

Changes in R

From version 3.5.0 to version 3.5.0 patched

by R Core Team

CHANGES IN R 3.5.0 patched

BUG FIXES

- `file("stdin")` is no longer considered seekable.
- `dput()` and `dump()` are no longer truncating when `options(deparse.max.lines = *)` is set.
- Calls with an S3 class are no longer evaluated when printed, fixing part of [PR#17398](#), thanks to a patch from Lionel Henry.
- Allow file argument of `Rscript` to include space even when it is first on the command line.
- `callNextMethod()` uses the generic from the environment of the calling method. Reported by Hervé Pagès with well documented examples.
- Compressed file connections are marked as blocking.
- `optim(*, lower = c(-Inf, -Inf))` no longer warns (and switches the method), thanks to a suggestion by John Nash.
- `predict(fm, newdata)` is now correct also for models where the formula has terms such as `splines::ns(..)` or `stats::poly(..)`, fixing [PR#17414](#), based on a patch from Duncan Murdoch.
- `simulate.lm(glm(*, gaussian(link = <non-default>)))` has been corrected, fixing [PR#17415](#) thanks to Alex Courtiol.
- `unlist(x)` no longer fails in some cases of nested empty lists. Reported by Steven Nydick.
- `qr.coef(qr(<all 0, w/ colnames>))` now works. Reported by Kun Ren.
- The radix sort is robust to vectors with >1 billion elements (but long vectors are still unsupported). Thanks to Matt Dowle for the fix.
- Terminal connections (e.g., `stdin`) are no longer buffered. Fixes [PR#17432](#).
- `deparse(x)`, `dput(x)` and `dump()` now respect `c()`'s argument names `recursive` and `use.names`, e.g., for `x <-setNames(0, "recursive")`, thanks to Suharto Anggono's [PR#17427](#).
- Unbuffered connections now work with encoding conversion. Reported by Stephen Berman.
- `'.Renviron'` on Windows with `Rgui` is again by default searched for in user documents directory when invoked *via* the launcher icon. Reported by Jeroen Ooms.
- `printCoefmat()` now also works with explicit `right=TRUE`.
- `print.noquote()` now also works with explicit `quote=FALSE`.
- The default method for `pairs(.., horInd=*, verInd=*)` now gets the correct order, thanks to reports by Chris Andrews and Gerrit Eichner. Additionally, when `horInd` or `verInd` contain only a subset of variables, all the axes are labeled correctly now.

- `agrep("...|...", ... , fixed=FALSE)` now matches when it should, thanks to a reminder by Andreas Kolter.
- `str(ch)` now works for more invalid multibyte strings.

CHANGES IN R 3.5.0

SIGNIFICANT USER-VISIBLE CHANGES

- All packages are by default byte-compiled on installation. This makes the installed packages larger (usually marginally so) and may affect the format of messages and tracebacks (which often exclude `.Call` and similar).

NEW FEATURES

- `factor()` now uses `order()` to sort its levels, rather than `sort.list()`. This allows `factor()` to support custom vector-like objects if methods for the appropriate generics are defined. It has the side effect of making `factor()` succeed on empty or length-one non-atomic vector(-like) types (e.g., `"list"`), where it failed before.
- `diag()` gets an optional `names` argument: this may require updates to packages defining S4 methods for it.
- `chooseCRANmirror()` and `chooseBioCmirror()` no longer have a `useHTTPS` argument, not needed now all R builds support `'https://'` downloads.
- New `summary()` method for `warnings()` with a (somewhat experimental) `print()` method.
- (**methods** package.) `.self` is now automatically registered as a global variable when registering a reference class method.
- `tempdir(check = TRUE)` recreates the `tempdir()` directory if it is no longer valid (e.g. because some other process has cleaned up the `'/tmp'` directory).
- New `askYesNo()` function and `"askYesNo"` option to ask the user binary response questions in a customizable but consistent way. (Suggestion of [PR#17242](#).)
- New low level utilities `...elt(n)` and `...length()` for working with `...` parts inside a function.
- `isTRUE()` is more tolerant and now true in

```
x <- rlnorm(99)
isTRUE(median(x) == quantile(x)["50%"])
```

New function `isFALSE()` defined analogously to `isTRUE()`.

