Package 'econid'

May 8, 2026

Title Economic Entity Identifier Standardization

Version 0.0.3

Description Provides utility functions for standardizing economic entity (economy, aggregate, institution, etc.) name and id in economic datasets such as those published by the International Monetary Fund and World Bank. Aims to facilitate consistent data analysis, reporting, and joining across datasets. Used as a foundational building block in the 'EconDataverse' family of packages (<https://www.econdataverse.org>).

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports cli, dplyr, purrr, rlang, stringr, tibble, tidyr

Suggests testthat (>= 3.0.0), withr

Config/testthat/edition 3

URL https://teal-insights.github.io/r-econid/, https://github.com/Teal-Insights/r-econid

BugReports https://github.com/Teal-Insights/r-econid/issues

Depends R (>= 4.1.0)

LazyData true

NeedsCompilation no

Author L. Teal Emery [cre], Christopher C. Smith [aut], Christoph Scheuch [ctb], Teal Insights [cph]

Maintainer L. Teal Emery <lte@tealinsights.com>

Repository CRAN

Date/Publication 2026-01-08 06:11:30 UTC

Contents

`add_entity_pattern` *Add a custom entity pattern*

Description

This function allows users to extend the default entity patterns with a custom entry.

Usage

```
add_entity_pattern(
  entity_id,
  entity_name,
  entity_type,
  aliases = NULL,
  entity_regex = NULL
)
```

Arguments

`entity_id`	A unique identifier for the entity.
`entity_name`	The standard (canonical) name of the entity.
`entity_type`	A character string describing the type of entity ("economy", "organization", "aggregate", or "other").
`aliases`	An optional character vector of alternative names identifying the entity. If provided, these are automatically combined (using the pipe operator, "|") with `entity_name` and `entity_id` to construct a regular expression pattern.
`entity_regex`	An optional custom regular expression pattern. If supplied, it overrides the regex automatically constructed from `aliases`.

Details

Custom entity patterns can be added at the top of a script (or interactively) and will be appended to the built-in patterns when using `list_entity_patterns()`. This makes it possible for users to register alternative names (aliases) for entities that might appear in their economic datasets.

The custom entity patterns are kept separately and are appended to the default patterns when retrieving the entity_patterns via `list_entity_patterns()`. The custom patterns will only persist for the length of the R session.

Value

NULL. As a side effect of the function, the custom pattern is stored in an internal tibble for the current session.

Examples

```
add_entity_pattern(
  "ASN",
  "Association of Southeast Asian Nations",
  "economy",
  aliases = c("ASEAN")
)
patterns <- list_entity_patterns()
print(patterns[patterns$entity_id == "ASN", ])
```

---

entity_patterns *Entity Patterns*

Description

A dataset containing patterns for matching entity names. This dataset is accessible through list_entity_patterns.

Usage

```
entity_patterns
```

Format

A data frame with the following columns:

entity_id Unique identifier for the entity

entity_name entity name

iso3c ISO 3166-1 alpha-3 code

iso2c ISO 3166-1 alpha-2 code

entity_type Type of entity ("economy", "organization", "aggregate", or "other")

entity_regex Regular expression pattern for matching entity names

Source

Data manually prepared by Teal L. Emery

list_entity_patterns *List entity patterns*

Description

This function returns a tibble containing regular expression patterns for identifying economic indicators. It combines the patterns from the built-in `entity_patterns` dataset with any custom patterns stored in the `.econid_env` environment.

Usage

```
list_entity_patterns()
```

Value

A data frame with the following columns:

entity_id entity id

entity_name entity name

iso2c ISO 3166-1 alpha-2 code

iso3c ISO 3166-1 alpha-3 code

entity_type entity type

entity_regex Regular expression pattern for matching entity names

Examples

```
patterns <- list_entity_patterns()
```

reset_custom_entity_patterns *Reset custom entity patterns*

Description

This function resets all custom entity patterns that have been added during the current R session.

Usage

```
reset_custom_entity_patterns()
```

Value

Invisibly returns NULL.

Examples

```
add_entity_pattern("EU", "European Union", "economy")
reset_custom_entity_patterns()
patterns <- list_entity_patterns()
print(patterns[patterns$entity_id == "EU", ])
```

`standardize_entity` *Standardize Entity Identifiers*

Description

Standardizes entity identifiers (e.g., name, ISO code) in an economic data frame by matching them against a predefined list of regex patterns to add columns containing standardized identifiers to the data frame.

Usage

```
standardize_entity(
  data,
  ...,
  output_cols = c("entity_id", "entity_name", "entity_type"),
  prefix = NULL,
  fill_mapping = NULL,
  default_entity_type = NA_character_,
  warn_ambiguous = TRUE,
  overwrite = TRUE,
  warn_overwrite = TRUE,
  .before = NULL
)
```

Arguments

`data`	A data frame or tibble containing entity identifiers to standardize
`...`	Columns containing entity names and/or IDs. These can be specified using unquoted column names (e.g., `entity_name`, `entity_id`) or quoted column names (e.g., `"entity_name"`, `"entity_id"`). Must specify at least one column. If two columns are specified, the first is assumed to be the entity name and the second is assumed to be the entity ID.
`output_cols`	Character vector specifying desired output columns. Options are "entity_id", "entity_name", "entity_type", "iso3c", "iso2c". Defaults to c("entity_id", "entity_name", "entity_type").
`prefix`	Optional character string to prefix the output column names. Useful when standardizing multiple entities in the same dataset (e.g., "country", "counterpart"). If provided, output columns will be named prefix_entity_id, prefix_entity_name, etc. (with an underscore automatically inserted between the prefix and the column name).

`fill_mapping` Named character vector specifying how to fill missing values when no entity match is found. Names should be output column names (without prefix), and values should be input column names (from `...`). For example, `c(entity_id = "country_code", entity_name = "country_name")` will fill missing entity_id values with values from the country_code column and missing entity_name values with values from the country_name column.

`default_entity_type` Character or NA; the default entity type to use for entities that do not match any of the patterns. Options are "economy", "organization", "aggregate", "other", or NA_character_. Defaults to NA_character_. This argument is only used when "entity_type" is included in output_cols.

`warn_ambiguous` Logical; whether to warn about ambiguous matches

`overwrite` Logical; whether to overwrite existing entity_* columns

`warn_overwrite` Logical; whether to warn when overwriting existing entity_* columns. Defaults to TRUE.

`.before` Column name or position to insert the standardized columns before. If NULL (default), columns are inserted at the beginning of the dataframe. Can be a character vector specifying the column name or a numeric value specifying the column index. If the specified column is not found in the data, an error is thrown.

Value

A data frame with standardized entity information merged with the input data. The standardized columns are placed directly to the left of the first target column.

Examples

```
# Standardize entity names and IDs in a data frame
test_df <- tibble::tribble(
  ~entity,         ~code,
  "United States",  "USA",
  "united.states",  NA,
  "us",             "US",
  "EU",             NA,
  "NotACountry",    NA
)

standardize_entity(test_df, entity, code)

# Standardize with fill_mapping for unmatched entities
standardize_entity(
  test_df,
  entity, code,
  fill_mapping = c(entity_id = "code", entity_name = "entity")
)

# Standardize multiple entities in sequence with a prefix
df <- data.frame(
  country_name = c("United States", "France"),
```

```
  counterpart_name = c("China", "Germany")
)
df |>
  standardize_entity(
    country_name
  ) |>
  standardize_entity(
    counterpart_name,
    prefix = "counterpart"
  )
```

Index