

Who Did What? The Roles of R Package Authors and How to Refer to Them

Hornik, Kurt; Murdoch, Duncan; Zeileis, Achim

DOI:

[10.57938/16b64daf-2e02-4beb-a3fd-3aebc8dafdfd](https://doi.org/10.57938/16b64daf-2e02-4beb-a3fd-3aebc8dafdfd)

Published: 01/11/2011

Document Version

Publisher's PDF, also known as Version of record

[Link to publication](#)

Citation for published version (APA):

Hornik, K., Murdoch, D., & Zeileis, A. (2011). *Who Did What? The Roles of R Package Authors and How to Refer to Them*. Research Report Series / Department of Statistics and Mathematics No. 114
<https://doi.org/10.57938/16b64daf-2e02-4beb-a3fd-3aebc8dafdfd>

Who Did What? The Roles of R Package Authors and How to Refer to Them

Kurt Hornik, Duncan Murdoch, Achim Zeileis

Research Report Series
Report 114, November 2011

Institute for Statistics and Mathematics
<http://statmath.wu.ac.at/>



Who Did What? The Roles of R Package Authors and How to Refer to Them

Kurt Hornik
WU Wirtschafts-
universität Wien

Duncan Murdoch
University of
Western Ontario

Achim Zeileis
Universität Innsbruck

Abstract

Computational infrastructure for representing persons and citations has been available in R for several years, but has been restructured through enhanced classes `"person"` and `"bibentry"` in recent versions of R. The new features include support for the specification of the roles of package authors (e.g., maintainer, author, contributor, translator, etc.) and more flexible formatting/printing tools among various other improvements. Here, we introduce the new classes and their methods and indicate how this functionality is employed in the management of R packages. Specifically, we show how the authors of R packages can be specified along with their roles in package `'DESCRIPTION'` and/or `'CITATION'` files and the citations produced from it.

Keywords: persons, roles, bibliography, citations, bibliometrics, `BIBTEX`, R.

1. Introduction

R packages are the result of scholarly activity and as such constitute scholarly resources which must be clearly identifiable for the respective scientific communities and, more generally, today's information society. In particular, packages published by standard repositories can be regarded as *reliable sources* which can and should be referenced (i.e., cited) by scientific works such as articles or other packages. This requires conceptual frameworks and computational infrastructure for describing bibliographic resources, general enough to encompass the needs of communities with an interest in R. These needs include support for exporting bibliographic metadata in standardized formats such as `BIBTEX` (Berry and Patashnik 2010), but also facilitating bibliometric analyses and investigations of the social fabric underlying the creation of scholarly knowledge.

The latter require a richer vocabulary than commonly employed by reference management software such as `BIBTEX`, identifying *persons* and their *roles* in relation to bibliographic resources. E.g., a thesis typically has an author and advisors. Software can have an (original) author and a translator to another language (such as from S to R). The maintainer of an R package is not necessarily an author.

In this paper, we introduce the base R infrastructure for representing and manipulating such scholarly data: objects of class `"person"` (hereafter, "person objects") hold informations about persons, possibly including their roles; objects of class `"bibentry"` (hereafter, "bibentry objects") hold bibliographic information in enhanced `BIBTEX` style, ideally using person objects

when referring to persons (such as authors or editors). Furthermore, we indicate how this functionality is employed in the management of R packages, in particular in their ‘CITATION’ and ‘DESCRIPTION’ files.

```
## specifying the authors of the "boot" package
R> p <- c(
+   person("Angelo", "Canty", role = "aut", comment =
+     "S original, http://statwww.epfl.ch/davison/BMA/library.html"),
+   person(c("Brian", "D."), "Ripley", role = c("aut", "trl", "cre"),
+     comment = "R port", email = "ripley@stats.ox.ac.uk"))
R> p

[1] "Angelo Canty [aut] (S original, http://statwww.epfl.ch/davison/BMA/library.html)"
[2] "Brian D. Ripley <ripley@stats.ox.ac.uk> [aut, trl, cre] (R port)"

## subsetting
R> p[-1]

[1] "Brian D. Ripley <ripley@stats.ox.ac.uk> [aut, trl, cre] (R port)"

## extracting information (as list for multiple persons)
R> p$given

[[1]]
[1] "Angelo"

[[2]]
[1] "Brian" "D."

## extracting information (as vector for single person)
R> p[2]$given

[1] "Brian" "D."

## flexible formatting
R> format(p, include = c("family", "given", "role"),
+   braces = list(family = c("", ","), role = c("[Role(s): ", "]")),
+   collapse = list(role = "/"))

[1] "Canty, Angelo [Role(s): aut]"          "Ripley, Brian D. [Role(s): aut/trl/cre]"

## formatting for BibTeX
R> toBibtex(p)

[1] "Angelo Canty and Brian D. Ripley"

## alternative BibTeX formatting "by hand"
R> paste(format(p, include = c("family", "given"),
+   braces = list(family = c("", ","))), collapse = " and ")

[1] "Canty, Angelo and Ripley, Brian D."
```

Table 1: Example for `person()`, specifying the authors of the `boot` package.

