

Package ‘pslr’

July 19, 2026

Type Package

Title Public Suffix List Engine

Version 1.1.1

Description A focused implementation of the Public Suffix List (PSL). Bundles a reproducible, pinned PSL snapshot and implements the official prevailing-rule algorithm to answer public-suffix (eTLD) and registrable-domain (eTLD+1) queries. Distinguishes ICANN and PRIVATE rule sections, accepts Unicode and ASCII hostnames via 'punycode' canonicalization, and supports an explicit, validated offline refresh path. The matcher is compiled with 'cpp11' and requires no external system library. Used as the PSL engine by the 'rurl' package.

License MIT + file LICENSE

Language en-US

URL <https://bart-turczynski.github.io/pslr/>,
<https://github.com/bart-turczynski/pslr>,
<https://CRAN.R-project.org/package=pslr>,
<https://bart-turczynski.r-universe.dev>

BugReports <https://github.com/bart-turczynski/pslr/issues>

Encoding UTF-8

Depends R (>= 4.1.0)

Imports punycode (>= 1.1.0), tools, utils

LinkingTo cpp11

Suggests cucumber (>= 2.0.0), curl, digest, knitr, oysteR, rmarkdown,
rosv, testthat (>= 3.0.0), withr

Config/testthat/edition 3

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

X-schema.org-keywords public suffix list, PSL, eTLD, eTLD+1,
registrable domain, ICANN, domain parsing, domain extraction,
domain names, TLD, cookie domains, IDNA, public suffix

NeedsCompilation yes
Author Bart Turczynski [aut, cre] (ORCID:
 <<https://orcid.org/0000-0002-8788-7980>>)
Maintainer Bart Turczynski <bartek@turczynski.pl>
Repository CRAN
Date/Publication 2026-07-19 07:20:02 UTC

Contents

is_public_suffix	2
psl_cache_prune	3
psl_outdated	5
psl_refresh	6
psl_rules	7
psl_use	8
psl_version	9
public_suffix	10
public_suffix_rule	11
registrable_domain	12
suffix_extract	14

Index	16
--------------	-----------

is_public_suffix	<i>Is a host itself a public suffix?</i>
------------------	--

Description

TRUE exactly when the valid canonical host equals its own public suffix under the selected policy. Returns NA whenever `public_suffix()` would return NA (missing or invalid input, or an unresolved host under `unknown = "na"`). Under the default `unknown = "default"`, an unlisted single label such as `"madeuptld"` is TRUE via the implicit `*` rule; ask `unknown = "na"` to test explicit membership instead.

Usage

```
is_public_suffix(  
  domain,  
  section = "all",  
  unknown = "default",  
  invalid = "na",  
  engine = psl_default_engine()  
)
```

Arguments

domain	Character vector of DNS hostnames (not URLs). Each element may be a mixed-case ASCII, Unicode, or A-label hostname, a single label, or a hostname with exactly one terminal root dot. See Input contract .
section	Which rule sections are eligible: "all" (default; ICANN and PRIVATE), "icann", or "private". Section filtering happens before prevailing-rule selection, so "private" does not silently add ICANN rules; a host matching no rule in the section falls through to the implicit default rule unless unknown = "na".
unknown	"default" (default) applies the spec's implicit * rule, so an unlisted single label is its own public suffix; "na" returns NA when no explicit rule in the selected section matches.
invalid	"na" (default) returns NA for each invalid element without a warning; "error" aborts on the first invalid element, reporting its 1-based index.
engine	The psl_engine to query against; defaults to the session-global engine selected by psl_use(), so most callers never set it. Pass an engine from psl_engine() to resolve hosts against a specific snapshot in isolation.

Value

A logical vector with length(domain), preserving the names of domain.

Input contract

NA is treated as missing (returns NA), not invalid. Invalid elements include empty or whitespace-only strings, leading or consecutive dots, URL syntax, IPv6 addresses, canonical dotted-decimal IPv4 literals, and labels that fail hostname/IDNA validation. Wrong argument types and non-scalar or unknown option values always abort regardless of invalid.

See Also

[public_suffix\(\)](#)

Examples

```
is_public_suffix("com")
is_public_suffix("example.com")
is_public_suffix("madeuptld")
is_public_suffix("madeuptld", unknown = "na")
```

psl_cache_prune

Prune stale on-disk PSL cache snapshots

Description

Removes superseded psl-<hex>.dat snapshot files from the user cache directory, always keeping the snapshot named by the active commit marker plus the keep most-recent other snapshots by modification time.

