

# Package: rspm (via r-universe)

October 30, 2024

**Type** Package

**Title** 'RStudio' Package Manager

**Version** 0.6.1

**Description** Enables binary package installations on Linux distributions. Provides access to 'RStudio' public repositories at <https://packagemanager.posit.co>, and transparent management of system requirements without administrative privileges. Currently supported distributions are 'CentOS' / 'RHEL' 7-9, and several 'RHEL' derivatives ('Rocky Linux', 'AlmaLinux', 'Oracle Linux', and 'Amazon Linux'), 'openSUSE' / 'SLES' 15.4-5, 'Debian' 11-12, and 'Ubuntu' LTS 20-22.

**License** GPL-2 | GPL-3 | MIT + file LICENSE

**Copyright** file COPYRIGHTS

**Encoding** UTF-8

**OS\_type** unix

**Suggests** renv, tinytest

**URL** <https://cran4linux.github.io/rspm/>

**BugReports** <https://github.com/cran4linux/rspm/issues>

**RoxygenNote** 7.3.1

**NeedsCompilation** no

**Author** Iñaki Ucar [aut, cph, cre]  
(<https://orcid.org/0000-0001-6403-5550>), R Core Team [aut, cph]

**Maintainer** Iñaki Ucar <iucar@fedoraproject.org>

**Repository** CRAN

**Date/Publication** 2024-10-29 11:10:02 UTC

## Contents

rspm-package . . . . .	2
integration . . . . .	2
manager . . . . .	3
renv_init . . . . .	4
<b>Index</b>	<b>6</b>

---

rspm-package	<b>rspm: RStudio Package Manager</b>
--------------	--------------------------------------

---

### Description

Enables binary package installations on Linux distributions. Provides access to RStudio public repositories at <https://packagemanager.posit.co>, and transparent management of system requirements without administrative privileges. Currently supported distributions are CentOS / RHEL 7-9, and several RHEL derivatives (Rocky Linux, AlmaLinux, Oracle Linux, Amazon Linux), openSUSE / SLES 15.4-5, Debian 11-12, and Ubuntu LTS 20-22.

### Author(s)

Iñaki Ucar

### References

<https://cran4linux.github.io/rspm/>

### See Also

[manager](#), [integration](#)

---

integration	<i>Enable/Disable RStudio Package Manager</i>
-------------	---

---

### Description

Functions to enable or disable RSPM repos as well as the integration of `install_sysreqs` into `install.packages` and `update.packages`. When enabled, binary packages are installed from RSPM if available, and system requirements are transparently resolved and installed without root privileges.

### Usage

`enable()`

`disable()`

### Details

To enable **rspm** permanently, include the following into your `.Rprofile`:

```
suppressMessages(rspm::enable())
```

### Value

No return value, called for side effects.

### See Also

[renv\\_init](#) for **renv** projects.

### Examples

```
## Not run:  
# install 'units' and all its dependencies from the system repos  
rspm::enable()  
install.packages("units")  
  
# install packages again from CRAN  
rspm::disable()  
install.packages("errors")  
  
## End(Not run)
```

---

manager

*Manage System Requirements*

---

### Description

Detect, install and configure system requirements. This function is automatically called when the package is enabled via [enable](#). It can also be called manually at any time to update the system requirements.

### Usage

```
install_sysreqs()
```

```
missing_sysreqs()
```

### Value

`install_sysreqs`: No return value, called for side effects.

`missing_sysreqs`: A list of missing libraries, for debugging.

## Examples

```
## Not run:
# install 'units' without enabling the integration
install.packages("units")
# then trigger the installation of system requirements manually
rspm::install_sysreqs()

## End(Not run)
```

---

renv\_init

*Initialize an **renv** Project*

---

## Description

Substitutes `renv::init()` to initialize a new **renv** project with **rspm** enabled. This function sets the appropriate infrastructure to activate the integration. Then, packages can be installed normally via `install.packages` and `update.packages`.

## Usage

```
renv_init(...)
```

## Arguments

... Arguments to be passed to `renv::init()`.

## Details

Note that, if `renv::install` or `renv::update` are called directly, then `rspm::install_sysreqs()` needs to be called manually.

## Value

The project directory, invisibly. This function is called for its side effects.

## Examples

```
## Not run:
# initialize a new project (with an empty R library)
rspm::renv_init()

# install 'units' and all its dependencies from the system repos
install.packages("units")

# install a specific version and install dependencies manually
renv::install("units@0.8-0")
rspm::install_sysreqs()
```

*renv\_init*

5

## End(Not run)

# Index

`disable (integration)`, 2

`enable`, 3

`enable (integration)`, 2

`install_sysreqs`, 2

`install_sysreqs (manager)`, 3

`integration`, 2, 2

`manager`, 2, 3

`missing_sysreqs (manager)`, 3

`renv_init`, 3, 4

`rspm (rspm-package)`, 2

`rspm-package`, 2