Package 'i18n'

July 22, 2025

Type Package

Version 0.2.0

Title Internationalization Data from the 'Unicode CLDR' in Tabular
Form

Description Up-to-date data from the 'Unicode CLDR Project' (where 'CLDR'
stands for 'Common Locale Data Repository') are available here as a series
of easy-to-parse datasets. Several functions are provided for extracting
key elements from the tabular datasets.

License MIT + file LICENSE

URL <https://github.com/rich-iannone/i18n>

BugReports <https://github.com/rich-iannone/i18n/issues>

Encoding UTF-8

LazyData true

LazyDataCompression xz

RoxygenNote 7.2.1

Depends R (>= 3.2)

Imports magrittr

Suggests lintr, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Richard Iannone [aut, cre] (ORCID:
<https://orcid.org/0000-0003-3925-190X>)

Maintainer Richard Iannone <riannone@me.com>

Repository CRAN

Date/Publication 2022-09-07 06:20:02 UTC

Contents

---

all_currency_codes   *A vector containing every currency code*

Description

This is a vector of the 305 currency codes that are used in the [currencies](#) dataset within the **i18n** package.

Usage

```
all_currency_codes
```

Format

An object of class `character` of length 305.

---

all_locales *A vector containing all locale names*

Description

This is a vector of the 574 locale names that are used throughout the tabular datasets within the **i18n** package.

Usage

```
all_locales
```

Format

An object of class `character` of length 574.

---

characters *A table with localized character data*

Description

The `characters` table contains localized character data across 574 locales. There are 574 rows and the following 12 columns:

- locale(character)
- exemplar_characters (character)
- auxiliary (character)
- index (character)
- numbers (character)
- punctuation (character)
- more_info (character)
- ellipsis (named list [length of 6])
- leninent_scope_general (named list [length of 9])
- leninent_scope_date (named list [length of 2])
- leninent_scope_number (named list [length of 3])
- stricter_scope_number (named list [length of 2])

Usage

```
characters
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 574 rows and 12 columns.

---

character_labels   *A table with localized character labels and descriptors*

Description

The character_labels table contains localized data for character labels across 574 locales. There are 574 rows and the following 3 columns:

- locale (character)
- character_label_patterns (named list [variable length])
- character_labels (named list [variable length])

Usage

```
character_labels
```

Format

An object of class tbl_df (inherits from tbl, data.frame) with 574 rows and 3 columns.

---

cldr_characters   *Get localized values from the [characters](#) dataset*

Description

The [characters](#) table contains information on the usage of characters and exemplar character sets across 574 locales. The cldr_characters() function allows one to extract element values from the table by supplying the locale and one of the following element names:

- "exemplar_characters"
- "auxiliary"
- "index"
- "numbers"
- "punctuation"
- "more_info"
- "ellipsis"
- "leninent_scope_general"
- "leninent_scope_date"
- "leninent_scope_number"
- "stricter_scope_number"

Usage

```
cldr_characters(
  locale = "en",
  element = characters_elements$exemplar_characters
)
```

Arguments

`locale`	The locale ID for which to obtain the data from the [characters](#) table.
`element`	The element from which information will be obtained for the specified `locale`.

Value

Either a named list or a length one character vector, depending on the `element` value.

---

cldr_character_labels   *Get localized values from the [character_labels](#) dataset*

Description

The [character_labels](#) table contains information on character patterns and character labels across 574 locales. The cldr_character_labels() function allows one to extract element values from the table by supplying the `locale` and one of the following element names:

- `"patterns"`: corresponds to the character_label_patterns column in [character_labels](#)
- `"labels"`: is the character_labels column in [character_labels](#)

Usage

```
cldr_character_labels(locale = "en", element = c("patterns", "labels"))
```

Arguments

`locale`	The locale ID for which to obtain the data from the `character_labels` table.
`element`	The element from which information will be obtained for the specified `locale`.

Value

A named list.

cldr_currencies	*Get a single localized value from the [currencies](#) dataset*

Description

The [currencies](#) table contains information of currency codes and localized display names and symbols across 574 locales. The cldr_currencies() function allows one to extract a single element value from the table by supplying the locale, the currency code (currency), and one of the following element names:

- "currency_symbol"
- "currency_symbol_narrow"
- "currency_display_name"
- "currency_display_name_count_1"
- "currency_display_name_count_other"

Usage

