# The *checkpoint* Package

*7/24/2020*

## Purpose

Main purpose of the *checkpoint* package is **reproducibility in R**. This is achieved by setting up a **frozen repository**. To do this *checkpoint* modifies the R library path (pointing to a local directory) and changes the *repos* option to reference the specified MRAN snapshot.

The reproducibility issue is already solved on DaVinci, since MRAN snapshots are currently used there. Nevertheless DaVinci users can benefits from *checkpoint* usage, because the package provides some functionality that might be very useful.

## Expected Benefits

### All contributed R packages are stored in one place

*checkpoint* stores all contributed R packages in the *checkpoint* library path. By default this is a subdirectory of the users home directory (~/.checkpoint/).

But this behaviour can be modified and *checkpoint* advised to choose a different local directory, which is especially usefull when doing GxP related work where all used contributed packages have to be stored in a special repository like ClearCase.

Let us assume the following case:

- a DaVinci users work is GxP related and
- all packages used by the project have to be stored in ClearCase

In this situation the *checkpoint* package comes in handy. By using the `checkpoint()` function it is relatively easy to not only define which MRAN snapshot to use, but also to configure your library folder to be stored in your project directory. This is achieved by setting the `checkpointLocation` parameter:

```
library(checkpoint)
checkpoint('2019-11-22', checkpointLocation='~/project_dir/')
```

In this case it is also an advantage that the R library path is shrunk down to the location of your R installation and the *checkpoint* library folder only. All used packages (except base R and recommended packages) are then stored in your project directory and can be checked-in to ClearCase.

### R version pre-defined in your script

As a precaution even your R version can be determined when calling the `checkpoint()` function:

```
library(checkpoint)
checkpoint('2019-11-22', R.version='3.6.1', checkpointLocation='~/project_dir/')
```

This will result in an error message when you are not in your proper environment (for whatever reasons) or the environment has changed to a different R version.

## Possible Issues

### Packages are re-installed to your local *checkpoint* library

DaVinci provides many contributed packages. Using *checkpoint* even packages that are already installed on DaVinci will be re-installed to your *checkpoint* directory.

This has pros and cons.

#### PROs:

- only a **limited number of packages** namely the ones that are used in your project will be contained in your *checkpoint* library. Especially GxP related work where all used source code (including used packages) has to be checked in to ClearCase is then easy to reproduce. The only component that is then not checked in to ClearCase is the R version used. And even this information can be easily recorded (using the *R.version* parameter).

#### CONs:

- local re-installation of packages probably has to be covered by some kind of **installation testing** (even when the packages are available pre-installed on DaVinci)

### Default location of *checkpoint* library path

The default location for the *checkpoint* library path is the users home directory and there is no way for *checkpoint* find a library path that differs from the default.

## Background Information

### CRAN, MRAN and checkpoint

#### CRAN

CRAN follows a rolling release concept. R packages are always provided in the latest version.

This leads to issues:

- custom-built R scripts can break in the future when used packages are updated
- additional package installations can break dependencies of another already installed package
- results are not reproducible since the environment in which they were obtained changes over time

#### MRAN

The Microsoft R Application Network (MRAN) solves part of these issues by the creation of daily CRAN snapshots that are provided back to the R community as a service. Every day since September 2014 at midnight UTC a snapshot of the CRAN main server in Austria is rsynced to MRAN.

#### checkpoint

The `checkpoint` package makes it possible to install package versions from one of the MRAN specific date in the past referencing the MRAN snapshots.

## FAQ

### What snapshot dates are recommended?

A suitable snapshot should meet the following criteria:

- *availabiltiy* - there is not for every day a snapshot available on MRAN. Valid snapshot dates can be retrieved via `getValidSnapshots()`.
- *intended R version* - to prevent incompatibilities a snapshot for the used R version should be chosen.
- *latest bugfixes included* - in case the R version is fixed (like on DaVinci), it is reasonable to choose a later snapshot to include a maximum of package bugfixes available for your R version.

## What happens in the background

### Library Path

Using *checkpoint('2019-11-22')* the output of `.libPaths()` changes from:

```
R 3.6.1> .libPaths()
[1] "/CHBS/apps/EB/software/R-bundle-Novartis/0.2-gomkl-2019a-R-3.6.1"
[2] "/CHBS/apps/EB/software/R-bundle-Bioconductor/3.9-gomkl-2019a-R-3.6.1"
[3] "/CHBS/apps/EB/software/ncdf4/1.16.1-gomkl-2019a-R-3.6.1"
[4] "/CHBS/apps/EB/software/R/3.6.1-gomkl-2019a/lib64/R/library"
```

to:

```
R 3.6.1> .libPaths()
[1] "/home/schmis1m/.checkpoint/2019-12-12/lib/x86_64-pc-linux-gnu/3.6.1"
[2] "/home/schmis1m/.checkpoint/R-3.6.1"
[3] "/CHBS/apps/EB/software/R/3.6.1-gomkl-2019a/lib64/R/library"
```

That means the number of available packages has also changed from 1'530 to only 705 (plus user installed packages). By the way, the *checkpoint* package itself is now no longer available when running command `installed.packages()`.

### Package Repositories

Running the *checkpoint* function on DaVinci makes package repositories (command *options('repos')*) change from :

```
R 3.6.1> options('repos')
$repos
                                                  CRAN
      "https://cran.microsoft.com/snapshot/2019-11-22/"
                                               BioCsoft
           "https://bioconductor.org/packages/3.9/bioc"
                                                BioCann
"https://bioconductor.org/packages/3.9/data/annotation"
                                                BioCexp
"https://bioconductor.org/packages/3.9/data/experiment"
```

to:

```
R 3.6.1> options('repos')
$repos
[1] "https://mran.microsoft.com/snapshot/2019-12-12"
```

## Rollback with `unCheckpoint()`

Executing `unCheckpoint()` will restore the library path back to:

```
R 3.6.1> .libPaths()
[1] "/CHBS/apps/EB/software/R-bundle-Novartis/0.2-gomkl-2019a-R-3.6.1"
[2] "/CHBS/apps/EB/software/R-bundle-Bioconductor/3.9-gomkl-2019a-R-3.6.1"
[3] "/CHBS/apps/EB/software/ncdf4/1.16.1-gomkl-2019a-R-3.6.1"
[4] "/CHBS/apps/EB/software/R/3.6.1-gomkl-2019a/lib64/R/library"
```

But package repositories will continue to link to the MRAN snapshot only.

## Open Questions

### How can non-CRAN packages be installed reproducibly?

*checkpoint* internally uses `utils::install.packages()` for package installation. Can packages from other repositories (BioConductor, GitHub) also be installed in a reproducible way?

### Are a packages system requirements a problem?

Probably yes, if a new manually installed packages expects some system libraries to be available that have not already been installed by one of the DaVinci pre-installed packages.

## Notes

### checkpoint 1.0 ante portas

In April 2020 Hong Ooi, the package maintainer, announced checkpoint version 1.0 beta to be available on GitHub.