

Collaborative Software Development Using R-Forge. Special invited paper on "The Future of R"

Theußl, Stefan; Zeileis, Achim

Published in:
The R Journal

Published: 01/05/2009

Document Version
Publisher's PDF, also known as Version of record

[Link to publication](#)

Citation for published version (APA):
Theußl, S., & Zeileis, A. (2009). Collaborative Software Development Using R-Forge. Special invited paper on "The Future of R". *The R Journal*, 1(1), 9-14.

Collaborative Software Development Using R-Forge

Special invited paper on “The Future of R”

by Stefan Theußl and Achim Zeileis

Introduction

Open source software (OSS) is typically created in a decentralized self-organizing process by a community of developers having the same or similar interests (see the famous essay by Raymond, 1999). A key factor for the success of OSS over the last two decades is the Internet: Developers who rarely meet face-to-face can employ new means of communication, both for rapidly writing and deploying software (in the spirit of Linus Torvald’s “release early, release often paradigm”). Therefore, many tools emerged that assist a collaborative software development process, including in particular tools for source code management (SCM) and version control.

In the R world, SCM is not a new idea; in fact, the R Development Core Team has always been using SCM tools for the R sources, first by means of Concurrent Versions System (CVS, see Cederqvist et al., 2006), and then via Subversion (SVN, see Pilato et al., 2004). A central repository is hosted by ETH Zürich mainly for managing the development of the base R system. Mailing lists like R-help, R-devel and many others are currently the main communication channels in the R community.

Also beyond the base system, many R contributors employ SCM tools for managing their R packages, e.g., via web-based SVN repositories like SourceForge (<http://SourceForge.net/>) or Google Code (<http://Code.Google.com/>). However, there has been no central SCM repository providing services suited to the specific needs of R package developers. Since early 2007, the R-project offers such a central platform to the R community. R-Forge (<http://R-Forge.R-project.org/>) provides a set of tools for source code management and various web-based features. It aims to provide a platform for collaborative development of R packages, R-related software or further projects. R-Forge is closely related to the most famous of such platforms—the world’s largest OSS development website—namely <http://SourceForge.net/>.

The remainder of this article is organized as follows. First, we present the core features that R-Forge offers to the R community. Second, we give a hands-on tutorial on how users and developers can get started with R-Forge. In particular, we illustrate how people can register, set up new projects, use R-Forge’s SCM facilities, provide their packages on R-

Forge, host a project-specific website, and how package maintainers submit a package to the Comprehensive R Archive Network (CRAN, <http://CRAN.R-project.org/>). Finally, we summarize recent developments and give a brief outlook to future work.

R-Forge

R-Forge offers a central platform for the development of R packages, R-related software and other projects.

R-Forge is based on GForge (Copeland et al., 2006) which is an open source fork of the 2.61 SourceForge code maintained by Tim Perdue, one of the original SourceForge authors. GForge has been modified to provide additional features for the R community, namely a CRAN-style repository for hosting development releases of R packages as well as a quality management system similar to that of CRAN. Packages hosted on R-Forge are provided in source form as well as in binary form for Mac OS X and Windows. They can be downloaded from the website of the corresponding project on R-Forge or installed directly in R; for a package `foo`, say, `install.packages("foo", repos = "http://R-Forge.R-project.org")`.

On R-Forge, developers organize their work in so-called “Projects”. Every project has various tools and web-based features for software development, communication and other services. All features mentioned in the following sections are accessible via so-called “Tabs”: e.g., user accounts can be managed in the *My Page* tab or a list of available projects can be displayed using the *Project Tree* tab.

Since starting the platform in early 2007, more and more interested users registered their projects on R-Forge. Now, after more than two years of development and testing, around 350 projects and more than 900 users are registered on R-Forge. This and the steadily growing list of feature requests show that there is a high demand for centralized source code management tools and for releasing prototype code frequently among the R community.

In the next three sections, we summarize the core features of R-Forge and what R-Forge offers to the R community in terms of collaborative development of R-related software projects.

Source code management

When carrying out software projects, source files change over time, new files get created and old files deleted. Typically, several authors work on several computers on the same and/or different files and

keeping track of every change can become a tedious task. In the open source community, the general solution to this problem is to use version control, typically provided by the majority of SCM tools. For this reason R-Forge utilizes SVN to facilitate the developer's work when creating software.