```
cldr_currencies(
  locale = "en",
  currency = currency_code_list$USD,
  element = currencies_elements$currency_symbol
)
```

Arguments

locale	The locale ID for which to obtain the data from the [currencies](#) table.
currency	The currency code (e.g., "USD", "EUR", etc.). A valid set of currency codes can be accessed through the [currency_code_list](#) object.
element	The element from which information will be obtained for the specified locale. A valid set of currency elements can be accessed through the [currencies_elements](#) list object.

Value

A length one character vector.

Examples

If you would like to get the currency display name for the British Pound ("GBP") currency while in the "de" locale, the following invocation of cldr_currencies() can be used.

```
cldr_currencies(
  locale = "de",
  currency = currency_code_list$GBP,
  element = currencies_elements$currency_display_name
)
```

```
#> [1] "Britisches Pfund"
```

---

cldr_dates   *Get a single localized value from the [dates](#) dataset*

Description

The [dates](#) table contains information on how to express dates and this data is localized across 574 locales. The cldr_dates() function allows one to extract a named list using a locale and a specific element. The element values are:

- "months_format_abbrev"
- "months_format_narrow"
- "months_format_wide"
- "days_standalone_narrow"
- "days_standalone_short"
- "days_standalone_wide"
- "quarters_format_abbrev"
- "quarters_format_narrow"
- "quarters_format_wide"
- "quarters_standalone_abbrev"
- "quarters_standalone_narrow"
- "quarters_standalone_wide"
- "dayperiods_format_abbrev"
- "dayperiods_format_narrow"
- "dayperiods_format_wide"
- "dayperiods_standalone_abbrev"
- "dayperiods_standalone_narrow"
- "dayperiods_standalone_wide"
- "eras_abbrev"
- "eras_names"
- "eras_narrow"
- "date_formats"
- "date_skeletons"
- "time_formats"
- "time_skeletons"
- "date_time_available_formats"
- "date_time_append_items"
- "date_time_interval_formats"

Usage

```
cldr_dates(locale = "en", element = dates_elements$months_format_abbrev)
```

Arguments

locale	The locale ID for which to obtain the data from the dates table.
element	The element from which information will be obtained for the specified locale.

Value

A named list.

---

cldr_locale_names   *Get a localized list of locale names from the locale_names dataset*

Description

The locale_names table contains information on how to express components of locale codes and this is localized across 574 locales. The cldr_locale_names() function allows one to extract a named list using a locale and one of the following element names:

- "langs": corresponds to the lang_names column in locale_names
- "scripts": is the script_names column in locale_names
- "territories": is territory_names
- "variants": is variant_names

Usage

```
cldr_locale_names(locale = "en", element = locale_names_elements$lang_names)
```

Arguments

locale	The locale ID for which to obtain the data from the locale_names table.
element	The element from which information will be obtained for the specified locale. A valid set of locale_names elements can be accessed through the locale_names_elements list object.

Value

A named list.

---

cldr_numbers *Get a single localized value from the [numbers](#) dataset*

Description

The [numbers](#) table contains localization data for number usage and this data is available for 574 locales. The cldr_numbers() function allows one to extract a named list using a locale and a specific element. The element values are:

- "default_numbering_system"
- "other_numbering_systems"
- "minimum_grouping_digits"
- "decimal"
- "group"
- "list"
- "percent_sign"
- "plus_sign"
- "minus_sign"
- "approx_sign"
- "exp_sign"
- "sup_exp"
- "per_mille"
- "infinity"
- "nan"
- "time_sep"
- "approx_pattern"
- "at_least_pattern"
- "at_most_pattern"
- "range_pattern"
- "decimal_format"
- "sci_format"
- "percent_format"
- "currency_format"
- "accounting_format"

Usage

