

Package ‘rappdirs’

August 29, 2016

Type Package

Title Application Directories: Determine Where to Save Data, Caches,
and Logs

Version 0.3.1

Depends R (>= 2.14), methods

Suggests testthat, roxygen2

Description An easy way to determine which directories on the users computer
you should use to save data, caches and logs. A port of Python's 'Appdirs'
([\{\}url{https://github.com/ActiveState/appdirs}](https://github.com/ActiveState/appdirs)) to R.

BugReports <https://github.com/hadley/rappdirs/issues>

URL <https://github.com/hadley/rappdirs>
<https://github.com/ActiveState/appdirs>

License MIT + file LICENSE

Copyright Original python appdirs module copyright (c) 2010
ActiveState Software Inc. R port copyright Hadley Wickham,
RStudio. See file LICENSE for details.

RoxygenNote 5.0.1

NeedsCompilation yes

Author Hadley Wickham [trl, cre, cph],
RStudio [cph],
Sridhar Ratnakumar [aut],
Trent Mick [aut],
ActiveState [cph] (R/appdir.r, R/cache.r, R/data.r, R/log.r translated
from appdirs),
Eddy Petrisor [ctb],
Trevor Davis [trl, aut],
Gabor Csardi [ctb],
Gregory Jefferis [ctb]

Maintainer Hadley Wickham <hadley@rstudio.com>

Repository CRAN

Date/Publication 2016-03-28 23:10:10

R topics documented:

rappdirs-package	2
app_dir	3
site_data_dir	4
user_cache_dir	5
user_data_dir	7
user_log_dir	8

Index	10
--------------	-----------

rappdirs-package	<i>Application directories: determine where to save data, caches and logs.</i>
------------------	--

Description

rappdirs solves the problem of where to save persistent data outside of the R library or the R per-session `tempdir`.

main functions

- user data dir (`user_data_dir`)
- user config dir (`user_config_dir`)
- user cache dir (`user_cache_dir`)
- site data dir (`site_data_dir`)
- user log dir (`user_log_dir`)

single entry function

The `app_dir` provides a convenient single entry point.

Caveats

Note that if you use rappdir's `user_data_dir` and friends to define a storage location for files you must be aware of **race conditions** when more than one R process tries to create/write files in this directory. This is in contrast to using the `tempdir`, `tempfile` base functions which should be unique for each R process. In general the directories provided by rappdirs are most suitable for storing data that is rarely written but might need to be shared across R sessions.

Note also that the **CRAN Policies** have the following to say about storage of data by packages:

- Packages should not write in the users' home filesystem, nor anywhere else on the file system apart from the R session's temporary directory (or during installation in the location pointed to by `TMPDIR`: and such usage should be cleaned up). Installing into the system's R installation (e.g., scripts to its bin directory) is not allowed.

Limited exceptions may be allowed in interactive sessions if the package obtains confirmation from the user.

See Also

[app_dir](#), [user_data_dir](#), [user_config_dir](#), [user_cache_dir](#), [site_data_dir](#), [user_log_dir](#)

Examples

```
dirs <- app_dir("SuperApp", "Acme")
dirs$config()
```

app_dir

Convenience wrapper for getting app dirs.

Description

Has methods:

Usage

```
app_dir(appname = NULL, appauthor = appname, version = NULL,
        expand = TRUE, os = get_os())
```

Arguments

appname	is the name of application. If NULL, just the system directory is returned.
appauthor	(only required and used on Windows) is the name of the appauthor or distributing body for this application. Typically it is the owning company name. This falls back to appname.
version	is an optional version path element to append to the path. You might want to use this if you want multiple versions of your app to be able to run independently. If used, this would typically be "<major>.<minor>". Only applied when appname is not NULL.
expand	If TRUE (the default) will expand the R_LIBS specifiers with their equivalents. See R_LIBS for list of all possibly specifiers.
os	Operating system whose conventions are used to construct the requested directory. Possible values are "win", "mac", "unix". If NULL (the default) then the convention of the current operating system (as determined by <code>rappdirs::get_os</code>) will be used. This argument is unlikely to find much use outside package testing (see details section of user_data_dir).

