

R-Forge 用户手册，测试版

SVN 修订号: 102, 2009 年 3 月 10 日

R-Forge 管理和开发小组

Copyright 2006–2008 R-Forge 管理和开发小组



本手册所含内容采用知识共享署名 - 相同方式共享 2.0 许可协议授权。

R-Forge 管理和开发小组选择使用知识共享公共授权协议 (Creative Commons Attribution License, CCAL) 授权此文档, 即, 在知识共享公共授权协议下, 作者保留此手册的版权所有, 在注明原始作者和来源的前提下, 作者允许任何人下载, 重新使用, 重印, 修改, 分发, 和 / 或复制此手册的内容. 另外, 仅在相同协议或兼容于管理作者作品的协议下, 作者许可他人分发衍生品. 使用这份不受限制的协议意在促进开放取用和自由使用所有类型的作品. 将这份标准协议应用到此作品可确保作者保持此作品可被自由和公开取得的权利. (参考 <http://creativecommons.org/licenses/by-sa/2.0/>以获取详情).

R-Forge 管理和开发小组的现有成员为 Kurt Hornik, Martin Kober, David Meyer, Stefan Theußl and Achim Zeileis. 联系以上成员, 请发送电子邮件至 R-Forge@R-project.org.

目 录

1	引言	1
2	注册	2
2.1	注册新用户	2
2.2	参与项目	2
2.3	注册新项目	2
3	源代码管理	2
3.1	简介	2
3.2	如何使 SVN 正常工作	3
3.2.1	Windows	3
3.2.2	Linux/Unix	4
3.2.3	Mac OS	4
4	开发 R 包	5
4.1	构建与检查	5
4.2	外部 SVN	6
4.3	提交 R 包到 CRAN	6
5	附加功能	6
5.1	项目主页	6
5.2	SCM 选项	6
5.3	邮件列表	6
6	从现有 CVS/SVN 版本库迁移	7
7	致谢	7

1 引言

十年以来, R 核心开发小组都在使用如 Subversion (SVN, 参见 Pilato et al., 2004) 或由苏黎世联邦理工学院提供的 Concurrent Versions System (CVS, 参见 Cederqvist et al., 2006) 一类的协作开发工具. 由于协作开发是许多开源项目成功的关键因素之一, 所以世界范围内许多 R 包开发者都使用类似的, 基于各自解决方案的协作开发平台. 因此, **R-project** 不仅要为 R 核心开发小组, 更要为整个 R 社区提供协作开发平台.

R-Forge (<http://R-Forge.R-project.org>) 提供了一系列源代码管理工具和多种基于 Web 页面的功能. 它旨在为 R 包、R 相关软件及更多项目提供一个协作开发平台. 它与此类平台的个中翘楚——世界上规模最大的开源软件 (OSS) 开发网站——<http://SourceForge.net> 密切相关. 区别在于, 我们提供的工具集更加专注于 R 社区的需求. 总而言之, R-Forge 是一个可以让所有 R 开发者和用户聚集并交换知识的地方. 加入这项开源社区服务可以让你能够使用我们提供的所有功能.

大致来说, R-Forge 是基于 GForge (Copeland et al., 2006) 的. 后者是集成了许多类似于实现代码协作的 SVN, 邮件列表, bug 跟踪等许多工具于同一平台的一个框架. 在 R-Forge 上, 所有的工作都使用所谓的“项目” (Project) 来进行组织. 在这些“项目”中可以主持编写一个或多个包. 这样做的主要优点是, 您的包不仅每天会在最新版本的 R 下被构建 (build) 和检查 (check), 还会在一个新版本的 R 发布之前的候选版本下被构建和检查. 开发者不得不在本地建立平台无关的构建和检查环境的时代已经一去不返了. 并且, 已通过 R-Forge 质量管理体系测试的包, 能够以一种标准化的方式被直接发布至 CRAN.

此外, R-Forge 还特别为更大型项目的项目成员工作协调及沟通提供了其他工具.

