Package ‘checkpoint’

March 17, 2015

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 0.3.9

Date 2015-03-17

Author Revolution Analytics

Maintainer Andrie de Vries <andrie@revolutionanalytics.com>

Copyright Revolution Analytics

License GPL-2

URL http://projects.revolutionanalytics.com/documents/rrt/rrtpkgs/

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

Imports utils

Depends R(>= 3.1.1)

Suggests knitr, testthat(>= 0.9), MASS

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2015-03-17 22:42:49
**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` 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.

**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` function can ensure the reproducibility of your scripts or projects at any time.
Usage

```r
checkpoint(snapshotDate, project = getwd(), R.version,
          scanForPackages = TRUE, checkpointLocation = "~/",
          verbose = TRUE,
          use.knitr = system.file(package = "knitr") != "")
```

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()`.
- **R.version**: Optional character string, e.g. "3.1.2". If specified, compares the current `R.version` to the specified `R.version`. If these differ, stops processing with an error, making no changes to the system. Specifically, if the check fails, the library path is NOT modified. This argument allows the original script author to specify a specific version of R to obtain the desired results.
- **scanForPackages**: If TRUE, scans for packages in project folder (see details). If FALSE, skips the scanning process. A use case for `scanForPackages = FALSE` is to skip the scanning and installation process, e.g. in production environments with a large number of R scripts in the project. Only set `scanForPackages = FALSE` if you are certain that all package dependencies are already in the checkpoint folder.
- **checkpointLocation**: File path where the checkpoint library is stored. Default is "~/", i.e. the user's home directory. A use case for changing this is to create a checkpoint library on a portable drive (e.g. USB drive).
- **verbose**: If TRUE, displays progress messages.
- **use.knitr**: If TRUE, uses parses all Rmarkdown files using the knitr package.

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 `require()` statements. In addition, scans for occurrences of code that accesses functions in namespaces using package:::foo() and package:::foo(). Finally, any occurrences of the functions `setClass`, `setRefClass`, `setMethod` or `setGeneric` will also identify the methods package as a dependency.

Specifically, the function will:
- Create a new local snapshot library to install packages. By default this library folder is at ~/.checkpoint but you can modify the path using the checkpointLocation argument.
- 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**

```r
## 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)\nlibrary(foreach)\n", 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
  ::, 3
  :::, 3
  checkpoint, 2, 2
  checkpoint-package, 2
  getwd, 3
  install.packages, 4
  library, 3
  options, 4
  R.version, 3
  require, 3
  setClass, 3
  setGeneric, 3
  setMethod, 3
  setRefClass, 3