Package: feck (via r-universe)

June 1, 2026

Type Package

Title Forward Error Correction and Erasure Coding

Version 1.0.0

Maintainer Mike Cheng <mikefc@coolbutuseless.com>

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

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

Description Forward error correction and erasure coding. Repair damaged files using pre-generated repair blocks.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Copyright See 'inst/COPYRIGHTS' file for original license for included code from 'zfec'

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

Date/Publication 2025-04-06 11:04:39 UTC

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

RemoteRef HEAD

RemoteSha b5ae3f841219ca819896eb6e9b39bd1dfcf8d78e

Contents

chiblihash64 *Calculate the 64-bit chibli hash for the given data*

Description

Calculate the 64-bit chibli hash for the given data

Usage

```
chiblihash64(raw_vec, skip = 0, len = length(raw_vec))
```

Arguments

raw_vec	raw vector
skip	number of bytes to skip at start
len	Number of bytes to hash

Value

String containing the chibli hash

Examples

```
zz <- as.raw(1:100)
chiblihash64(zz)
chiblihash64(zz)
zz[1] <- as.raw(199)
chiblihash64(zz)
```

fec_prepare_file *Create blocks for a file*

Description

Create blocks for a file

Usage

```
fec_prepare_file(
  filename,
  blocks_filename = NULL,
  k = 10,
  n = 2,
  verbosity = 0L
)
```

Arguments

filename	filename
blocks_filename	Default: NULL, suffix with ".feck"
k	number of chunks to split data into. Larger 'k' will mean the file will be chunked into more, smaller blocks. This will help isolate errors, but can increase computation time.
n	number of repair chunks to create. Each repair chunk will be able to repair one bad chunk in the data.
verbosity	Default: 0

Value

Invisibly return the raw vector of repair blocks

---

fec_prepare_raw *Create recovery blocks*

Description

Create recovery blocks

Usage

```
fec_prepare_raw(raw_vec, k = 10, n = 2, verbosity = 0L)
```

Arguments

raw_vec	raw vector
k	number of chunks to split data into. Larger 'k' will mean the file will be chunked into more, smaller blocks. This will help isolate errors, but can increase computation time.
n	number of repair chunks to create. Each repair chunk will be able to repair one bad chunk in the data.
verbosity	Default: 0

Value

Raw vector of meta-data and recovery blocks

Examples

```
set.seed(1)
dat <- as.raw(sample(0:255, 1e3, replace = TRUE))
rblocks <- fec_prepare_raw(dat, verbosity = 1)
```

fec_repair_file *Repair file*

Description

Repair file

Usage

```
fec_repair_file(filename, blocks_filename = NULL, verbosity = 0L)
```

Arguments

filename	filename. Repaired file will have suffix ".repaired"
blocks_filename	Default: NULL, suffix with ".feck"
verbosity	Default: 0

Value

Invisibly return a raw vector of the contents of the repaired file. If repair is not possible, an error will be raised.

fec_repair_raw *Recover damaged data in a raw vector*

Description

Recover damaged data in a raw vector

Usage

```
fec_repair_raw(raw_vec, blocks, verbosity = 0L)
```

Arguments

raw_vec	raw vector
blocks	raw vector of recovery blocks as created by fec_prepare_raw()
verbosity	Default: 0

Value

Repaired data, or NULL if repair not possible

Examples

```
set.seed(1)
dat0 <- dat <- as.raw(sample(0:255, 1e4, replace = TRUE))
rblocks <- fec_prepare_raw(dat, verbosity = 1)
# simulate damange to data
dat[1:100] <- as.raw(0)
identical(dat, dat0)
dat_repaired <- fec_repair_raw(dat, rblocks, verbosity = 1)
identical(dat_repaired, dat0)
```

---

print.fec_blocks *Print repair blocks*

Description

Print repair blocks

Usage

```
## S3 method for class 'fec_blocks'
print(x, ...)
```

Arguments

x	Repair blocks created by fec_prepare_raw()
...	ignored

Value

None

Examples

```
rblocks <- fec_prepare_raw(as.raw(1:1000))
print(rblocks)
```

Index