

marshal: Saving and Loading Objects that Otherwise Cannot be Saved or Exported to Parallel Workers

Student Name - Damodar Dhakad

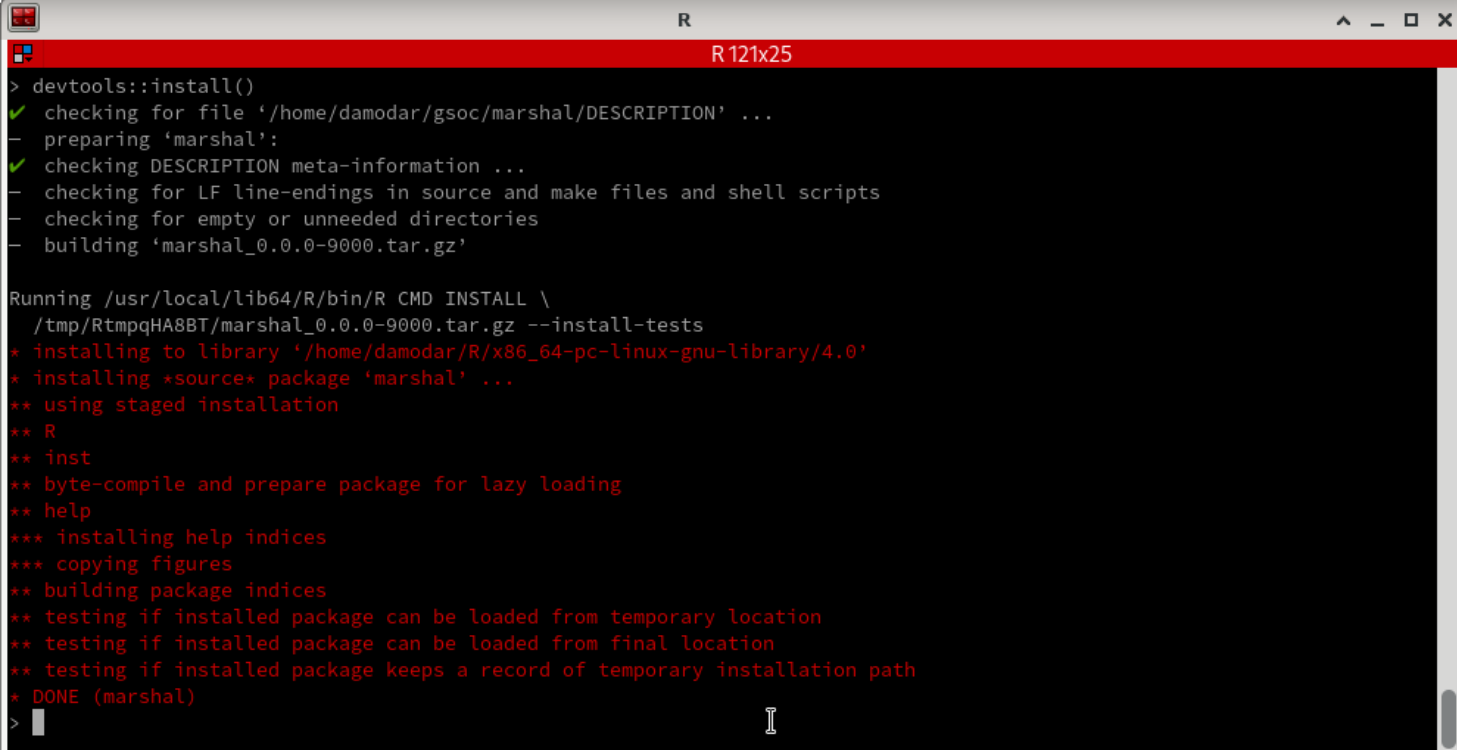
1. Easy Task :Git and R package testing

Fork the marshal repository and build the package and check it with both devtools::check() and R CMD check --as-cran.

