

Package ‘sessioninfo’

July 23, 2025

Title R Session Information

Version 1.2.3

Maintainer Gábor Csárdi <csardi.gabor@gmail.com>

Description Query and print information about the current R session. It is similar to 'utils::sessionInfo()', but includes more information about packages, and where they were installed from.

License GPL-2

URL <https://github.com/r-lib/sessioninfo#readme>,
<https://sessioninfo.r-lib.org>

BugReports <https://github.com/r-lib/sessioninfo/issues>

Depends R (>= 3.4)

Imports cli (>= 3.1.0), tools, utils

Suggests callr, covr, gh, reticulate, rmarkdown, testthat (>= 3.2.0),
withr

Config/Needs/website pkgdown, tidyverse/tidytemplate

Config/testthat/edition 3

Config/testthat/parallel true

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Gábor Csárdi [cre],
Hadley Wickham [aut],
Winston Chang [aut],
Robert Flight [aut],
Kirill Müller [aut],
Jim Hester [aut],
R Core team [ctb],
Posit Software, PBC [cph, fnd]

Repository CRAN

Date/Publication 2025-02-05 13:40:02 UTC

Contents

external_info	2
os_name	3
package_info	3
platform_info	5
python_info	6
session_diff	6
session_info	7
Index	10

external_info	<i>Information about related software</i>
---------------	---

Description

Information about related software

Usage

external_info()

Details

Note that calling this function will attempt to load the tcltk and grDevices packages.

Value

A list with elements:

- cairo: The cairo version string.
- libpng: The png version string.
- jpeg: The jpeg version string.
- tiff: The tiff library and version string used.
- tcl: The tcl version string.
- curl: The curl version string.
- zlib: The zlib version string.
- bzlib: The zlib version string.
- xz: The zlib version string.
- PCRE: The Perl Compatible Regular Expressions (PCRE) version string.
- ICU: The International Components for Unicode (ICU) version string.
- TRE: The TRE version string.
- iconv: The iconv version string.
- readline: The readline version string.
- BLAS: The path with the implementation of BLAS in use.
- LAPACK: The path with the implementation of LAPACK in use.

See Also

Similar functions and objects in the base packages: [utils::sessionInfo\(\)](#), [base::extSoftVersion](#), [tcltk::tclVersion\(\)](#) [base::La_library](#), [base::La_version\(\)](#), [base::libcurlVersion\(\)](#).

Examples

```
external_info()
```

os_name	<i>Human readable name of the current operating system</i>
---------	--

Description

For example Windows 8.1 instead of Windows version 6.3.9600. On macOS it includes the code names, on Linux it includes the distribution names and codenames if appropriate.

Usage

```
os_name()
```

Details

It uses [utils::sessionInfo\(\)](#), but simplifies its output a bit on Windows, to make it more concise.

Value

A character scalar.

package_info	<i>Information about the currently loaded packages, or about a chosen set</i>
--------------	---

Description

Information about the currently loaded packages, or about a chosen set

Usage

```
package_info(
  pkgs = c("loaded", "attached", "installed")[1],
  include_base = FALSE,
  dependencies = NA
)
```

Arguments

pkgs	Which packages to show. It may be: <ul style="list-style-type: none"> • NULL or "loaded": show all loaded packages, • "attached": show all attached packages, • "installed": show all installed packages, • a character vector of package names. Their (hard) dependencies are also shown by default, see the dependencies argument.
include_base	Include base packages in summary? By default this is false since base packages should always match the R version.
dependencies	Whether to include the (recursive) dependencies as well. See the dependencies argument of <code>utils::install.packages()</code> .

Value

A data frame with columns:

- package: package name.
- ondiskversion: package version (on the disk, which is sometimes not the same as the loaded version).
- loadedversion: package version. This is the version of the loaded namespace if pkgs is NULL, and it is the version of the package on disk otherwise. The two of them are almost always the same, though.
- path: path to the package on disk.
- loadedpath: the path the package was originally loaded from.
- attached: logical, whether the package is attached to the search path.
- is_base: logical, whether the package is a base package.
- date: the date the package was installed or built, in UTC.
- source: where the package was installed from. E.g. CRAN, GitHub, local (from the local machine), etc.
- md5ok: Whether MD5 hashes for package DLL files match, on Windows. NA on other platforms.
- library: factor, which package library the package was loaded from. For loaded packages, this is (the factor representation of) loadedpath, for others path.

See `session_info()` for the description of the *printed* columns by `package_info` (as opposed to the *returned* columns).

Examples

```
package_info()
package_info("sessioninfo")
```

platform_info	<i>Information about the current platform</i>
---------------	---

Description

Information about the current platform

Usage

```
platform_info()
```

Value

A list with elements:

- `version`: the R version string.
- `os`: the OS name in human readable format, see [os_name\(\)](#).
- `system`: CPU, and machine readable OS name, separated by a comma.
- `ui`: the user interface, e.g. Rgui, RTerm, etc. see GUI in [base::Platform](#).
- `hostname`: the name of the machine known on the network, see `nodename` in [base::Sys.info\(\)](#). For privacy, it is only included if the `sessioninfo.include_hostname` option is set to TRUE.
- `language`: The current language setting. The LANGUAGE environment variable, if set, or (EN) if unset.
- `collate`: Collation rule, from the current locale.
- `ctype`: Native character encoding, from the current locale.
- `tz`: The current time zone.
- `date`: The current date.
- `rstudio`: RStudio format string, only added in RStudio.
- `pandoc`: pandoc version and path
- `quarto`: quarto version and path

See Also

Similar functions and objects in the base packages: [base::R.version.string](#), [utils::sessionInfo\(\)](#), [base::version](#), [base::Platform](#), [base::Sys.getlocale\(\)](#), [base::Sys.timezone\(\)](#).