```
cldr_numbers(
  locale = "en",
  element = numbers_elements$default_numbering_system
)
```

Arguments

locale	The locale ID for which to obtain the data from the [numbers] table.
element	The element from which information will be obtained for the specified locale.

Value

Either a named list or a length one character vector, depending on the element value.

---

currencies   *A table with localized currency attributes and descriptors*

Description

The currencies table contains localized data for number-related entities across 574 locales. This table has 173,013 rows, one per distinct combination of locale and currency (currency_code), and the following 7 columns:

- locale (character)
- currency_code (character)
- currency_display_name (character)
- currency_symbol (character)
- currency_symbol_narrow (character)
- currency_display_name_count_1 (character)
- currency_display_name_count_other (character)

Usage

    currencies

Format

An object of class tbl_df (inherits from tbl, data.frame) with 175070 rows and 7 columns.

---

dates	*A table with localized date attributes and descriptors*

Description

The dates table contains localized data for constructing dates and times across 574 locales. There are 574 rows and the following 38 columns:

- locale (character)
- months_format_abbrev (named list [length of 12])
- months_format_narrow (named list [length of 12])
- months_format_wide (named list [length of 12])
- months_standalone_abbrev (named list [length of 12])
- months_standalone_narrow (named list [length of 12])
- months_standalone_wide (named list [length of 12])
- days_format_abbrev (named list [length of 7])
- days_format_narrow (named list [length of 7])
- days_format_short (named list [length of 7])
- days_format_wide (named list [length of 7])
- days_standalone_abbrev (named list [length of 7])
- days_standalone_narrow (named list [length of 7])
- days_standalone_short (named list [length of 7])
- days_standalone_wide (named list [length of 7])
- quarters_format_abbrev (named list [length of 4])
- quarters_format_narrow (named list [length of 4])
- quarters_format_wide (named list [length of 4])
- quarters_standalone_abbrev (named list [length of 4])
- quarters_standalone_narrow (named list [length of 4])
- quarters_standalone_wide (named list [length of 4])
- dayperiods_format_abbrev (named list [variable length])
- dayperiods_format_narrow (named list [variable length])
- dayperiods_format_wide (named list [variable length])
- dayperiods_standalone_abbrev (named list [variable length])
- dayperiods_standalone_narrow (named list [variable length])
- dayperiods_standalone_wide (named list [variable length])
- eras_abbrev (named list [length of 4])
- eras_names (named list [length of 4])
- eras_narrow (named list [length of 4])

- date_formats (named list [variable length])
- date_skeletons (named list [length of 4])
- time_formats (named list [variable length])
- time_skeletons (named list [variable length])
- date_time_patterns (named list [length of 4])
- date_time_available_formats (named list [variable length])
- date_time_append_items (named list [length of 11])
- date_time_interval_formats (named list [variable length])

Usage

```
dates
```

Format

An object of class tbl_df (inherits from tbl, data.frame) with 574 rows and 38 columns.

---

dates_generic	*A table with localized generic date attributes and descriptors*

Description

The dates_generic table contains localized data for constructing dates and times across 574 locales. There are 574 rows and the following 38 columns:

- locale (character)
- months_format_abbrev (named list [length of 12])
- months_format_narrow (named list [length of 12])
- months_format_wide (named list [length of 12])
- months_standalone_abbrev (named list [length of 12])
- months_standalone_narrow (named list [length of 12])
- months_standalone_wide (named list [length of 12])
- days_format_abbrev (named list [length of 7])
- days_format_narrow (named list [length of 7])
- days_format_short (named list [length of 7])
- days_format_wide (named list [length of 7])
- days_standalone_abbrev (named list [length of 7])
- days_standalone_narrow (named list [length of 7])
- days_standalone_short (named list [length of 7])
- days_standalone_wide (named list [length of 7])
- quarters_format_abbrev (named list [length of 4])

- quarters_format_narrow (named list [length of 4])

- quarters_format_wide (named list [length of 4])

- quarters_standalone_abbrev (named list [length of 4])

- quarters_standalone_narrow (named list [length of 4])

- quarters_standalone_wide (named list [length of 4])

- dayperiods_format_abbrev (named list [variable length])

- dayperiods_format_narrow (named list [variable length])

- dayperiods_format_wide (named list [variable length])

- dayperiods_standalone_abbrev (named list [variable length])

- dayperiods_standalone_narrow (named list [variable length])

- dayperiods_standalone_wide (named list [variable length])

- eras_abbrev (named list [length of 4])

- eras_names (named list [length of 4])

- eras_narrow (named list [length of 4])

- date_formats (named list [variable length])

- date_skeletons (named list [length of 4])

- time_formats (named list [variable length])

- time_skeletons (named list [variable length])

- date_time_patterns (named list [length of 4])

- date_time_available_formats (named list [variable length])

- date_time_append_items (named list [length of 11])

- date_time_interval_formats (named list [variable length])

Usage

