



Extending your .Rprofile with {envsetup} & {renv}

PHUSE US Connect 2023

Nick Masel (Janssen R&D), Mike Stackhouse (Atorus)

Mimi Jordan, *Hope*

Mimi's painting process was a form of meditation, training her mind to focus and redirect her thoughts. She passed away in 2020.

The opinions in this presentation are my own and do not necessarily reflect the views and policies of J&J



What is {renv}?

The renv package helps you create reproducible **environments** for your R projects. Use renv to make your R projects more:

- **Isolated:** Installing a new or updated package for one project won't break your other projects, and vice versa. That's because renv gives each project its own private package library.
- **Portable:** Easily transport your projects from one computer to another, even across different platforms. renv makes it easy to install the packages your project depends on.
- **Reproducible:** renv records the exact package versions you depend on, and ensures those exact versions are the ones that get installed wherever you go.



What is {envsetup}?

The purpose of this package is to support the setup of the R environment. The two main features are:

- **Autos:** to automatically source files and/or directories into your environment
- **Paths:** to consistently set path objects across projects for I/O

Both are implemented using a configuration file to allow easy, custom configurations that can be used for multiple or all projects.



Autos and Paths

What are *autos*?

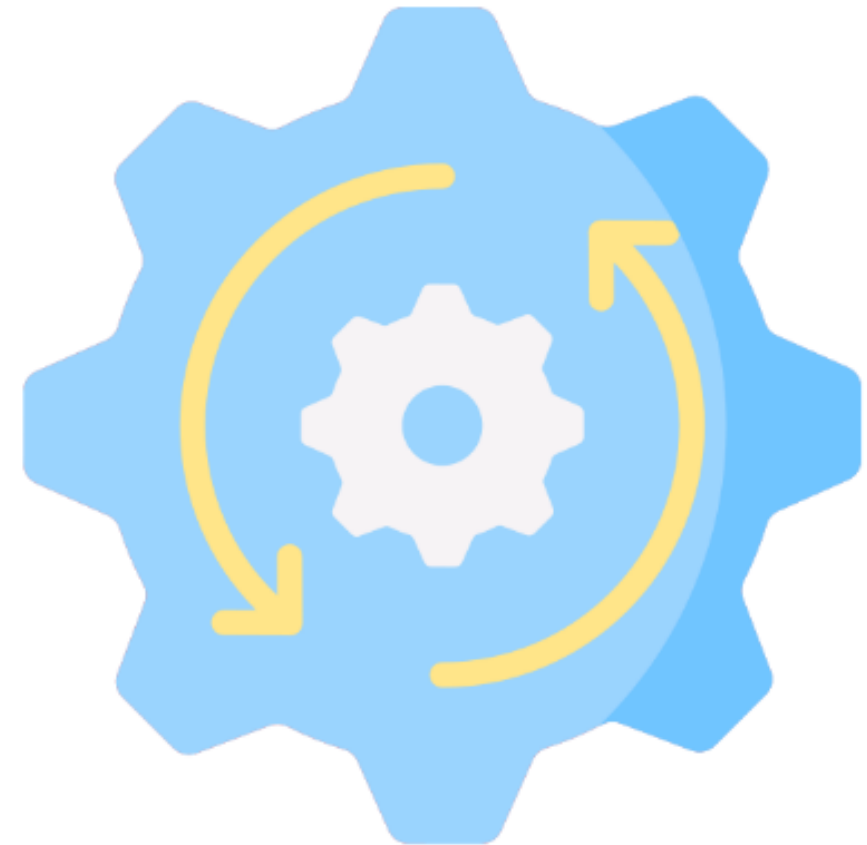
- Replicates concept of sasautos
- Files or directories that should be sourced on session starts
- Functions inserted into separate namespaces
 - Behind global
 - In front of loaded packages

What are *paths*?

- Replicates some aspects of SAS® libnames
- Abstracts read and write paths to avoid hardcoding
- Enable hierarchy search paths for files
- Messages which directory files were found in or written

What is a .Rprofile?

- At Start-up, R will search for a profile and if found, it will source this into your workspace
- Just an R script that gets executed, so you can put any R code you in your .Rprofile
- Perfect!
 - Put things here that everyone needs all the time

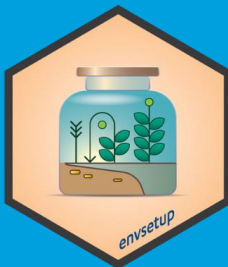


Bring them together

.Rprofile



Activate package
management



Configure path and source
non-package functions

How can a user implement this on a project?

Viral exacerbation at 40x magnification

Project Setup

For Individual Users

Use a Package Manager & Repository Snapshots

- MRAN
 - <https://mran.microsoft.com>
 - Will be decommissioned
- Posit Package Manager
 - <https://packagemanager.rstudio.com>
- `options(repos = c("insert snapshot path"))`

`renv::init()`

- Activates so packages are managed for the project
- Take snapshots of your packages to record in `renv.lock`
- Use `restore` and `snapshot` to manage package landscape for your project

`envsetup::init()`

Three scenarios for initializing

1. Default `envsetup` config used and directories are built w/in a project to match
2. User defined `envsetup` config used and directories are built to match
3. User defined `envsetup` config used to match an existing standard directory tree

How can an organization implement this for all of their projects?

Viral exacerbation at 40x magnification

Project Setup

For Organizations

Release cycles

Disclaimer

If you aren't managing releases, some of this isn't relevant.

Why? ...because there is no pre-specified set of packages for users

Generic Setup Files

- **One** envsetup configuration can work for all projects!
- **One** .Rprofile
 - or you could have one per release
- **One** renv.lock **per release**
 - This will restore the snapshot repos as well as the packages

System Setup

- Store files per release w/access restrictions
- Link projects to releases
- Add mechanism to setup projects for teams / users
 - For example
 - RStudio Addin
 - Custom tooling



Wrap-up

{renv} allows you to

- Have a consistent package set for your projects
- Reconstruct that environment as needed

{envsetup} allows you to

- Abstract directory paths to avoid hard coded directories
- Source project level functions
- Build a repeatable study process to use as a default configuration

The .Rprofile file allows you to

- Automatically carry out these actions at sessions start
- Build a process where programmers can open the project and get coding



PHARMACEUTICAL COMPANIES OF
Johnson & Johnson