Changes in R Version 2.7.0

by the R Core Team

User-visible changes

- The default graphics device in non-interactive use is now `pdf()` rather than `postscript()`. [PDF viewers are now more widely available than PostScript viewers.]

  The default width and height for `pdf()` and `bitmap()` have been changed to 7 (inches) to match the screen devices.

- Most users of the `X11()` device will see a new device that has different fonts, anti-aliasing of lines and fonts and supports semi-transparent colours.

- Considerable efforts have been made to make the default output from graphics devices as similar as possible (and in particular close to that from postscript/pdf). Many devices were misinterpreting 'pointsize' in some way, for example as being in device units (pixels) rather than in points.

- Packages which include graphics devices need to be re-installed for this version of R, with recently updated versions.

New features

- The apse code used by `agrep()` has been updated to version 0.16, with various bug fixes.

  `agrep()` now supports multibyte character sets.

- `any()` and `all()` avoid coercing zero-length arguments (which used a surprising amount of memory) since they cannot affect the answer.

  Coercion of other than integer arguments now gives a warning as this is often a mistake (e.g. writing `all(pr)` > 0 instead of `all(pr > 0)`).

- `as.Date()`, `as.POSIXct()` and `as.POSIXlt()` now convert numeric arguments (days or seconds since some epoch) provided the 'origin' argument is specified.

- New function `as.octmode()` to create objects such as file permissions.

- `as.POSIXlt()` is now generic, and it and `as.POSIXct()` gain a '...' argument. The character/factor methods now accept a 'format' argument (analogous to that for as.Date).

- New function `browseVignettes()` lists available vignettes in an HTML browser with links to PDF, Rnw, and R files.

- There are new capabilities "aqua" (for the AQUA GUI and `quartz()` device on Mac OS X) and "cairo" (for cairo-based graphics devices).

- New function `checkNEWS()` in package 'tools' that detects common errors in NEWS file formatting.

- `deparse()` gains a new argument 'nlines' to limit the number of lines of output, and this is used internally to make several functions more efficient.

- `deriv()` now knows the derivatives of `digamma(x)`, `trigamma(x)` and `psigamma(x, deriv)` (wrt to x).

- `dir.create()` has a new argument 'mode', used on Unix-alikes (only) to set the permissions on the created directory.

- Where an array is dropped to a length-one vector by `drop()` or [, drop = TRUE], the result now has names if exactly one of the dimensions was named. (This is compatible with S.) Previously there were no names.

- The 'incomparables' argument to `duplicated()`, `unique()` and `match()` is now implemented, and passed to `match()` from `merge()`.

- `dyn.load()` gains a 'DLLpath' argument to specify the path for dependent DLLs: currently only used on Windows.

- The spreadsheet `edit()` methods (and used by `fix()`) for data frames and matrices now warn when classes are discarded.

  When editing a data frame, columns of unknown type (that is not numeric, logical, character or factor) are now converted to character (instead of numeric).

- `file.create()` has a new argument 'showWarnings' (default TRUE) to show an informative warning when creation fails, and `dir.create()` warns under more error conditions.

- New higher-order functions `Find()`, `Negate()` and `Position()`.

- `[dpqr]gamma(*, shape = 0)` now work as limits of 'shape -> 0', corresponding to the point distribution with all mass at 0.

- An informative warning (in addition to the error message) will be given when the basic, extended or perl mode of `grep()`, `strsplit()` and friends fails to compile the pattern.

- More study is done of perl=TRUE patterns in `grep()` and friends when `length(x)` > 10: this should improve performance on long vectors.

- `grep()`, `strsplit()` and friends with fixed=TRUE or perl=TRUE work in UTF-8 and preserve the UTF-8 encoding for UTF-8 inputs where supported.

- `help.search()` now builds the database about 3x times faster.

- `iconv()` now accepts "UTF8" on all platforms (many did, but not e.g. libiconv as used on Windows).

- `identity()` convenience function to be used for programming.

- In addition to warning when 'pkgs' is not found, `install.packages()` now reports if it finds a valid package with only a case mismatch in the name.

- `intToUtf8()` now marks the Encoding of its output.

- The function `is()` now works with S3 inheritance; that is, with objects having multiple strings in the class attribute.

- Extensions to condition number computation for matrices, notably complex ones are provided, both in `kappa()` and the new `rcond()`.

- `list.files()` gains a 'ignore.case' argument, to allow case-insensitive matching on some Windows/MacOS file systems.