2. Persons and their roles

Person objects can hold information about an arbitrary positive number of persons. These can be obtained by one call to `person()` with list arguments, or by first creating objects representing single persons and combining these via `c()`.

Every person has at least one *given* name, and natural persons typically (but not necessarily, [Wikipedia 2011](#)) have a *family* name. These names are specified by arguments `given` and `family`, respectively. For backward compatibility with versions of R prior to 2.12.0, there are also arguments `first`, `middle` and `last` with obvious (Western-centric) meaning; however, these are deprecated as not universally applicable: some cultures place the given after the family name, or use no family name.

An optional `email` argument allows specifying persons' email addresses, which may serve as unique identifiers for the persons.

Whereas person names by nature may be arbitrary non-empty character strings, the state of the art in library science is to employ standardized vocabularies for person roles. R uses the "Relator and Role" codes and terms (<http://www.loc.gov/marc/relators/relaterm.html>) from MARC (MACHine-Readable Cataloging, [Library of Congress 2011](#)), one of the leading collections of standards used by librarians to encode and share bibliographic information. Argument `role` specifies the roles using the MARC codes; if necessary, the `comment` argument can be used to provide (human-readable) character strings complementing the standardized role information.

As an example (see [Table 1](#) and also `?person`), consider specifying the authors of package `boot` ([Canty and Ripley 2011](#)): Angelo Canty is the first author ("`aut`") who wrote the S original; Brian D. Ripley is the second author ("`aut`"), who translated the code to R ("`trl`"), and maintains the package ("`cre`").

Note that the MARC definition of the relator term "Creator" is "Use for a person or organization responsible for the intellectual or artistic content of a work.", which maps nicely to the notion of the maintainer of an R package. In general, while all MARC relator codes are supported, the following usage is suggested when giving the roles of persons in the context of authoring R packages:

`"aut"` (Author): Full authors who have made substantial contributions to the package and should show up in the package citation.

`"com"` (Compiler): Package maintainers who collected code (potentially in other languages) but did not make further substantial contributions to the package.

`"ctb"` (Contributor): Authors who have made smaller contributions (such as code patches etc.) but should not show up in the package citation.

`"cph"` (Copyright holder): Copyright holders.

`"cre"` (Creator): Package maintainer.

`"ths"` (Thesis advisor): Thesis advisor, if the package is part of a thesis.

`"trl"` (Translator): Translator to R, if the R code is a translation from another language (typically S).

Person objects can be subscripted by field (using `$`) or by position (using `[]`), and allow for flexible formatting and printing. By default, email, role and comment entries are suitably “embraced” by enclosing into angle brackets, square brackets and parentheses, respectively, and suitably collapsed. There is also a `toBibtex()` method which creates a suitable `BIBTEX` representation. Finally, the default method for `as.person()` is capable of inverting the default person formatting, and also can handle formatted person entries collapsed by comma or ‘and’ (with appropriate white space). See Table 1 and `?person` for various illustrative usages of these methods.

Starting from R 2.14.0, (R code specifying) person objects can also be used in the `Authors@R` field of ‘DESCRIPTION’ files. This information is used by the R package management system (specifically, `R CMD build` and `R CMD INSTALL`) to create the `Author` and `Maintainer` fields from `Authors@R`, unless these are still defined (statically) in ‘DESCRIPTION’. Also, the person information is used for reliably creating the author information in auto-generated package citations (more on this later). Note that there are no default roles for `Authors@R`: all authors must be given an author role (“aut”), and the maintainer must be identified by the creator role (“cre”).

3. Bibliographic information

Bibentry objects can hold information about an arbitrary positive number of bibliography entries. These can be obtained by one call to `bibentry()` with list arguments, or by first creating objects representing single bibentries and combining these via `c()`.