- 项目站点是开发者在 R-Forge 的二级域名上 (例如 <http://foo.R-forge.R-project.org>) 展示各自工作的一种手段. 您也可以在项目概述页面 (project summary page) 提供到另外一个站点的链接作为替代.
- 邮件列表: 当建立一个项目时, 系统将默认自动创建一个邮件列表. 额外的邮件列表也可轻松建立.
- 不同的项目可被归类到不同的主题内 (例如, 生物统计, 金融, 回归分析, ...). 这使得他人能够迅速找到想要的东西. 人们可以在所谓的“项目树” (Project Tree) 标签中浏览这些类别. 默认设置下, 项目树与 CRAN 类似地依字母顺序列出所有项目及其简短描述. 很明显, 这个列表不可能很完整, 欢迎大家提出改进建议.
- 项目管理员可以单独设立论坛 (Forums).
- 更新动态 (News) 除了可以被提交至项目概述页面, 还可以被提交至网站首页. 后者需要得到 R-Forge 管理和开发小组中的某个成员批准. 您还可以通过 RSS feeds 订阅这些更新动态.

这份手册介绍了开始使用 R-Forge 的所有相关步骤. 如果您需要更多帮助或持有关于 R-Forge 的高见, 请发送邮件至 R-Forge@R-project.org. 关于底层 GForge 系统的详细文档, 请参考 Copeland et al. (2006).

2 注册

2.1 注册新用户

要注册一个新用户, 请前往 <http://R-Forge.R-project.org>, 点击页面右上角的 “New Account”. 填写表单 (本站的每个文本框后均有说明和提示), 最后点击 “Submit”. 您指定的电子邮件地址将收到一封包含一个链接的电子邮件. 点击此链接后, 您的账户即被成功验证. 此时即可登录.

特别注意: “My Page” 是 R-Forge 上最重要的页面. 在这里您可以进行账户设置, 了解被分配项目的情况和相关事项 (即 bug 反馈, 功能需求反馈, 等等.).

现在, 您可以开始着手建立自己的项目 (详情参阅第 2.3 节) 或成为已有项目的开发成员 (第 2.2 节).

2.2 参与项目

要参与一个已有项目, 步骤如下:

1. 首先您需要获取您要参与项目的名称. 您可以搜索项目 (页面最上方中间部分) 或者点击 “Project Tree” 中显示的某个项目. 后者包含 R-Forge 上的所有项目 (依字母顺序排列).
2. 前往项目概述页面 (应该为默认的入口页面). 此页面右侧有一个叫做 “Developer Info” 的小窗口. 要参与这个项目首先需要得到该项目管理者的许可. 所以我们点击 “Request to join” 给项目管理者发送一封电子邮件.
3. 如果项目管理者决定添加您为项目开发者, 您将收到一封确认您的开发者账户的邮件. 这样, 您就拥有了完整的 SVN 访问权限 (详情参阅第 3 节).

2.3 注册新项目

注册一个新项目很简单: 打开 R-Forge 网站, 登录后进入 “My Page”. 在页面顶端的菜单有一个 “Register Project” 链接. 填好表格就提交您的项目吧. 经过 R-Forge 管理员的人工审核后您会收到通知邮件, 之后就可以开始在 R-Forge 上开发您的项目了.

3 源代码管理

3.1 简介

进行软件项目开发时, 源文件经常被修改, 文件增删更是常见的事情. 一般来说, 数位开发人员在各自的计算机上编辑相同的和 (或) 不同的文件, 并跟踪每一个修改会是一项乏味的工作. 在开源社区中, 这个问题通行的解决方案是使用版本控制系统, 这一般是由多数软件配置管理工具提供的. 为此, R-Forge 使用了 SVN 来促进开发人员的软件编写工作.

一个中央版本库 (repository) 保证了开发人员一直能够访问项目源代码的最新版本. 任何已被授权的合作开发者均可以 “检出” (checkout, 即下载) 或 “更新”(update) 项目的文件

结构, 做必要的修改或者从当前的修订版本增加、删除文件并最终“提交”(commit) 您的修改和增删项到版本库中. 不仅如此, SVN 还会跟踪项目文件结构的完整历史. 在开发阶段的任何时间点, 回溯到历史上的任何阶段并检查和还原旧文件成为了可能. 由于每个开发阶段均会被自动分配一个随时间增长且独一无二的版本号, 所以我们称其为版本控制.

在 R-Forge 上, 像这样版本可控的版本库是为每个项目自动创建的. 要开始使用, 项目开发成员只需安装各自选择的客户端 (例如, Windows 平台的 Tortoise SVN 或 Mac OSX 平台的 svnX) 并检出对应版本库即可. 除了版本库原生的各个版本的备份功能之外, 一个完整的版本库备份将在每天自动生成一次.