- `ls.str()` and `lsf.str()` have slightly changed arguments and defaults such that `ls.str()` no arguments works when debugging.

- Under Unix, `utils::make.packages.html()` can now be used directly to set up linked HTML help pages, optionally without creating the package listing and search database (which can be much faster).

- `new.packages()` now knows about the frontend package gnomeGUI (which does not install into a library).

- `optim(*, control = list(...))` now warns when '...' contains unexpected names, instead of silently ignoring them.

- The options "browser" and "editor" may now be set to functions, just as "pager" already could.

- `packageDescription()` makes use of installed metadata where available (for speed, e.g. in `make.packages.html()`).

- `pairwise.t.test()` and `pairwise.wilcox.-test()` now more explicitly allow paired tests. In the former case it is now flagged as an error if both 'paired' and 'pool.SD' are set TRUE (formerly, 'paired' was silently ignored), and one-sided tests are generated according to 'alternative' also if 'pool.SD' is TRUE.

- `paste()` and `file.path()` are now completely internal, for speed. (This speeds up `make.packages.html(packages=FALSE)` severalfold, for example.)

- `paste()` now sets the encoding on the result under some circumstances (see ?paste).

- `predict.loess()` now works when `loess()` was fitted with transformed explanatory variables, e.g, `loess(y ~ log(x)+ log(z))`.

- `print(<data.frame>)`'s new argument 'row.names' allows to suppress printing rownames.

- `print()` and `str()` now also "work" for 'logLik' vectors longer than one.

- Progress-bar functions `txtProgressBar()`, `tkProgressBar()` in package tcltk and `winProgressBar()` (Windows only).

- `readChar()` gains an argument 'useBytes' to allow it to read a fixed number of bytes in an MBCS locale.

- `readNEWS()` has been moved to the tools package.

- `round()` and `signif()` now do internal argument matching if supplied with two arguments and at least one is named.

- New function `showNonASCII()` in package tools to aid detection of non-ASCII characters in .R and .Rd files.

- The `[dpq]signrank()` functions now typically use considerably less memory than previously, thanks to a patch from Ivo Ugrina.

- `spec.ar()` now uses `frequency(x)` when calculating the frequencies of the estimated spectrum, so that for monthly series the frequencies are now per year (as for spec.pgram) rather than per month as before.

- `spline()` gets an 'xout' argument, analogously to `approx()`.

- sprintf() now does all the conversions needed in a first pass if length(fmt) == 1, and so can be many times faster if called with long vector arguments.

- [g]sub(useBytes = FALSE) now sets the encoding on changed elements of the result when working on an element of known encoding. (This was previously done only for perl = TRUE.)

- New function Sys.chmod(), a wrapper for 'chmod' on platforms which support it. (On Windows it handles only the read-only bit.)

- New function Sys.umask(), a wrapper for 'umask' on platforms which support it.

- New bindings ttk*() in package tcltk for the 'themed widgets' of Tk 8.5. The tcltk demos make use of these widgets where available.

- write.table(d, row.names=FALSE) is faster when 'd' has millions of rows; in particular for a data frame with automatic row names. (Suggestion from Martin Morgan.)

- The parser limit on string size has been removed.

- If a NEWS file is present in the root of a source package, it is installed (analogously to LICENSE, LICENCE and COPYING).

- Rd conversion to 'example' now quotes aliases which contain spaces.

- The handling of DST on dates outside the range 1902-2037 has been improved. Dates after 2037 are assumed to have the same DST rules as currently predicted for the 2030's (rather than the 1970s), and dates prior to 1902 are assumed to have no DST and the same offset as in 1902 (if known, otherwise as in the 1970s).

- On platforms where we can detect that mktime sets errno (e.g. Solaris and the code used on Windows but not Linux nor Mac OS X), 1969-12-31 23:59:59 GMT is converted from POSIXlt to POSIXct as -1 and not NA.

- The definition of 'whitespace' used by the parser is slightly wider: it includes Unicode space characters on Windows and in UTF-8 locales on machines which use Unicode wide characters.

- The src/extra/intl sources have been updated to those from gettext 0.17.

- New flag –interactive on Unix-alikes forces the session to be interactive (as –ess does on Windows).

- x[<zero-length>] <- NULL is always a no-op: previously type-checking was done on the replacement value and so this failed, whereas we now assume NULL can be promoted to any zero-length vector-like object.

