

Package ‘stow’

August 30, 2026

Title Durable Managed Local Copies of Remote Files

Version 0.3.0

Description Turns remote file URLs into paths to durable managed local copies stored in a fixed subdirectory of platform-appropriate, package-specific user data directories. Matching copies are reused across R sessions and remain available for offline use. Supports optional ETag-based versions, content validation, and staged replacement that prevents failed downloads and files that fail a supplied validator from becoming managed local copies. Includes tools to inspect, conservatively prune, and explicitly remove retained copies.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 8.0.0

Depends R (>= 4.0.0)

Imports curl, digest

Suggests testthat (>= 3.2.0), withr

Config/testthat/edition 3

URL <https://github.com/cole-brokamp/stow>

BugReports <https://github.com/cole-brokamp/stow/issues>

NeedsCompilation no

Author Cole Brokamp [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0002-0289-3151>>)

Maintainer Cole Brokamp <cole@colebrokamp.com>

Repository CRAN

Date/Publication 2026-08-30 10:00:02 UTC

Contents

stow	2
stow_info	4

stow_path	5
stow_prune	7
stow_remove	8

Index	10
--------------	-----------

stow	<i>Download a managed local copy</i>
------	--------------------------------------

Description

stow() turns a remote file URL into an absolute path to a durable managed local copy. It downloads the file when a matching copy is not available and otherwise reuses the existing copy, including across R sessions. The directory is created and located by stow_path().

Usage

```
stow(  
  url,  
  package = "stow",  
  subdir = NULL,  
  overwrite = FALSE,  
  offline = FALSE,  
  quiet = FALSE,  
  etag = TRUE,  
  validate = NULL  
)
```

Arguments

url	A single https, http, ftp, or ftps URL. Its path must end in a filename. Query strings and fragments are not supported.
package	The name of the R package that owns the managed local copy. It must be a syntactically valid R package name: at least two characters, beginning with an ASCII letter, containing only ASCII letters, numbers, and periods, and ending with a letter or number. The default, "stow", is intended for direct use. A package that uses stow should supply its own package name. Copies are stored in a fixed stow subdirectory beneath the package-specific user data directory returned by tools::R_user_dir().
subdir	An optional relative subdirectory within the package's fixed stow directory. Each component must begin with a letter or number, may otherwise contain letters, numbers, dots, underscores, or hyphens, and must not end in a dot. Absolute paths, empty components, . components, and . . components are rejected.
overwrite	Whether to download again and replace the managed local copy that matches the current URL and ETag. Copies for other ETags are retained.
offline	Whether stow() must make no network requests and use only a managed local copy. It cannot be combined with overwrite = TRUE.

<code>quiet</code>	Whether to suppress informational messages and download progress. Warnings and errors are never suppressed.
<code>etag</code>	Whether online calls should ask the server for an ETag and use it to distinguish versions. If an ETag is unavailable or the request fails, <code>stow()</code> continues with a URL-derived filename.
<code>validate</code>	An optional content-validation function called with one candidate file path. It must return exactly <code>TRUE</code> ; an error or any other result marks the file invalid.

Value

The absolute path to the managed local copy as a visible character scalar. `stow()` does not read or interpret the file's contents.

Managed local copy identity and versions

Managed local copy filenames retain the URL basename and include a 64-bit xxHash of the URL directory. This distinguishes URLs that have the same basename. When a server-provided ETag is available, its 64-bit xxHash is inserted before the file extension. A changed ETag therefore creates a new managed local copy path, and earlier ETag variants are retained until explicitly managed with `stow_prune()` or `stow_remove()`.

Online calls reuse an existing matching copy unless `overwrite = TRUE` or the copy fails `validate`. If the ETag request fails or is unsupported, downloading continues with the URL-derived, non-ETag name.

In offline mode, `stow()` makes no network requests. It first looks for the non-ETag copy and otherwise chooses the newest matching ETag variant by modification time, using lexical filename order to break ties.