A central repository ensures that the developer always has access to the current version of the project's source code. Any of the authorized collaborators can "check out" (i.e., download) or "update" the project file structure, make the necessary changes or additions, delete files from the current revision and finally "commit" changes or additions to the repository. More than that, SVN keeps track of the complete history of the project file structure. At any point in the development stage it is possible to go back to any previous stage in the history to inspect and restore old files. This is called version control, as every stage automatically is assigned a unique version number which increases over time.

On R-Forge such a version-controlled repository is automatically created for each project. To get started, the project members just have to install the client of their choice (e.g., Tortoise SVN on Windows or svnX on Mac OS X) and check out the repository. In addition to the inherent backup of every version within the repository a backup of the whole repository is generated daily.

A rights management system assures that, by default, anonymous users have read access and developers have write access to the data associated with a certain project on R-Forge. More precisely, registered users can be granted one of several roles: e.g., the "Administrator" has all rights including the right to add new users to the project or release packages directly to CRAN. He/she is usually the package maintainer, the project leader or has registered the project originally. Other members of a project typically have either the role "Senior Developer" or "Junior Developer" which both have permission to commit to the project SVN repository and examine the log files in the *R Packages* tab (the differences between the two roles are subtle, e.g., senior developers additionally have administrative privileges in several places in the project). When we speak of developers in subsequent sections we refer to project members having the rights at least of a junior developer.

Release and quality management

Development versions of a software project are typically prototypes and are subject to many changes. Thus, R-Forge offers two tools which assist the developers in improving the quality of their source code.

First, it offers a quality management system similar to that of CRAN. Packages on R-Forge are checked in a standardized way on different platforms based on `R CMD check` at least once daily. The resulting log files can be examined by the project de-

velopers so that they can improve the package to pass all tests on R-Forge and subsequently on CRAN.

Second, bug tracking systems allow users to notify package authors about problems they encounter. In the spirit of OSS—given enough eyeballs, all bugs are shallow (Raymond, 1999)—peer code review leads to an overall improvement of the quality of software projects.

Additional features

A number of further tools, of increased interest for larger projects, help developers to coordinate their work and to communicate with their user base. These tools include:

- **Project websites:** Developers may present their work on a subdomain of R-Forge, e.g., `http://foo.R-Forge.R-project.org/`, or via a link to an external website.
- **Mailing lists:** By default a list `foo-commits@lists.R-Forge.R-project.org` is automatically created for each project. Additional mailing lists can be set up easily.
- **Project categorization:** Administrators may categorize their project in several so-called "Trove Categories" in the *Admin* tab of their project (under *Trove Categorization*). For each category three different items can be selected. This enables users to quickly find what they are looking for using the *Project Tree* tab.
- **News:** Announcements and other information related to a project can be put on the project summary page as well as on the home page of R-Forge. The latter needs approval by one of the R-Forge administrators. All items are available as RSS feeds.
- **Forums:** Discussion forums can be set up separately by the project administrators.

How to get started

This section is intended to be a hands-on tutorial for new users. Depending on familiarity with the systems/tools involved the instructions might be somewhat brief. In case of problems we recommend consulting the user's manual (R-Forge Administration and Development Team, 2008) which contains detailed step-by-step instructions.

When accessing the URL `http://R-Forge.R-project.org/` the home page is presented (see Figure 1). Here one can

