Fedora Packages of R Software

Last updated by Iñaki Úcar on 2024-09-24

Source: https://pagure.io/R/fedora-r-packages (https://pagure.io/R/fedora-r-packages)

Installation

Fedora

The newest R release (including recommended packages as well as development headers and tools) can be installed by running

```
$ sudo dnf install R
```

or `yum` instead of `dnf` for older EPEL versions. This 'R' RPM is a meta-package. It has no content but ensures that the following components are installed

Component	Description
R-core	The minimal R components necessary for a functional runtime
R-core-devel	Core files for development of R packages (no Java)
R-java	R with Fedora-provided Java Runtime Environment
R-java-devel	Development package for use with Java enabled R components
libRmath	Standalone math library from the R project
libRmath-devel	Headers from the R standalone math library

This division enables minimal installations (e.g., with no Java, with no development tools…), but generally R users will need all the components to be able to install any package from source. Therefore, it is recommended to install the 'R' meta-package.

EPEL for CentOS and RHEL

The Fedora RPMs for R have been ported to CentOS/RHEL by the project Extra Packages for Enterprise Linux (https://docs.fedoraproject.org/en-US/epel/) (EPEL). These RPMs are also compatible with distributions derived from CentOS/RHEL.

To use the EPEL repository, it is sufficient to download and install the appropriate "epel-release" RPM, as described in the EPEL FAQ (https://docs.fedoraproject.org/en-US/epel/epel-faq/#how_can_i_install_the_packages_from_the_epel_software_repository). Then R can be installed as described above.

Administration and maintenance

The R installation is divided in two directories:

- `/usr/lib/R` (`/usr/lib64/R` in 64-bit architectures) contains R binaries and libraries.
- `/usr/share/R` contains documentation, licenses and other non-binary files.

In the same way,

- `/usr/lib/R/library` (`/usr/lib64/R/library` in 64-bit architectures) contains system-provided packages with binary code.
- `/usr/share/R/library` contains system-provided packages without binary code.

Additionally, the R installation adds the following paths:

- `/usr/local/lib/R/library` (the same for 64-bit architectures), which is not used by any package in the official repositories.

- `/home/<user>/R/<architecture>-redhat-linux-gnu-library/<version>` , which is the destination for any package installed from the R console using `install.packages` .

For example, these are the library paths for a x86_64 machine with R 4.4.x installed:

```
.libPaths()
#> [1] "/home/<user>/R/x86_64-redhat-linux-gnu-library/4.4"
#> [2] "/usr/local/lib/R/library"
#> [3] "/usr/lib64/R/library"
#> [4] "/usr/share/R/library"
```

Supported packages

Recommended R packages are included as part of the R-core component. A number of add-on packages from CRAN, Bioconductor and other sources are readily available via the official repositories. Hence, running

```
$ dnf repoquery --repo=fedora-source R-*
```

provides a comprehensive list.

The listing below shows all RPMs available for R packages on Fedora Linux 40 (Forty), classified by the R repository that would normally be used to install the package from within R (see the help page `?chooseRepositories`).

▶ Click to toggle the list of packages

Note that the classification is not mutually exclusive (e.g. R-RCurl appears several times) and that there are RPMs that are not available from any standard R repository. These are listed under “Other”.

Additional packages

The cran2copr (https://copr.fedorainfracloud.org/coprs/iucar/cran/) project maintains binary RPM repositories for the current and previous stable Fedora version for most of CRAN (more than 20k packages as of September 2024) in an automated way using Fedora Copr (https://copr.fedorainfracloud.org/).

These repositories are automatically synchronized with CRAN every day at 00:00 UTC through a GitHub Action that removes archived packages and builds the most recent updates. To ensure compatibility with the official repositories, these set of packages are named “R-CRAN-pkgname” (instead of “R-pkgname”), and are installed into `/usr/local/lib/R/library` .

To enable this Copr repository in your system:

```
$ sudo dnf copr enable iucar/cran
$ sudo dnf install R-CoprManager
```

The last command is optional, but recommended, because the CoprManager package integrates binary package installation into your R session. In this way, you can install or update packages in R as you normally do, e.g.,

```
install.packages("car")
update.packages(ask=FALSE)
```

in the R console, and packages will be automatically installed from the Copr repository. If a package is not available, then it just falls back to normal installation from CRAN.