Validation

When `validate` is supplied, `stow()` calls it before reusing an existing managed local copy and after downloading new content but before committing that content. It is also applied to a managed local copy created by another process before that copy is reused. A validator error, `FALSE`, or any value other than exactly `TRUE` marks the candidate invalid.

New downloads are validated at a temporary path, so validators should inspect file contents rather than rely on the temporary filename or its extension. A validator can, for example, check a file size, parse expected metadata, open a serialized object, or verify a checksum.

An invalid online managed local copy triggers a replacement download. In offline mode, an invalid copy produces an error and is left unchanged. When `validate = NULL`, `stow()` checks that a download produced a regular file but makes no claim about its format, integrity, or meaning.

Durable managed local copy updates

Downloads are written to a temporary file inside the destination directory. A failed or incomplete download is cleaned up and never becomes a managed local copy. Newly downloaded content that fails `validate` is likewise never committed to its destination filename.

When replacing an existing copy, `stow()` keeps the existing file until the replacement has downloaded and validated successfully. If committing the replacement fails, it attempts to restore the

existing file. Thus a failed download or invalid replacement does not overwrite an existing managed local copy. An existing copy that fails validation may remain on disk if its replacement fails, but it is not returned as valid.

See Also

[stow_path\(\)](#) to locate managed local copies, [stow_info\(\)](#) to list them, and [stow_prune\(\)](#) to manage retained copies.

Examples

```
url <- "https://github.com/geomarker-io/addr/releases/download/v1.3.0/addr-taf-v1-2025.json"

tryCatch(
  withr::with_envvar(c(R_USER_DATA_DIR = tempfile("stow-data-")), {
    path <- stow(url)
    readLines(path, n = 3)

    is_addr_manifest <- function(path) {
      text <- paste(readLines(path, warn = FALSE), collapse = "\n")
      grepl('"artifact_type": "addr-taf-fuel"', text, fixed = TRUE)
    }
    stow(url, validate = is_addr_manifest)
  }),
  error = function(error) {
    message("Skipping remote example: ", conditionMessage(error))
  }
)
```

stow_info

List managed local copies

Description

`stow_info()` lists regular files recursively below [stow_path\(\)](#). It reports storage metadata only: it does not read the files, apply a validator, or make network requests.

Usage

```
stow_info(package = "stow", subdir = NULL)
```

Arguments

package	The name of the R package that owns the managed local copy. It must be a syntactically valid R package name: at least two characters, beginning with an ASCII letter, containing only ASCII letters, numbers, and periods, and ending with a letter or number. The default, "stow", is intended for direct use. A package that uses stow should supply its own package name. Copies are stored
---------	--

in a fixed `stow` subdirectory beneath the package-specific user data directory returned by `tools::R_user_dir()`.

`subdir` An optional relative subdirectory within the package's fixed `stow` directory. Each component must begin with a letter or number, may otherwise contain letters, numbers, dots, underscores, or hyphens, and must not end in a dot. Absolute paths, empty components, `.` components, and `..` components are rejected.

Value

A base data frame with the absolute file path, size in bytes, and last-modified time in `modified`. An empty managed local copy directory returns a zero-row data frame with the same columns.

See Also

`stow()` to download a managed local copy, `stow_path()` to locate the directory, and `stow_prune()` to manage retained copies.

Examples

```
withr::with_envvar(
  c(R_USER_DATA_DIR = tempfile("stow-data-")),
  stow_info()
)
```

`stow_path`

Locate or create the managed local copy directory

Description

`stow_path()` creates and returns the directory where managed local copies downloaded by `stow()` are saved. The directory is a fixed `stow` subdirectory of the platform-appropriate, package-specific user data location returned by `tools::R_user_dir()` with `which = "data"`, so it remains available across R sessions and is not treated as disposable.

