Package 'bspm'

January 9, 2023

Type Package

Title Bridge to System Package Manager

Version 0.4.1

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://enchufa2.github.io/bspm/

BugReports https://github.com/Enchufa2/bspm/issues

RoxygenNote 7.2.2

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 2023-01-09 22:30:07 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://enchufa2.github.io/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, 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.

See Also

`manager`

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)

available_sys()

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.

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.

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

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)
```

Index

available_sys *(manager)*, 3

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