On the other hand, `remove.packages` will still remove only packages installed in your user library. If you want to remove system packages, run:

```
CoprManager::remove_copr("car")
```

If you want to disable the CoprManager, so that `install.packages` only works with CRAN again, then run:

```
CoprManager::disable()
install.packages("car") # from CRAN to user lib
```

BLAS/LAPACK switching

Since Fedora 33, R (as well as Numpy, Octave and all the other BLAS/LAPACK consumers) is linked against FlexiBLAS (https://www.mpi-magdeburg.mpg.de/projects/flexiblas), a BLAS/LAPACK wrapper library that enables runtime switching of the optimized backend (see the change proposal (https://fedoraproject.org/wiki/Changes/FlexiBLAS_as_BLAS/LAPACK_manager) for further details), and the OpenMP version of OpenBLAS is set as the default backend.

The accompanying `flexiblas` R package enables BLAS/LAPACK switching without leaving the R session, as well as setting the number of threads for parallel backends (see the package’s README (https://github.com/Enchufa2/r-flexiblas) for further information).

```
$ sudo dnf install R-flexiblas # install FlexiBLAS API interface for R
$ sudo dnf install flexiblas-* # install all available optimized backends
```

Then, in an R session we see:

```
library(flexiblas)

# check whether FlexiBLAS is available
flexiblas_avail()
#> [1] TRUE

# get the current backend
flexiblas_current_backend()
#> [1] "OPENBLAS-OPENMP"

# list all available backends
flexiblas_list()
#> [1] "NETLIB"           "__FALLBACK__"     "BLIS-THREADS"     "OPENBLAS-OPENMP"
#> [5] "BLIS-SERIAL"      "ATLAS"            "OPENBLAS-SERIAL"  "OPENBLAS-THREADS"
#> [9] "BLIS-OPENMP"

# get/set the number of threads
flexiblas_set_num_threads(12)
flexiblas_get_num_threads()
#> [1] 12
```

This is an example of GEMM benchmark for all the backends available:

```
library(flexiblas)

n <- 2000
runs <- 10
ignore <- "__FALLBACK__"

A <- matrix(runif(n*n), nrow=n)
B <- matrix(runif(n*n), nrow=n)

# load backends
backends <- setdiff(flexiblas_list(), ignore)
idx <- flexiblas_load_backend(backends)

# benchmark
timings <- sapply(idx, function(i) {
  flexiblas_switch(i)

  # warm-up
  C <- A[1:100, 1:100] %*% B[1:100, 1:100]

  unname(system.time({
    for (j in seq_len(runs))
      C <- A %*% B
  })[3])
})

results <- data.frame(
  backend = backends,
  `timing [s]` = timings,
  `performance [GFlops]` = (2 * (n / 1000)^3) / timings,
  check.names = FALSE)

results[order(results$performance),]
#>             backend timing [s] performance [GFlops]
#> 1           NETLIB     56.776           0.2818092
#> 5            ATLAS      5.988           2.6720107
#> 2     BLIS-THREADS      3.442           4.6484602
#> 8      BLIS-OPENMP      3.408           4.6948357
#> 4      BLIS-SERIAL      3.395           4.7128130
#> 6  OPENBLAS-SERIAL      3.206           4.9906425
#> 7 OPENBLAS-THREADS      0.773          20.6985770
#> 3  OPENBLAS-OPENMP      0.761          21.0249671
```

Add-ons

RStudio IDE

The RStudio IDE (https://posit.co/products/open-source/rstudio/), built for Fedora and accompanied by the latest stable release of Quarto, is available via the iucar/rstudio (https://copr.fedorainfracloud.org/coprs/iucar/rstudio/) Copr repository. To enable this Copr repository in your system:

```
$ sudo dnf copr enable iucar/rstudio
$ sudo dnf install rstudio-desktop     # installs the Desktop IDE
$ sudo dnf install rstudio-server      # installs the web-based IDE
```

Note that these packages are not compatible with the official RStudio packages, so you should not mix them.

Others

Additionally, the following add-ons are available in the official repositories:

Component	Description
rkward	Graphical front-end for the R language
emacs-ess	Emacs Speaks Statistics under GNU Emacs

Containerized environments

Fedora Docker images

There are official Fedora Docker images (https://hub.docker.com/_/fedora), maintained by the Fedora Release Engineering team, that can be used as base images for containerized R applications (for cloud deployments, CI/CD systems…). All the instructions above apply too for installation and maintenance of R software in these Docker images. However, you may see the following warning when installing R packages from source:

```
Warning in file.create(f.tg) :
 cannot create file '/usr/share/doc/R/html/packages.html', reason 'No
such file or directory'
Warning in utils::make.packages.html(.Library, docdir = R.home("doc")) :
 cannot update HTML package index
```

This is expected, because base Docker images add `tsflags=nodocs` to `/etc/dnf/dnf.conf` in order to minimize image sizes, and thus some documentation is missing in the R installation. However, this warning is completely harmless and can be safely ignored. If you still want to silence this warning, there are two options:

- Installing R-core with `--setopt=tsflags=` will reset `tsflags` and thus will install R's docs.
- Installing source R packages with `--no-docs` does not issue any warning.