Usage

```
stow_path(package = "stow", subdir = NULL)
```

Arguments

`package` The name of the R package that owns the managed local copy. It must be a syntactically valid R package name: at least two characters, beginning with an ASCII letter, containing only ASCII letters, numbers, and periods, and ending with a letter or number. The default, `"stow"`, is intended for direct use. A package that uses `stow` should supply its own package name. Copies are stored in a fixed `stow` subdirectory beneath the package-specific user data directory returned by `tools::R_user_dir()`.

subdir An optional relative subdirectory within the package's fixed stow directory. Each component must begin with a letter or number, may otherwise contain letters, numbers, dots, underscores, or hyphens, and must not end in a dot. Absolute paths, empty components, . components, and . . components are rejected.

Value

The absolute directory path for managed local copies as a character scalar. The directory is created if it does not already exist.

Environment variables

R_USER_DATA_DIR and XDG_DATA_HOME are environment variables, not R options. `tools::R_user_dir()` first uses R_USER_DATA_DIR; if it is unset, it uses XDG_DATA_HOME when available, followed by the platform-specific default. `stow_path()` creates the package-specific data directory, a fixed stow directory beneath it, and any requested subdir.

Environment variables can be set before R starts, either through the operating system environment or with a line in a user or project .Renviron file:

```
R_USER_DATA_DIR=/path/to/data
```

They can also be set after R has started with `Sys.setenv()`:

```
Sys.setenv(R_USER_DATA_DIR = "/path/to/data")
stow_path()
```

A value set with `Sys.setenv()` affects subsequent calls in the current R process; it does not move files that were already downloaded. Use `withr::with_envvar()` for a temporary, scoped change (see examples).

See Also

`stow()` to download a managed local copy, `stow_info()` to list managed local copies, and `stow_prune()` to manage retained copies.

Examples

```
withr::with_envvar(
  c(R_USER_DATA_DIR = tempfile("stow-data-")),
  stow_path(subdir = "inputs/raw")
)
```

stow_prune

*Prune outdated and orphaned managed local copies***Description**

`stow_prune()` actively manages files created by `stow()` without deleting the current offline fallback for a URL. It makes no network requests and never removes directories or unrecognized files.

Usage

```
stow_prune(
  url = NULL,
  package = "stow",
  subdir = NULL,
  max_age = 30,
  keep = NULL,
  dry_run = FALSE,
  quiet = FALSE
)
```

Arguments

<code>url</code>	NULL or one URL accepted by <code>stow()</code> . When supplied, outdated ETag variants for that URL are eligible for removal. With <code>url = NULL</code> , only orphaned stow temporary and backup files are considered.
<code>package</code>	The name of the R package that owns the managed local copy. It must be a syntactically valid R package name: at least two characters, beginning with an ASCII letter, containing only ASCII letters, numbers, and periods, and ending with a letter or number. The default, "stow", is intended for direct use. A package that uses stow should supply its own package name. Copies are stored in a fixed stow subdirectory beneath the package-specific user data directory returned by <code>tools::R_user_dir()</code> .
<code>subdir</code>	An optional relative subdirectory within the package's fixed stow directory. Each component must begin with a letter or number, may otherwise contain letters, numbers, dots, underscores, or hyphens, and must not end in a dot. Absolute paths, empty components, . components, and . . components are rejected.
<code>max_age</code>	A non-negative number of days. A file must be at least this old to be eligible for removal. The default retains eligible files for 30 days. Use 0 to remove eligible files regardless of age.
<code>keep</code>	NULL or managed local copy paths for <code>url</code> that must be retained in addition to the protections applied automatically. A path returned by a recent <code>stow(url, ...)</code> call can be supplied when the server has returned to an older ETag version. Each path must be an existing managed local copy for <code>url</code> in the selected package and subdirectory.
<code>dry_run</code>	Whether to report eligible files without removing them.
<code>quiet</code>	Whether to suppress informational cleanup messages. Warnings and errors are never suppressed.