R-Forge 使用的权限管理系统默认保证了对于某个项目相关的数据, 匿名用户拥有读取权限, 开发者拥有写权限. 准确地讲, 已注册用户可以被授予一种或多种角色 (role): 例如, “管理员”(Administrator) 拥有所有权限, 包括向项目添加新成员或向 CRAN 直接发布 R 包等. 他 (她) 通常是此包的维护者 (maintainer), 项目主管 (project leader) 或该项目的原始注册者. 一个项目的其他成员通常会拥有 “高级开发者”(Senior Developer) 或初级开发者 (Junior Developer) 的角色, 这两种角色均有向 SVN 版本库提交和检查 *R Packages* 标签下日志文件的权限. 这两种角色之间的区别是很小的, 例如, 高级开发者在项目的其他几处还拥有额外的管理权限. 随后章节中提及的 “开发者” 是指至少拥有初级开发者权限的项目成员.

3.2 如何使 SVN 正常工作

R-Forge 使用 **Subversion** (SVN, <http://subversion.tigris.org>) 进行源代码管理. 您需要一个 SVN 客户端 (如 Windows 平台上的 “Tortoise SVN”, <http://tortoisesvn.tigris.org>) 来充分利用您的 SVN 版本库. 出于安全考虑, 我们为开发者账户使用了安全外壳 (secure shell, SSH) 隧道, 这意味着所有的网络传输均是加密的. 这也意味着您需要根据相应情况配置您的系统.

3.2.1 Windows

本节提到的软件可在如下地址找到:

<http://www.chiark.greenend.org.uk/~sgtatham/putty/>.

1. 从 <http://tortoisesvn.net/downloads> 获取并安装最新的 TortoiseSVN 客户端. 注意: Windows Vista 用户需要注意选择对应自身系统架构的安装包 (即, Windows Vista 64 位版本的用户需要下载 64 位版本的安装包).
2. 使用密码认证来写入项目版本库已经足够 (如果您决定这样做, 请直接跳至步骤 5). 但为了方便起见 (避免询问密码), 生成一个 SSH 密钥对并上传公钥 (**public key**) 到 R-Forge 可能是更好的选择:

- 下载并运行 `puttygen.exe`.
- 使 `ssh-2 rsa` 保持勾选状态并点击 “generate” (然后随便移动一下鼠标).
- 点击相应的按钮以保存私钥 (private key) (不需要设定密码).
- 高亮选中文本框 “Public key for pasting into OpenSSH authorized_keys file” 中的文本.

- 复制并粘贴这段文本到您的 shell 账户信息设置 (shell account information configuration) 中: 打开 <http://R-forge.R-project.org> 并登录. 进入 “My Page” 然后点击 “Account maintenance”. 在页面的底部的 “Shell Account Information” 窗口点击 “edit keys” (**特别注意:** 以上操作需要该项目的管理员或开发成员权限, 否则选项 “Shell Account Information” 不会出现, 参阅第 2.2 节以参与一个项目或第 2.3 节以注册新项目). 您在这里输入密钥的典型形式如下:

```
ssh-rsa AAAA... foo@bar
```

第一个区域描述了密钥类型, 第二个区域即为密钥本身, 第三个区域是一段注释. 保证一个密钥中没有换行是非常重要的.

3. 现在您需要等待一个小时. 密钥每隔一小时才激活一次.
4. 下一步您需要一个类似于 `pageant.exe` 的认证工具. 使用 `pageant.exe` 读取您的私钥 (**private key**) (右键点击 `pageant` 的托盘图标然后点击 “add key”).
5. 最后, 使用 “Tortoise SVN” 或您选择的 SVN 客户端配合项目网站上给出的 URL(见 “developer account” 下的标签 “SCM”) 检出版本库. (注意: 如果这最后一步由于某种原因失败, 等待一个小时后再尝试一次. 因为每小时仅可登录一次.)

3.2.2 Linux/Unix

1. 使用密码认证来写入项目版本库已经足够 (如果您决定这样做, 请直接跳至步骤 4). 但为了方便起见 (避免询问密码), 上传一个 SSH 密钥对可能是更好的选择: 在命令行中使用 `ssh-keygen` 生成并保存密钥或使用您已有的密钥对.
2. 使用如下网站上传公钥 (**public key**) 到 R-Forge: 打开 “My Page”, 然后点击 “Account maintenance”. 在此页面底部点击 “Shell Account Information” 窗口中的 “edit keys” (**特别注意:** 以上操作需要该项目的管理员或开发成员权限, 否则 “Shell Account Information” 不会出现, 参阅第 2.2 节以参与一个项目或第 2.3 节以注册新项目). 您在这里输入密钥的典型形式如下:

```
ssh-dsa AAAA... foo@bar
```

第一个区域描述了密钥类型, 第二个区域即为密钥本身, 第三个区域是一段注释. 保证一个密钥中没有换行是非常重要的.