```
dates_generic
```

Format

An object of class tbl_df (inherits from tbl, data.frame) with 574 rows and 38 columns.

---

day_periods   *A table with rule sets for naming periods of a day*

Description

The day_periods table contains rules for naming periods of time throughout a day. There are 519 rows that comprise a day period name and rule for a locale. There may be only two rows associated with a locale but many more if a locale has many names for periods of a day. The following columns are included:

- locale (character)
- period (character)
- from (character)
- to (character)
- at (character)

The period value provides an identifier for the period of time. For a given locale there may typically be "afternoon1" and "evening1" period identifiers. Some may have quite a few periods defined (perhaps with "morning1" and "morning2" rules). A period is either a block of time defined by the from and to columns, or, a set time (like "noon" and "midnight") found in the at column. The period values are typically obtained from this dataset in order to obtain localized text from the dates and dates_generic datasets (within the dayperiods_* columns).

Usage

day_periods

Format

An object of class tbl_df (inherits from tbl, data.frame) with 519 rows and 5 columns.

---

default_locales   *A table containing a mapping of default locale names to base locales*

Description

This is a table that contains base locale names (e.g., "en", "de") alongside their default locale names. This indicates that "en" maps to "en-US" and "de" should map to "de-DE". Throughout the **i18n** datasets, base names are used instead of their expanded equivalents.

There are 228 rows and the following 2 columns:

- default_locale (character)
- base_locale (character)

The default_locale column contains the expanded locale names (e.g., "en-US") that do not normally appear within the *CLDR* datasets but are valid aliases for the base locale names (e.g., "en") found in the base_locale column.

Usage

```
default_locales
```

Format

An object of class tbl_df (inherits from tbl, data.frame) with 228 rows and 2 columns.

---

delimiters   *A table with localized delimiter values*

Description

The delimiters table contains localized information on the preferred and alternate sets of quotation marks across 574 locales. There are 574 rows and the following 5 columns:

- locale (character)
- quotation_start (character)
- quotation_end (character)
- alt_quotation_start (character)
- alt_quotation_end (character)

Usage

```
delimiters
```

Format

An object of class tbl_df (inherits from tbl, data.frame) with 574 rows and 5 columns.

---

element_lists   *Element lists for different* CLDR *data tables*

Description

Several element lists are available for use in the various cldr_*() functions. The list object names, the number of elements they hold, and the functions they nicely pair with are:

Usage

```
locale_list

currency_code_list

locale_names_elements

currencies_elements

dates_elements

numbers_elements

characters_elements
```

Format

An object of class list of length 574.

An object of class list of length 305.

An object of class list of length 4.

An object of class list of length 5.

An object of class list of length 37.

An object of class list of length 26.

An object of class list of length 12.

Details

locale_list (574) -> several cldr_*() functions currency_code_list (305) -> cldr_currency()
currencies_elements (5) -> cldr_currency() locale_names_elements (4) -> cldr_locale_names()
dates_elements (28) -> cldr_dates() numbers_elements (26) -> cldr_numbers() characters_elements
(12) -> cldr_characters()

---

layout   *A table with localized layout data*

Description

The layout table contains data on text layout across 574 locales. There are 574 rows and the
following 3 columns:

- locale (character)
- character_order (character)
- line_order (character)

Usage

```
layout
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 574 rows and 3 columns.

---

locale_names *A table with localized language, script, and territory names*

Description

The `locale_names` table contains localized names for languages, script names, names for territories, and names for variants. There are 574 rows and the following 5 columns:

- `locale` (character)
- `lang_names` (named list [variable length])
- `script_names` (named list [variable length])
- `territory_names` (named list [variable length])
- `variant_names` (named list [variable length])

The `lang_names` column contains named lists for all localized language names. The `script_names` column holds named lists for all localized script names, and `territory_names` has all of the localized territory names per locale. The `variant_names` list column containing named lists for all localized variant names.

