

# Package ‘ctv’

January 18, 2018

**Version** 0.8-4

**Date** 2018-01-18

**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 (>= 2.13.0), utils

**Suggests** XML

**License** GPL-2 | GPL-3

**NeedsCompilation** no

**Author** Achim Zeileis [aut, cre] (0000-0003-0918-3766),  
Kurt Hornik [aut] (0000-0003-4198-9911)

**Maintainer** Achim Zeileis <Achim.Zeileis@R-project.org>

**Repository** CRAN

**Date/Publication** 2018-01-18 22:15:52 UTC

## R topics documented:

ctv-client	2
ctv-server	3

<b>Index</b>	<b>6</b>
--------------	----------

## Description

Client-side tools for installing CRAN task views.

## Usage

```
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

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 <code>available.views</code> ) 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 <code>getOption("repos")</code> is tried and otherwise <code>getOption("CRAN")</code> 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 <code>available.packages</code> , 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 <code>install.packages</code> or <code>download.packages</code> 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 names of the 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 `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.

### References

Zeileis A (2005). CRAN Task Views. *R News*, 5(1), 39–40. <https://CRAN.R-project.org/doc/Rnews/>.

### See Also

`install.packages`

### Examples

```
## Not run:
## query names 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
update.views("Econometrics")

## End(Not run)
```

### Description

Server-side tools for maintaining CRAN task views.

**Usage**

```
read.ctv(file)

ctv2html(x, file = NULL, css = "../CRAN_web.css",
         packageURL = "../packages/", reposname = "CRAN")

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

repos_update_views(repos = ".", css = "../CRAN_web.css", reposname = "CRAN", ...)
```

**Arguments**

file	character specifying a file path: for <code>read.ctv</code> a CRAN task view <code>‘.ctv’</code> file, and for <code>ctv2html</code> the corresponding output <code>‘.html’</code> file.
x	an object of class <code>"ctv"</code> as returned by <code>read.ctv</code> .
css	character specifying the path and name of the cascade style sheet that should be included in the HTML files.
packageURL	character specifying the path (relative to the view directory) to the package descriptions.
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 <code>‘Views.rds’</code> and <code>‘.html’</code> files should be installed. The <code>‘.ctv’</code> files should be located in the <code>‘web/views/’</code> directory.
...	further arguments passed to <code>available.packages</code> or <code>ctv2html</code> , respectively.

**Details**

CRAN Task views are generated from an XML-based format `‘.ctv’` that is described in the vignette of this package.

`read.ctv` can read a `‘.ctv’` file with a CRAN task view specification and returns an object of class `"ctv"`. This functions requires the availability of the **XML** package.

`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 `‘.ctv’` file are consistent with each other and whether all packages are available from the repository.

`repos_update_views` reads all `‘.ctv’` 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**

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

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

`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).
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.

## References

Zeileis A (2005). CRAN Task Views. *R News*, **5**(1), 39–40. <https://CRAN.R-project.org/doc/Rnews/>.

## See Also

[install.views](#)

## Examples

```
## read .ctv file
x <- read.ctv(system.file("ctv", "Econometrics.ctv", package = "ctv"))
x

## Not run:
## generate corresponding .html file
ctv2html(x)

## check packagelist
check_ctv_packages(x)

## End(Not run)
```

# Index

## \*Topic **utilities**

ctv-client, [2](#)

ctv-server, [3](#)

available.packages, [2](#), [3](#)

available.views (ctv-client), [2](#)

check\_ctv\_packages (ctv-server), [3](#)

CRAN.views, [4](#)

CRAN.views (ctv-client), [2](#)

ctv-client, [2](#)

ctv-server, [3](#)

ctv2html (ctv-server), [3](#)

download.packages, [2](#), [3](#)

download.views (ctv-client), [2](#)

install.packages, [2](#), [3](#)

install.views, [4](#), [5](#)

install.views (ctv-client), [2](#)

installed.packages, [3](#)

print.ctv (ctv-client), [2](#)

print.ctvlist (ctv-client), [2](#)

read.ctv (ctv-server), [3](#)

repos\_update\_views (ctv-server), [3](#)

update.views (ctv-client), [2](#)