- The default symbol table size has been increased from 4119 to 49157; this may improve the performance of symbol resolution when many packages are loaded. (Suggested by Jim Hester.)
- `line()` gets a new option `iter = 1`.
- Reading from connections in text mode is buffered, significantly improving the performance of `readLines()`, as well as `scan()` and `read.table()`, at least when specifying `colClasses`.
- `order()` is smarter about picking a default sort method when its arguments are objects.

- `available.packages()` has two new arguments which control if the values from the per-session repository cache are used (default `true`, as before) and if so how old cached values can be to be used (default one hour).

These arguments can be passed from `install.packages()`, `update.packages()` and functions calling that: to enable this `available.packages()`, `packageStatus()` and `download.file()` gain a `...` argument.

- `packageStatus()`'s `upgrade()` method no longer ignores its `...` argument but passes it to `install.packages()`.
- `installed.packages()` gains a `...` argument to allow arguments (including `noCache`) to be passed from `new.packages()`, `old.packages()`, `update.packages()` and `packageStatus()`.
- `factor(x, levels, labels)` now allows duplicated labels (not duplicated levels!). Hence you can map different values of `x` to the same level directly.
- Attempting to use `names<-()` on an S4 derivative of a basic type no longer emits a warning.
- The `list` method of `within()` gains an option `keepAttrs = FALSE` for some speed-up.
- `system()` and `system2()` now allow the specification of a maximum elapsed time ('timeout').
- `debug()` supports debugging of methods on any object of S4 class "genericFunction", including group generics.
- Attempting to increase the length of a variable containing `NULL` using `length(<-)` still has no effect on the target variable, but now triggers a warning.
- `type.convert()` becomes a generic function, with additional methods that operate recursively over `list` and `data.frame` objects. Courtesy of Arni Magnusson ([PR#17269](#)).
- `lower.tri(x)` and `upper.tri(x)` only needing `dim(x)` now work via new functions `.row()` and `.col()`, so no longer call `as.matrix()` by default in order to work efficiently for all kind of matrix-like objects.
- `print()` methods for "xgettext" and "xngettext" now use `encodeString()` which keeps, e.g. "\n", visible. (Wish of [PR#17298](#).)
- `package.skeleton()` gains an optional encoding argument.
- `approx()`, `spline()`, `splinefun()` and `approxfun()` also work for long vectors.

- `deparse()` and `dump()` are more useful for S4 objects, `dput()` now using the same internal C code instead of its previous imperfect workaround R code. S4 objects now typically `deparse` perfectly, i.e., can be recreated identically from `deparse`d code.

`dput()`, `deparse()` and `dump()` now print the `names()` information only once, using the more readable (`tag = value`) syntax, notably for `list()`s, i.e., including data frames.

These functions gain a new control option "niceNames" (see `.deparseOpts()`), which when set (as by default) also uses the (`tag = value`) syntax for atomic vectors. On the other hand, without `deparse` options "showAttributes" and "niceNames", names are no longer shown also for lists. `as.character(list(c (one = 1)))` now includes the name, as `as.character(list(list(one = 1)))` has always done.

`m:n` now also `deparse`s nicely when $m > n$.

The "quoteExpressions" option, also part of "all", no longer `quote()`s formulas as that may not re-parse identically. ([PR#17378](#))