Usage

```
locale_names
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 574 rows and 5 columns.

---

numbers   *A table with localized numerical attributes and descriptors*

Description

The numbers table contains localized data for number-related entities across 574 locales. This table
has 574 rows (one per locale) and the following 26 columns:

- locale (character)
- default_numbering_system (character)
- other_numbering_systems (named list [variable length])
- minimum_grouping_digits (integer)
- decimal (character)
- group (character)
- list (character)
- percent_sign (character)
- plus_sign" (character)
- minus_sign (character)
- approx_sign (character)
- exp_sign (character)
- sup_exp (character)
- per_mille (character)
- infinity (character)
- nan (character)
- time_sep (character)
- approx_pattern (character)
- at_least_pattern (character)
- at_most_pattern (character)
- range_pattern (character)
- decimal_format (character)
- sci_format (character)
- percent_format (character)
- currency_format (character)
- accounting_format (character)

The first column, locale, is the locale name (e.g., "en", "de-AT", etc.). The remaining 25 columns
will be explained in separate sections.

Usage

numbers

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 574 rows and 26 columns.

Default Numbering System

The `default_numbering_system` (*CLDR*: `'defaultNumberingSystem'`) column provides an element that indicates which numbering system should be used for presentation of numeric quantities in the given locale.

Other Numbering Systems

The `other_numbering_systems` (*CLDR*: `'otherNumberingSystems'`) column provides an element that defines general categories of numbering systems that are sometimes used in the given locale for formatting numeric quantities. These additional numbering systems are often used in very specific contexts, such as in calendars or for financial purposes. There are currently three defined categories, as follows:

native

Defines the numbering system used for the native digits, usually defined as a part of the script used to write the language. The native numbering system can only be a numeric positional decimal-digit numbering system, using digits with General_Category=Decimal_Number. In locales where the native numbering system is the default, it is assumed that the numbering system `"latn"` (Western Digits `0`-`9`) is always acceptable, and can be selected using the `"-nu"` keyword as part of a Unicode locale identifier.

traditional

Defines the traditional numerals for a locale. This numbering system may be numeric or algorithmic. If the traditional numbering system is not defined, applications should use the native numbering system as a fallback.

finance

Defines the numbering system used for financial quantities. This numbering system may be numeric or algorithmic. This is often used for ideographic languages such as Chinese, where it would be easy to alter an amount represented in the default numbering system simply by adding additional strokes. If the financial numbering system is not specified, applications should use the default numbering system as a fallback.

The categories defined for other numbering systems can be used in a Unicode locale identifier to select the proper numbering system without having to know the specific numbering system by name. To select the Hindi language using the native digits for numeric formatting, use locale ID `"hi-IN-u-nu-native"`. To select the Chinese language using the appropriate financial numerals, use locale ID: `"zh-u-nu-finance"`. With the Tamil language using the traditional Tamil numerals, use locale ID `"ta-u-nu-traditio"`. As a last example, to select the Arabic language using western digits `0`-`9`, use locale ID `"ar-u-nu-latn"`.

Minimum Grouping Digits

The `minimum_grouping_digits` (*CLDR*: `'minimumGroupingDigits'`) value can be used to suppress groupings below a certain value. This is used for languages such as Polish, where one would only write the grouping separator for values above `9999`. The `minimum_grouping_digits` value contains the default for the locale.

Number Symbols

Number symbols define the localized symbols that are commonly used when formatting numbers in a given locale. These symbols can be referenced using a number formatting pattern.

The decimal (*CLDR*: 'decimal') symbol separates the integer and fractional part of the number. The group (*CLDR*: 'group') symbol separates clusters of integer digits to make large numbers more legible; commonly used for thousands (grouping size 3, e.g. "100,000,000") or in some locales, ten-thousands (grouping size 4, e.g. "1,0000,0000"). There may be two different grouping sizes: The primary grouping size used for the least significant integer group, and the secondary grouping size used for more significant groups; these are not the same in all locales (e.g. "12,34,56,789"). If a pattern contains multiple grouping separators, the interval between the last one and the end of the integer defines the primary grouping size, and the interval between the last two defines the secondary grouping size. All others are ignored, so "#,##,###,####" == "###,###,####" == "##,#,###,####".

