Inserting references in Rd and roxygen2 documentation

Georgi N. Boshnakov

Abstract

Package **Rdpack** provides a straightforward way to insert BibTeX references in the documentation of an R package, using the RdMacros feature introduced in R 3.2.0. It works for 'roxygen2' documentation, as well.

This vignette is part of package Rdpack, version 2.6.1.

Keywords: bibliographic references, Rd, bibtex, citations, R.

1. Setup

To prepare a package for importing BibTeX references it is necessary to tell the package management tools that package **Rdpack** and its Rd macros are needed. The references should be put in file `inst/REFERENCES.bib`. These steps are enumerated below in somewhat more detail for convenince:

1. Add the following lines to file 'DESCRIPTION':

   ```
   Imports: Rdpack
   RdMacros: Rdpack
   ```

   Make sure the capitalisation of `RdMacros` is as shown. If the field 'RdMacros' is already present, add 'Rdpack' to the list on that line. Similarly for field 'Imports'.

2. Add the following line to file 'NAMESPACE'[1]:

   ```
   importFrom(Rdpack,reprompt)
   ```

   The equivalent line for 'roxygen2' is

   ```
   #' @importFrom Rdpack reprompt
   ```

3. Create file `REFERENCES.bib` in subdirectory `inst/` of your package and put the bibtex references in it.

2. Inserting references in package documentation

Once the steps outlined in the previous section are done, references can be inserted in the documentation as `\insertRef{key}{package}`, where `key` is the bibtex key of the reference and `package` is your package. This works in Rd files and in roxygen documentation chunks.

---

[1] Any function for package **Rdpack** will do. This is to avoid getting a warning from 'R CMD check'.

In fact, argument 'package' can be any installed R package[2], not necessarily the current one. This means that you don't need to copy references from other packages to your `"REFERENCES.bib"` file. This works for packages that have `"REFERENCES.bib"` in their installation directory and for the default packages. See also the help pages `?Rdpack::insertRef` and `?Rdpack::Rdpack-package`. For example, the help page `?Rdpack::insertRef` contains the following lines in section "References" of the Rd file:

```
\insertRef{Rpack:bibtex}{Rdpack}
```

The first line above inserts the reference labeled `Rpack:bibtex` in Rdpack's `REFERENCES.bib`. The second line inserts the reference labeled `R` in file `REFERENCES.bib` in package 'bibtex'.

A roxygen2 documentation chunk might look like this:

```
#' @references
#' \insertRef{Rpack:bibtex}{Rdpack}
```

3. Inserting citations

From version 0.7 of **Rdpack**, additional Rd macros are available for citations[3]. They can be used in both Rd and roxygen2 documentation. If you are using these, it will be prudent to require at least this version of Rdpack in the "Imports:" directive in file DESCRIPTION: `Rdpack (>= 0.7)`.

3.1. Macros for citations

`\insertCite{key}{package}` cites the key and records it for use by `\insertAllCited{}`, see below. The style of the citations is author-year. The "et al" convention is used when there are more than two authors[4].

`key` can contain more keys separated by commas. Here are some examples (on the left is the code in the documentation chunk, on the right the rendered citation):

Documentation source	rendered
`\insertCite{parseRd}{Rdpack}`	(Murdoch 2010)
`\insertCite{Rpack:bibtex}{Rdpack}`	(Francois 2014)
`\insertCite{parseRd,Rpack:bibtex}{Rdpack}`	(Murdoch 2010; Francois 2014)

By default the citations are parenthesised `\insertCite{parseRd}{Rdpack}` produces (Murdoch 2010), as in the examples above. To get textual citations, like Murdoch (2010), put the string `;textual` at the end of the key. Here are the examples from the table above, rendered as textual citations:

Documentation source	rendered
`\insertCite{parseRd;textual}{Rdpack}`	Murdoch (2010)
`\insertCite{Rpack:bibtex;textual}{Rdpack}`	Francois (2014)
`\insertCite{parseRd,Rpack:bibtex;textual}{Rdpack}`	Murdoch (2010); Francois (2014)

The last line in the table demonstrates that this also works with several citations.

To mix the citations with other text, such as "see also" and "chapter 3", write the list of keys as a free text, starting it with the symbol `@` and prefixing each key with it. The `@` symbol will not appear in the output. For example, the following code:

---

[2]There is of course the risk that the referenced entry may be removed from the other package. So this is probably only useful for one's own packages. Also, the other package would better be one of the packages mentioned in DESCRIPTION.}

[3]They were introduced in the development version 0.6-1, but 0.7 is the first version with them released on CRAN.