The devtools way -

Each repository has its own way to install a package from them, devtools packages helps to simplify this process. For example `install_github()` use to install packages from GitHub.

To install Marshal package available at [GitHub](#), type `devtools::install_github("HenrikBengtsson/marshal")` onto R command prompt It will download the repository into temporary folder and proceed with installation. Other way is to cd into your self git clone repository and run `devtools::install()` Output will look like this -

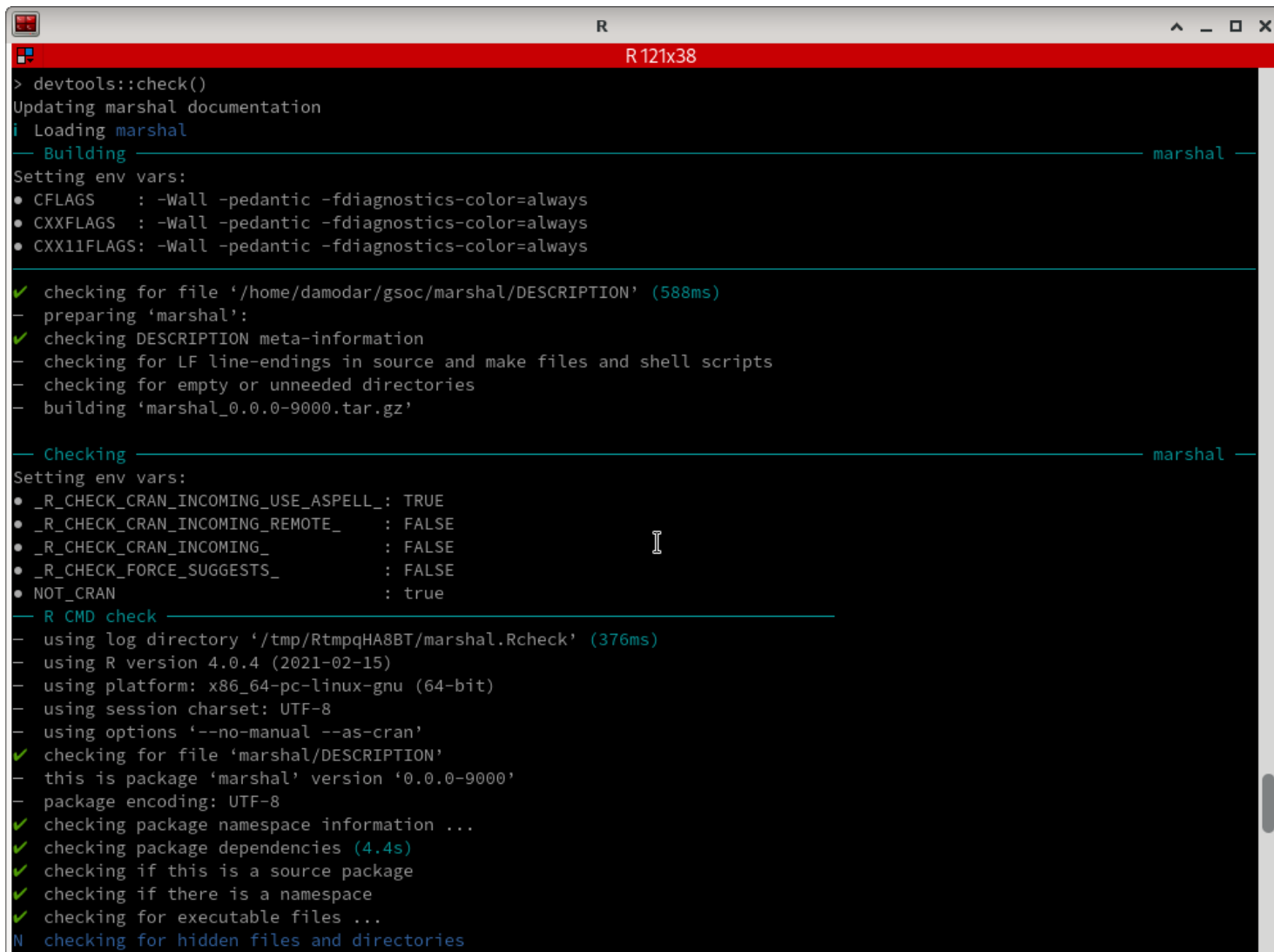


```
R
R 121x25
> devtools::install()
✓ checking for file '/home/damodar/gsoc/marshal/DESCRIPTION' ...
- preparing 'marshal':
✓ checking DESCRIPTION meta-information ...
- checking for LF line-endings in source and make files and shell scripts
- checking for empty or unneeded directories
- building 'marshal_0.0.0-9000.tar.gz'

Running /usr/local/lib64/R/bin/R CMD INSTALL \
/tmp/RtmpqHA8BT/marshal_0.0.0-9000.tar.gz --install-tests
* installing to library '/home/damodar/R/x86_64-pc-linux-gnu-library/4.0'
* installing *source* package 'marshal' ...
** using staged installation
** R
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
*** copying figures
** building package indices
** testing if installed package can be loaded from temporary location
** testing if installed package can be loaded from final location
** testing if installed package keeps a record of temporary installation path
* DONE (marshal)
> |
```