Details

- cache
- log
- data
- config
- site_data
- site_config

Examples

```
ggplot2_app <- app_dir("ggplot2", "hadley")
ggplot2_app$cache()
ggplot2_app$log()
ggplot2_app$data()
ggplot2_app$config()
ggplot2_app$site_config()
ggplot2_app$site_data()
```

site_data_dir	<i>Return full path to the user-shared data dir for this application.</i>
---------------	---

Description

site_data_dir returns full path to the user-shared data dir for this application. site_config_dir returns full path to the user-specific configuration directory for this application which returns the same path as site data directory in Windows and Mac but a different one for Unix. Typical user-shared data directories are:

Usage

```
site_data_dir(appname = NULL, appauthor = appname, version = NULL,
  multipath = FALSE, expand = TRUE, os = get_os())

site_config_dir(appname = NULL, appauthor = appname, version = NULL,
  multipath = FALSE, expand = TRUE, os = get_os())
```

Arguments

appname	is the name of application. If NULL, just the system directory is returned.
appauthor	(only required and used on Windows) is the name of the appauthor or distributing body for this application. Typically it is the owning company name. This falls back to appname.
version	is an optional version path element to append to the path. You might want to use this if you want multiple versions of your app to be able to run independently. If used, this would typically be "<major>.<minor>". Only applied when appname is not NULL.
multipath	is an optional parameter only applicable to *nix which indicates that the entire list of data dirs should be returned. By default, the first directory is returned.
expand	If TRUE (the default) will expand the R_LIBS specifiers with their equivalents. See R_LIBS for list of all possibly specifiers.
os	Operating system whose conventions are used to construct the requested directory. Possible values are "win", "mac", "unix". If NULL (the default) then the convention of the current operating system (as determined by rappdirs::get_os) will be used. This argument is unlikely to find much use outside package testing (see details section of user_data_dir).

Details

- Mac OS X: `‘/Library/Application Support/<AppName>’`
- Unix: `‘/usr/local/share:/usr/share/’`
- Win XP: `‘C:\Documents and Settings\All Users\Application Data\<AppAuthor>\<AppName>’`
- Vista: (Fail! `‘C:\ProgramData’` is a hidden *system* directory on Vista.)
- Win 7: `‘C:\ProgramData\<AppAuthor>\<AppName>’`. Hidden, but writeable on Win 7.

Unix also specifies a separate location for user-shared configuration data in `$XDG_CONFIG_DIRS`.

- Unix: `‘/etc/xdg/<AppName>’`, in `$XDG_CONFIG_HOME` if defined

For Unix, this returns the first default. Set the `multipath=TRUE` to guarantee returning all directories.

Warning

Do not use this on Windows. See the note above for why.

<code>user_cache_dir</code>	<i>Return full path to the user-specific cache dir for this application.</i>
-----------------------------	--

Description

Typical user cache directories are:

Usage

```
user_cache_dir(appname = NULL, appauthor = appname, version = NULL,
  opinion = TRUE, expand = TRUE, os = get_os())
```

Arguments

- | | |
|------------------------|--|
| <code>appname</code> | is the name of application. If <code>NULL</code> , just the system directory is returned. |
| <code>appauthor</code> | (only required and used on Windows) is the name of the appauthor or distributing body for this application. Typically it is the owning company name. This falls back to <code>appname</code> . |
| <code>version</code> | is an optional version path element to append to the path. You might want to use this if you want multiple versions of your app to be able to run independently. If used, this would typically be <code>"<major>.<minor>"</code> . Only applied when <code>appname</code> is not <code>NULL</code> . |
| <code>opinion</code> | (logical) can be <code>FALSE</code> to disable the appending of ‘Cache’ to the base app data dir for Windows. See discussion below. |
| <code>expand</code> | If <code>TRUE</code> (the default) will expand the <code>R_LIBS</code> specifiers with their equivalents. See R_LIBS for list of all possibly specifiers. |

os Operating system whose conventions are used to construct the requested directory. Possible values are "win", "mac", "unix". If NULL (the default) then the convention of the current operating system (as determined by `rappdirs::get_os`) will be used. This argument is unlikely to find much use outside package testing (see details section of `user_data_dir`).