If you are using an old version of `devtools` or `remotes` to install R packages, the warning above may have turned into an error like the following:

```
Error in file.copy(file.path(R.home("doc"), "html", "R.css"), outman) :
  (converted from warning) problem copying /usr/share/doc/R/html/R.css
to /usr/lib64/R/library/00LOCK-<package>/00new/<package>/html/R.css:
No such file or directory
```

This is a known issue (https://github.com/r-lib/devtools/issues/2084) that should be fixed in recent releases. Otherwise, setting the environment variable `R_REMOTES_NO_ERRORS_FROM_WARNINGS=true` should avoid turning installation warnings to errors.

Toolbox: container-based development

Toolbox (https://docs.fedoraproject.org/en-US/fedora-silverblue/toolbox/) enables a rootless containerized environment for everyday software development. It is best suited for immutable operating systems, such as Fedora Silverblue (https://silverblue.fedoraproject.org/), but it can be used in any Fedora or CentOS/RedHat base system. It could be used as the main development environment to avoid installing anything in the base system, or e.g. to test a new R release only available in Fedora rawhide without polluting the main installation.

As an example, let us suppose that the base system runs R 4.3 on Fedora 39, and we want to test R 4.4, available in rawhide, before Fedora 40 is released:

```
$ sudo dnf install toolbox
$ toolbox enter --release 40
[toolbox]$ sudo dnf install R
[toolbox]$ R

R version 4.4.1 (2024-06-14) -- "Race for Your Life"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-redhat-linux-gnu
```

Note that users do not need any administrative rights to install anything in their toolboxes. Once R is installed in the toolbox, it can be directly executed without entering first:

```
$ toolbox run --release 40 R
```

Your favorite IDE must be installed in the toolbox too in order to connect to the R installation there. GUIs such as RStudio Desktop should just work:

```
[toolbox]$ sudo dnf copr enable iucar/rstudio
[toolbox]$ sudo dnf install rstudio-desktop
[toolbox]$ rstudio
```

See this introduction (https://fedoramagazine.org/a-quick-introduction-to-toolbox-on-fedora/) to Toolbox on Fedora as well as the documentation (`man toolbox`) for further information.

Reporting Issues

- The best place to report issues with these packages or ask R questions specific to Fedora/EPEL is the R-SIG-Fedora mailing list (see https://stat.ethz.ch/mailman/listinfo/r-sig-fedora (https://stat.ethz.ch/mailman/listinfo/r-sig-fedora) for more information).
- To report issues about R packages in the iucar/cran (https://copr.fedorainfracloud.org/coprs/iucar/cran/) Copr repository, please open an issue at https://github.com/cran4linux/cran2copr/issues (https://github.com/cran4linux/cran2copr/issues).
- To report issues about the RStudio packages in the iucar/rstudio (https://copr.fedorainfracloud.org/coprs/iucar/rstudio/) Copr repository, please open an issue at https://github.com/Enchufa2/specs/issues (https://github.com/Enchufa2/specs/issues).
- To report issues about this page, please write to the mailing list or open a ticket at https://pagure.io/R/fedora-r-packages (https://pagure.io/R/fedora-r-packages).

Acknowledgements

Thanks to Martyn Plummer for maintaining a previous version of this page. The R stack for Fedora/EPEL is maintained by Tom “Spot” Callaway, Elliott Sales de Andrade, José Abílio Matos and Mattias Ellert among others. Additional CRAN RPMs for Fedora are maintained by Iñaki Ucar, built and distributed through Fedora Copr (https://copr.fedorainfracloud.org/). The Copr Build System is maintained by Miroslav Suchý, Pavel Raiskup, Jakub Kadlcik and many others. Thanks to Martin Koehler and Fabio Valentini for their invaluable assistance in bringing FlexiBLAS to Fedora.

mirror server hosted at Truenetwork (https://truenetwork.ru), Russian Federation.