3. 现在您需要等待一个小时 (等待密钥被激活).
4. 最后, 使用 `svn checkout` 或您选择的 SVN 客户端配合项目网站上给出的 URL(见 “developer account” 下的标签 “SCM”) 检出版本库. (注意: 如果这最后一步由于某种原因失败, 等待一个小时后再尝试一次. 因为每小时仅可登录一次.)

3.2.3 Mac OS

Mac OS 用户请参考 Unix 的相关章节 (3.2.2). 您亦可选择使用 GUI 程序来管理 SVN 版本库及 SSH 密钥对.

1. 您可以使用 SSHKeychain (<http://www.sshkeychain.org/>) 来管理 SSH 密钥对。您可以在 SSHKeychain 的 'Preferences...' 对话框中生成您的密钥链 (keychains), 但是您必须提供一个至少有 5 位字符的口令 (passphrase)。如果您不想使用口令这种形式, 在终端中使用 'ssh-keygen' (使用 SCPlugin 要求有一个不含口令的密钥链, 见第 4 步)。在 SSHKeychain 中, 选择 'Agent' - 'Add single key...' 然后选择您刚刚生成的密钥。
2. 如针对 Unix 的 3.2.2 节阐述的一样, 上传公钥 (public key) 到 R-Forge。公钥被保存在和私钥相同目录下的文件 [keyfile].pub 中。如果您在生成密钥时使用了默认的目录和名称, 这些文件会在 home 路径下的一个名为 '.ssh' 的隐藏目录下, 并因此在 Finder 中是不可见的。要复制公钥到您的桌面上, 请在终端键入以下命令 (替换为您自己的文件名):

```
cp ~/.ssh/id_rsa.pub /Desktop/id_rsa.pub
```

然后, 将此文件上传到 R-Forge。
3. 等待一个小时等待密钥被激活。
4. 如果您想在 Finder 中方便地管理您的 SVN 版本库, 请安装 SCPlugin。要安装此软件, 请根据 <http://scplugin.tigris.org/installation.html> 页面的说明安装。一旦安装了 SCPlugin, 在 Finder 中前往您想放置 SVN 版本库本地副本的目录。创建一个以您的项目名称命名的文件夹, 右键点击此文件夹并选择 *Subversion -> Checkout*。输入项目网站上给出的 URL (见 “developer account” 下的标签 “SCM”)。(注意: 如果这最后一步失败, 等待一个小时后再试一次。因为每小时仅可登录一次)。

Mac OS 平台上安装 SVN 的另外一个参考页面是 <http://www.wikihow.com/Install-Subversion-on-Mac-OS-X>。在此有提及可选的 SVN 软件。

4 开发 R 包

R-Forge 旨在为 R 包、与 R 相关的软件和其他更多项目提供一个协作开发平台。特别地, 为 R 包开发提供良好的基础设施是 R-Forge 管理和开发小组的主要任务。被提交到 SVN 版本库的包会在多种平台上构建 (build) 和检查 (check), 包括 Linux, Windows(32 位) 以及 Mac OS。

4.1 构建与检查

要构建 (build) 您自己的包, 仅仅把您的 (多个) 包放入 SVN 版本库中的 **pkg/** 目录 (参阅第 3.2 节的实现方法) 并提交 (check in) 就可以了。通常此目录同时包含了 R 包 (和它一起惯有的 DESCRIPTION 文件) 以及 **R/**, **man/**, **data/** 目录 (参考 R Development Core Team (2006) 获取更多细节), 或包含了两个或更多的存放包源码的目录, 即, 通过在不同的子文件夹中放置包, 您可以同时构建多于一个包。如 **pkg/foo/**, **pkg/bar/**, 等。

每个包将在每天晚上 (欧洲中部时间) 被自动检出 (check out), 在数种平台上被构建和检查。(确认您的 DESCRIPTION 文件格式规范且包含一个正确的包名称, 否则您的包可能不会被检出)。所有被成功检出的包将在 R packages 标签中的项目主页出现。项目成员也拥有查看在所有支持平台上**检查和构建日志**的权限。

此外, 这些包将在一个类似于 CRAN 风格的版本库——R-Forge package repository 中被设为可用. 这样就可以在 R 命令行中简单地键入如下命令来安装这些包

```
install.packages("foo", repos = "http://R-Forge.R-project.org")
```