Details

- Mac OS X: `~/Library/Caches/<AppName>`
- Unix: `~/ .cache/<AppName>`, `$XDG_CACHE_HOME` if defined
- Win XP: `C:\Documents and Settings\<username>\Local Settings\Application Data\<AppAuthor>\<AppName>`
- Vista: `C:\Users\<username>\AppData\Local\<AppAuthor>\<AppName>\Cache`

On Windows the only suggestion in the MSDN docs is that local settings go in the `'CSIDL_LOCAL_APPDATA'` directory. This is identical to the non-roaming app data dir (the default returned by `'user_data_dir'` above). Apps typically put cache data somewhere **under** the given dir here. Some examples: `'...\Mozilla\Firefox\Profiles\<ProfileName>\Cache'`, `'...\Acme\SuperApp\Cache\1.0'`

Opinion

This function appends `'Cache'` to the `'CSIDL_LOCAL_APPDATA'` value. This can be disabled with `opinion = FALSE` option.

See Also

[tempdir](#) for a non-persistent temporary directory.

Examples

```
user_cache_dir("rappdirs")
## Not run:
# Throw this in your R profile to store a R history file in standard cache location
if(capabilities("cledit")) {
  cache_dir <- rappdirs::user_cache_dir("R")
  history_file <- file.path(cache_dir, "Rhistory")
  .First <- function() utils::loadhistory(history_file)
  .Last <- function() {
    if (!file.exists(cache_dir)) dir.create(cache_dir, recursive = TRUE)
    try(savehistory(history_file))
  }
}
## End(Not run)
```

user_data_dir	<i>Return path to user data directories.</i>
---------------	--

Description

user_data_dir returns full path to the user-specific data dir for this application. user_config_dir returns full path to the user-specific configuration directory for this application which returns the same path as user data directory in Windows and Mac but a different one for Unix.

Usage

```
user_data_dir(appname = NULL, appauthor = appname, version = NULL,
             roaming = FALSE, expand = TRUE, os = get_os())
```

```
user_config_dir(appname = NULL, appauthor = appname, version = NULL,
               roaming = TRUE, expand = TRUE, os = get_os())
```

Arguments

appname	is the name of application. If NULL, just the system directory is returned.
appauthor	(only required and used on Windows) is the name of the appauthor or distributing body for this application. Typically it is the owning company name. This falls back to appname.
version	is an optional version path element to append to the path. You might want to use this if you want multiple versions of your app to be able to run independently. If used, this would typically be "<major>.<minor>". Only applied when appname is not NULL.
roaming	(logical, default FALSE) can be set TRUE to use the Windows roaming appdata directory. That means that for users on a Windows network setup for roaming profiles, this user data will be sync'd on login. See http://technet.microsoft.com/en-us/library/cc766489(WS.10).aspx for a discussion of issues.
expand	If TRUE (the default) will expand the R_LIBS specifiers with their equivalents. See R_LIBS for list of all possibly specifiers.
os	Operating system whose conventions are used to construct the requested directory. Possible values are "win", "mac", "unix". If NULL (the default) then the convention of the current operating system (as determined by rappdirs::get_os) will be used. This argument is unlikely to find much use outside package testing (see details section of user_data_dir).