  Other cases of a zero-length index are done more efficiently.

- There is a new option in Rd markup of \donttest{} to mark example code that should be run by example() but not tested (e.g. because it might fail in some locales).

- The error handler in the parser now reports line numbers for more syntax errors (MBCS and Unicode encoding errors, line length and context stack overflows, and mis-specified argument lists to functions).

- The "MethodsList" objects originally used for method selection are being phased out. New utilities provide simpler alternatives (see ?findMethods), and direct use of the mangled names for the objects is now deprecated.

- Creating new S4 class and method definitions in an environment that could not be identified (as package, namespace or global) previously generated an error. It now results in creating and using an artificial package name from the current date/time, with a warning. See ?getPackageName.

- Unix-alikes now give a warning on startup if locale settings fail. (The Windows port has long done so.)

- Parsing and scanning of numerical constants is now done by R's own C code. This ensures cross-platform consistency, and mitigates the effects of setting LC_NUMERIC (within base R it only applies to output – packages may differ).

  The format accepted is more general than before and includes binary exponents in hexadecimal constants: see ?NumericConstants for details.

- Dependence specifications for R or packages in the Depends field in a DESCRIPTION file can now make use of operators < > == and != (in addition to <= and >=): such packages will not be installable nor loadable in R < 2.7.0.

  There can be multiple mentions of R or a package in the Depends field in a DESCRIPTION file: only the first mention will be used in R < 2.7.0.

GRAPHICS CHANGES

- The default graphics devices in interactive and non-interactive sessions are now configurable via environment variables R_INTERACTIVE_DEVICE and R_DEFAULT_DEVICE respectively.

- New function `dev.new()` to launch a new copy of the default graphics device (and taking care if it is "pdf" or "postscript" not to trample on the file of an already running copy).

- `dev.copy2eps()` uses `dev.displaylist()` to detect screen devices, rather than list them in the function.

- New function `dev.copy2pdf()`, the analogue of `dev.copy2eps()`.

- `dev.interactive()` no longer treats a graphics device as interactive if it has a display list (but devices can still register themselves on the list of interactive devices).

- The `X11()` and `windows()` graphics devices have a new argument 'title' to set the window title.

- `X11()` now has the defaults for all of its arguments set by the new function `X11.options()`, inter alia replacing options "gamma", "colortype" and "X11fonts".

- `ps.options()` now warns on unused option 'append'.

  `xfig()` no longer takes default arguments from `ps.options()`. (This was not documented prior to 2.6.1 patched.)

  `pdf()` now takes defaults from the new function `pdf.options()` rather that from `ps.options()` (and the latter was not documented prior to 2.6.1 patched).

  The defaults for all arguments other than 'file' in `postscript()` and `pdf()` can now be set by `ps.options()` or `pdf.options()`

- New functions `setEPS()` and `setPS()` as wrappers to `ps.options()` to set appropriate defaults for figures for inclusion in other documents and for spooling to a printer respectively.

- The meaning of numeric 'pch' has been extended where MBCSes are supported. Now negative integer values indicate Unicode points, integer values in 32-127 represent ASCII characters, and 128-255 are valid only in single-byte locales. (Previously what happened with negative pch values was undocumented: they were replaced by the current setting of `par("pch")`.)

- Graphics devices can say if they can rotate text well (e.g. `postscript()` and `pdf()` can) and if so the device's native text becomes the default for contour labels rather than using Hershey fonts.

- The setting of the line spacing (`par("cra")`[2]) on the `X11()` and `windows()` devices is now comparable with `postscript()` etc, and roughly 20% smaller than before (it used to depend on the locale for X11). (So is the `pictex()` device, now 20% larger.) This affects the margin size in plots, and should result in better-looking plots.

- There is a per-device setting for whether new frames need confirmation. This is controlled by either `par("ask")` or `grid.prompt()` and affects all subsequent plots on the device using base or grid graphics.

- There is a new version of the `X11()` device based on cairo graphics which is selected by type "cairo" or "nbcairo", and is available on machines with cairo installed and preferably pango (which most machines with gtk+ >= 2.8 will have). This version supports translucent colours and normally does a better job of font selection so it has been possible to display (e.g.) English, Polish, Russian and Japanese text on a single `X11()` window. It is the default where available.

  There is a companion function, `savePlot()`, to save the current plot to a PNG file.