Usage

```
psl_cache_prune(keep = 1L)
```

Arguments

keep	Number of previous snapshots to retain <i>in addition to</i> the active one, as a single non-negative whole number. The default 1 keeps the current snapshot and one previous snapshot (two .dat files). 0 keeps only the active snapshot; the active snapshot is never removed, even then.
------	---

Details

Each `psl_refresh()` that finds changed upstream content writes a new content-addressed snapshot and repoints the commit marker at it, but never removes the snapshot it supersedes; across many refreshes these accumulate. `psl_cache_prune()` reclaims that space.

This operates on the *on-disk* snapshot files and is distinct from `psl_cache_clear()`, which flushes the in-memory match-result cache for the current session: pruning deletes stale .dat files from disk to reclaim space, whereas clearing only discards computed query results. Pruning never changes which list is active and never removes the active snapshot, so the active matcher and a later `psl_use("cache")` keep working.

When there is no cache directory or no commit marker (nothing has been published yet), there is no active snapshot to anchor retention on, so the call is a no-op that returns an empty vector rather than an error.

Value

Invisibly, a character vector of the removed snapshot file paths, empty when nothing was pruned.

See Also

`psl_refresh()`, which writes the snapshots this prunes; `psl_use()`.

Examples

```
if (interactive()) {  
  psl_refresh(force = TRUE)  
  psl_cache_prune() # keep the current snapshot and one previous  
  psl_cache_prune(keep = 0) # keep only the active snapshot  
}
```

psl_outdated	<i>Is the active Public Suffix List snapshot stale?</i>
--------------	---

Description

An offline staleness check: compares the `list_date` of the list currently active in this session (see [psl_version\(\)](#)) against the current time and reports whether it is older than `max_age` days. The Public Suffix List changes continually upstream, so a long-lived bundled snapshot drifts from the live list over time; this is the signal to consider [psl_refresh\(\)](#).

Usage

```
psl_outdated(max_age = 180)
```

Arguments

<code>max_age</code>	Maximum age, in days, before the active snapshot is considered outdated. A single positive number; defaults to 180.
----------------------	---

Details

The check reads only the already-loaded active metadata: it never touches the network and never activates a different list. When the active list's `list_date` is unknown (NA) or cannot be parsed, the result is NA – staleness is undetermined rather than assumed either way.

The age of the active snapshot in days is attached to the result as the `"age_days"` attribute (a double, or NA when the date is unknown), so a caller that wants the magnitude as well as the verdict need not recompute it.

Value

A single logical, with an `"age_days"` attribute: TRUE when the active list is more than `max_age` days old, FALSE when it is fresher, and NA when its date is unknown.

See Also

[psl_version\(\)](#), [psl_refresh\(\)](#)

Examples

```
# Is the active snapshot older than the default threshold?
psl_outdated()

# The age in days is available without recomputing it:
attr( "psl_outdated()", "age_days" )

# Use a stricter threshold:
psl_outdated(max_age = 30)
```

psl_refresh

Refresh the cached Public Suffix List from upstream

Description

Downloads, validates, and publishes a fresh Public Suffix List into the user cache. This is the only function in the package that accesses the network, and only when you call it explicitly.

Usage

```
psl_refresh(
  url = "https://publicsuffix.org/list/public_suffix_list.dat",
  force = FALSE,
  activate = FALSE
)
```

Arguments

url	Absolute https URL of the list source. Defaults to the official list. URLs with another scheme or embedded credentials are rejected, and a redirect to a non-HTTPS URL is refused.
force	When FALSE (default), a successfully validated cache younger than 24 hours is reused without a download, respecting upstream download guidance. TRUE forces a fresh download.
activate	When TRUE, the resulting snapshot becomes the active list for the session, exactly as <code>psl_use()</code> would activate it. When FALSE (default), the cache is updated but the active list is unchanged.

Details

Cache age is measured from the successful network retrieval timestamp; reusing a fresh cache does not advance that timestamp. The download goes to a temporary file in binary mode and must be no larger than a documented maximum (16 MiB). The source is then fully validated – UTF-8, section markers, rule grammar, conflicting rules, and successful canonicalization of every rule – and exact same-section duplicates warn once and are deduplicated. Source and metadata are published only after validation succeeds, using an atomic commit that never exposes a partial or mismatched snapshot. A failed refresh never replaces a valid cache or the active matcher.

Value

Invisibly, a one-row `data.frame` shaped like `psl_version()` describing the selected cache snapshot, whether or not it was activated.

See Also

`psl_use()`, `psl_version()`

Examples

```
if (interactive()) {  
  psl_refresh()  
  psl_refresh(force = TRUE, activate = TRUE)  
}
```

psl_rules

Rules of the active Public Suffix List

Description

Returns the explicit rules of the active list as a base [data.frame](#), one row per rule. The implicit default * rule is not included.

