

Package: rwasml (via r-universe)

August 5, 2026

Title Build R Packages for WebAssembly

Version 0.3.0

Description Utility functions to help build R packages compiled for WebAssembly (Wasm), manage Wasm binary R package libraries and repositories, and prepare webR compatible filesystem images for static web hosting of data files and R packages.

License MIT + file LICENSE

URL <https://r-wasm.github.io/rwasm/>, <https://github.com/r-wasm/rwasm>

BugReports <https://github.com/r-wasm/rwasm/issues>

Depends R (>= 4.0)

Imports dplyr, fs, jsonlite, lifecycle, pak, pkgdepends, rlang (>= 1.1.0), utils, withr, zip

Suggests knitr, rmarkdown

VignetteBuilder knitr

Config/Needs/website tidyverse/tidytemplate

Config/usethis/last-upkeep 2025-04-25

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

SystemRequirements webR development installation
(<https://webr.r-wasm.org/>)

Config/pak/sysreqs cmake make libuv1-dev libssl-dev

Repository <https://r-multiverse.r-universe.dev>

Date/Publication 2025-08-13 09:30:37 UTC

RemoteUrl <https://github.com/r-wasm/rwasm>

RemoteRef v0.3.0

RemoteSha 005588296e03080357c7140c1eaf8b47ccabe0bc

Contents

add_list	2
add_pkg	3
add_repo	4
add_tar_index	4
build	5
file_packager	6
make_library	7
make_vfs_library	7
make_vfs_repo	8
rm_pkg	9
write_packages	9

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

add_list	<i>Add one or more packages from a file</i>
----------	---

Description

Downloads and builds the list of [R package references](#) in the file `list_file`, compiling each package for use with WebAssembly and webR. The resulting WebAssembly binary packages are added to the repository directory `repo_dir`. The repository directory will be created if it does not already exist.

Usage

```
add_list(list_file, ...)
```

Arguments

- `list_file` Path to a file containing a list of R package references.
- `...` Arguments passed on to [add_pkg](#)
- `repo_dir` The package repository directory. Defaults to `"./repo"`.
- `remotes` A character vector of package references to prefer as a remote source. Defaults to `NA`, meaning prefer a built-in list of references to packages pre-modified for use with webR.
- `dependencies` Dependency specification for packages to additionally add to the repository. Defaults to `FALSE`, meaning no additional packages. Use `NA` to install only hard dependencies whereas `TRUE` installs all optional dependencies as well. See [pkgdepends::as_pkg_dependencies](#) for details.
- `compress` When `TRUE`, add and compress Emscripten virtual filesystem meta-data in the resulting R package binary `.tgz` files. Otherwise, [file_packager\(\)](#) is used to create uncompressed virtual filesystem images included in the output binary package repository. Defaults to `TRUE`.

Details

The R package references should be listed in the file `list_file`, one line per package reference.

<code>add_pkg</code>	<i>Add R package reference(s) to a package repository</i>
----------------------	---

Description

Downloads and builds the [R package references](#) given by packages, compiling each package for use with WebAssembly and webR. The resulting WebAssembly binary packages are added to the repository directory `repo_dir`. The repository directory will be created if it does not already exist.

Usage

```
add_pkg(  
  packages,  
  repo_dir = "./repo",  
  remotes = NA,  
  dependencies = FALSE,  
  compress = TRUE  
)
```

Arguments

<code>packages</code>	A character vector of one or more package references.
<code>repo_dir</code>	The package repository directory. Defaults to <code>"./repo"</code> .
<code>remotes</code>	A character vector of package references to prefer as a remote source. Defaults to NA, meaning prefer a built-in list of references to packages pre-modified for use with webR.
<code>dependencies</code>	Dependency specification for packages to additionally add to the repository. Defaults to FALSE, meaning no additional packages. Use NA to install only hard dependencies whereas TRUE installs all optional dependencies as well. See pkgdepends::as_pkg_dependencies for details.
<code>compress</code>	When TRUE, add and compress Emscripten virtual filesystem metadata in the resulting R package binary <code>.tgz</code> files. Otherwise, file_packager() is used to create uncompressed virtual filesystem images included in the output binary package repository. Defaults to TRUE.

add_repo	<i>Add all packages from a CRAN-like repository to a package repository</i>
----------	---

Description

Downloads and builds all available source R packages from the R package repositories given by `repos`, compiling each package for use with WebAssembly and webR. The resulting WebAssembly binary packages are added to the repository directory `repo_dir`. The repository directory will be created if it does not already exist.