- login,
- register a user or a project,

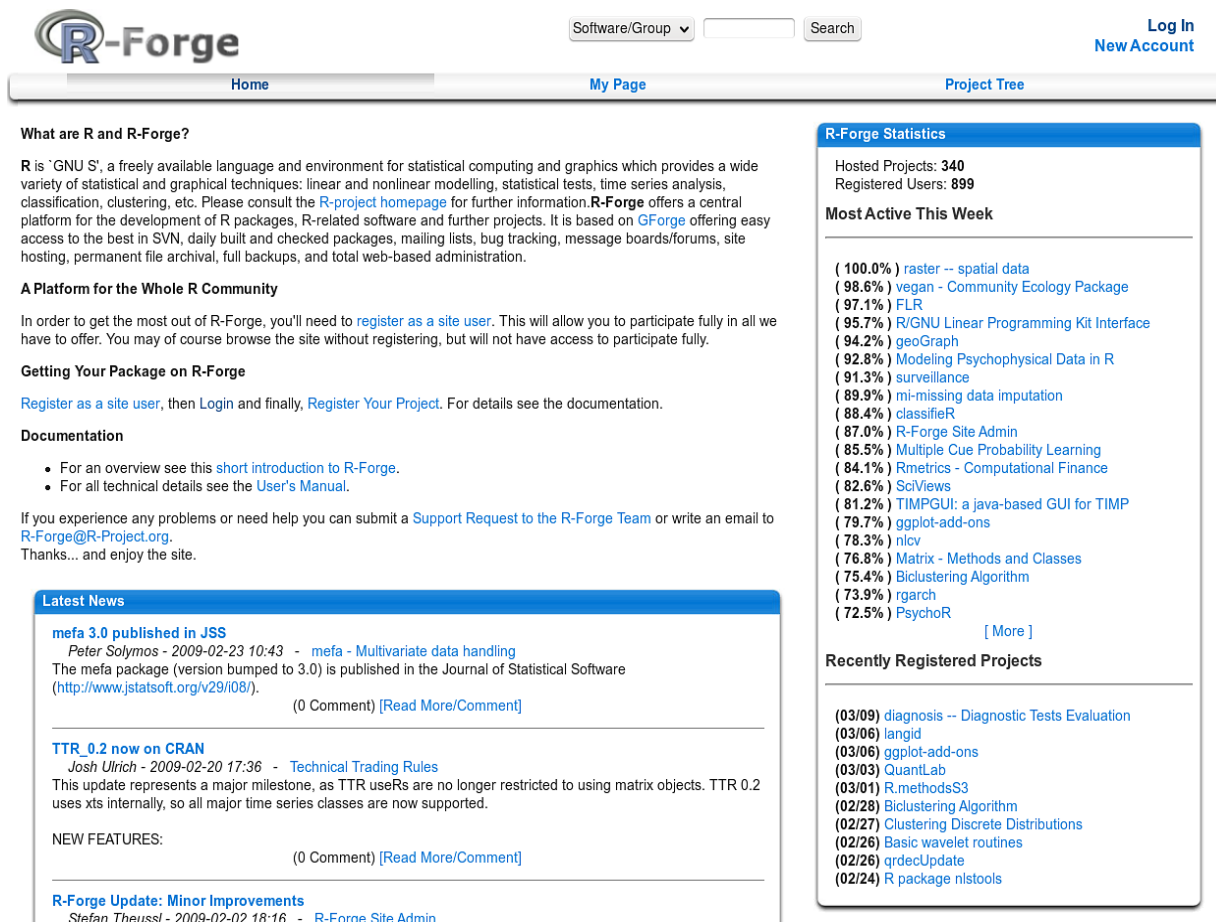


Figure 1: Home page of R-Forge on 2009-03-10

- download the documentation,
- examine the latest news and changes,
- go to a specific project website either by searching for available projects (top middle of the page), by clicking on one of the projects listed on the right, or by going through the listing in the *Project Tree* tab.

Registered users can access their personal page via the tab named *My Page*. Figure 2 shows the personal page of the first author.

Registering as a new user

To use R-Forge as a developer, one has to register as a site user. A link on the main web site called *New Account* on the top right of the home page leads to the corresponding registration form.

After submitting the completed form, an e-mail is sent to the given address containing a link for activating the account. Subsequently, all R-Forge features, including joining existing or creating new projects, are available to logged-in users.

Registering a project

There are two possibilities to register a project: Clicking on *Register Your Project* on the home page or going to the *My Page* tab and clicking on *Register Project* (see Figure 2 below the main tabs). Both links lead to a form which has to be filled out in order to finish the registration process. In the text field “Project Public Description” the registrant is supposed to enter a concise description of the project. This text and the “Project Full Name” will be shown in several places on R-Forge (see e.g., Figure 1). The text entered in the field “Project Purpose And Summarization” is additional information for the R-Forge administrators, inspected for approval. “Project Unix Name” refers to the name which uniquely determines the project. In the case of a project that contains a single R package, the project Unix name typically corresponds to the package name (in its lower-case version). Restrictions according to the Unix file system convention force Unix names to be in lower case (and will be converted automatically if they are typed in upper case).

