

# Package ‘uchardet’

November 7, 2022

**Type** Package

**Title** The Universal Character Encoding Detector

**Description** R bindings of the 'uchardet', encoding detector library from Mozilla (<<https://www.freedesktop.org/wiki/Software/uchardet/>>). It takes a sequence of bytes in an unknown character encoding and without any additional information, and attempts to get the encoding of the text. All return names of the encodings are iconv-compatible.

**Version** 1.1.1

**License** GPL-2

**Copyright** file COPYRIGHTS

**URL** <https://artemklevtsov.gitlab.io/uchardet>,  
<https://gitlab.com/artemklevtsov/uchardet>

**BugReports** <https://gitlab.com/artemklevtsov/uchardet/-/issues>

**Depends** R (>= 3.1.0)

**Suggests** tinytest, knitr, rmarkdown

**SystemRequirements** C++11, GNU make

**NeedsCompilation** yes

**ByteCompile** yes

**Encoding** UTF-8

**RoxygenNote** 7.2.1

**VignetteBuilder** knitr

**Author** Artem Klevtsov [aut, cre] (<<https://orcid.org/0000-0003-0492-6647>>),  
Philipp Upravitelev [ctb]

**Maintainer** Artem Klevtsov <a.a.klevtsov@gmail.com>

**Repository** CRAN

**Date/Publication** 2022-11-07 13:20:02 UTC

## R topics documented:

detect_file_enc . . . . .	2
detect_raw_enc . . . . .	3
detect_str_enc . . . . .	4
uchardet . . . . .	5

<b>Index</b>	<b>8</b>
--------------	----------

---

detect_file_enc	<i>File encoding detection</i>
-----------------	--------------------------------

---

### Description

This function tries to detect character encoding.

### Usage

```
detect_file_enc(x)
```

### Arguments

x                      Character vector, containing file names or paths.

### Value

A character vector of length equal to the length of x and contains guessed iconv-compatible encodings names.

### Examples

```
# detect character vector with ASCII strings
ascii <- "I can eat glass and it doesn't hurt me."
detect_str_enc(ascii)

# detect character vector with UTF-8 strings
utf8 <- "\u4e0b\u5348\u597d"
print(utf8)
detect_str_enc(utf8)

# function to read ASCII or UTF-8 files
read_file <- function(x) readChar(x, file.size(x))
# path to examples
ex_path <- system.file("examples", package = "uchardet")

# russian text
ru_utf8 <- read_file(file.path(ex_path, "ru.txt"))
print(ru_utf8)
detect_str_enc(iconv(ru_utf8, "utf8", "ibm866"))
detect_str_enc(iconv(ru_utf8, "utf8", "koi8-r"))
```

```

detect_str_enc(iconv(ru_utf8, "utf8", "cp1251"))

# china text
zh_utf8 <- read_file(file.path(ex_path, "zh.txt"))
print(zh_utf8)
detect_str_enc(iconv(zh_utf8, "utf8", "big5"))
detect_str_enc(iconv(zh_utf8, "utf8", "gb18030"))

# korean text
ko_utf8 <- read_file(file.path(ex_path, "ko.txt"))
print(ko_utf8)
detect_str_enc(iconv(ko_utf8, "utf8", "uhc"))
detect_str_enc(iconv(ko_utf8, "utf8", "iso-2022-kr"))

```

---

detect_raw_enc	<i>Raw bytes encoding detection</i>
----------------	-------------------------------------

---

## Description

This function tries to detect raw bytes encoding.

## Usage

```
detect_raw_enc(x)
```

## Arguments

x                      Raw vector.

## Value

A character which contains a guessed iconv-compatible encoding name.

## Examples

```

# detect raw vector encoding with ASCII encoding
ascii <- "I can eat glass and it doesn't hurt me."
detect_raw_enc(charToRaw(ascii))

# detect raw vector with UTF-8 encoding
utf8 <- "\u4e0b\u5348\u597d"
detect_raw_enc(charToRaw(utf8))

# function to read file as raw bytes
read_bin <- function(x) readBin(x, raw(), file.size(x))

# detect encoding of files read as raw vector
ex_path <- system.file("examples", package = "uchardet")

# deutsch text as binary data

```

```
de_bin <- read_bin(file.path(ex_path, "de", "windows-1252.txt"))
detect_raw_enc(de_bin)

# russian text as binary data
ru_bin <- read_bin(file.path(ex_path, "ru", "windows-1251.txt"))
detect_raw_enc(ru_bin)

# china text as binary data
zh_bin <- read_bin(file.path(ex_path, "zh", "utf-8.txt"))
detect_raw_enc(zh_bin)
```

---

detect\_str\_enc            *String encoding detection*

---

## Description

This function tries to detect character encoding.

## Usage

```
detect_str_enc(x)
```

## Arguments

x                      Character vector.

## Value

A character vector of length equal to the length of x and contains guessed iconv-compatible encodings names.