Value

A base data frame describing files eligible for removal, with absolute path, type, last-modified time in modified, and logical removed. removed is NA during a dry run, TRUE after successful removal, and FALSE if removal failed. If no files are eligible, a zero-row data frame with the same columns is returned.

Retention and removal rules

For a supplied url, the URL-derived non-ETag copy is always retained. The newest ETag variant by modification time is also always retained, using lexical filename order to break ties. This is the same ETag fallback that `stow()` would select in offline mode when no non-ETag copy exists. Paths supplied through keep receive an additional explicit protection. Other ETag variants are removed only after they reach max_age.

Temporary files whose names begin with .stow-download- are incomplete staged downloads and become eligible after max_age. Replacement backups become eligible only when the corresponding destination exists. A backup whose destination is missing is retained because it may be the only recoverable copy. Orphan cleanup is recursive within the selected package's managed local copy directory or subdir.

Use `stow_remove()` when the intent is to remove every managed local copy for a known URL, including its current usable copy.

See Also

`stow_remove()` to remove all managed local copies for a URL, `stow_info()` to inspect managed local copies, and `stow_path()` to locate them.

Examples

```
withr::with_envvar(c(R_USER_DATA_DIR = tempfile("stow-data-")), {
  stow_prune(dry_run = TRUE)
})
```

stow_remove

Remove every managed local copy for a URL

Description

`stow_remove()` explicitly removes all regular managed local copies belonging to one URL in a selected package and subdirectory. This includes the non-ETag copy and every ETag variant, so a later `stow()` call must download the file again. It makes no network requests and does not remove directories, internal temporary or backup files, copies for other URLs, or unrecognized files.

Usage

```
stow_remove(
  url,
  package = "stow",
  subdir = NULL,
  dry_run = FALSE,
  quiet = FALSE
)
```

Arguments

<code>url</code>	A single https, http, ftp, or ftps URL. Its path must end in a filename. Query strings and fragments are not supported.
<code>package</code>	The name of the R package that owns the managed local copy. It must be a syntactically valid R package name: at least two characters, beginning with an ASCII letter, containing only ASCII letters, numbers, and periods, and ending with a letter or number. The default, "stow", is intended for direct use. A package that uses stow should supply its own package name. Copies are stored in a fixed stow subdirectory beneath the package-specific user data directory returned by <code>tools::R_user_dir()</code> .
<code>subdir</code>	An optional relative subdirectory within the package's fixed stow directory. Each component must begin with a letter or number, may otherwise contain letters, numbers, dots, underscores, or hyphens, and must not end in a dot. Absolute paths, empty components, . components, and . . components are rejected.
<code>dry_run</code>	Whether to report matching managed local copies without removing them.
<code>quiet</code>	Whether to suppress informational removal messages. Warnings and errors are never suppressed.

Value

A base data frame describing matching managed local copies, with absolute path, type, last-modified time in `modified`, and logical removed. `removed` is NA during a dry run, TRUE after successful removal, and FALSE if removal failed. If no copies match, a zero-row data frame with the same columns is returned.

See Also

`stow_prune()` for conservative cleanup that retains current usable copies, and `stow_info()` to inspect managed local copies.

Examples

```
withr::with_envvar(c(R_USER_DATA_DIR = tempfile("stow-data-")), {
  stow_remove(
    "https://example.com/files/data.csv",
    dry_run = TRUE
  )
})
```

Index

stow, [2](#)
stow(), [5–8](#)
stow_info, [4](#)
stow_info(), [4, 6, 8, 9](#)
stow_path, [5](#)
stow_path(), [2, 4, 5, 8](#)
stow_prune, [7](#)
stow_prune(), [3–6, 9](#)
stow_remove, [8](#)
stow_remove(), [3, 8](#)
Sys.setenv(), [6](#)

tools::R_user_dir(), [2, 5–7, 9](#)