Usage

```
add_repo(repos = ppm_config$cran_mirror, skip = FALSE, ...)
```

Arguments

<code>repos</code>	A character vector containing the base URL(s) of CRAN-like R package repositories. Defaults to the Posit Package Manager CRAN mirror.
<code>skip</code>	A character string containing a regular expression matching names of packages to skip. Defaults to <code>FALSE</code> , meaning keep all packages.
<code>...</code>	Arguments passed on to add_pkg
<code>repo_dir</code>	The package repository directory. Defaults to <code>"./repo"</code> .
<code>remotes</code>	A character vector of package references to prefer as a remote source. Defaults to <code>NA</code> , meaning prefer a built-in list of references to packages pre-modified for use with webR.
<code>dependencies</code>	Dependency specification for packages to additionally add to the repository. Defaults to <code>FALSE</code> , meaning no additional packages. Use <code>NA</code> to install only hard dependencies whereas <code>TRUE</code> installs all optional dependencies as well. See pkgdepends::as_pkg_dependencies for details.
<code>compress</code>	When <code>TRUE</code> , add and compress Emscripten virtual filesystem metadata in the resulting R package binary <code>.tgz</code> files. Otherwise, file_packager() is used to create uncompressed virtual filesystem images included in the output binary package repository. Defaults to <code>TRUE</code> .

add_tar_index	<i>Add Emscripten virtual filesystem metadata to a given tar archive</i>
---------------	--

Description

Calculates file offsets and other metadata for content stored in an (optionally gzip compressed) tar archive. Once added, the tar archive with metadata can be mounted as an Emscripten filesystem image, making the contents of the archive available to the WebAssembly R process.

Usage

```
add_tar_index(file, strip = 0)
```

Arguments

file	Filename of the tar archive for which metadata is to be added.
strip	Remove the specified number of leading path elements when mounting with webR. Defaults to 0.

Details

The virtual filesystem metadata is appended to the end of the tar archive, with the output replacing the original file. The resulting archive should be hosted online so that its URL can be provided to webR for mounting on the virtual filesystem.

If strip is greater than 0 the virtual filesystem metadata is generated such that when mounted by webR the specified number of leading path elements are removed. Useful for R package binaries where data files are stored in the original .tgz file under a subdirectory. Files with fewer path name elements than the specified amount are skipped.

build	<i>Build one or more R packages for WebAssembly</i>
-------	---

Description

Downloads and builds the [R package references](#) given by packages, compiling each package for use with WebAssembly and webR. The resulting WebAssembly binary packages are written to out_dir.

Usage

```
build(
  packages,
  out_dir = ".",
  remotes = NULL,
  dependencies = FALSE,
  compress = TRUE
)
```

Arguments

packages	A character vector of one or more package references.
out_dir	The output directory. Defaults to ".".
remotes	A character vector of package references to prefer as a remote source. If NA, use a built-in list of references to packages pre-modified for use with webR. Defaults to NULL, meaning no preference over the usual remote sources.

dependencies	Dependency specification for packages to additionally add to the repository. Defaults to FALSE, meaning no additional packages. Use NA to install only hard dependencies whereas TRUE installs all optional dependencies as well. See pkgdepends::as_pkg_dependencies for details.
compress	When TRUE, add and compress Emscripten virtual filesystem metadata in the resulting R package binary .tgz files. Otherwise, file_packager() is used to create uncompressed virtual filesystem images included in the output binary package repository. Defaults to TRUE.

file_packager	<i>Create an Emscripten filesystem image</i>
---------------	--

Description

Uses Emscripten's `file_packager` tool to build an Emscripten filesystem image that can be mounted by webR. The filesystem image may contain arbitrary data that will be made available for use by the WebAssembly R process once mounted.

Usage

```
file_packager(in_dir, out_dir = "./vfs", out_name = NULL, compress = FALSE)
```

Arguments

in_dir	Directory to be packaged into the filesystem image.
out_dir	Directory in which to write the output image files. Defaults to <code>"./vfs"</code> .
out_name	A character string for the output image base filename. If NULL, defaults to the final component of the input directory path.
compress	Logical. If TRUE, a compressed version of the filesystem data is included in the output. Defaults to FALSE.

Details

Outputs at least two files (named by `out_name`) in the `out_dir` directory: a data file with extension `".data"`, and a metadata file with extension `".js.metadata"`. Both files should be hosted online so that their URL can be provided to webR for mounting on the Emscripten virtual filesystem.