Examples

```
platform_info()
```

python_info	<i>Python configuration</i>
-------------	-----------------------------

Description

Python configuration

Usage

```
python_info()
```

Value

Returns a [reticulate::py_config](#) object, which also has the python_info class. It is a named list of values.

If reticulate is not installed or Python is not configured, then it return a python_info object that is a character vector, and it does not have a py_config class.

Examples

```
python_info()  
session_info(info = "all")
```

session_diff	<i>Compare session information from two sources</i>
--------------	---

Description

Compare session information from two sources

Usage

```
session_diff(  
  old = "local",  
  new = "clipboard",  
  packages = c("diff", "merge"),  
  ...  
)
```

Arguments

old, new	A session_info object (the return value of <code>session_info()</code>), or a pointer to <code>session_info()</code> output. See details below.
packages	How to compare the package info for old and new: <ul style="list-style-type: none"> • "diff": line diffs, in the style of <code>diff -u</code> • "merge": merge the information into one data frame, with one row per package. Only package version and source are compared.
...	Passed to any new <code>session_info()</code> calls.

Details

Various way to specify old and new:

- A session_info object.
- "local" runs `session_info()` in the current session, and uses its output.
- "clipboard" takes the session info from the system clipboard. If the clipboard contains a URL, it is followed to download the session info.
- The URL where you inspect the results for a GitHub Actions job. Typically has this form:

```
https://github.com/OWNER/REPO/actions/runs/RUN_ID/jobs/HTML_ID
```

Internally, this URL is parsed so we can look up the job id, get the log file, and extract session info.

- Any other URL starting with `http://` or `https://`. `session_diff()` searches the HTML (or text) page for the session info header to find the session info.

Examples

```
session_diff()
```

```
session_info          Print session information
```

Description

This is `utils::sessionInfo()` re-written from scratch to both exclude data that's rarely useful (e.g., the full collate string or base packages loaded) and include stuff you'd like to know (e.g., where a package was installed from).

Usage

```
session_info(
  pkgs = c("loaded", "attached", "installed")[1],
  include_base = FALSE,
  info = c("auto", "all", "platform", "packages", "python", "external"),
  dependencies = NA,
  to_file = FALSE
)
```

Arguments

pkgs	Which packages to show. It may be: <ul style="list-style-type: none"> • NULL or "loaded": show all loaded packages, • "attached": show all attached packages, • "installed": show all installed packages, • a character vector of package names. Their (hard) dependencies are also shown by default, see the dependencies argument.
include_base	Include base packages in summary? By default this is false since base packages should always match the R version.
info	What information to show, it can be "auto" to choose automatically, "all" to show everything, or a character vector with elements from: <ul style="list-style-type: none"> • "platform": show platform information via <code>platform_info()</code>, • "packages": show package information via <code>package_info()</code>, • "python": show Python configuration via <code>python_info()</code>, • "external": show information about external software, via <code>external_info()</code>.
dependencies	Whether to include the (recursive) dependencies as well. See the dependencies argument of <code>utils::install.packages()</code> .
to_file	Whether to print the session information to a file. If TRUE the name of the file will be <code>session-info.txt</code> , but <code>to_file</code> may also be a string to specify the file name.

Details

Columns in the *printed* package list:

- package: package name
- *: whether the package is attached to the search path
- version: package version. If the version is marked with (!) that means that the loaded and the on-disk version of the package are different.
- date: when the package was built, if this information is available. This is the Date/Publication or the Built field from DESCRIPTION. (These are usually added automatically by R.) Sometimes this data is not available, then it is NA.
- source: where the package was built or installed from, if available. Examples: CRAN (R 3.3.2), Github (r-lib/pkgbuild@8aab60b), Bioconductor, local.

See `package_info()` for the list of columns in the data frame that is *returned* (as opposed to *printed*).

Value

A `session_info` object.

If `to_file` is not FALSE then it is returned invisibly. (To print it to both a file and to the screen, use `(session_info(to_file = TRUE))`.)

Examples

```
session_info()  
session_info("sessioninfo")
```

Index

`base::Platform`, 5
`base::extSoftVersion`, 3
`base::La_library`, 3
`base::La_version()`, 3
`base::libcurlVersion()`, 3
`base::R.version.string`, 5
`base::Sys.getlocale()`, 5
`base::Sys.info()`, 5
`base::Sys.timezone()`, 5
`base::version`, 5

`external_info`, 2
`external_info()`, 8

`os_name`, 3
`os_name()`, 5

`package_info`, 3
`package_info()`, 8
`platform_info`, 5
`platform_info()`, 8
`python_info`, 6
`python_info()`, 8

`reticulate::py_config`, 6

`session_diff`, 6
`session_info`, 7
`session_info()`, 4, 7

`tcltk::tclVersion()`, 3

`utils::install.packages()`, 4, 8
`utils::sessionInfo()`, 3, 5, 7