To run the test cases located at `marshal/test` we use `devtools::check()` function from `devtools` package.

Output will look like this -



```
R
R 121x38
> devtools::check()
Updating marshal documentation
i Loading marshal
— Building — marshal —
Setting env vars:
• CFLAGS      : -Wall -pedantic -fdiagnostics-color=always
• CXXFLAGS    : -Wall -pedantic -fdiagnostics-color=always
• CXX11FLAGS  : -Wall -pedantic -fdiagnostics-color=always

✓ checking for file ‘/home/damodar/gsoc/marshal/DESCRIPTION’ (588ms)
— preparing ‘marshal’:
✓ checking DESCRIPTION meta-information
— checking for LF line-endings in source and make files and shell scripts
— checking for empty or unneeded directories
— building ‘marshal_0.0.0-9000.tar.gz’

— Checking — marshal —
Setting env vars:
• _R_CHECK_CRAN_INCOMING_USE_ASPELL_: TRUE
• _R_CHECK_CRAN_INCOMING_REMOTE_    : FALSE
• _R_CHECK_CRAN_INCOMING_           : FALSE
• _R_CHECK_FORCE_SUGGESTS_          : FALSE
• NOT_CRAN                          : true

— R CMD check —
— using log directory ‘/tmp/RtmpqHA8BT/marshal.Rcheck’ (376ms)
— using R version 4.0.4 (2021-02-15)
— using platform: x86_64-pc-linux-gnu (64-bit)
— using session charset: UTF-8
— using options ‘--no-manual --as-cran’
✓ checking for file ‘marshal/DESCRIPTION’
— this is package ‘marshal’ version ‘0.0.0-9000’
— package encoding: UTF-8
✓ checking package namespace information ...
✓ checking package dependencies (4.4s)
✓ checking if this is a source package
✓ checking if there is a namespace
✓ checking for executable files ...
N checking for hidden files and directories
```

```
R
R 121x38

Found the following hidden files and directories:
.github
These were most likely included in error. See section 'Package
structure' in the 'Writing R Extensions' manual.
✓ checking for portable file names
✓ checking for sufficient/correct file permissions
✓ checking whether package 'marshal' can be installed (3.3s)
✓ checking installed package size ... OK
✓ checking package directory ...
✓ checking for future file timestamps (2.7s)
✓ checking DESCRIPTION meta-information (538ms)
N checking top-level files
Non-standard file/directory found at top level:
'Makefile'
✓ checking for left-over files
✓ checking index information
✓ checking package subdirectories ...
✓ checking R files for non-ASCII characters ...
✓ checking R files for syntax errors ...
✓ checking whether the package can be loaded ...
✓ checking whether the package can be loaded with stated dependencies ...
✓ checking whether the package can be unloaded cleanly ...
✓ checking whether the namespace can be loaded with stated dependencies ...
✓ checking whether the namespace can be unloaded cleanly ...
✓ checking loading without being on the library search path ...
✓ checking dependencies in R code ...
✓ checking S3 generic/method consistency (1.2s)
✓ checking replacement functions ...
✓ checking foreign function calls ...
✓ checking R code for possible problems (2.6s)
✓ checking Rd files ...
✓ checking Rd metadata ...
✓ checking Rd line widths ...
✓ checking Rd cross-references (348ms)
✓ checking for missing documentation entries ...
✓ checking for code/documentation mismatches (530ms)
✓ checking Rd \usage sections (1.3s)
✓ checking Rd contents ...
```

```
R
R 121x38
✓ checking foreign function calls ...
✓ checking R code for possible problems (2.6s)
✓ checking Rd files ...
✓ checking Rd metadata ...
✓ checking Rd line widths ...
✓ checking Rd cross-references (348ms)
✓ checking for missing documentation entries ...
✓ checking for code/documentation mismatches (530ms)
✓ checking Rd \usage sections (1.3s)
✓ checking Rd contents ...
✓ checking for unstated dependencies in examples ...
✓ checking line endings in Makefiles
✓ checking for GNU extensions in Makefiles
✓ checking include directives in Makefiles
- checking examples ... NONE
✓ checking for non-standard things in the check directory
✓ checking for detritus in the temp directory

See
  '/tmp/RtmpqHA8BT/marshal.Rcheck/00check.log'
for details.

----- R CMD check results ----- marshal 0.0.0-9000 -----
Duration: 21s

> checking for hidden files and directories ... NOTE
Found the following hidden files and directories:
  .github
These were most likely included in error. See section 'Package
structure' in the 'Writing R Extensions' manual.

> checking top-level files ... NOTE
Non-standard file/directory found at top level:
  'Makefile'

0 errors ✓ | 0 warnings ✓ | 2 notes ✗
> 
```

