Package 'bspm'

November 6, 2026

Type Package

Title Bridge to System Package Manager

Version 0.6.0

Description Enables binary package installations on Linux distributions. Provides functions to manage packages via the distribution's package manager. Also provides transparent integration with R's install.packages() and a fallback mechanism. When installed as a system package, interacts with the system's package manager without requiring administrative privileges via an integrated D-Bus service; otherwise, uses sudo. Currently, the following backends are supported: DNF, APT, ALPM.

License MIT + file LICENSE

Encoding UTF-8

OS_type unix

SystemRequirements systemd, dbus-python, PyGObject, python-(libdnf5|dnf|apt|alpm)

Suggests tinytest

URL https://cran4linux.github.io/bspm/

BugReports https://github.com/cran4linux/bspm/issues

Config/roxygen2/version 8.1.1

NeedsCompilation no

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

Maintainer Iñaki Ucar <iucar@fedoraproject.org>

Repository CRAN

Date/Publication 2026-11-06 20:40:16 UTC

Contents

bspm-package 2
bspm-options 2

bspm-scripts 3
integration 3
manager 4
shadowed_packages 6
sysreqs 7

Index **8**

---

bspm-package **bspm**: *Bridge to System Package Manager*

Description

Enables binary package installations on Linux distributions. Provides functions to manage packages via the distribution's package manager. Also provides transparent integration with R's `install.packages` and a fallback mechanism. When installed as a system package, interacts with the system's package manager without requiring administrative privileges via an integrated D-Bus service; otherwise, uses sudo. Currently, the following backends are supported: DNF, APT, ALPM.

Author(s)

Iñaki Ucar

References

https://cran4linux.github.io/bspm/

See Also

`manager`, `integration`, `bspm-scripts`, `bspm-options`

---

bspm-options *Package Options*

Description

List of `options()` supported to configure **bspm**'s behavior. In general, these should be set *before* calling any package function.

Options specific to bspm

`bspm.backend.check`**:** logical, default `TRUE`. If false, the initial check on `enable()` is not performed.

`bspm.version.check`**:** logical, default `TRUE`. If false, as many binaries are installed as possible without any version check, and then installation from source is used as a fallback.

`bspm.sudo.autodetect`**:** logical, default `FALSE`. If true, enables autodetection and selection of password-less sudo.

`bspm.sudo`**:** logical, default `FALSE`. If true, forces `sudo` unconditionally as the preferred mechanism.

Options from base R

These are used in the same way as in base R. See `options` for a detailed description.

- `askYesNo`
- `install.packages.compile.from.source`

`bspm-scripts` *Call Internal Scripts*

Description

Internal scripts can be called via `Rscript` (see examples).

Examples

```
## Not run:
# get a list of available scripts with descriptions
Rscript -e bspm:::scripts

# see a script's help
Rscript -e bspm:::scripts <script_name> -h

# run a script
Rscript -e bspm:::scripts <script_name> [args]

## End(Not run)
```

`integration` *Enable/Disable Bridge to System Package Manager*

Description

Functions to enable or disable the integration of `install_sys` into `install.packages`. When enabled, packages are installed transparently from system repositories if available, including dependencies, and from the configured R repositories if not.

Usage

```
enable()

disable()
```

Details

To enable **bspm** system-wide by default, include the following:

```
suppressMessages(bspm::enable())
```

into the `Rprofile.site` file. To enable it just for a particular user, move that line to the user's `~/.Rprofile` instead.

By default, enabling **bspm** triggers a check of the backend, and a warning is raised if the system service is required but not available. To avoid this check, `options(bspm.backend.check=FALSE)` can be set.

Enabling **bspm** sets default installation `type` to `"both"`, which means 'use binary if available and current, otherwise try source'. The action if there are source packages which are preferred is controlled by `getOption("install.packages.compile.from.source")`. Set this option to `"never"` to always prefer binaries over source packages, with an informative message about newer versions available from source.

If binaries are always preferred and no message is required, a special *fast* mode can be enabled via `options(bspm.version.check=FALSE)`, (true by default) which completely skips version checking.

See Also

`manager`, `bspm-options`

Examples

```
## Not run:
# install 'units' and all its dependencies from the system repos
bspm::enable()
install.packages("units")

# install packages again from CRAN
bspm::disable()
install.packages("errors")

## End(Not run)
```

`manager` *Manage Packages from System Repositories*

Description

Talk to the system package manager to install/remove... packages from system repositories (see details for further options).