  On Unix-alikes, devices `jpeg()` and `png()` also accept type = "cairo", and with that option do not need a running X server. The meaning of `capabilities("jpeg")` and `capabilities("png")` has changed to reflect this. On MacOS X, there is a further type = "quartz". The default type is selected by the new option "bitmapType", and is "quartz" or "cairo" where available.

  Where cairo 1.2 or later is supported, there is a `svg()` device to write SVG files, and `cairo_pdf()` and `cairo_ps()` devices to write (possibly bitmap) PDF and postscript files via cairo.

  Some features require cairo >= 1.2, and some which are nominally supported under 1.2 seem to need 1.4 to work well.

- There are new `bmp()` and `tiff()` devices.

- New function `devSize()` to report the size of the current graphics device surface (in inches or device units). This gives the same information as `par("din")`, but independent of the graphics subsystem.

- New base graphics function `clip()` to set the clipping region (in user coordinates).

- New functions grconvertX() and grconvertY() to convert between coordinate systems in base graphics.

- identify() recycles its 'labels' argument if necessary.

- stripchart() is now a generic function, with default and formula methods defined. Additional graphics parameters may be included in the call. Formula handling is now similar to boxplot().

- strwidth() and strheight() gain 'font' and 'vfont' arguments and accept in-line pars such as 'family' in the same way as text() does. (Longstanding wish of PR#776)

- example(ask=TRUE) now applies to grid graphics (e.g. from lattice) as well as to base graphics.

- Option "device.ask.default" replaces "par.ask.-default" now it applies also to grid.prompt().

- plot.formula() only prompts between plots for interactive devices (it used to prompt for all devices).

- When plot.default() is called with y=NULL it now calls Axis() with the 'y' it constructs rather than use the default axis.

Deprecated & defunct

- In package installation, SaveImage: yes is defunct and lazyloading is attempted instead.

- $ on an atomic vector or S4 object is now defunct.

- Partial matching in [[is now only performed if explicitly requested (by exact=FALSE or exact=NA).

- Command-line completion has been moved from package 'rcompgen' to package 'utils': the former no longer exists as a separate package in the R distribution.

- The S4 pseudo-classes "single" and double have been removed. (The S4 class for a REALSXP is "numeric": for back-compatibility as(x, "double") coerces to "numeric".)

- gpar(gamma=) in the grid package is now defunct.

- Several S4 class definition utilities, get*(), have been said to be deprecated since R 1.8.0; these are now formally deprecated. Ditto for removeMethodsObject().

- Use of the graphics headers Rgraphics.h and Rdevices.h is deprecated, and these will be unavailable in R 2.8.0. (They are hardly used except in graphics devices, for which there is an updated API in this version of R.)

- options("par.ask.default") is deprecated in favour of "device.ask.default".

- The 'device-independent' family "symbol" is deprecated as it was highly locale- and device-dependent (it only did something useful in single-byte locales on most devices) and font=5 (base) or fontface=5 (grid) did the job it was intended to do more reliably.

- gammaCody() is now formally deprecated.

- Two low-level functions using MethodsList metadata objects (mlistMetaName() and getAllMethods()) are deprecated.

- Setting par(gamma=) is now deprecated, and the windows() device (the only known example) no longer allows it.

- The C macro 'allocString' will be removed in 2.8.0 – use 'mkChar', or 'allocVector' directly if really necessary.

Installation

- Tcl/Tk >= 8.3 (released in 2000) is now required to build package tcltk.

- configure first tries TCL_INCLUDE_SPEC and TK_INCLUDE_SPEC when looking for Tcl/Tk headers. (The existing scheme did not work for the ActiveTcl package on Mac OS X.)

- The Windows build only supports Windows 2000 or later (XP, Vista, Server 2003 and Server 2008).

- New option –enable-R-static-lib installs libR.a which can be linked to a front-end via 'R CMD config –ldflags'. The tests/Embedding examples now work with a static R library.

- Netscape (which was discontinued in Feb 2008) is no longer considered when selecting a browser.

- xdg-open (the freedesktop.org interface to kfmclient/gnome-open/...) is considered as a possible browser, after real browsers such as firefox, mozilla and opera.

- The search for tclConfig.sh and tkConfig.sh now only looks in directories with names containing $(LIBnn) in the hope of finding the version for the appropriate architecture (e.g. x86_64 or i386).

- libtool has been updated to version 2.2.
- Use of –with-system-zlib, –with-system-bzlib or –with-system-pcre now requires version >= 1.2.3, 1.0.5, 7.6 respectively, for security.