R CMD check way -

R CMD check runs all sorts of checks on the contents of an R package, and gives warnings and error messages when it finds things that aren't right. It also will run the examples in the .Rd files for each of your functions, as well as other automated tests that we've included. First we need to build package file which in this case by running `R CMD build marshal`. And then to verify the functionalities of package, run `R CMD check --as-cran marshal_0.0.0-9000.tar.gz`.

```
damodar@linux: ~/gsoc
damodar@linux: ~/gsoc 121x38
damodar@linux ~/gsoc (main) $ R CMD build marshal
* checking for file 'marshal/DESCRIPTION' ... OK
* preparing 'marshal':
* checking DESCRIPTION meta-information ... OK
* checking for LF line-endings in source and make files and shell scripts
* checking for empty or unneeded directories
* building 'marshal_0.0.0-9000.tar.gz'

damodar@linux ~/gsoc (main) $ R CMD check --as-cran marshal_0.0.0-9000.tar.gz
* using log directory '/home/damodar/gsoc/marshal.Rcheck'
* using R version 4.0.4 (2021-02-15)
* using platform: x86_64-pc-linux-gnu (64-bit)
* using session charset: UTF-8
* using option '--as-cran'
* checking for file 'marshal/DESCRIPTION' ... OK
* this is package 'marshal' version '0.0.0-9000'
* package encoding: UTF-8
* checking CRAN incoming feasibility ... NOTE
Maintainer: 'Henrik Bengtsson <henrikb@braju.com>'

New submission

Version contains large components (0.0.0-9000)
* checking package namespace information ... OK
* checking package dependencies ... OK
* checking if this is a source package ... OK
* checking if there is a namespace ... OK
* checking for executable files ... OK
* checking for hidden files and directories ... NOTE
Found the following hidden files and directories:
.github
These were most likely included in error. See section 'Package
structure' in the 'Writing R Extensions' manual.

CRAN-pack does not know about
.github
* checking for portable file names ... OK
* checking for sufficient/correct file permissions ... OK
```

```
damodar@linux: ~/gsoc
damodar@linux: ~/gsoc 121x38
* checking whether package 'marshal' can be installed ... OK
* checking installed package size ... OK
* checking package directory ... OK
* checking for future file timestamps ... OK
* checking DESCRIPTION meta-information ... OK
* checking top-level files ... NOTE
Files 'README.md' or 'NEWS.md' cannot be checked without 'pandoc' being installed.
Non-standard file/directory found at top level:
  'Makefile'
* checking for left-over files ... OK
* checking index information ... OK
* checking package subdirectories ... OK
* checking R files for non-ASCII characters ... OK
* checking R files for syntax errors ... OK
* checking whether the package can be loaded ... OK
* checking whether the package can be loaded with stated dependencies ... OK