Details

Typical user data directories are:

- Mac OS X: '~/Library/Application Support/<AppName>'
- Unix: '~/local/share/<AppName>', in \$XDG_DATA_HOME if defined

- Win XP (not roaming): 'C:\Documents and Settings\\Data\\'
- Win XP (roaming): 'C:\Documents and Settings\\Local Settings\Data\\'
- Win 7 (not roaming): 'C:\Users\\AppData\Local\\'
- Win 7 (roaming): 'C:\Users\\AppData\Roaming\\'

Unix also specifies a separate location for user configuration data in

- Unix: '~/.config/<AppName>', in \$XDG_CONFIG_HOME if defined

See for example <http://ploum.net/184-cleaning-user-preferences-keeping-user-data/> or <http://standards.freedesktop.org/basedir-spec/basedir-spec-latest.html> for more information. Arguably plugins such as R packages should go into the user configuration directory and deleting this directory should return the application to a default settings.

The `os` parameter allows the calculation of directories based on a convention other than the current operating system. This feature is designed with package testing in mind and is *not* recommended for end users. One possible exception is that some users on "mac" might wish to use the "unix" XDG convention.

Examples

```
user_data_dir("rappdirs")
user_config_dir("rappdirs", version="%p-platform/%v")
user_config_dir("rappdirs", roaming=TRUE, os="win")
user_config_dir("rappdirs", roaming=FALSE, os="win")
user_config_dir("rappdirs", os="unix")
user_config_dir("rappdirs", os="mac")
## Not run:
# you could try to use functions to store R libraries in a standard user directory
# by using the following in your .Rprofile file
# but unfortunately if rappdir package was stored in standard user directory then
# it won't be on R's search path any longer, so would need to be installed system-wide...
require("utils")
.libPaths(new=rappdirs::user_config_dir("R", version="%p-platform/%v"))

## End(Not run)
```

user_log_dir

Return full path to the user-specific log dir for this application.

Description

Typical user cache directories are:

Usage

```
user_log_dir(appname = NULL, appauthor = appname, version = NULL,
  opinion = TRUE, expand = TRUE, os = get_os())
```


Arguments

appname	is the name of application. If NULL, just the system directory is returned.
appauthor	(only required and used on Windows) is the name of the appauthor or distributing body for this application. Typically it is the owning company name. This falls back to appname.
version	is an optional version path element to append to the path. You might want to use this if you want multiple versions of your app to be able to run independently. If used, this would typically be "<major>.<minor>". Only applied when appname is not NULL.
opinion	(logical) can be FALSE to disable the appending of 'Logs' to the base app data dir for Windows, and 'log' to the base cache dir for Unix. See discussion below.
expand	If TRUE (the default) will expand the R_LIBS specifiers with their equivalents. See R_LIBS for list of all possibly specifiers.
os	Operating system whose conventions are used to construct the requested directory. Possible values are "win", "mac", "unix". If NULL (the default) then the convention of the current operating system (as determined by <code>rappdirs::get_os</code>) will be used. This argument is unlikely to find much use outside package testing (see details section of user_data_dir).

Details

- Mac OS X: `~/Library/Logs/<AppName>`
- Unix: `~/ .cache/<AppName>/log`, or under `\env$XDG_CACHE_HOME` if defined
- Win XP: `C:\Documents and Settings\<username>\Local Settings\Application Data\<AppAuthor>\<AppName>`
- Vista: `C:\Users\<username>\AppData\Local\<AppAuthor>\<AppName>\Logs`

On Windows the only suggestion in the MSDN docs is that local settings go in the `CSIDL_LOCAL_APPDATA` directory. (Note: I'm interested in examples of what some windows apps use for a logs dir.)

Opinion

This function appends 'Logs' to the `'CSIDL_LOCAL_APPDATA'` value for Windows and appends 'log' to the user cache dir for Unix. This can be disabled with the `opinion = FALSE` option.

Examples

```
user_log_dir()
```

Index

app_dir, 3, 3

R_LIBS, 3–5, 7, 9

rappdirs (rappdirs-package), 2

rappdirs-package, 2

site_config_dir (site_data_dir), 4

site_data_dir, 3, 4

tempdir, 2, 6

tempfile, 2

user_cache_dir, 3, 5

user_config_dir, 3

user_config_dir (user_data_dir), 7

user_data_dir, 2–4, 6, 7, 7, 9

user_log_dir, 3, 8