Utilities

- Rdconv now removes empty sections including alias and keyword entries, with a note.
- Keyword entries are no longer mandatory in Rd files.
- R CMD INSTALL now also installs tangled versions of all vignettes.
- R CMD check now warns if spaces or non-ASCII characters are used in file paths, since these are not in general portable.
- R CMD check (via massage-examples.pl) now checks all examples with a 7 inch square device region on A4 paper, for locale-independence and to be similar to viewing examples on an on-screen device.

  If a package declares an encoding in the DESCRIPTION file, the examples are assumed to be in that encoding when running the tests. (This avoids errors in running latin1 examples in a UTF-8 locale.)
- R CMD check uses pdflatex (if available) to check the typeset version of the manual, producing PDF rather than DVI. (This is a better check since the package reference manuals on CRAN are in PDF.)
- R CMD Rd2dvi gains a –encoding argument to be passed to R CMD Rdconv, to set the default encoding for conversions. If this is not supplied and the files are package sources and the DESCRIPTION file contains an Encoding field, that is used for the default encoding.
- `available.packages()` (and hence `install.packages()` etc.) now supports subdirectories in a repository, and `tools::write_PACKAGES()` can now produce PACKAGES files including subdirectories.
- The default for 'stylepath' in Sweave's (default) RweaveLatex driver can be set by the environment variable SWEAVE_STYLEPATH_DEFAULT: see ?RweaveLatex.

C-level facilities

- Both the Unix and Windows interfaces for embedding now make use of 'const char *' declarations where appropriate.
- `Rprintf()` and `REprintf()` now use 'const char *' for their format argument – this should reduce warnings when called from C++.
- There is a new description of the interface for graphics devices in the 'R Internals' manual, and several new entry points. The API has been updated to version R_GE_version = 5, and graphics devices will need to be updated accordingly.
- Graphics devices can now select to be sent text in UTF-8, even if the current locale is not UTF-8 (and so enable text entered in UTF-8 to be plotted). This is used by `postscript()`, `pdf()` and the `windows()` family of devices, as well as the new cairo-based devices.
- More Lapack routines are available (and declared in R_Ext/Lapack.h), notably for (reciprocal) condition number estimation of complex matrices.
- Experimental utility R_has_slot supplementing R_do_slot.
- There is a new public interface to the encoding info stored on CHARSXPs, getCharCE and mkCharCE using the enumeration type cetype_t.
- A new header 'R_ext/Visibility.h' contains some definitions for controlling the visibility of entry points, and how to control visibility is now documented in 'Writing R Extensions'.

Bug fixes

- `pt(x, df)` is now even more accurate in some cases (e.g. 12 instead of 8 significant digits), when $x^2 << df$, thanks to a remark from Ian Smith, related to PR#9945.
- `co[rv](use = "complete.obs")` now always gives an error if there are no complete cases: they used to give NA if method = "pearson" but an error for the other two methods. (Note that this is pretty arbitrary, but zero-length vectors always give an error so it is at least consistent.)

  `cor(use="pair")` used to give diagonal 1 even if the variable was completely missing for the rank methods but NA for the Pearson method: it now gives NA in all cases.

  `cor(use="pair")` for the rank methods gave a matrix result with dimensions > 0 even if one of the inputs had 0 columns.
- Supplying edit.row.names = TRUE when editing a matrix without row names is now an error and not a segfault. (PR#10500)

- The error handler in the parser reported unexpected & as && and | as ||.

- ps.options(reset = TRUE) had not reset for a long time.

- paste() and file.path() no longer allow NA_character_ for their 'sep' and 'collapse' arguments.

- by() failed for 1-column matrices and dataframes. (PR#10506) However, to preserve the old behaviour, the default method when operating on a vector still passes subsets of the vector to FUN, and this is now documented.

- Better behaviour of str.default() for non-default 'strict.width' (it was calling str() rather than str.default() internally); also, more useful handling of options("str").

- wilcox.test(exact=FALSE, conf.int=TRUE) could fail in some extreme two-sample problems. (Reported by Wolfgang Huber.)

- par(pch=) would accept a multi-byte string but only use the first byte. This would lead to incorrect results in an MBCS locale if a non-ASCII character was supplied.