[4]This feature was introduced in Rdpack 0.8-2.

```
\insertCite{@see also @parseRd and @Rpack:bibtex}{Rdpack}
\insertCite{@see also @parseRd; @Rpack:bibtex}{Rdpack}
\insertCite{@see also @parseRd and @Rpack:bibtex;textual}{Rdpack}
```

produces:

Rendered
(see also Murdoch 2010 and Francois 2014)
(see also Murdoch 2010; Francois 2014)
see also Murdoch (2010) and Francois (2014)

The text of the arguments of the macro in this free form should have no markup. For example, if you want to put the phrase `see also` in italic, enclosing it with `\emph{...}` (in Rd) or the equivalent `_..._` (in markdown) will not work[5]. For textual citations a workaround is to invoke `\insertCite` for each key and type the markup outside the macro arguments. For parenthetical citations the solutions is to ask `\insertCite` to omit the parentheses by putting `;nobrackets` at the end of the argument[6]. The parentheses can then be put manually where needed. For example,

```
(\emph{see also} \insertCite{@@parseRd and @Rpack:bibtex;nobrackets}{Rdpack})
```

produces: (*see also* Murdoch 2010 and Francois 2014).

Further macros for citations

The macro `\insertNoCite{key}{package}` records one or more references for `\insertAllCited` but does not cite it. Setting `key` to `*` will record all references from the specified package. For example, `\insertNoCite{R}{bibtex}` records the reference whose key is `R`, while `\insertNoCite{*}{utils}` records all references from package "utils" for inclusion by `\insertAllCited`.

`\insertCiteOnly{key}{package}` is as `\insertCite` but does not record the key for the list of references assembled by `\insertAllCited`.

3.2. Automatically generating lists of references

The macro `\insertAllCited{}` can be used to insert all references cited with `\insertCite` or `\insertNoCite`. A natural place to put this macro is the references section. The Rd section may look something like:

```
\references{
  \insertAllCited{}
}
```

The analogous documentation chunk in roxygen2 might look like this:

```
#' @references
#'   insertAllCited{}
```

Don't align the backslash with the second 'e' of `@references`, since roxygen2 may interpret it as verbatim text, not macro.

Rd macro `\insertCited{}` works like `\insertAllCited` but empties the references list after finishing its work. This means that the second and subsequent `\insertCited` in the same help page will list only citations done since the preceding `\insertCited`. Prompted by issue 27 on github to allow separate references lists for each method and the class in R6 documentation.

---

[5]For details see Github issue #23 raised by Martin R. Smith.

[6]With **Rdpack** versions greater than 2.1.3.

3.3. Changing the style of references

Package `Rdpack` supports bibliography styles for lists of references[7]. Currently the only alternative offered is to use long names (Georgi N. Boshnakov) in place of the default style (Boshnakov GN). More comprehensive alternatives can be included if needed or requested.

To cause all lists of references produced by `\insertAllCited` in a package to appear with full names, add `.onLoad()` function to your package. If you don't have `.onLoad()`, just copy the following definition:

```
> .onLoad <- function(lib, pkg){
+     Rdpack::Rdpack_bibstyles(package = pkg, authors = "LongNames")
+     invisible(NULL)
+ }
```

If you already have `.onLoad()`, add the line containing the `Rdpack::Rdpack_bibstyles` call to it.

After installling/reloading your package the lists of references should appear with long author names. "Rdpack" itself now uses this style.

4. Possible issues

4.1. Warning from 'R CMD build'

If 'R CMD build' or `devtools::build()` gives a warning along the lines of:

```
Warning: C:/temp/RtmpqWQqji/.../XXX.Rd:52: unknown macro '\insertRef'
```

then check the syntax in file DESCRIPTION — the most common cause of this is misspelling `RdMacros:`. Make sure in particular that 'M' is uppercase.

4.2. Development using 'devtools'

The described procedure works transparently in 'roxygen2' chunks and with Hadley Wickham's 'devtools'. Packages are built and installed properly with the 'devtools' commands and the references are processed as expected.

Currently (2017-08-04) if you run help commands `?xxx` for functions from the package you are working on and their help pages contain references, you may encounter some puzzling warning messages in 'developer' mode, something like:

```
1: In tools::parse_Rd(path) :
  ~/mypackage/man/abcde.Rd: 67: unknown macro '\insertRef'
```

These warnings are again about unknown macros but the reason is completely different: they pop up because "devtools" reroutes the help command to process the developer's Rd sources (rather than the documentation in the installed directory) but doesn't tell `parse_Rd` where to look for additional macros[8].