- If the option `setWidthOnResize` is set and `TRUE`, R run in a terminal using a recent readline library will set the width option when the terminal is resized. Suggested by Ralf Goertz.
- If multiple `on.exit()` expressions are set using `add = TRUE` then all expressions will now be run even if one signals an error.
- `mclapply()` gets an option `affinity.list` which allows more efficient execution with heterogeneous processors, thanks to Helena Kotthaus.
- The character methods for `as.Date()` and `as.POSIXlt()` are more flexible *via* new arguments `tryFormats` and `optional`: see their help pages.
- `on.exit()` gains an optional argument `after` with default `TRUE`. Using `after = FALSE` with `add = TRUE` adds an exit expression before any existing ones. This way the expressions are run in a first-in last-out fashion. (From Lionel Henry.)
- On Windows, `file.rename()` internally retries the operation in case of error to attempt to recover from possible anti-virus interference.
- Command line completion on `'::'` now also includes lazy-loaded data.
- If the `TZ` environment variable is set when date-time functions are first used, it is recorded as the session default and so will be used rather than the default deduced from the OS if `TZ` is subsequently unset.
- There is now a `[` method for class `"DLLInfoList"`.
- `glm()` and `glm.fit` get the same `singular.ok = TRUE` argument that `lm()` has had forever. As a consequence, in `glm(*,method = <your_own>)`, user specified methods need to accept a `singular.ok` argument as well.
- `aspell()` gains a filter for Markdown (`'.md'` and `'.Rmd'`) files.
- `intToUtf8(multiple = FALSE)` gains an argument to allow surrogate pairs to be interpreted.
- The maximum number of DLLs that can be loaded into R e.g. *via* `dyn.load()` has been increased up to 614 when the OS limit on the number of open files allows.
- `Sys.timezone()` on a Unix-alike caches the value at first use in a session: *inter alia* this means that setting `TZ` later in the session affects only the *current* time zone and not the *system* one.
`Sys.timezone()` is now used to find the system timezone to pass to the code used when R is configured with `'--with-internal-tzcode'`.
- When `tar()` is used with an external command which is detected to be GNU tar or libarchive tar (aka `bsdtar`), a different command-line is generated to circumvent line-length limits in the shell.
- `system(*,intern = FALSE)`, `system2()` (when not capturing output), `file.edit()` and `file.show()` now issue a warning when the external command cannot be executed.
- The “default” (`"lm"` etc) methods of `vcov()` have gained new optional argument `complete = TRUE` which makes the `vcov()` methods more consistent with the `coef()` methods in the case of singular designs. The former (back-compatible) behavior is given by `vcov(*,complete = FALSE)`.
- `coef()` methods (for `lm` etc) also gain a `complete = TRUE` optional argument for consistency with `vcov()`.
For `"aov"`, both `coef()` and `vcov()` methods remain back-compatibly consistent, using the *other* default, `complete = FALSE`.

- `attach(*, pos = 1)` is now an error instead of a warning.
- New function `getDefaultCluster()` in package **parallel** to get the default cluster set via `setDefaultCluster()`.
- `str(x)` for atomic objects `x` now treats both cases of `is.vector(x)` similarly, and hence much less often prints "atomic". This is a slight non-back-compatible change producing typically both more informative and shorter output.
- `write.dcf()` gets optional argument `useBytes`.
- New, partly experimental `packageDate()` which tries to get a valid "Date" object from a package 'DESCRIPTION' file, thanks to suggestions in [PR#17324](#).
- `tools::resaveRdaFiles()` gains a version argument, for use when packages should remain compatible with earlier versions of R.
- `ar.yw(x)` and hence by default `ar(x)` now work when `x` has NAs, mostly thanks to a patch by Pavel Krivitsky in [PR#17366](#). The `ar.yw.default()`'s AIC computations have become more efficient by using `determinant()`.
- New `warnErrList()` utility (from package **nlme**, improved).
- By default the (arbitrary) signs of the loadings from `princomp()` are chosen so the first element is non-negative.
- If '`--default-packages`' is not used, then `Rscript` now checks the environment variable `R_SCRIPT_DEFAULT_PACKAGES`. If this is set, then it takes precedence over `R_DEFAULT_PACKAGES`. If default packages are not specified on the command line or by one of these environment variables, then `Rscript` now uses the same default packages as R. For now, the previous behavior of not including **methods** can be restored by setting the environment variable `R_SCRIPT_LEGACY` to 'yes'.
- When a package is found more than once, the warning from `find.package(*, verbose=TRUE)` lists all library locations.
- POSIXt objects can now also be rounded or truncated to month or year.
- `stopifnot()` can be used alternatively via new argument `exprs` which is nicer and useful when testing several expressions in one call.
- The environment variable `R_MAX_VSIZE` can now be used to specify the maximal vector heap size. On macOS, unless specified by this environment variable, the maximal vector heap size is set to the maximum of 16GB and the available physical memory. This is to avoid having the R process killed when macOS over-commits memory.
- `sum(x)` and `sum(x1, x2, ..., x<N>)` with many or long logical or integer vectors no longer overflows (and returns NA with a warning), but returns double numbers in such cases.
- Single components of "POSIXlt" objects can now be extracted and replaced via `[` indexing with 2 indices.
- S3 method lookup now searches the namespace registry after the top level environment of the calling environment.
- Arithmetic sequences created by `1:n`, `seq_along`, and the like now use compact internal representations via the ALTREP framework. Coercing integer and numeric vectors to character also now uses the ALTREP framework to defer the actual conversion until first use.
- Finalizers are now run with interrupts suspended.