After the completed form is submitted, the project has to be approved by the R-Forge administrators and a confirmation e-mail is sent to the registrant upon approval. After that, the regis-

Figure 2: The *My Page* tab of the first author

trant automatically becomes the project administrator and the standardized web area of the project (<http://R-Forge.R-project.org/projects/foo/>) is immediately available on R-Forge. This web area includes a *Summary* page, an *Admin* tab visible only to the project administrators, and various other tabs depending on the features enabled for this project in the *Admin* tab. To present the new project to a broader community the name of the project additionally is promoted on the home page under “Recently Registered Projects” (see Figure 1).

Furthermore, within an hour after approval a default mailing list and the project’s SVN repository containing a ‘README’ file and two pre-defined directories called ‘pkg’ and ‘www’ are created. The content of these directories is used by the R-Forge server for creating R packages and a project website, respectively.

SCM and R packages

The first step after creation of a project is typically to start generation of content for one (or more) R package(s) in the ‘pkg’ directory. Developers can either start committing changes via SVN as usual or—

if the package is already version-controlled somewhere else—the corresponding parts of the repository including the history can be migrated to R-Forge (see Section 6 of the user’s manual).

The *SCM* tab of a project explains how the corresponding SVN repository located at `svn://svn.R-Forge.R-project.org/svnroot/foo` can be checked out. From this URL the sources are checked out either anonymously without write permission (enabled by default) or as developer using an encrypted authentication method, namely secure shell (ssh). Via this secure protocol (`svn+ssh://` followed by the registered user name) developers have full access to the repository. Typically, the developer is authenticated via the registered password but it is also possible to upload a secure shell key (updated hourly) to make use of public/private key authentication. Section 3.2 of the user’s manual explains this process in more detail.

To make use of the package builds and checks the package source code has to be put into the ‘pkg’ directory of the repository (i.e., ‘pkg/DESCRIPTION’, ‘pkg/R’, ‘pkg/man’, etc.) or, alternatively, a subdirectory of ‘pkg’. The latter structure allows developers to have more than one package in a single project; e.g.,

if a project consists of the packages `foo` and `bar`, then the source code is located in `'pkg/foo'` and `'pkg/bar'`, respectively.

R-Forge automatically examines the `'pkg'` directory of every repository and builds the package sources as well as the package binaries on a daily basis for Mac OS X and Windows (if applicable). The package builds are provided in the *R Packages* tab (see Figure 3) for download or can be installed directly in R using `install.packages("foo", repos="http://R-Forge.R-project.org")`. Furthermore, in the *R Packages* tab developers can examine logs of the build and check process on different platforms.

To release a package to CRAN the project administrator clicks on *Submit this package to CRAN* in the *R Packages* tab. Upon confirmation, a message will be sent to `CRAN@R-project.org` and the latest successfully built source package (the `'tar.gz'` file) is automatically copied to `ftp://CRAN.R-project.org/incoming/`. Note that packages are built once daily, i.e., the latest source package does not include more recent code committed to the SVN repository.

Figure 3 shows the *R Packages* tab of the `tm` project (<http://R-Forge.R-project.org/projects/tm/>) as one of the project administrators would see it. Depending on whether you are a member of the project or not and your rights you will see only parts of the information provided for each package.

Further steps

A customized project website, accessible through <http://foo.R-Forge.R-project.org/> where `foo` corresponds to the unix name of the project, is managed via the `'www'` directory. The website gets updated every hour.

The changes made to the project can be examined by entering the corresponding standardized web area. On entry, the *Summary* page is shown. Here, one can

- examine the details of the project including a short description and a listing of the administrators and developers,
- follow a link leading to the project homepage,
- examine the latest news announcements (if available),
- go to other sections of the project like *Forums*, *Tracker*, *Lists*, *R Packages*, etc.
- follow the download link leading directly to the available packages of the project (i.e., the *R Packages* tab).

Furthermore, meta-information about a project can be supplied in the *Admin* tab via so-called "Trove Categorization".

Recent and future developments

In this section, we briefly summarize the major changes and updates to the R-Forge system during the last few months and give an outlook to future developments.