Usage

```
psl_rules(section = "all")
```

Arguments

section Which rule sections to return: "all" (default), "icann", or "private".

Value

A base [data.frame](#) with columns, in order: rule (original source rule text), canonical_rule (the canonicalized rule, including the *. or ! marker), kind ("normal", "wildcard", or "exception"), section ("icann" or "private"), and labels (integer rule depth, counting a wildcard label). Rows are ordered first by section (ICANN before PRIVATE) and then by source-file order.

See Also

[psl_version\(\)](#), [public_suffix_rule\(\)](#)

Examples

```
head(psl_rules("icann"))  
nrow(psl_rules("private"))
```

psl_use

Choose the active Public Suffix List for this session

Description

Switches the list backing every query in the current R session. The change is session-only and is validated before any session state changes; a failure leaves the previously active list usable. A successful switch invalidates the match-result cache.

Usage

```
psl_use(source = "bundled", path = NULL)
```

Arguments

source	Where to load the list from: "bundled" (the pinned package snapshot), "cache" (the latest successfully validated snapshot from psl_refresh()), or "path" (a custom file).
path	For source = "path", a single readable PSL-format UTF-8 file containing one complete ICANN section and one complete PRIVATE section, using official markers. Must be NULL for any other source.

Details

A custom path is held to the same runtime duplicate policy as [psl_refresh\(\)](#): exact same-section duplicates warn once and are deduplicated, while conflicting rule kinds for the same labels are fatal. Cache and custom-path sources are read in source form and indexed under the runtime normalizer; they never reuse the bundled generated index.

Value

Invisibly, the [psl_version\(\)](#) row for the newly active list.

See Also

[psl_refresh\(\)](#), [psl_version\(\)](#), [psl_rules\(\)](#)

Examples

```
psl_use("bundled")
if (interactive()) {
  psl_use("cache")
  psl_use("path", path = "my_list.dat")
}
```

psl_version

Identity of the active Public Suffix List

Description

Returns a one-row [data.frame](#) describing the list currently active in this R session: its source-snapshot provenance and the normalization identifiers actually used to index the active matcher. Reproducing a query result requires both the active-list identity and these normalization identifiers (PRD s10), so a reproducibility-sensitive workflow should record this row.

Usage

```
psl_version()
```

Details

The columns, in order, are:

source "bundled", "cache", or "path".

url Source URL of the active snapshot: the upstream download URL for the bundled list; NA for a "cache" or "path" source.

path File path of a "cache" or "path" source; NA otherwise.

retrieved_at Network retrieval timestamp, or NA.

list_date Upstream list date, or NA when unknown.

commit Upstream commit SHA, or NA when unknown.

size Source byte size (integer).

checksum Source checksum, including its algorithm prefix (e.g. "sha256: ...").

normalizer The dependency providing canonicalization, currently "punycode".

normalizer_version Its installed package version.

normalization_profile Its stable case-mapping / IDNA / validation profile identifier.

unicode_version The Unicode data version used by that profile.

Unavailable metadata is a typed NA, never omitted. The normalization identifiers describe the implementation used by the current session, whether the active list came from the bundled snapshot, the user cache, or a custom path; an in-memory compatibility rebuild (PRD s8.3) updates them without altering the shipped source identity or checksum.

Value

A one-row base [data.frame](#) with the columns described in Details.

See Also

[psl_use\(\)](#), [psl_refresh\(\)](#), [psl_rules\(\)](#)

Examples

```
psl_version()
```

public_suffix	<i>Public suffix of a host</i>
---------------	--------------------------------

Description

Returns the public suffix (effective top-level domain, eTLD) of each host under the selected Public Suffix List policy, following the official prevailing-rule algorithm.

Usage

```
public_suffix(
  domain,
  section = "all",
  output = "ascii",
  unknown = "default",
  invalid = "na",
  engine = psl_default_engine()
)
```

Arguments

domain	Character vector of DNS hostnames (not URLs). Each element may be a mixed-case ASCII, Unicode, or A-label hostname, a single label, or a hostname with exactly one terminal root dot. See Input contract .
section	Which rule sections are eligible: "all" (default; ICANN and PRIVATE), "icann", or "private". Section filtering happens before prevailing-rule selection, so "private" does not silently add ICANN rules; a host matching no rule in the section falls through to the implicit default rule unless unknown = "na".
output	"ascii" (default) returns lowercase A-labels; "unicode" decodes them after matching. A terminal root dot is preserved either way.
unknown	"default" (default) applies the spec's implicit * rule, so an unlisted single label is its own public suffix; "na" returns NA when no explicit rule in the selected section matches.
invalid	"na" (default) returns NA for each invalid element without a warning; "error" aborts on the first invalid element, reporting its 1-based index.
engine	The psl_engine to query against; defaults to the session-global engine selected by psl_use(), so most callers never set it. Pass an engine from psl_engine() to resolve hosts against a specific snapshot in isolation.

