Package: xxhashlite (via r-universe)

September 8, 2024

Type Package

Title Extremely Fast Hashing of R Objects, Raw Data and Files using 'xxHash' Algorithms

Version 0.2.2

Maintainer Mike Cheng <mikefc@coolbutuseless.com>

Description Extremely fast hashing of R objects using 'xxHash'. R objects are hashed via the standard serialization mechanism in R. Raw byte vectors and strings can be handled directly for compatibility with hashes created on other systems. This implementation is a wrapper around the 'xxHash' 'C' library which is available from <https://github.com/Cyan4973/xxHash>.

License MIT + file LICENSE

URL https://github.com/coolbutuseless/xxhashlite

BugReports https://github.com/coolbutuseless/xxhashlite/issues

Encoding UTF-8

RoxygenNote 7.3.1

Suggests testthat

Depends R (>= 3.5.0)

Copyright This package includes code from the 'xxhash' written Yann Collet. See file 'inst/LICENSE-xxhash' for copyright information of the original library.

Repository https://coolbutuseless.r-universe.dev

RemoteUrl https://github.com/coolbutuseless/xxhashlite

RemoteRef HEAD

RemoteSha 1ded4178580d6214e118ab242624b97025209b25

Contents

`xxhash` *Calculate the hash of an arbitrary R object.*

Description

This function will calculate the hash of any object understood by `base::serialize()`.

Usage

```
xxhash(robj, algo = "xxh128", as_raw = FALSE)
```

Arguments

`robj`	Any R object
`algo`	Select the specific xxhash algorithm. Default: 'xxh128'. (the latest algorithm in the xxhash family) Valid values: 'xxh32', 'xxh64', 'xxh128', 'xxh3'
`as_raw`	Return the hash as a raw vector of bytes instead of string? Default: FALSE. If TRUE, then the raw bytes are returned in big-endian order - which is what `xxHash` considers the *canonical* form.

Value

String representation of hash. If `as_raw = TRUE` then a raw vector is returned instead.

Examples

```
xxhash(mtcars)
xxhash(mtcars, algo = 'xxh3', as_raw = TRUE)
```

`xxhash_con` *Calculate the hash of data from a connection object*

Description

Calculate the hash of data from a connection object

Usage

```
xxhash_con(con, algo = "xxh128", as_raw = FALSE)
```

Arguments

`con`	connection
`algo`	Select the specific xxhash algorithm. Default: 'xxh128'. (the latest algorithm in the xxhash family) Valid values: 'xxh32', 'xxh64', 'xxh128', 'xxh3'
`as_raw`	Return the hash as a raw vector of bytes instead of string? Default: FALSE. If TRUE, then the raw bytes are returned in big-endian order - which is what `xxHash` considers the *canonical* form.

Value

String representation of hash. If `as_raw` = `TRUE` then a raw vector is returned instead.

Examples

```
filename <- system.file('DESCRIPTION', package = 'base', mustWork = TRUE)
xxhash_con(file(filename))
```

`xxhash_file` *Calculate the hash of a file*

Description

Calculate the hash of a file

Usage

```
xxhash_file(file, algo = "xxh128", as_raw = FALSE)
```

Arguments

`file`	filename
`algo`	Select the specific xxhash algorithm. Default: 'xxh128'. (the latest algorithm in the xxhash family) Valid values: 'xxh32', 'xxh64', 'xxh128', 'xxh3'
`as_raw`	Return the hash as a raw vector of bytes instead of string? Default: FALSE. If TRUE, then the raw bytes are returned in big-endian order - which is what `xxHash` considers the *canonical* form.

Value

String representation of hash. If `as_raw` = `TRUE` then a raw vector is returned instead.

Examples

```
filename <- system.file('DESCRIPTION', package = 'base', mustWork = TRUE)
xxhash_file(filename)
```

xxhash_raw — *Calculate the hash of a raw vector or string*

Description

This performs a hash of the raw bytes - not of the serialized representation.

Usage

```
xxhash_raw(vec, algo = "xxh128", as_raw = FALSE)
```

Arguments

vec	raw vector or single character string
algo	Select the specific xxhash algorithm. Default: 'xxh128'. (the latest algorithm in the xxhash family) Valid values: 'xxh32', 'xxh64', 'xxh128', 'xxh3'
as_raw	Return the hash as a raw vector of bytes instead of string? Default: FALSE. If TRUE, then the raw bytes are returned in big-endian order - which is what `xxHash` considers the *canonical* form.

Value

String representation of hash. If `as_raw = TRUE` then a raw vector is returned instead.

Examples

```
vec <- "hello"
xxhash_raw(vec)
vec <- as.raw(c(0x01, 0x02, 0x99))
xxhash_raw(vec)
```

Index