Package: ctv (via r-universe)

October 8, 2026

Version 1.0-1

Date 2026-10-08

Title CRAN Task Views

Description Infrastructure for task views to CRAN-style repositories: Querying task views and installing the associated packages (client-side tools), generating HTML pages and storing task view information in the repository (server-side tools).

Depends R (>= 3.0.0), utils

Suggests xml2, knitr, rmarkdown

VignetteBuilder knitr

License GPL-2 | GPL-3

URL https://github.com/cran-task-views/ctv/, https://zeileis.codeberg.page/ctv/

BugReports https://codeberg.org/zeileis/ctv/issues

Repository https://zeileis.r-universe.dev

Date/Publication 2026-10-08 01:27:03 UTC

RemoteUrl https://codeberg.org/zeileis/ctv

RemoteRef HEAD

RemoteSha c69e285e73bfbbb7ac80fbc1cf6bce103efa3141

Contents

ctv-client 2
ctv-server 4

Index **7**

ctv-client *CRAN Task Views: Installing/Updating/Downloading Packages*

Description

Client-side tools for installing CRAN task views.

Usage

```
ctv(view, repos = NULL, ...)

available.views(repos = NULL, ...)
install.views(views, coreOnly = FALSE, repos = NULL, ...)
update.views(views, coreOnly = FALSE, repos = NULL, lib.loc = NULL, filters = NULL, ...)
download.views(views, destdir, coreOnly = FALSE, repos = NULL, ...)

## S3 method for class 'ctv'
print(x, packagelist = TRUE, ...)
## S3 method for class 'ctvlist'
print(x, packagelist = FALSE, ...)
```

Arguments

view, views	character vector with the short names of the task views whose associated packages should be downloaded and installed. Alternatively, views can also be an object of class "ctvlist" (as returned by available.views) or an object of class "ctv" (i.e., an element of a "ctvlist").
coreOnly	logical. Should all packages or only core packages be installed? (recycled to the same length as views)
repos	character, the base URL of the repository. By default getOption("repos") is tried and otherwise getOption("CRAN") is used.
lib.loc	character vector describing the location of R library trees to search through (and update packages therein).
filters	a character vector or list to filter available.packages, e.g., for filtering with respect to operating system type or free and open-source software license.
destdir	directory where downloaded packages are to be stored.
...	further arguments passed to install.packages or download.packages respectively.
x	an object of class "ctv" or "ctvlist" respectively.
packagelist	logical. Should the packagelist also be printed?

Details

`install.views` queries the file 'Views.rds' located at the 'src/contrib' directory of 'repos' and then simply calls `install.packages` to install the packages associated with the view specified. For each view it can be specified whether all packages or only the core packages should be installed.

`available.views` returns the list of task views currently available in the file 'Views.rds'. In earlier versions, this was called `CRAN.views` (which still works and provides the same functionality).

`update.views` queries which packages from a view are not yet installed (using `installed.packages`) and which of the installed packages are older than the packages available (using `available.packages`). It subsequently installs only the packages that are not current or not installed yet.

`download.views` works exactly as `install.views` except that it calls `download.packages` instead of `install.packages`.

For a more detailed description of the arguments see also `install.packages`.

Value

`available.views` returns an object of class `"ctvlist"` of the available task views whose elements are of class `"ctv"`.

`install.views` and `update.views` have no return value.

See Also

`install.packages`

Examples

```
## Not run:
## query only a single view (with known name)
ctv("Econometrics")

## extract corresponding citation
ctv("Econometrics")$citation

## query list of CRAN task views available
available.views()

## install Econometrics view
install.views("Econometrics")
## only with core packages
install.views("Econometrics", coreOnly = TRUE)

## update Econometrics view (i.e. only packages that are not current anyway)
update.views("Econometrics")

## End(Not run)
```

ctv-server *CRAN Task Views: Tools for Maintainers*

Description

Server-side tools for maintaining CRAN task views.