Value

A character vector with length(domain), preserving the names of domain. Other attributes are dropped.

Input contract

NA is treated as missing (returns NA), not invalid. Invalid elements include empty or whitespace-only strings, leading or consecutive dots, URL syntax, IPv6 addresses, canonical dotted-decimal IPv4 literals, and labels that fail hostname/IDNA validation. Wrong argument types and non-scalar or unknown option values always abort regardless of invalid.

See Also

[registrable_domain\(\)](#), [is_public_suffix\(\)](#), [suffix_extract\(\)](#), [public_suffix_rule\(\)](#)

Examples

```
public_suffix("www.example.com")
public_suffix("example.co.uk")
public_suffix("example.com.")
public_suffix("madeuptld", unknown = "na")
```

public_suffix_rule	<i>Inspect the prevailing PSL rule for each host</i>
--------------------	--

Description

Inspect the prevailing PSL rule for each host

Usage

```
public_suffix_rule(
  domain,
  section = "all",
  unknown = "default",
  invalid = "na",
  engine = psl_default_engine()
)
```

Arguments

domain	Character vector of DNS hostnames (not URLs). Each element may be a mixed-case ASCII, Unicode, or A-label hostname, a single label, or a hostname with exactly one terminal root dot. See Input contract .
section	Which rule sections are eligible: "all" (default; ICANN and PRIVATE), "icann", or "private". Section filtering happens before prevailing-rule selection, so "private" does not silently add ICANN rules; a host matching no rule in the section falls through to the implicit default rule unless unknown = "na".
unknown	"default" (default) applies the spec's implicit * rule, so an unlisted single label is its own public suffix; "na" returns NA when no explicit rule in the selected section matches.

invalid	"na" (default) returns NA for each invalid element without a warning; "error" aborts on the first invalid element, reporting its 1-based index.
engine	The psl_engine to query against; defaults to the session-global engine selected by psl_use(), so most callers never set it. Pass an engine from psl_engine() to resolve hosts against a specific snapshot in isolation.

Value

A base `data.frame` with one row per input and columns, in order: `input` (original), `host_ascii` (canonical A-label host), `rule` (the canonical rule including `*`, `!`, `"*"` for the implicit default), `kind` ("normal", "wildcard", "exception", or "default"), `rule_section` ("icann", "private", or NA for the default/no result), and `public_suffix_ascii` (the derived A-label public suffix). Invalid rows are NA in every derived column. A valid host left unresolved by unknown = "na" keeps `host_ascii` while the rule and suffix columns are NA. An exception rule retains its `!` for auditability. Zero-length input returns a zero-row frame; all-invalid input keeps one row per input.

Input contract

NA is treated as missing (returns NA), not invalid. Invalid elements include empty or whitespace-only strings, leading or consecutive dots, URL syntax, IPv6 addresses, canonical dotted-decimal IPv4 literals, and labels that fail hostname/IDNA validation. Wrong argument types and non-scalar or unknown option values always abort regardless of invalid.

See Also

`public_suffix()`, `suffix_extract()`

Examples

```
public_suffix_rule("www.example.co.uk")
public_suffix_rule("madeuptld")
```

registrable_domain	<i>Registrable domain of a host</i>
--------------------	-------------------------------------

Description

Returns the registrable domain (eTLD+1) of each host: its public suffix plus one host label to the left. It is NA when no such label exists (the host is itself a public suffix) or when the public suffix is NA.

Usage

```
registrable_domain(
    domain,
    section = "all",
    output = "ascii",
    unknown = "default",
    invalid = "na",
    engine = psl_default_engine()
)
```

Arguments

domain	Character vector of DNS hostnames (not URLs). Each element may be a mixed-case ASCII, Unicode, or A-label hostname, a single label, or a hostname with exactly one terminal root dot. See Input contract .
section	Which rule sections are eligible: "all" (default; ICANN and PRIVATE), "icann", or "private". Section filtering happens before prevailing-rule selection, so "private" does not silently add ICANN rules; a host matching no rule in the section falls through to the implicit default rule unless unknown = "na".
output	"ascii" (default) returns lowercase A-labels; "unicode" decodes them after matching. A terminal root dot is preserved either way.
unknown	"default" (default) applies the spec's implicit * rule, so an unlisted single label is its own public suffix; "na" returns NA when no explicit rule in the selected section matches.
invalid	"na" (default) returns NA for each invalid element without a warning; "error" aborts on the first invalid element, reporting its 1-based index.
engine	The psl_engine to query against; defaults to the session-global engine selected by psl_use(), so most callers never set it. Pass an engine from psl_engine() to resolve hosts against a specific snapshot in isolation.

