(any) Object to check.
arg	(character(1)) Argument name for error messages.

Value

NULL (invisibly). Called for side effects.

DataType	<i>DataType</i>
----------	-----------------

Description

DataType is the S3 class representing a tensor element type. It is an enum: one singleton object per supported dtype, wrapping the canonical dtype string. The members are the element types supported by the r-xla stack:

- boolean: bool
- signed integers: i8, i16, i32, i64
- unsigned integers: ui8, ui16, ui32, ui64
- floats: f16, bf16, f32, f64
- complex: c64, c128

Construct a DataType with [as_dtype\(\)](#). Inspect it with [dtype_width\(\)](#) and the [is_dtype_float\(\)](#) family of predicates.

See Also

[as_dtype\(\)](#), [dtype_width\(\)](#), [is_dtype_float\(\)](#)

device	<i>Device</i>
--------	---------------

Description

Get the device of a tensor.

Usage

device(x, ...)

Arguments

- x (any) Object.
- ... (any) Additional arguments.

Value

(any)

dtype	<i>Data Type</i>
-------	------------------

Description

Get the data type of a tensor. The return type depends on the specific tensor implementation.

Usage

dtype(x, ...)

Arguments

- x (any) Object.
- ... (any) Additional arguments.

Value

(any)

dtype_category	<i>Category of a DataType</i>
----------------	-------------------------------

Description

The category of a [DataType](#): one of "bool", "int", "uint", "float", or "complex".

Usage

dtype_category(x)

Arguments

x ([DataType](#))

Value

character(1)

See Also

[is_dtype_float\(\)](#) and friends for per-category predicates

dtype_width	<i>Width of a DataType</i>
-------------	----------------------------

Description

The width in bits of one element of the given [DataType](#). For bool this is 1; for complex types it is the total width of the element (c64 -> 64).

Usage

dtype_width(x)

Arguments

x ([DataType](#))

Value

integer(1)

is_dtype	<i>Is DataType</i>
----------	--------------------

Description

Check if an object is a [DataType](#).

Usage

is_dtype(x)

Arguments

x	(any) Object to check.
---	---------------------------

Value

logical(1)

is_dtype_float	<i>DataType Category Predicates</i>
----------------	-------------------------------------

Description

Check which category a [DataType](#) belongs to:

- is_dtype_float(): floating point, including bf16.
- is_dtype_int(): signed integers.
- is_dtype_uint(): unsigned integers.
- is_dtype_bool(): bool.
- is_dtype_complex(): c64, c128.

Usage

is_dtype_float(x)

is_dtype_int(x)

is_dtype_uint(x)

is_dtype_bool(x)

is_dtype_complex(x)

Arguments

x (DataType)

Value

logical(1)

naxes	Number of Axes
-------	----------------

Description

Get the number of axes of a tensor.

Usage

naxes(x)

Arguments

x (any)
Object.

Value

(integer)

nelts	Number of Elements
-------	--------------------

Description

Get the number of elements of a tensor.

Usage

nelts(x)

Arguments

x (any)
Object.

Value

(integer)

shape	<i>Shape</i>
-------	--------------

Description

Get the shape of a tensor.

Usage

shape(x, ...)

Arguments

- x (any) Object.
- ... (any) Additional arguments.

Value

(integer())
The shape of the tensor.

Index

`as_array`, [2](#)
`as_dtype`, [3](#)
`as_dtype()`, [4](#)
`as_raw`, [3](#)
`assert_dtype`, [4](#)

`DataType`, [3](#), [4](#), [4](#), [6–8](#)
`device`, [5](#)
`dtype`, [5](#)
`dtype_category`, [6](#)
`dtype_width`, [6](#)
`dtype_width()`, [4](#)

`is_dtype`, [7](#)
`is_dtype_bool(is_dtype_float)`, [7](#)
`is_dtype_complex(is_dtype_float)`, [7](#)
`is_dtype_float`, [7](#)
`is_dtype_float()`, [4](#), [6](#)
`is_dtype_int(is_dtype_float)`, [7](#)
`is_dtype_uint(is_dtype_float)`, [7](#)

`naxes`, [8](#)
`nelts`, [8](#)

`shape`, [9](#)