Package: rwasm (via r-universe)

April 16, 2025

Title Build R Packages for WebAssembly

Version 0.2.0

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`

Imports dplyr, fs, jsonlite, pak, pkgdepends, rlang, utils, withr, zip

Suggests knitr, rmarkdown

VignetteBuilder knitr

Config/Needs/website tidyverse/tidytemplate

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

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

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

Repository https://r-multiverse-staging.r-universe.dev

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

RemoteRef v0.2.0

RemoteSha a7079cbfd10428e18486578d3c309c9fd69f6d97

Contents

`add_list` *Add one or more packages from a file*

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 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`.

Details

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

add_pkg *Add R package reference(s) to a package repository*

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

`packages`	A character vector of one or more package references.
`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 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`.

add_repo *Add all packages from a CRAN-like repository to a package repository*

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

`repos` A character vector containing the base URL(s) of CRAN-like R package repositories. Defaults to the Posit Package Manager CRAN mirror.

`skip` A character string containing a regular expression matching names of packages to skip. Defaults to `FALSE`, meaning keep all packages.

`...` 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 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`.

`add_tar_index` *Add Emscripten virtual filesystem metadata to a given* `tar` *archive*

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` *Build one or more R packages for WebAssembly*

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 *Create an Emscripten filesystem image*

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 `"./vfs"`.
`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

repo_dir	The package repository directory. Defaults to "./repo".
lib_dir	Package library output directory. Defaults to "./lib".
strip	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 "./vfs".
out_name	A character string for the output library image filename.
repo_dir	The package repository directory. Defaults to "./repo".
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.
...	Arguments passed on to make_library 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 `"./repo"`.

`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`.

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` *Remove R package(s) from a package repository*

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 `"./repo"`.

`write_packages` *Write the `PACKAGES` file for a package repository*

Description

Write the `PACKAGES` file for a package repository

Usage

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

Arguments

`repo_dir`	The package repository directory. Defaults to `"./repo"`.

Index