- There are some checks for valid C-style formats in, e.g. png(filename=). (PR#10571)

- vector() was misinterpreting some double 'length' values, e.g, NaN and NA_real_ were interpreted as zero. Also, invalid types of 'length' were interpreted as -1 and hence reported as negative. (length<- shared the code and hence the same misinterpretations.)

- A basic class "S4" was added to correspond to the "S4" object type, so that objects with this type will print, etc. The class is VIRTUAL, since all actual S4 objects must have a real class.

- Classes with no slots that contain only VIRTUAL classes are now VIRTUAL, as was intended but confused by having an empty S4 object as prototype. ## backed out temporarily ##

- format.AsIs() discarded dimnames, causing dataframes with matrix variables to be printed without using the column names, unlike what happens in S-PLUS (Tim Hesterberg, PR#10730).

- xspline() and grid::grid.xspline() work in device coordinates and now correct for anisotropy in the device coordinate system.

- grid.locator() now indicates to the graphics device that it is is in 'graphics input' mode (as locator() and identify() always have). This means that devices can now indicate the 'graphics input' mode by e.g. a change of cursor.

- Locales without encoding specification and non-UTF-8 locales now work properly on Mac OS X. Note that locales without encoding specification always use UTF-8 encoding in Mac OS X (except for specials "POSIX" and "C") - this is different from other operating systems.

- iconv() now correctly handles to="" and from="" on Mac OS X.

- In diag()'s argument list, drop the explicit default (' = n') for 'ncol' which is ugly when making diag() generic.

- S4 classes with the same name from different packages were not recognized because of a bug in caching the new definition.

- jpeg() and png() no longer maintain a display list, as they are not interactive devices.

- Using attr(x, "names") <- value (instead of the correct names<-) with 'value' a pairlist (instead of the correct character vector) worked incorrectly. (PR#10807)

- Using [<- to add a column to a data frame dropped other attributes whereas [[<- and $<- did not: now all preserve attributes. (PR#10873)

- File access functions such as file.exists(), file.info(), dirname() and unlink() now treat an NA filename as a non-existent file and not the file "NA".

- r<foo>(), the random number generators, are now more consistent in warning when NA's (specifically NaN's) are generated.

- rnorm(n, mu = Inf) now returns rep(Inf, n) instead of NaN; similar changes are applied to rlnorm(), rexp(), etc.

- [l]choose() now warns when rounding non-integer 'k' instead of doing so silently. (May help confused users such as PR#10766.)

- gamma() was warning incorrectly for most negative values as being too near a negative integer. This also affected other functions making use of its C-level implementation.

- dumpMethod() and dumpMethods() now work again.

- package.skeleton() now also works for code_files with only metadata (e.g. S4 setClass) definitions; it handles S4 classes and methods, producing documentation and NAMESPACE exports if requested.

- Some methods package utilities (`implicitGeneric()`, `makeGeneric()`) will be more robust in dealing with primitive functions (not a useful idea to call them with primitives, though).

- Making a MethodsList from a function with no methods table will return an empty list, rather than cause an error (questionably a bug, but caused some obscure failures).

- `setAs()` now catches 2 arguments in the method definition, if they do not match the arguments of `coerce()`.

- S4 methods with missing arguments in the definition are handled correctly when non-signature arguments exist, and check for conflicting local names in the method definition.

- `qgamma()` and `qchisq()` could be inaccurate for small p, e.g. `qgamma(1.2e-10, shape = 19)` was 2.52 rather than 2.73.

- `dbeta(.., ncp)` is now more accurate for large ncp, and typically no longer underflows for give.log = TRUE.

- `coerce()` is now a proper S4 object and so prints correctly.

- @ now checks it is being applied to an S4 object, and if not gives a warning (which will become an error in 2.8.0).

- `dump()` and friends now warn that all S4 objects (even those based on vectors) are not `source()`able, with a stronger wording.

- `read.dcf(all = TRUE)` was leaking connections.

- `scan()` with a non-default separator could skip nul bytes, including those entered as code 0 with allowEscapes=TRUE. This was different from the default separator.

- `determinant(matrix(,0,0))` now returns a correct "det" result; also value 1 or 0 depending on 'logarithm', rather than `numeric(0)`.

- Name space 'grDevices' was not unloading its DLL when the name space was unloaded.

- `getNativeSymbolInfo()` was unaware of non-registered Fortran names, because one of the C support routines ignored them.

- `load()` again reads correctly character strings with embedded nuls. (This was broken in 2.6.x, but worked in earlier versions.)