

# Package: b32 (via r-universe)

May 24, 2026

**Title** Fast and Vectorized Base32 Encoding

**Version** 0.1.0

**Description** Fast, dependency free, and vectorized base32 encoding and decoding. 'b32' supports the Crockford, Z, RFC 4648 lower, hex, and lower hex alphabets.

**License** MIT + file LICENSE

**URL** <https://github.com/extendr/b32>, <http://extendr.rs/b32/>

**Depends** R (>= 4.2)

**Suggests** blob, testthat (>= 3.0.0)

**Config/rextenr/version** 0.4.2.9000

**Config/testthat/edition** 3

**Encoding** UTF-8

**Language** en

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**SystemRequirements** Cargo (Rust's package manager), rustc >= 1.65.0, xz

**Config/pak/sysreqs** xz-utils libclang-dev

**Repository** <https://extendr.r-universe.dev>

**Date/Publication** 2026-01-19 18:51:14 UTC

**RemoteUrl** <https://github.com/extendr/b32>

**RemoteRef** HEAD

**RemoteSha** 15c830e400451d613731d2eec35c7fbb14036625

## Contents

encode . . . . .	2
<b>Index</b>	<b>3</b>

---

 encode *Base32 Encode and Decode*


---

**Description**

Base32 Encode and Decode

**Usage**

```
encode(x, alphabet = "crockford", padded = TRUE)
```

```
decode(x, alphabet = "crockford", padded = TRUE)
```

```
decode_as_string(x, alphabet = "crockford", padded = TRUE)
```

**Arguments**

x	for encode(), must be either a character or raw vector. For decode(), a character vector.
alphabet	default "crockford". The base32 encoding alphabet to use. Must be one of "crockford", "rfc4648", "rfc4648lower", "rfc4648hex", "rfc4648hexlower", or "z".
padded	default TRUE. Appends = to ensure that the final encoded chunk is 8 blocks of characters.

**Details**

decode() returns a blob object which is a list of raw vectors. If an error is encountered in decode() the result will be an NA.

**Value**

decode() returns a blob object. encode() returns a character vector. decode\_as\_string() returns a character vector.

**Examples**

```
encode("hello")
encode("hello", padded = FALSE)
encode("hello", alphabet = "rfc4648")

# Decode base32 string
decode("NBSWY3DPEBLW64TMMQ=====")
decode_as_string("NBSWY3DPEBLW64TMMQ=====")
decode("NBSWY3DPEBLW64TMMQ", padded = FALSE)
```

# Index

`decode (encode)`, [2](#)  
`decode_as_string (encode)`, [2](#)  
`encode`, [2](#)