Bibentry objects represent bibliographic information in “enhanced `BIBTEX` style”, using the same entry types (such as ‘Article’ or ‘Book’) and field names (such as ‘author’ and ‘year’) as `BIBTEX`. A single bibentry is created by calling `bibentry()` with arguments `bibtype` (typically used positionally) and `key` (optionally) giving the type of the entry and a key for it, and further fields given in `tag = value` form, with `tag` and `value` giving the name and value of the field, respectively. See `?bibentry` for details on types and fields. The author and editor field values can be person objects as discussed above (and are coerced to such using `as.person()` if they are not). In addition, it is possible to specify header and footer information for the individual entries (arguments `header` and `footer`) and the collection of entries (arguments `mheader` and `mfooter`).

Bibentry objects can be subscripted by field (using `$`) or by position (using `[]`), and have `print()` and `format()` methods providing a choice between at least six different styles (as controlled by a `style` argument): plain text, HTML, `LATEX`, `BIBTEX`, R (added in R 2.14.0 for formatting bibentries as (high-level) R code), and citation information (including headers/footers and a mixture of plain text and `BIBTEX`). The first three make use of a `.bibstyle` argument using the `bibstyle()` function in package `tools` which defines the display style of the citation by providing a collection of functions to format the individual parts of the entry. Currently the default style “JSS”, which corresponds to the bibliography style employed by the *Journal of Statistical Software* (<http://www.jstatsoft.org/>), is the only available style, but it can be modified by replacing the component functions, or replaced with a completely new style. A style is implemented as an environment, and is required to contain functions to format each type of bibliographic entry (`formatArticle()`, `formatBook()`, etc.) as well a function `sortKeys()` to sort entries. The “JSS” style includes several dozen other functions to imple-

```
R> b <- c(
+   bibentry(
+     bibtype = "Manual",
+     title = "{boot}: Bootstrap {R} ({S-Plus}) Functions",
+     author = p,
+     year = "2011",
+     note = "R package version 1.3-3",
+     url = "http://CRAN.R-project.org/package=boot",
+     key = "boot-package"
+   ),
+
+   bibentry(
+     bibtype = "Book",
+     title = "Bootstrap Methods and Their Applications",
+     author = "Anthony C. Davison [aut], David V. Hinkley [aut]",
+     year = "1997",
+     publisher = "Cambridge University Press",
+     address = "Cambridge",
+     isbn = "0-521-57391-2",
+     url = "http://statwww.epfl.ch/davison/BMA/",
+     key = "boot-book"
+   )
+ )
R> b
```

Canty A and Ripley BD (2011). `_boot: Bootstrap R (S-Plus) Functions_`. R package version 1.3-3, <URL: <http://CRAN.R-project.org/package=boot>>.

Davison AC and Hinkley DV (1997). `_Bootstrap Methods and Their Applications_`. Cambridge University Press, Cambridge. ISBN 0-521-57391-2, <URL: <http://statwww.epfl.ch/davison/BMA/>>.

```
## formatting for BibTeX
```

```
R> toBibtex(b[1])
```

```
@Manual{boot-package,
  title = {{boot}: Bootstrap {R} ({S-Plus}) Functions},
  author = {Angelo Canty and Brian D. Ripley},
  year = {2011},
  note = {R package version 1.3-3},
  url = {http://CRAN.R-project.org/package=boot},
}
```

Table 2: Example for `bibentry()`, specifying citations for the `boot` package (i.e., the package itself and the book which it implements).

ment the required ones. Users modifying it should read the help page `?tools::bibstyle`, which includes an example that changes the sort order. Those implementing a completely new style should consult the source code.

Table 2 shows an illustrative example for using `bibentry()`: A vector with two bibentries is created for the `boot` package and the book whose methods are implemented. In the first bibentry the person object `p` from Table 1 is reused while the second bibentry applies `as.person()` to a string with author and role information. Subsequently, the bibentry is printed in "JSS" style and transformed to `BIBTEX`. Note that the roles of the persons in the author field are *not* employed in `toBibtex()` as `BIBTEX` does not support that concept. (Hence, the role information in the `as.person()` call creating `b[2]` would not have been necessary for producing `BIBTEX` output.)

Currently, the primary use of bibentry objects is in package ‘CITATION’ files, which function `citation()` collects into one bibentry object for creating package citations, primarily for visual inspection and export to `BIBTEX`. (Technically, `citation()` creates an object of class "citation" inheriting from "bibentry", for which the `print()` method uses a different default style.)