The list (*CLDR*: 'list') symbol is used to separate numbers in a list intended to represent structured data such as an array. It must be different from the decimal value. This list separator is for non-linguistic usage as opposed to the list patterns for linguistic lists (e.g. "Bob, Carol, and Ted").

The plus_sign (*CLDR*: 'plusSign') is the preferred symbol for expressing a positive value and the minus_sign (*CLDR*: 'minusSign') is for negative values. It can be used to produce modified patterns, so that "3.12" is formatted as "+3.12", for example. The standard number patterns (except for accounting notation) will contain the minus_sign, explicitly or implicitly. In the explicit pattern, the value of the plus_sign can be substituted for the value of the minus_sign to produce a pattern that has an explicit plus sign.

The approx_sign (*CLDR*: 'approximatelySign') element contains a symbol used to denote an approximate value. The symbol is substituted in place of the minus_sign using the same semantics as plus_sign substitution.

The exp_sign (*CLDR*: 'exponential') provides a symbol used for separating the mantissa and exponent values. The exponential notation in sup_exp (*CLDR*: 'superscriptingExponent') could alternatively be used to show a format like "1.23 × 104". The superscripting can use markup, such as ⁴ in HTML, or for the special case of Latin digits, use superscripted numeral characters.

The percent_sign (*CLDR*: 'percentSign') is a symbol used to indicate a percentage (1/100th) amount. If present, the value might require multiplication by 100 before formatting. The per_mille (*CLDR*: 'perMille') symbol used to indicate a per mille (1/1000th) amount. If present, the value might need to be multiplied by 1000 before formatting.

The infinity sign is provided in the infinity (*CLDR*: 'infinity') element. The nan element (*CLDR*: 'nan') has the NaN (not a number) sign. These elements both correspond to the IEEE bit patterns for infinity and NaN.

The time_sep (*CLDR*: 'timeSeparator') pattern allows the same time format to be used for multiple number systems when the time separator depends on the number system. For example, the time format for Arabic should be a colon when using the Latin numbering system, but when the Arabic numbering system is used, the traditional time separator in older print styles was often Arabic comma.

Miscellaneous Patterns

There are several miscellaneous patterns for special purposes. The `approx_pattern` (*CLDR*: `'approximately'`) indicates an approximate number, such as: `"~99"`. With the pattern called `at_most_pattern` (*CLDR*: `'atMost'`) we can describe an upper limit. This indicates that, for example, there are 99 items or fewer. The `at_least_pattern` (*CLDR*: `'atLeast'`) describes a lower limit. This might be `"99+"` to indicate that there are 99 items or more. With the `range_pattern` (*CLDR*: `'range'`), a range of numbers, such as `"99–103"`, can be used to indicate that there are from 99 to 103 items.

Number Formats

Number formats are used to define the rules for formatting numeric quantities. Different formats are provided for different contexts. The `decimal_format` (*CLDR*: `'decimalFormats'`) is the prescribed locale-specific way to write a base 10 number. Variations of the `decimal_format` pattern are provided that allow compact number formatting. The `percent_format` (*CLDR*: `'percentFormats'`) is the pattern to use for percentage formatting. The pattern for use with scientific (exponent) formatting is provided as `sci_format` (*CLDR*: `'scientificFormats'`). The pattern for use with currency formatting is found in `currency_format` (*CLDR*: `'currencyFormats'`). This format contains a few additional structural options that allow proper placement of the currency symbol relative to the numeric quantity. The `accounting_format` (*CLDR*: `'accountingFormats'`) pattern is to be used to generate accounting-style formatting.

---

script_metadata	*A table with metadata for a wide variety of script types*

Description

The `script_metadata` table contains metadata for various script types. There are 170 rows and the following 11 columns:

- `script` (character)
- `sample_char` (character)
- `rank` (integer)
- `script` (character)
- `rtl` (character)
- `lb_letters` (character)
- `has_case` (character)
- `shaping_req` (character)
- `ime` (character)
- `density` (integer)
- `origin_country` (character)
- `likely_lang` (character)

Usage