## Examples

```
# detect character vector with ASCII strings
ascii <- "I can eat glass and it doesn't hurt me."
detect_str_enc(ascii)

# detect character vector with UTF-8 strings
utf8 <- "\u4e0b\u5348\u597d"
print(utf8)
detect_str_enc(utf8)

# function to read ASCII or UTF-8 files
read_file <- function(x) readChar(x, file.size(x))
# path to examples
ex_path <- system.file("examples", package = "uchardet")

# russian text
ru_utf8 <- read_file(file.path(ex_path, "ru.txt"))
print(ru_utf8)
```

```
detect_str_enc(iconv(ru_utf8, "utf8", "ibm866"))
detect_str_enc(iconv(ru_utf8, "utf8", "koi8-r"))
detect_str_enc(iconv(ru_utf8, "utf8", "cp1251"))

# china text
zh_utf8 <- read_file(file.path(ex_path, "zh.txt"))
print(zh_utf8)
detect_str_enc(iconv(zh_utf8, "utf8", "big5"))
detect_str_enc(iconv(zh_utf8, "utf8", "gb18030"))

# korean text
ko_utf8 <- read_file(file.path(ex_path, "ko.txt"))
print(ko_utf8)
detect_str_enc(iconv(ko_utf8, "utf8", "uhc"))
detect_str_enc(iconv(ko_utf8, "utf8", "iso-2022-kr"))
```

---

uchardet

*The Universal Character Encoding Detector*

---

## Description

R bindings for the uchardet library (<<https://www.freedesktop.org/wiki/Software/uchardet/>>), that is the encoding detector library of Mozilla. It takes a sequence of bytes in an unknown character encoding without any additional information, and attempts to determine the encoding of the text. Returned encoding names are iconv-compatible.

## Author(s)

**Maintainer:** Artem Klevtsov <a.a.klevtsov@gmail.com> ([ORCID](#))

Other contributors:

- Philipp Upravitelev <upravitelev@gmail.com> [contributor]

## References

uchardet page: <https://www.freedesktop.org/wiki/Software/uchardet/>

## See Also

Useful links:

- <https://artemklevtsov.gitlab.io/uchardet>
- <https://gitlab.com/artemklevtsov/uchardet>
- Report bugs at <https://gitlab.com/artemklevtsov/uchardet/-/issues>

**Examples**

```

# detect character vector with ASCII strings
ascii <- "I can eat glass and it doesn't hurt me."
detect_str_enc(ascii)

# detect character vector with UTF-8 strings
utf8 <- "\u4e0b\u5348\u597d"
print(utf8)
detect_str_enc(utf8)

# function to read ASCII or UTF-8 files
read_file <- function(x) readChar(x, file.size(x))
# path to examples
ex_path <- system.file("examples", package = "uchardet")

# russian text
ru_utf8 <- read_file(file.path(ex_path, "ru.txt"))
print(ru_utf8)
detect_str_enc(iconv(ru_utf8, "utf8", "ibm866"))
detect_str_enc(iconv(ru_utf8, "utf8", "koi8-r"))
detect_str_enc(iconv(ru_utf8, "utf8", "cp1251"))

# china text
zh_utf8 <- read_file(file.path(ex_path, "zh.txt"))
print(zh_utf8)
detect_str_enc(iconv(zh_utf8, "utf8", "big5"))
detect_str_enc(iconv(zh_utf8, "utf8", "gb18030"))

# korean text
ko_utf8 <- read_file(file.path(ex_path, "ko.txt"))
print(ko_utf8)
detect_str_enc(iconv(ko_utf8, "utf8", "uhc"))
detect_str_enc(iconv(ko_utf8, "utf8", "iso-2022-kr"))
# detect ASCII file encoding
detect_file_enc(system.file("DESCRIPTION", package = "uchardet"))

# paths to examples files
ex_path <- system.file("examples", package = "uchardet")
# various languages and encodings examples files
ex_files <- Sys.glob(file.path(ex_path, "*", "*"))
# detect files encodings
detect_file_enc(head(ex_files, 10))
# detect raw vector encoding with ASCII encoding
ascii <- "I can eat glass and it doesn't hurt me."
detect_raw_enc(charToRaw(ascii))

# detect raw vector with UTF-8 encoding
utf8 <- "\u4e0b\u5348\u597d"
detect_raw_enc(charToRaw(utf8))

# function to read file as raw bytes
read_bin <- function(x) readBin(x, raw(), file.size(x))

```

```
# detect encoding of files read as raw vector
ex_path <- system.file("examples", package = "uchardet")

# deutsch text as binary data
de_bin <- read_bin(file.path(ex_path, "de", "windows-1252.txt"))
detect_raw_enc(de_bin)

# russian text as binary data
ru_bin <- read_bin(file.path(ex_path, "ru", "windows-1251.txt"))
detect_raw_enc(ru_bin)

# china text as binary data
zh_bin <- read_bin(file.path(ex_path, "zh", "utf-8.txt"))
detect_raw_enc(zh_bin)
```

# Index

`detect_file_enc`, [2](#)

`detect_raw_enc`, [3](#)

`detect_str_enc`, [4](#)

`uchardet`, [5](#)

`uchardet-package (uchardet)`, [5](#)