Recently added features and major changes include:

- New defaults for freshly registered projects: Only the tabs *Lists*, *SCM* and *R packages* are enabled initially. *Forums*, *Tracker* and *News* can be activated separately (in order not to overwhelm new users with too many features) using *Edit Public Info* in the *Admin* tab of the project. Experienced users can decide which features they want and activate them.
- An enhanced structure in the SVN repository allowing multiple packages in a single project (see above).
- The R package **RForgeTools** (Theußl, 2009) contains platform-independent package building and quality management code used on the R-Forge servers.
- A modified *News* submit page offering two types of submissions: project-specific and global news. The latter needs approval by one of the R-Forge administrators (default: project-only submission).
- Circulation of SVN commit messages can be enabled in the *SCM Admin* tab by project administrators. The mailing list mentioned in the text field is used for delivering the SVN commit messages (default: off).
- Mailing list search facilities provided by the Swish-e engine which can be accessed via the *List* tab (private lists are not included in the search index).

Further features and improvements which are currently on the wishlist or under development include

- a Wiki,
- task management facilities,
- a re-organized tracker more compatible with R package development and,
- an update of the underlying GForge system to its successor FusionForge (<http://FusionForge.org/>).

For suggestions, problems, feature requests, and other questions regarding R-Forge please contact `R-Forge@R-project.org`.

R-Forge Search the entire project Search [Advanced search](#) [Log Out My Account](#)

Home My Page Project Tree **tm - Text Mining Package**

Summary Admin Lists SCM R Packages

R Development Page | Admin

Contributed R Packages

Below is a list of all packages provided by project **tm - Text Mining Package**.

Important note for package binaries: R-Forge provides these binaries only for versions of R that were released since the package was first submitted to R-Forge, but not for older versions. In order to use more recent packages from R-Forge you may need to switch to a newer version of R or, alternatively, try to build the package with an older version of R. Packages are built/checked according to this [schedule](#).

tm - Text Mining Package

A framework for text mining applications within R.

Download: [Package source \(.tar.gz\)](#) | [Windows binary \(.zip\)](#) | [MacOS X universal binary \(.tgz\)](#)

Logs:	Linux x86_32	Linux x86_64	Windows x86_32	MacOS X universal
Daily build:	patched	patched	patched devel	patched devel (N/A)
Daily check:	patched devel	patched devel	patched devel	patched devel (N/A)

[\[Submit this package to CRAN\]](#)

To install this package directly within R type: `install.packages("tm", repos="http://R-Forge.R-project.org")`

Version: 0.3-3.2 | Last change: 2009-03-03 17:23:22+01 | Rev.: 894
Stable Release: [Get tm 0.3-3 from CRAN](#)

Figure 3: R Packages tab of the **tm** project

Acknowledgments

Setting up this project would not have been possible without Douglas Bates and the University of Wisconsin as they provided us with a server for hosting this platform. Furthermore, the authors would like to thank the Computer Center of the Wirtschaftsuniversität Wien for their support and for providing us with additional hardware as well as a professional server infrastructure.

Bibliography

- P. Cederqvist et al. *Version Management with CVS*. Network Theory Limited, Bristol, 2006. Full book available online at <http://ximbiot.com/cvs/manual/>.
- T. Copeland, R. Mas, K. McCullagh, T. Perdue, G. Smet, and R. Spisser. *GForge Manual*, 2006. URL http://gforge.org/gf/download/docmanfileversion/8/1700/gforge_manual.pdf.
- C. M. Pilato, B. Collins-Sussman, and B. W. Fitzpatrick. *Version Control with Subversion*. O'Reilly,

2004. Full book available online at <http://svnbook.red-bean.com/>.

R-Forge Administration and Development Team. *R-Forge User's Manual*, 2008. URL <http://download.R-Forge.R-project.org/R-Forge.pdf>.

E. S. Raymond. The Cathedral and the Bazaar. *Knowledge, Technology, and Policy*, 12(3):23–49, 1999.

S. Theußl. *RForgeTools: R-Forge Build and Check Tools*, 2009. URL <http://R-Forge.R-project.org/projects/site/>. R package version 0.3-3.

Stefan Theußl
Department of Statistics and Mathematics
WU Wirtschaftsuniversität Wien
Austria
stefan.theussl@wu.ac.at

Achim Zeileis
Department of Statistics and Mathematics
WU Wirtschaftsuniversität Wien
Austria
achim.zeileis@wu.ac.at