These warnings are harmless - the help pages are built properly and no warnings appear outside "developer" mode, e.g. in a separate R~session. You may also consider using the function `viewRd()`, discussed below, for viewing Rd files.

---

[7]Support for styles is available since `Rdpack (>= 0.8)`.

[8]The claims in this sentence can be deduced entirely from the informative message. Indeed, (1)~the error is in processing a source Rd file in the development directory of the package, and (2)~the call to `\parse_Rd` specifies only the file.

4.3. Latex markup in BibTeX entries

In principle, BibTeX entries may contain arbitrary Latex markup, while the Rd format supports only a subset. As a consequence, some BibTeX entries may need some editing when included in REFERENCES.bib[9]. Only do this for entries that do not render properly or cause errors, since most of the time this should not be necessary. For encoding related issues of REFERENCES.bib see the dedicated subsection below.

If mathematics doesn't render properly replace the Latex dollar syntax with Rd's `\eqn`, e.g. `$x^2$` with `\eqn{x^2}`. This should not be needed for versions of Rdpack 0.8-4 or later.

Some Latex macros may have to be removed or replaced with suitable Rd markup. Again, do this only if they cause problems, since some are supported, e.g. `\doi`.

See also the overview help page, `help("Rdpack-package")`, of **Rdpack**. Among other things, it contains some dummy references which illustrate the above.

4.4. Encoding of file REFERENCES.bib

If a package has a declared encoding (in file `DESCRIPTION`), `REFERENCES.bib` is read-in with that encoding[10]. Otherwise, the encoding of `REFERENCES.bib` is assumed to be UTF-8 (which includes ASCII as a subset).

Note that BibTeX entries downloaded from online databases and similar sources may contain unexpected characters in other encodings, e.g. 'latin1'. In such cases the check tools in R-devel (since about 2018-10-01) may give warnings like:

```
prepare_Rd: input string 1 is invalid in this locale
```

To resolve this, convert the file to the declared encoding or UTF-8. Alternatively, replace the offending symbols with their classic TeX/LaTeX equivalents (which are ASCII). Non-ASCII symbols in BibTeX entries obtained from online databases are often in fields like "Abstract", which are normally not included in lists of references and can be deleted from REFERENCES.bib.

One way to check for non-ASCII symbols in a file is `tools::showNonASCIIfile()`.

Since **Rdpack** switched to the bibtex parser in package **rbibutils**, if the bib file contains Latex escape sequences standing for accented Latin characters, such as `\'e` and `\"o`, they are imported as is. They are converted to UTF-8 only when the text is rendered for output. If R's checking tools complain about non-ASCII characters add the following encoding declaration to file DESCRIPTION[11]:

```
Encoding: UTF-8
```

Needless to say, make sure that there are really no characters from encodings like 'latin1'.

With the previous bibtex parser (before Rdpack version 1.1.0) the conversion was done earlier, which resulted in confusing messages about non-ASCII characters, even when the file REFERENCES.bib was pure ASCII. This should no longer happen.

5. Viewing Rd files

A function, `viewRd`, to view Rd files in the source directory of a package was introduced in version 0.4-23 of **Rdpack**. A typical user call would look something like:

---

[9]Thanks to Michael Dewey for suggesting the discussion of this.

[10]From `Rdpack (>=0.9-1)` The issue of not handling the encoding was raised by Professor Brian Ripley.

[11]Admittedly, this is not ideal since the user should not need to care how things are processed internally but I haven't pinpointed the exact cause for this.

```
Rdpack::viewRd("./man/filename.Rd")
```

By default the requested help page is shown in text format. To open the page in a browser, set argument `type` to `"html"`:

```
Rdpack::viewRd("./man/filename.Rd", type = "html")
```

Users of 'devtools' can use `viewRd()` in place of `help()` to view Rd sources[12].

Affiliation:

Georgi N. Boshnakov
School of Mathematics
The University of Manchester
Oxford Road, Manchester M13 9PL, UK
URL: http://www.maths.manchester.ac.uk/~gb/

---

[12]Yes, your real sources are the `*.R` files but `devtools::document()` transfers the roxygen2 documentation chunks to Rd files (and a few others), which are then rendered by **R** tools.