When `compress` is TRUE, an additional file with extension `".data.gz"` is also output containing a compressed version of the filesystem data. The metadata file is also changed to reflect the availability of a compressed version of the data.

make_library	Create an R package library
--------------	-----------------------------

Description

Extracts all R packages contained in the repository directory `repo_dir` and writes them to a library directory `lib_dir`.

Usage

```
make_library(repo_dir = "./repo", lib_dir = "./lib", strip = NULL)
```

Arguments

<code>repo_dir</code>	The package repository directory. Defaults to <code>"./repo"</code> .
<code>lib_dir</code>	Package library output directory. Defaults to <code>"./lib"</code> .
<code>strip</code>	A character vector of directories to strip from each R package.

Details

The `lib_dir` directory will be created if it does not already exist.

The `strip` argument may be used to strip certain directories from packages installed to the library directory `lib_dir`. This can be used to reduce the total library file size by removing directories that are not strictly necessary for the R package to run, such as directories containing documentation and vignettes.

make_vfs_library	Create an Emscripten filesystem image of an R package library
------------------	---

Description

Extracts all binary R packages contained in the repository directory `repo_dir` and creates an Emscripten filesystem image containing the resulting package library.

Usage

```
make_vfs_library(  
  out_dir = "./vfs",  
  out_name = "library.data",  
  repo_dir = "./repo",  
  compress = FALSE,  
  ...  
)
```

Arguments

out_dir	Directory in which to write the output image files. Defaults to <code>"./vfs"</code> .
out_name	A character string for the output library image filename.
repo_dir	The package repository directory. Defaults to <code>"./repo"</code> .
compress	When TRUE, add and compress Emscripten virtual filesystem metadata in the resulting R package binary <code>.tgz</code> files. Otherwise, <code>file_packager()</code> is used to create uncompressed virtual filesystem images included in the output binary package repository. Defaults to TRUE.
...	Arguments passed on to <code>make_library</code>
strip	A character vector of directories to strip from each R package.

Details

A single filesystem image is generated using Emscripten's `file_packager()` tool and the output `.data` and `.js.metadata` filesystem image files are written to the directory `out_dir`.

When `compress` is TRUE, an additional file with extension `".data.gz"` is also output containing a compressed version of the filesystem data.

The resulting image can be downloaded by webR and mounted on the Emscripten virtual filesystem as an efficient way to provide a pre-configured R library, without installing each R package individually.

make_vfs_repo

Add Emscripten filesystem images to an R package repository

Description

Creates an Emscripten filesystem image for each R package that exists in the package repository directory `repo_dir`.

Usage

```
make_vfs_repo(repo_dir = "./repo", compress = FALSE)
```

Arguments

repo_dir	The package repository directory. Defaults to <code>"./repo"</code> .
compress	When TRUE, add and compress Emscripten virtual filesystem metadata in the resulting R package binary <code>.tgz</code> files. Otherwise, <code>file_packager()</code> is used to create uncompressed virtual filesystem images included in the output binary package repository. Defaults to TRUE.

Details

Each filesystem image is generated using Emscripten's `file_packager()` tool and the output `.data` and `.js.metadata` filesystem image files are written to the repository in the same directory as the package binary `.tgz` files.

The resulting filesystem images may then be used by webR to download and install R packages by mounting the `.data` images to the Emscripten virtual filesystem.

When `compress` is `TRUE`, an additional file with extension `".data.gz"` is also output containing a compressed version of the filesystem data.

rm_pkg	<i>Remove R package(s) from a package repository</i>
--------	--

Description

Remove R package(s) from a package repository

Usage

```
rm_pkg(packages, repo_dir = "./repo")
```

Arguments

packages	A character vector of one or more package names.
repo_dir	The package repository directory. Defaults to <code>"./repo"</code> .

write_packages	<i>Write the PACKAGES file for a package repository</i>
----------------	---

Description

Write the PACKAGES file for a package repository

Usage

```
write_packages(repo_dir = "./repo")
```

Arguments

repo_dir	The package repository directory. Defaults to <code>"./repo"</code> .
----------	---

Index

`add_list`, [2](#)
`add_pkg`, [2](#), [3](#), [4](#)
`add_repo`, [4](#)
`add_tar_index`, [4](#)

`build`, [5](#)

`file_packager`, [6](#)
`file_packager()`, [2–4](#), [6](#), [8](#), [9](#)

`make_library`, [7](#), [8](#)
`make_vfs_library`, [7](#)
`make_vfs_repo`, [8](#)

`pkgdepends::as_pkg_dependencies`, [2–4](#), [6](#)

R package references, [2](#), [3](#), [5](#)
`rm_pkg`, [9](#)

`write_packages`, [9](#)