Package citations can also be auto-generated from the package ‘DESCRIPTION’ metadata, see Table 3 for an example: The bibentry title field is generated from the package metadata fields `Package` and `Title` which should be capitalized (in title case), and not use markup or end in a period. (Hence, it may need manual touch-ups, e.g., protecting `{S}` or `{B}ayesian` etc.) The year is taken from `Date/Publication` (for CRAN packages) or `Date` (if available), and the note field is generated from the package’s `Version`. If the package was installed from CRAN, the official stable CRAN package URL (here: <http://CRAN.R-project.org/package=boot>)¹ is given in the `url` field. Finally, the bibentry author field is generated from the description `Author` field (unless there is an `Authors@R`, see below). As the `Author` field in ‘DESCRIPTION’ is a plain text field intended for human readers, but not necessarily for automatic processing, it may happen that the auto-generated `BIBTEX` is incorrect (as in Table 3). Hence, `citation()` provides an ATTENTION note to this end. For R at least 2.14.0, one can overcome this problem by providing an `Authors@R` field in ‘DESCRIPTION’, from which the names of all persons with author roles can reliably be determined and included in the author field. Note again that author roles ("aut") are not “implied” as defaults, and must be specified explicitly.

As illustrated in Table 3, one can inspect the auto-generated package citation bibentry by giving `auto = TRUE` when calling `citation()` on a package – even if the package provides a ‘CITATION’ file in which case `auto` defaults to `FALSE`. The motivation is that it must be possible to refer to the package *itself*, even if the preferred way of citing the package in publications is *via* the references given in the package ‘CITATION’ file. Of course, both approaches might also be combined as in the `boot` package where one of the bibentries in ‘CITATION’ corresponds essentially to the auto-generated package bibentry (see `citation("boot")`). To facilitate incorporating the auto-generated citations into ‘CITATION’, R at least 2.14.0 allows to include a `citation(auto = meta)` entry: when evaluating the code in the ‘CITATION’ file, `citation()` automatically sets `meta` to the package metadata. This eliminates the need for more elaborate programmatic constructions of the package citation bibentry (or even worse, manually duplicating the relevant package metadata), provided that auto-generation is reliable (i.e., correctly processes author names and does not need additional `LATEX` markup or protection in the title). To check whether this is the case for a particular package’s metadata, constructs such as `citation(auto = packageDescription("boot"))` can be used.

¹ Not that the current physical URLs, e.g., <http://CRAN.R-project.org/web/packages/boot/index.html> are not guaranteed to be stable and may change in the future (as they have in the past).

```
R> citation("boot", auto = TRUE)
```

To cite package 'boot' in publications use:

```
S original by Angelo Canty <cantya@mcmaster.ca>. R port by Brian Ripley
<ripley@stats.ox.ac.uk>. Note that maintainers are not available to give
advice on using a package they did not author. (2011). boot: Bootstrap
Functions (originally by Angelo Canty for S). R package version 1.3-3.
http://CRAN.R-project.org/package=boot
```

A BibTeX entry for LaTeX users is

```
@Manual{,
  title = {boot: Bootstrap Functions (originally by Angelo Canty for S)},
  author = {S original by Angelo Canty <cantya@mcmaster.ca>. R port by Brian
    Ripley <ripley@stats.ox.ac.uk>. Note that maintainers are not available
    to give advice on using a package they did not author.},
  year = {2011},
  note = {R package version 1.3-3},
  url = {http://CRAN.R-project.org/package=boot},
}
```

ATTENTION: This citation information has been auto-generated from the package DESCRIPTION file and may need manual editing, see 'help("citation")' .

Table 3: Citation for the **boot** package auto-generated from its ‘DESCRIPTION’ file. (Note that this is not the recommended citation of the package authors, see `citation("boot")` without `auto = TRUE` for that.)

4. Actions for package maintainers

Package maintainers should provide an `Authors@R` entry in their ‘DESCRIPTION’ files to allow computations on the list of authors, in particular to ensure auto-generating bibliographic references to their packages reliably works. Typically, the `Author` and `Maintainer` entries in ‘DESCRIPTION’ can then be omitted in the sources (as these are automatically added by `R CMD build`).

Packages wanting to employ the auto-generated citation as (one of) the officially recommended reference(s) to their package, can employ the new simplified `citation(auto = meta)` to do so. (Note that if this is used, the package’s citation then depends on R at least 2.14.0 which should be declared in the `Depends` field of ‘DESCRIPTION’.)

As of 2011-09-27, CRAN had 3309 packages in total, with about 20% (643) having ‘CITATION’ files, but less than 1% already providing an `Authors@R` entry in their package metadata: so clearly, there is substantial room for future improvement.

5. Conclusions and outlook

R has provided functionality for representing persons and generating package citations for several years now. Earlier versions used ‘CITATION’ files with calls to `citEntry()` (and possibly `citHeader()` and `citFooter()`). R 2.12.0 introduced the new `bibentry` functionality for improved representation and manipulation of bibliographic information (the old mechanism is still supported and implemented by the new one). R 2.12.0 also added the new person functionality (changing to the more appropriate given/family scheme and adding roles). R 2.14.0 officially supports the `Authors@R` field in package ‘DESCRIPTION’ files from which `Author` and `Maintainer` fields in this file as well as package `bibentries` can reliably be auto-generated.