* checking whether the package can be unloaded cleanly ... OK
* checking whether the namespace can be loaded with stated dependencies ... OK
* checking whether the namespace can be unloaded cleanly ... OK
* checking loading without being on the library search path ... OK
* checking use of S3 registration ... OK
* checking dependencies in R code ... OK
* checking S3 generic/method consistency ... OK
* checking replacement functions ... OK
* checking foreign function calls ... OK
* checking R code for possible problems ... OK
* checking Rd files ... OK
* checking Rd metadata ... OK
* checking Rd line widths ... OK
* checking Rd cross-references ... OK
* checking for missing documentation entries ... OK
* checking for code/documentation mismatches ... OK
* checking Rd \usage sections ... OK
* checking Rd contents ... OK
* checking for unstated dependencies in examples ... OK
* checking line endings in Makefiles ... OK
* checking for GNU extensions in Makefiles ... OK
* checking include directives in Makefiles ... OK
```

```
damodar@linux: ~/gsoc
damodar@linux: ~/gsoc 121x38
* checking for missing documentation entries ... OK
* checking for code/documentation mismatches ... OK
* checking Rd \usage sections ... OK
* checking Rd contents ... OK
* checking for unstated dependencies in examples ... OK
* checking line endings in Makefiles ... OK
* checking for GNU extensions in Makefiles ... OK
* checking include directives in Makefiles ... OK
* checking examples ... NONE
* checking PDF version of manual ... WARNING
LaTeX errors when creating PDF version.
This typically indicates Rd problems.
LaTeX errors found:
! LaTeX Error: File `inconsolata.sty' not found.

Type X to quit or <RETURN> to proceed,
or enter new name. (Default extension: sty)

! Emergency stop.
<read *>

l.287 ^^M

! ==> Fatal error occurred, no output PDF file produced!
* checking PDF version of manual without hyperrefs or index ... OK
* checking for non-standard things in the check directory ... NOTE
Found the following files/directories:
  'marshal-manual.tex'
* checking for detritus in the temp directory ... OK
* DONE

Status: 1 WARNING, 4 NOTES
See
  '/home/damodar/gsoc/marshal.Rcheck/00check.log'
for details.

damodar@linux ~/gsoc (main) $
```

Reference -

https://kbroman.org/pkg_primer/pages/build.html

https://kbroman.org/pkg_primer/pages/check.html

2. Easy: Prototyping in R

2.a Write `XML_serialize_refhook()` and `XML_unserialize_refhook()` functions that are simple wrappers that calls `XML::xmlSerializeHook()` and `XML::xmlDeserializeHook()` internally such that they can be used in place of the latter two function.

2.b Write a package help page with an example illustrating how to use your new functions for saving and reading XML objects to file.

2.c Try to make the package passes R CMD check will all OKs.