Package ‘bspm’

January 4, 2022

Type Package
Title Bridge to System Package Manager
Version 0.3.9
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-(dnf|apt|alpm)
Suggests tinytest
URL https://github.com/Enchufa2/bspm
BugReports https://github.com/Enchufa2/bspm/issues
RoxygenNote 7.1.2
NeedsCompilation no
Author Iñaki Ucar [aut, cph, cre] (<https://orcid.org/0000-0001-6403-5550>)
Maintainer Iñaki Ucar <iucar@fedoraproject.org>
Repository CRAN
Date/Publication 2022-01-04 15:30:08 UTC

R topics documented:

bspm-package ......................................................... 2
integration ........................................................... 2
manager ............................................................... 3

Index 5
**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://github.com/Enchufa2/bspm

**See Also**
`manager`, `integration`

---

**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, 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.
See Also

manager

Examples

## Not run:
## 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)
discover()

Arguments

pkgs character vector of names of packages.

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.
By default, if a package is not available in the system repositories, it is installed from R’s configured repositories along with all its dependencies. This behavior can be changed via `options(bspm.always.install.deps=TRUE)` which tries to install from system repositories recursive dependencies of those packages that are not available. For example, if A depends on B, and B is available in the system repositories but A is not, then only A will be installed from CRAN with this option enabled, and both will be installed from CRAN with this option disabled (default).

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

**See Also**

`integration`

**Examples**

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

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

## End(Not run)
```
Index

bspm-package, 2

disable (integration), 2
discover (manager), 3

enable (integration), 2

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

manager, 2, 3, 3

remove_sys (manager), 3