- `merge()` gains new option `no.dups` and by default suffixes the second of two duplicated column names, thanks to a proposal by Scott Ritchie (and Gabe Becker).
- `scale.default(x, center, scale)` now also allows `center` or `scale` to be “numeric-alike”, i.e., such that `as.numeric(.)` coerces them correctly. This also eliminates a wrong error message in such cases.
- `par*apply` and `par*applyLB` gain an optional argument `chunk.size` which allows to specify the granularity of scheduling.
- Some `as.data.frame()` methods, notably the `matrix` one, are now more careful in not accepting duplicated or NA row names, and by default produce unique non-NA row names. This is based on new function `.rowNamesDF(x, make.names = *) <- rNms` where the logical argument `make.names` allows to specify *how* invalid row names `rNms` are handled. `.rowNamesDF()` is a “workaround” compatible default.
- R has new serialization format (version 3) which supports custom serialization of ALTREP framework objects. These objects can still be serialized in format 2, but less efficiently. Serialization format 3 also records the current native encoding of unflagged strings and converts them when de-serialized in R running under different native encoding. Format 3 comes with new serialization magic numbers (RDA3, RDB3, RDX3). Format 3 can be selected by `version = 3` in `save()`, `serialize()` and `saveRDS()`, but format 2 remains the default for all serialization and saving of the workspace. Serialized data in format 3 cannot be read by versions of R prior to version 3.5.0.
- The “Date” and “date-time” classes “POSIXlt” and “POSIXct” now have a working `length<-()` method, as wished in [PR#17387](#).
- `optim(*, control = list(warn.1d.NelderMead = FALSE))` allows to turn off the warning when applying the default “Nelder-Mead” method to 1-dimensional problems.
- `matplot(..., panel.first = .)` etc now work, as `log` becomes explicit argument and `...` is passed to `plot()` unevaluated, as suggested by Sebastian Meyer in [PR#17386](#).
- Interrupts can be suspended while evaluating an expression using `suspendInterrupts`. Subexpression can be evaluated with interrupts enabled using `allowInterrupts`. These functions can be used to make sure cleanup handlers cannot be interrupted.
- R 3.5.0 includes a framework that allows packages to provide alternate representations of basic R objects (ALTREP). The framework is still experimental and may undergo changes in future R releases as more experience is gained. For now, documentation is provided in <https://svn.r-project.org/R/branches/ALTREP/ALTREP.html>.

UTILITIES

- `install.packages()` for source packages now has the possibility to set a ‘timeout’ (elapsed-time limit). For serial installs this uses the `timeout` argument of `system2()`; for parallel installs it requires the `timeout` utility command from GNU **coreutils**.
- It is now possible to set ‘timeouts’ (elapsed-time limits) for most parts of R CMD check *via* environment variables documented in the ‘R Internals’ manual.
- The ‘BioC extra’ repository which was dropped from Bioconductor 3.6 and later has been removed from `setRepositories()`. This changes the mapping for 6–8 used by `setRepositories(ind=)`.
- R CMD check now also applies the settings of environment variables `_R_CHECK_SUGGESTS_ONLY_` and `_R_CHECK_DEPENDS_ONLY_` to the re-building of vignettes.

- R CMD check with environment variable `_R_CHECK_DEPENDS_ONLY_` set to a true value makes test-suite-management packages available and (for the time being) works around a common omission of `markdown` from the 'VignetteBuilder' field.