Having reliable comprehensive information on package “authors” and their roles which can be processed automatically is also highly relevant for the purpose of understanding the roles of persons in the social fabric underlying the remarkable growth of the R package multiverse, opening up fascinating possibilities for research on R (of course, with R), e.g., using social network analysis and related methods.

Since R 2.10.0, R help files have been able to contain R code in `\Sexpr` macros to be executed before the help text is displayed. In the future, it is planned to use this facility to auto-generate `\references` sections based on citations in the text, much as `LATEX` and `BIBTEX` can automatically generate bibliographies.

These enhancements will clearly make dealing with bibliographic information in Rd files much more convenient and efficient, and eliminate the need to “manually” (re)format entries in the `\references` sections. Of course, this should not come at the price of needing to manually generate `bibentries` for references already available in other bibliographic formats. For `BIBTEX` format ‘.bib’ databases (presumably the format typically employed by R users), conversion is straightforward: one can read the entries in ‘.bib’ files into `bibentry` objects using `read.bib()`² in CRAN package `bibtex` (François 2010), use (for R at least 2.14.0) `format()` with `style = "R"` to obtain a character vector with `bibentry()` calls for each `bibentry`, and use `writeLines()` to write (or add) this to a `bibentry` database file (see `?bibentry` for examples for the last two steps). For other formats, one could first transform to `BIBTEX` format: e.g., `Bibutils` (Putnam 2008) can convert between the COPAC, EndNote refer, EndNote XML, Pubmed XML, ISI Web of Science, US Library of Congress MODS XML, RIS, and Word 2007 bibliography formats. However, this may lose useful information: e.g., the MODS format can provide role data which (as discussed above) will be dropped when converting to `BIBTEX` format. We are thus planning to provide a `bibutils` package which instead converts to the MODS XML format first, and creates `bibentry` objects from this. (As this functionality requires both external software and package `XML`, Temple Lang 2011, it cannot be made available within base R.)

With these functionalities in place, managing bibliographic information in R documentation files will become much more convenient, and R will become a much sharper knife for cutting-edge bibliometric and scientometric analyses. In particular, it will become possible to programmatically analyze package citations, and use these to learn about the processes of knowledge dissemination driving the creation of R packages, and the relatedness of package authors and concepts.

²However, note that `read.bib()` currently segfaults when reading backslashes, and incorrectly adds class “citation” to the returned “bibentry” object.

References

- Berry K, Patashnik O (2010). `BIBTEX`. URL <http://www.ctan.org/tex-archive/biblio/bibtex/base>.
- Canty A, Ripley BD (2011). *boot: Bootstrap R (S-Plus) Functions*. R package version 1.3-3, URL <http://CRAN.R-project.org/package=boot>.
- François R (2010). *bibtex: BIBTEX Parser*. R package version 0.2-1, URL <http://CRAN.R-project.org/package=bibtex>.
- Library of Congress (2011). “MARC Standards.” URL <http://www.loc.gov/marc/>, accessed 2011-11-14.
- Putnam C (2008). *Bibutils: Bibliography Conversion Utilities*. The Scripps Research Institute. URL <http://www.scripps.edu/~cdputnam/software/bibutils/>.
- Temple Lang D (2011). *XML: Tools for Parsing and Generating XML within R and S-Plus*. R package version 3.4-3, URL <http://CRAN.R-project.org/package=XML>.
- Wikipedia (2011). “Personal Name — Wikipedia, The Free Encyclopedia.” URL http://en.wikipedia.org/wiki/Personal_name, accessed 2011-11-14.

Affiliation:

Kurt Hornik
Department of Finance, Accounting and Statistics
WU Wirtschaftsuniversität Wien
Augasse 2–6
1090 Wien, Austria
E-mail: Kurt.Hornik@R-project.org
URL: <http://statmath.wu-wien.ac.at/~hornik/>

Duncan Murdoch
Department of Statistical and Actuarial Sciences
University of Western Ontario
London, Ontario, Canada
E-mail: murdoch@stats.uwo.ca
URL: <http://www.stats.uwo.ca/faculty/murdoch/>

Achim Zeileis
Faculty of Economics and Statistics
Universität Innsbruck
Universitätsstr. 15
6020 Innsbruck, Austria
E-mail: Achim.Zeileis@R-project.org
URL: <http://eeecon.uibk.ac.at/~zeileis/>