

Package ‘checkpoint’

October 1, 2014

Title Install Packages from Snapshots on the Checkpoint Server for Reproducibility

Description The goal of checkpoint is to solve the problem of package reproducibility in R. Specifically, checkpoint allows you to install packages as they existed on CRAN on a specific snapshot date as if you had a CRAN time machine. To achieve reproducibility, the `checkpoint()` function installs the packages required or called by your project and scripts to a local library exactly as they existed at the specified point in time. Only those packages are available to your project, thereby avoiding any package updates that came later and may have altered your results. In this way, anyone using `checkpoint's` `checkpoint()` can ensure the reproducibility of your scripts or projects at any time. To create the snapshot archives, once a day (at midnight UTC) we refresh the Austria CRAN mirror, on the "Managed R Archived Network" server (<http://mran.revolutionanalytics.com/>). Immediately after completion of the rsync mirror process, we take a snapshot, thus creating the archive. Snapshot archives exist starting from 2014-09-17.

Version 0.3.2

Date 2014-10-01

Author Revolution Analytics

Maintainer Andrie de Vries <andrie@revolutionanalytics.com>

Copyright Revolution Analytics

License GPL-2

URL <https://github.com/RevolutionAnalytics/checkpoint>

BugReports <http://www.github.com/RevolutionAnalytics/checkpoint/issues>

Imports knitr

Suggests testthat(>= 0.9)

NeedsCompilation no

Repository CRAN

Date/Publication 2014-10-01 18:44:47

R topics documented:

| | |
|------------------------------|---|
| checkpoint-package | 2 |
| checkpoint | 2 |

| | |
|--------------|----------|
| Index | 5 |
|--------------|----------|

| | |
|--------------------|---|
| checkpoint-package | <i>Install packages from snapshots on the checkpoint server for reproducibility</i> |
|--------------------|---|

Description

The goal of checkpoint is to solve the problem of package reproducibility in R. Specifically, checkpoint allows you to install packages as they existed on CRAN on a specific snapshot date as if you had a CRAN time machine.

Details

To achieve reproducibility, the `checkpoint()` function installs the packages required or called by your project and scripts to a local library exactly as they existed at the specified point in time. Only those packages are available to your project, thereby avoiding any package updates that came later and may have altered your results. In this way, anyone using the `checkpoint` `checkpoint()` function can ensure the reproducibility of your scripts or projects at any time.

To create the snapshot archives, once a day (at midnight UTC) we refresh the Austria CRAN mirror, on the checkpoint server (<http://mran.revolutionanalytics.com/>). Immediately after completion of the `rsync` mirror process, we take a snapshot, thus creating the archive. Snapshot archives exist starting from 2014-09-17.

checkpoint exposes only a single function:

`checkpoint` Configures R session to use packages as they existed on CRAN at time of snapshot.

| | |
|------------|--|
| checkpoint | <i>Configures R session to use packages as they existed on CRAN at time of snapshot.</i> |
|------------|--|

Description

Together, the `checkpoint` package and the checkpoint server act as a CRAN time machine. The `checkpoint()` function installs the packages referenced in the specified project to a local library exactly as they existed at the specified point in time. Only those packages are available to your session, thereby avoiding any package updates that came later and may have altered your results. In this way, anyone using the `checkpoint` `checkpoint()` function can ensure the reproducibility of your scripts or projects at any time.

Usage

```
checkpoint(snapshotDate, project = getwd(), verbose = TRUE)
```

Arguments

| | |
|--------------|--|
| snapshotDate | Date of snapshot to use in YYYY-MM-DD format, e.g. "2014-09-17". Specify a date on or after "2014-09-17". MRAN takes one snapshot per day. |
| project | A project path. This is the path to the root of the project that references the packages to be installed from the MRAN snapshot for the date specified for snapshotDate. Defaults to current working directory using getwd() . |
| verbose | If TRUE, displays progress messages. |

Value

NULL. See the Details section for side effects.

Details

`checkpoint()` creates a local library into which it installs a copy of the packages required by your project as they existed on CRAN on the specified snapshot date. Your R session is updated to use only these packages.

To automatically determine all packages used in your project, the function scans all R code (`.R`, `.Rmd`, and `.Rpres` files) for `library()` and `requires()` statements.

Specifically, the function will:

- Create a new local snapshot library to install packages. This library folder is at `~/ .checkpoint`
- Update the options for your CRAN mirror and point to an MRAN snapshot using `options(repos)`
- Scan your project folder for all required packages and install them from the snapshot using [install.packages](#)

Resetting the checkpoint

To reset the checkpoint, simply restart your R session.

Examples

```
## Not run:  
  
# Create temporary project and set working directory  
  
example_project <- paste0("~/checkpoint_example_project_", Sys.Date())  
  
dir.create(example_project, recursive = TRUE)  
oldwd <- setwd(example_project)  
  
# Write dummy code file to project
```

```
cat("library(MASS)", "library(foreach)",
    sep="\n",
    file="checkpoint_example_code.R")

# Create a checkpoint by specifying a snapshot date

library(checkpoint)
checkpoint("2014-09-17")

# Check that CRAN mirror is set to MRAN snapshot
getOption("repos")

# Check that library path is set to ~/.checkpoint
.libPaths()

# Check which packages are installed in checkpoint library
installed.packages()

# cleanup
unlink(example_project, recursive = TRUE)
setwd(oldwd)

## End(Not run)
```

Index

*Topic **package**

checkpoint-package, 2

checkpoint, 2, 2

checkpoint-package, 2

getwd, 3

install.packages, 3