INSTALLATION on a UNIX-ALIKE

- Support for a system Java on macOS has been removed — install a fairly recent Oracle Java (see 'R Installation and Administration' §C.3.2).
- `configure` works harder to set additional flags in 'SAFE_FFLAGS' only where necessary, and to use flags which have little or no effect on performance.
In rare circumstances it may be necessary to override the setting of 'SAFE_FFLAGS'.
- C99 functions `expm1`, `hypot`, `log1p` and `nearbyint` are now required.
- `configure` sets a '-std' flag for the C++ compiler for all supported C++ standards (e.g., '-std=gnu++11' for the C++11 compiler). Previously this was not done in a few cases where the default standard passed the tests made (e.g. `clang 6.0.0` for C++11).

C-LEVEL FACILITIES

- 'Writing R Extensions' documents macros `MAYBE_REFERENCED`, `MAYBE_SHARED` and `MARK_NOT_MUTABLE` that should be used by package C code instead `NAMED` or `SET_NAMED`.
- The object header layout has been changed to support merging the ALTREP branch. This requires re-installing packages that use compiled code.
- 'Writing R Extensions' now documents the `R_tryCatch`, `R_tryCatchError`, and `R_UnwindProtect` functions.
- `NAMEDMAX` has been raised to 3 to allow protection of intermediate results from (usually ill-advised) assignments in arguments to BUILTIN functions. Package C code using `SET_NAMED` may need to be revised.

DEPRECATED AND DEFUNCT

- `Sys.timezone(location = FALSE)` is defunct, and is ignored (with a warning).
- `methods:::bind_activation()` is defunct now; it typically has been unneeded for years.
The undocumented 'hidden' objects `.__H__.cbind` and `.__H__.rbind` in package `base` are deprecated (in favour of `cbind` and `rbind`).
- The declaration of `pythag()` in 'Rmath.h' has been removed — the entry point has not been provided since R 2.14.0.

BUG FIXES

- `printCoefmat()` now also works without column names.
- The S4 methods on `Ops()` for the "structure" class no longer cause infinite recursion when the structure is not an S4 object.
- `nlm(f, ...)` for the case where `f()` has a "hessian" attribute now computes $LL' = H + \mu I$ correctly. (PR#17249).
- An S4 method that "rematches" to its generic and overrides the default value of a generic formal argument to NULL no longer drops the argument from its formals.

- Rscript can now accept more than one argument given on the '#!' line of a script. Previously, one could only pass a single argument on the '#!' line in Linux.
- Connections are now written correctly with encoding "UTF-16LE". (PR#16737).
- Evaluation of `..0` now signals an error. When `..1` is used and `...` is empty, the error message is more appropriate.
- (Windows mainly.) Unicode code points which require surrogate pairs in UTF-16 are now handled. All systems should properly handle surrogate pairs, even those systems that do not need to make use of them. (PR#16098)
- `stopifnot(e, e2, ...)` now evaluates the expressions sequentially and in case of an error or warning shows the relevant expression instead of the full `stopifnot(...)` call.
- `path.expand()` on Windows now accepts paths specified as UTF-8-encoded character strings even if not representable in the current locale. (PR#17120)
- `line(x, y)` now correctly computes the medians of the left and right group's x-values and in all cases reproduces straight lines.
- Extending S4 classes with slots corresponding to special attributes like `dim` and `dimnames` now works.
- Fix for `legend()` when `fill` has multiple values the first of which is NA (all colours used to default to `par(fg)`). (PR#17288)
- `installed.packages()` did not remove the cached value for a library tree that had been emptied (but would not use the old value, just waste time checking it).
- The documentation for `installed.packages(noCache = TRUE)` incorrectly claimed it would refresh the cache.
- `aggregate(<data.frame>)` no longer uses spurious names in some cases. (PR#17283)
- `object.size()` now also works for long vectors.
- `packageDescription()` tries harder to solve re-encoding issues, notably seen in some Windows locales. This fixes the `citation()` issue in PR#17291.
- `poly(<matrix>, 3)` now works, thanks to prompting by Marc Schwartz.
- `readLines()` no longer segfaults on very large files with embedded '\0' (aka 'nul') characters. (PR#17311)
- `ns()` (package **splines**) now also works for a single observation. `interpSpline()` gives a more friendly error message when the number of points is less than four.
- `dist(x, method = "canberra")` now uses the correct definition; the result may only differ when `x` contains values of differing signs, e.g. not for 0-1 data.
- `methods:::cbind()` and `methods:::rbind()` avoid deep recursion, thanks to Suharto Anggono via PR#17300.
- Arithmetic with zero-column data frames now works more consistently; issue raised by Bill Dunlap.
Arithmetic with data frames gives a data frame for `^` (which previously gave a numeric matrix).
- `pretty(x, n)` for large `n` or large `diff(range(x))` now works better (though it was never meant for large `n`); internally it uses the same rounding fuzz (`1e-10`) as `seq.default()` — as it did up to 2010-02-03 when both were `1e-7`.

