

# Package ‘needs’

October 13, 2022

**Type** Package

**Title** Attaches and Installs Packages

**Version** 0.0.3

**Description** A simple function for easier package loading and auto-installation.

**URL** <https://github.com/joshkatz/needs>

**BugReports** <https://github.com/joshkatz/needs/issues>

**Depends** R (>= 3.2.0)

**Imports** utils

**License** MIT + file LICENSE

**LazyData** TRUE

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Author** Josh Katz [aut, cre]

**Maintainer** Josh Katz <josh.katz@nytimes.com>

**Repository** CRAN

**Date/Publication** 2016-03-28 11:04:09

## R topics documented:

needs-package . . . . .	2
needs . . . . .	2
prioritize . . . . .	3
toProfile . . . . .	4

<b>Index</b>	<b>5</b>
--------------	----------

---

needs-package	<i>Easier package loading / auto-installation</i>
---------------	---

---

### Description

**needs** is a simple R function to make package loading / installation hassle-free — use it in place of `library` to attach packages and automatically install any that are missing. You can also supply a minimum version number, and it will update old packages as needed. No more changing your code to reinstall packages every time you update R — `needs` does it for you.

### Author(s)

Josh Katz

### References

Source repo: <http://www.github.com/joshkatz/needs>

---

needs	<i>Attach/install packages</i>
-------	--------------------------------

---

### Description

`needs` loads and attaches packages, automatically installing (and attaching) any it can't find in your libraries. It accepts any number of arguments, given as names or character strings. Optionally, supply a minimum version on a per-package basis to update old packages as needed.

### Usage

```
needs(..., .printConflicts = F)
```

### Arguments

<code>...</code>	Packages, given as unquoted names or character strings. Specify a required package version as <code>package = "version"</code> .
<code>.printConflicts</code>	Logical, specifying whether to print a summary of objects that exist in multiple places on the search path along with their respective locations. Set to <code>TRUE</code> to identify any masked functions. Objects in the base package and the global environment are ignored. Defaults to <code>FALSE</code> .

### Details

Recommended use is to allow the function to autoload when prompted the first time the package is loaded interactively. To change this setting later, run `needs:::autoload(TRUE)` or `needs:::autoload(FALSE)` to turn autoloading on or off, respectively.

**See Also**[needs-package](#)**Examples**

```
## Not run:
needs() # returns NULL

needs(foo, bar)

# require a minimum version
needs(foo,
      bar = "0.9.1",
      baz = "0.4.3")

## End(Not run)
```

---

prioritize	<i>Re-attach packages to prevent masking</i>
------------	--

---

**Description**

prioritize detaches packages from the search path, then re-attaches them, placing them at the beginning of the search path to prevent masking. This allows for the loading of packages with conflicting function names in any order.

**Usage**

```
prioritize(...)
```

**Arguments**

... Packages, given as unquoted names or character strings. Earlier arguments will be attached later (and therefore get a higher priority).

**Details**

If you find yourself calling this function a lot, you're probably doing something wrong.

**Examples**

```
## Not run:

# loading plyr after dplyr causes badness
needs(dplyr, plyr)
```

```
# prioritize the functions in dplyr
prioritize(dplyr)

## End(Not run)
```

---

toProfile                      *Append to Rprofile in current working directory.*

---

### Description

Exports the package functionality to .Rprofile in the current working directory. Now the same code you run can be run on another system without requiring any extra installation or throwing errors for uninstalled packages.

### Usage

```
toProfile(dir = NULL, append = T)
```

### Arguments

dir	Target directory for Rprofile. Defaults to working directory.
append	Whether to append to the current Rprofile, if one exists. Set to false to overwrite.

### Examples

```
## Not run:
toProfile()

## End(Not run)
```

# Index

needs, [2](#)  
needs-package, [2](#)  
prioritize, [3](#)  
toProfile, [4](#)