Value

A character vector with length(domain), preserving the names of domain. Other attributes are dropped.

Input contract

NA is treated as missing (returns NA), not invalid. Invalid elements include empty or whitespace-only strings, leading or consecutive dots, URL syntax, IPv6 addresses, canonical dotted-decimal IPv4 literals, and labels that fail hostname/IDNA validation. Wrong argument types and non-scalar or unknown option values always abort regardless of invalid.

See Also

[public_suffix\(\)](#), [is_public_suffix\(\)](#), [suffix_extract\(\)](#)

Examples

```
registrable_domain("www.example.co.uk")
registrable_domain("com")
registrable_domain("foo.madeupld", unknown = "na")
```

suffix_extract

Split hosts into subdomain, registrant label, and public suffix

Description

Split hosts into subdomain, registrant label, and public suffix

Usage

```
suffix_extract(
  domain,
  section = "all",
  output = "ascii",
  unknown = "default",
  invalid = "na",
  engine = psl_default_engine()
)
```

Arguments

domain	Character vector of DNS hostnames (not URLs). Each element may be a mixed-case ASCII, Unicode, or A-label hostname, a single label, or a hostname with exactly one terminal root dot. See Input contract .
section	Which rule sections are eligible: "all" (default; ICANN and PRIVATE), "icann", or "private". Section filtering happens before prevailing-rule selection, so "private" does not silently add ICANN rules; a host matching no rule in the section falls through to the implicit default rule unless unknown = "na".
output	"ascii" (default) returns lowercase A-labels; "unicode" decodes them after matching. A terminal root dot is preserved either way.
unknown	"default" (default) applies the spec's implicit * rule, so an unlisted single label is its own public suffix; "na" returns NA when no explicit rule in the selected section matches.
invalid	"na" (default) returns NA for each invalid element without a warning; "error" aborts on the first invalid element, reporting its 1-based index.
engine	The psl_engine to query against; defaults to the session-global engine selected by psl_use(), so most callers never set it. Pass an engine from psl_engine() to resolve hosts against a specific snapshot in isolation.

Value

A base [data.frame](#) with one row per input and columns, in order: `input` (original, unchanged), `host` (canonical host in output form), `subdomain` (labels left of the registrable domain; "" when none), `domain` (the single registrant label left of the suffix), `suffix` (the public suffix), and `registrable_domain` (eTLD+1). `domain`, `subdomain`, and `registrable_domain` are NA when the host is itself a public suffix. If public-suffix resolution is NA, every derived column except `input` and a successfully normalized host is NA. Zero-length input returns a zero-row frame; all-invalid input keeps one row per input. Root dots are preserved on `host`, `suffix`, and `registrable_domain` only.

Input contract

NA is treated as missing (returns NA), not invalid. Invalid elements include empty or whitespace-only strings, leading or consecutive dots, URL syntax, IPv6 addresses, canonical dotted-decimal IPv4 literals, and labels that fail hostname/IDNA validation. Wrong argument types and non-scalar or unknown option values always abort regardless of invalid.

See Also

[public_suffix\(\)](#), [public_suffix_rule\(\)](#)

Examples

```
suffix_extract("www.example.co.uk")
suffix_extract(c("example.com", "com", NA))
```

Index

`data.frame`, [6](#), [7](#), [9](#), [12](#), [15](#)

`is_public_suffix`, [2](#)

`is_public_suffix()`, [11](#), [13](#)

`psl_cache_prune`, [3](#)

`psl_outdated`, [5](#)

`psl_refresh`, [6](#)

`psl_refresh()`, [4](#), [5](#), [8](#), [9](#)

`psl_rules`, [7](#)

`psl_rules()`, [8](#), [9](#)

`psl_use`, [8](#)

`psl_use()`, [4](#), [6](#), [9](#)

`psl_version`, [9](#)

`psl_version()`, [5–8](#)

`public_suffix`, [10](#)

`public_suffix()`, [2](#), [3](#), [12](#), [13](#), [15](#)

`public_suffix_rule`, [11](#)

`public_suffix_rule()`, [7](#), [11](#), [15](#)

`registrable_domain`, [12](#)

`registrable_domain()`, [11](#)

`suffix_extract`, [14](#)

`suffix_extract()`, [11–13](#)