Usage

```
install_sys(pkgs)

remove_sys(pkgs)

moveto_sys(lib, newer = FALSE)

available_sys()

discover()
```

Arguments

pkgs	character vector of names of packages.
lib	a character vector giving the library directories to remove the packages from. If missing, defaults to the first element in R_LIBS_USER.
newer	whether to move newer packages from the user library. The special value "ask" is also supported.

Details

If R runs with root privileges (e.g., in a docker container), these functions talk directly to the system package manager. Regular users are also able to install/remove packages without any administrative permission via the accompanying D-Bus service if **bspm** is installed as a system package. If not, these methods fall back on using sudo to elevate permissions (or pkexec in GUIs such as RStudio) in interactive sessions. Note that, if you want to fall back to sudo in a non-interactive session, you need to set options(bspm.sudo=TRUE).

If options(bspm.sudo.autodetect=TRUE), **bspm** tries to detect whether it is running in an environment where password-less sudo can be used (e.g., in a containerized environment such as a Fedora Toolbox) for every call, and then uses sudo accordingly.

The moveto_sys method moves existing user packages to the system library to avoid *package shadowing* (i.e., installs the available system packages and removes copies from the user library; see shadowed_packages). This provides a mechanism to easily deploy **bspm** on an existing R installation with a populated user library.

The discover method is only needed when e.g. a new repository is added that contains packages with different prefixes (for example, your system repositories may provide packages called r-cran-* and r-bioc-* and then you add a new repository that provides packages called r-github-*). Otherwise, it will not have any effect besides regenerating the internal configuration files.

Value

Functions install_sys, remove_sys, and moveto_sys return, invisibly, a character vector of the names of packages not available in the system.

Function available_sys returns a matrix with one row per package. Row names are the package names, and column names include "Package", "Version", "Repository".

See Also

integration, bspm-scripts

Examples

```
## Not run:
# install 'units' and all its dependencies from the system repos
bspm::install_sys("units")

# now remove it
bspm::remove_sys("units")

# get available packages
bspm::available_sys()

## End(Not run)
```

---

shadowed_packages *Find Shadowed Packages*

Description

Find packages that are *shadowed* by others in library locations with a higher priority.

Usage

```
shadowed_packages(lib.loc = NULL)
```

Arguments

lib.loc	character vector describing the location of the R library trees to search through, or NULL for all known trees (see .libPaths).

Details

R supports setting several locations for library trees. This is a powerful feature, but many times packages end up installed in multiple locations, and in such cases R silently uses the one in the path with the highest priority (appearing first in .libPaths), thus *shadowing* packages in locations with a lower priority.

For **bspm** installations, this means that outdated user packages may break system libraries. This utility reports packages that are shadowed (one per row) with information on which location ("Shadow.LibPath") and version ("Shadow.Version") has priority over it. The moveto_sys method is a great complement to move such outdated versions to the system libraries.

Value

A data.frame with one row per package, row names the package names and column names (currently) "Package", "LibPath", "Version", "Shadow.LibPath", "Shadow.Version", "Shadow.Newer".

See Also

moveto_sys

---

sysreqs *Find System Requirements*

Description

Compute the system requirements (system libraries; operating system packages) required by a set of R packages.

Usage

```
sysreqs(pkgs, ..., type = c("build", "run"), distro = NULL,
  collapse = FALSE)
```

Arguments

pkgs	character vector of R package names.
...	unused, reserved for future expansion.
type	type of requirements: build- or run-time requirements, or both.
distro	name of the Linux distribution; if nothing is provided, the function tries to detect it automatically.
collapse	whether to collapse the requirements into a single line.

Details

This function relies on https://github.com/cran4linux/sysreqs.

Value

A character vector of system requirements.

Index

.libPaths, *6*

available_sys *(*manager*)*, 4

bspm *(*bspm-package*)*, 2
bspm-options, 2
bspm-package, 2
bspm-scripts, 3

disable *(*integration*)*, 3
discover *(*manager*)*, 4

enable, *2*
enable *(*integration*)*, 3

install.packages, *3*
install_sys, *3*
install_sys *(*manager*)*, 4
integration, *2*, 3, *6*

manager, *2*, *4*, 4
moveto_sys, *6, 7*
moveto_sys *(*manager*)*, 4

options, *2, 3*

remove_sys *(*manager*)*, 4

shadowed_packages, *5*, 6
sysreqs, 7