```
script_metadata
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 170 rows and 11 columns.

---

tz_bcp_id   *A table with BCP47 Olson/IANA-style and canonical time zone IDs*

Description

The `tz_bcp_id` table provides a lookup for converting between BCP47 Olson/IANA-style time zone IDs and the canonical forms (according to BCP47). There are 593 rows and the following 3 columns:

- `tz_bcp_id` (character)
- `tz_canonical` (character)
- `description` (character)

Usage

```
tz_bcp_id
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 593 rows and 3 columns.

---

tz_exemplar   *A table with localized names for all time zone exemplar cities*

Description

The `tz_exemplar` table contains localized names for all exemplar cities used in time zone names. There are 574 rows and a column for each exemplar city name (comprising 442 columns; the `locale` column is first). To have syntactical column names, all slashes in exemplar city names are instead represented with period characters (e.g., `Indiana/Vincennes` is `Indiana.Vincennes`). Some exemplar cities are not actually cities and these are: `UTC.long.standard` (*en*: `"Coordinated Universal Time"`), `UTC.short.standard` (*en*: `"UTC"`), and `Unknown` (*en*: `"Unknown City"`).

Usage

```
tz_exemplar
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 574 rows and 443 columns.

`tz_formats`	*A table with localized time zone formatting information*

Description

The `tz_formats` table contains localized formatting information across all locales. There are 574 rows and the following 7 columns:

- `locale` (character)
- `hour_format` (character)
- `gmt_format` (character)
- `gmt_zero_format` (character)
- `region_format` (character)
- `region_format_daylight` (character)
- `region_format_standard` (character)
- `region_format_fallback` (character)

Usage

```
tz_formats
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 574 rows and 8 columns.

`tz_map`	*A table with names of map-based time zones*

Description

The `tz_map` table contains names for all map-based time zone names. There are 598 rows and the following 4 columns:

- `canonical_tz_name` (character)
- `territory` (character)
- `tz_name` (character)

Usage

```
tz_map
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 598 rows and 3 columns.

---

tz_metazone_names   *A table with localized time zone names for all metazones*

Description

The tz_metazone_names table contains localized time zone names for all metazones (e.g., America/Eastern).
There can be a variety of time zone names, comprising long and short forms (e.g., Eastern Time
and ET) and this is further segmented by generic, standard, and daylight forms (an example, using
short forms, is ET, EST, and EDT). There are 465 rows and a column for each metazone (comprising
159 columns; the locale column is first).

Usage

```
tz_metazone_names
```

Format

An object of class tbl_df (inherits from tbl, data.frame) with 465 rows and 160 columns.

---

tz_metazone_users   *A table that links canonical tz names with their metazone*

Description

The tz_metazone_users table allows for a lookup of canonical time zone name to which meta-
zone each uses. As an example, the canonical time zone America/Vancouver corresponds to the
America_Pacific metazone (this is the long ID, but there is often a short ID available as well). The
metazone_long_id can be used to get a localized metazone name by use of the tz_metazone_names
table.

There are 293 rows and the following 4 columns:

- canonical_tz_name (character)
- territory (character)
- metazone_long_id (character)
- metazone_short_id (character)

Usage

```
tz_metazone_users
```

Format

An object of class tbl_df (inherits from tbl, data.frame) with 293 rows and 4 columns.

---

units   *A table with localized data on units*

Description

The `units` table contains localized character data across 574 locales. There are 1722 rows and 1281 columns. Each row represents a display type (`"long"`, `"short"`, or `"narrow"`) for each of the locales.

Following the `locale` and `type` columns, each unit and its subelements are provided as a cluster of columns in the form `"<<category>-unit name>.<subelement name>"`. The subelement names are:

- `"displayName"`
- `"unitPattern-count-one"`
- `"unitPattern-count-other"`
- `"unitPattern-count-zero"`
- `"unitPattern-count-two"`
- `"unitPattern-count-few"`
- `"unitPattern-count-many"`

The `"displayName"` is the localized name for a unit when displayed outside of a pattern. The `"unitPattern-count-*"` subelements provide the localized forms of the unit when the value is exactly 0 (`"unitPattern-count-one"`), 1 (`"unitPattern-count-one"`), 2 (`"unitPattern-count-two"`), and, when the value constitutes a few (`"unitPattern-count-few"`) or many (`"unitPattern-count-many"`) units. Every other case is handled by `"unitPattern-count-other"`.

Usage

```
units
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 1722 rows and 1281 columns.

Index