Usage

```
read.ctv(file, ...)

ctv2html(x, file = NULL, cran = FALSE,
  css = NULL, packageURL = NULL, reposname = "CRAN")

check_ctv_packages(file, repos = TRUE, ...)

repos_update_views(repos = ".", cran = TRUE,
  css = NULL, reposname = "CRAN", ...)
```

Arguments

file	character specifying a file path: for read.ctv a CRAN task view file, and for ctv2html the corresponding output '.html' file.
x	an object of class "ctv" as returned by read.ctv.
cran	logical. Is the HTML to be hosted on CRAN?
css	character specifying the path and name of the cascade style sheet that should be included in the HTML files (if any). The default is "../CRAN_web.css" if cran = TRUE and NULL otherwise.
packageURL	character specifying the path (relative to the view directory) to the package descriptions. The default is "https://CRAN.R-project.org/package=%s" if cran = FALSE and "../packages/" otherwise.
reposname	character giving the name of the CRAN-style repository, used for generating HTML pages.
repos	character, the base URL of the CRAN-style repository where the 'Views.rds' and '.html' files should be installed. The task view files should be located in the 'web/views/' directory. If repos = TRUE (default) the official CRAN repository is used and checks for archived packages are included.
...	further arguments passed to available.packages or ctv2html, respectively.

Details

CRAN Task views are generated from an R/Markdown-based file format that is described in the vignette of this package. (Previous versions used an XML-based format which is still fully supported in all functions.)

`read.ctv` can read a file with a CRAN task view specification and returns an object of class `"ctv"`. This functions requires the availability of the **knitr/rmarkdown** packages (or the **xml2** package, respectively, for the legacy XML format).

`ctv2html` generates a '`.html`' file with the information contained in a `"ctv"` object.

`check_ctv_packages` checks whether the info and packagelist sections of the task view file are consistent with each other and whether all packages are available from the repository.

`repos_update_views` reads all task view files in a specified directory, generates a '`.html`' file for each and an index '`.html`' file. Furthermore, it stores all `"ctv"` objects in a `"ctvlist"` object in a file '`Views.rds`' that can be queried by `install.views` or `CRAN.views`.

Value

`repos_update_views` returns an object of class `"ctvlist"` containing the `"ctv"` objects available.

`ctv2html` returns invisibly a vector with the HTML code generated.

`check_ctv_packages` returns a list of character vectors with package names.

`read.ctv` returns a list of class `"ctv"` with elements:

`name`	character, name of the task view (must be a valid name for an R object).
`topic`	character, topic of the task view.
`maintainer`	character, maintainer of the task view.
`email`	character, valid e-mail address (optional).
`version`	character, version specified via date in ISO format.
`url`	character, valid task view URL (optional).
`source`	character, source repository where the task view file is maintained (optional).
`info`	character, HTML code with informations about the task view.
`packagelist`	data frame with the columns `name` (character, name of package) and `core` (logical, Is priority core?).
`links`	character vector, HTML code with links for the task view.
`otherlinks`	character vector, HTML code with links to other resources for the task view.
`citation`	citation object inheriting from `bibentry` with information how to cite the task view.

See Also

`install.views`

Examples

```
## task view file shipped within the package
ctv <- system.file("ctv", "Econometrics.md", package = "ctv")

if(require("knitr") & require("rmarkdown")) {
## read task view file
x <- read.ctv(ctv)
```

```
## print
x
}

## Not run:
## generate corresponding .html file from task view object
ctv2html(x)
## or equivalently from task view file
ctv2html(ctv)

## check packagelist
check_ctv_packages(ctv)

## End(Not run)
```

Index

∗ **utilities**
ctv-client, 2
ctv-server, 4

available.packages, *2, 3*
available.views *(*ctv-client*)*, 2

bibentry, *5*

check_ctv_packages *(*ctv-server*)*, 4
CRAN.views, *5*
CRAN.views *(*ctv-client*)*, 2
ctv *(*ctv-client*)*, 2
ctv-client, 2
ctv-server, 4
ctv2html *(*ctv-server*)*, 4

download.packages, *2, 3*
download.views *(*ctv-client*)*, 2

install.packages, *2, 3*
install.views, *5*
install.views *(*ctv-client*)*, 2
installed.packages, *3*

print.ctv *(*ctv-client*)*, 2
print.ctvlist *(*ctv-client*)*, 2

read.ctv *(*ctv-server*)*, 4
repos_update_views *(*ctv-server*)*, 4

update.views *(*ctv-client*)*, 2