- Internal C-level `R_check_class_and_super()` and hence `R_check_class_etc()` now also consider non-direct super classes and hence return a match in more cases. This e.g., fixes behaviour of derived classes in package [Matrix](#).
- Reverted unintended change in behavior of return calls in `on.exit` expressions introduced by stack unwinding changes in R 3.3.0.
- Attributes on symbols are now detected and prevented; attempt to add an attribute to a symbol results in an error.
- `fisher.test(*, workspace = <n>)` now may also increase the internal stack size which allows larger problem to be solved, fixing [PR#1662](#).
- The **methods** package no longer directly copies slots (attributes) into a prototype that is of an “abnormal” (reference) type, like a symbol.
- The **methods** package no longer attempts to call `length<-()` on NULL (during the bootstrap process).
- The **methods** package correctly shows methods when there are multiple methods with the same signature for the same generic (still not fully supported, but at least the user can see them).
- `sys.on.exit()` is now always evaluated in the right frame. (From Lionel Henry.)
- `seq.POSIXt(*, by = "<n>DSTdays")` now should work correctly in all cases and is faster. ([PR#17342](#))
- `.C()` when returning a logical vector now always maps values other than FALSE and NA to TRUE (as documented).
- Subassignment with zero length vectors now coerces as documented ([PR#17344](#)). Further, `x <-numeric(); x[1] <-character()` now signals an error ‘replacement has length zero’ (or a translation of that) instead of doing nothing.
- (Package **parallel**.) `mclapply()`, `pvec()` and `mcpipeline()` (when `mccollect()` is used to collect results) no longer leave zombie processes behind.
- `R CMD INSTALL <pkg>` now produces the intended error message when, e.g., the `LazyData` field is invalid.
- `as.matrix(dd)` now works when the data frame `dd` contains a column which is a data frame or matrix, including a 0-column matrix/d.f. .
- `mclapply(X, mc.cores)` now follows its documentation and calls `lapply()` in case `mc.cores = 1` also in the case `mc.preschedule` is false. ([PR#17373](#))
- `aggregate(<data.frame>, drop=FALSE)` no longer calls the function on <empty> parts but sets corresponding results to NA. (Thanks to Suharto Anggono’s patches in [PR#17280](#)).
- The `duplicated()` method for data frames is now based on the `list` method (instead of string coercion). Consequently `unique()` is better distinguishing data frame rows, fixing [PR#17369](#) and [PR#17381](#). The methods for matrices and arrays are changed accordingly.
- Calling `names()` on an S4 object derived from “environment” behaves (by default) like calling `names()` on an ordinary environment.
- `read.table()` with a non-default separator now supports quotes following a non-whitespace character, matching the behavior of `scan()`.

- `parLapplyLB` and `parSapplyLB` have been fixed to do load balancing (dynamic scheduling). This also means that results of computations depending on random number generators will now really be non-reproducible, as documented.
- Indexing a list using dollar and empty string (`l[i]""`) returns `NULL`.
- Using `\usage{ data(<name>, package="<pkg>") }` no longer produces R CMD check warnings.
- `match.arg()` more carefully chooses the environment for constructing default choices, fixing [PR#17401](#) as proposed by Duncan Murdoch.
- Deparsing of consecutive `!` calls is now consistent with deparsing unary `-` and `+` calls and creates code that can be reparsed exactly; thanks to a patch by Lionel Henry in [PR#17397](#). (As a side effect, this uses fewer parentheses in some other deparsing involving `!` calls.)