要从其他版本库中安装其依赖的包, 也可以使用类似的命令, 如

```
install.packages("mypackage", repos = c("http://R-Forge.R-project.org",  
"http://your.nearest.CRAN.mirror.org"), dep = TRUE)
```

4.2 外部 SVN

希望使用自己的外部版本库的开发者可以使用 R packages Admin 标签中的 **external SVN** 功能 (按照网站上的说明操作). 请注意, 只有可匿名访问并使用 R-Forge 版本库结构的版本库 (即, pkg 目录在版本库中可以访问) 可被支持. 并行使用外部和内部 SVN 版本库是可行的. 不论来源于哪里, 所有的包都将出现在 R 的 packages 标签下.

如果您决定将您现有的版本库迁移到 R-Forge, 请参阅第 6 节.

4.3 提交 R 包到 CRAN

通过 Web 界面, 您的 R 包可以被轻松提交至 **CRAN**. 注意, 要执行此任务, 您必须是该项目的管理者, 即该包的维护者 (maintainer).

5 附加功能

5.1 项目主页

R-Forge 上的每个项目均拥有自己的主页 (Homepage). 对应的 URL 为 `http://foo.r-forge.r-project.org/`, 其中 foo 为此项目的 Unix name¹. 使用您的 SVN 版本库中预定义的 **www** 目录来创建或修改您的主页. 注意, 此页面仅在每天被检出 (check out) 一次, 所以请您记住, 在刚刚提交 (commit) 你的修改或更新之后, 这些修改不会即时生效. 同时也请注意, 未来将仅允许使用 html 或 xhtml 构建页面.

5.2 SCM 选项

可以通过 SCM Admin tab 设置两个选项:

1. Enable anonymous access 其意自明. 一旦禁用, 则仅有项目维护成员可以访问 SVN 版本库且**包将不会被检出及进行自动构建**. 但项目主页仍然可用. 默认启用.
2. Delivery of Commit Messages 此项可以允许向您选择的邮件地址直接发送消息邮件, 例如, 此项目的 commit 邮件列表 (参考下节). 默认禁用.

5.3 邮件列表

R-Forge 为 Lists 标签下可以访问的所有项目提供了邮件列表服务.

一个名为 `foo-commits` 的邮件列表随每个项目的建立而被创建 (foo 为项目的 Unix name), 并且可被用于分发提交通知 (commit messages) (参阅第 5.2 节以了解如何启用此功能). 项目管理员可以为他们的项目创建和管理额外的邮件列表.

¹即一个符合 Unix 命名规则的名字——译者注.

此外, 搜索邮件列表 (**searching the lists**) 功能是由 Swish-e engine 提供的, 并且可以通过 List 标签访问. 请注意, 私密列表不在搜索范围内.

6 从现有 CVS/SVN 版本库迁移

在您的 R-Forge 项目中包含您现有版本库中完整历史的步骤是:

1. 首先在 R-Forge 上注册新项目.
2. 如果您使用 CVS, 请从 CVS 转换到 SVN (参考示例 <http://www.xs4all.nl/~carlo17/svn/cvs2svn.html> 的实现方法).
3. Dump 完整的 SVN 版本库 (使用 `svnadmin dump > foo.dump`).
4. 确认 dump 下来的文件可以被一个新建立的 SVN 版本库读取 (`svnadmin load newrep < foo.dump`).
5. 把文件发送给我们: 如果 dump 文件小于 10 MB, 您可以通过电子邮件发送其至 `R-Forge@R-project.org`, 否则您需要上传到某处并提供下载地址.
6. 最后一步, 执行 `svn move` 以移动您的包到 /pkg 目录 (如果此步未被自动执行).

7 致谢

本项目若没有 Douglas Bates 和威斯康辛大学的帮助是不可能完成的, 他们赞助了一台服务器让此平台得以运行. 此外, 感谢维也纳经济大学计算机科学服务中心的支持, 他们还提供了额外的硬件和专业的服务器基础设施. 感谢 Philippe Grosjean 提供了针对 Mac OS 的说明 (第 3.2.3 节).

参考文献

- Per Cederqvist et al. *Version Management with CVS*. Network Theory Limited, Bristol, 2006. Full book available online at <http://ximbiot.com/cvs/manual/>.
- Tom Copeland, Roland Mas, Ken McCullagh, Tim Perdue, Guillaume Smet, and Reinhard Spisser. *GForge Manual*, 2006. URL http://gforge.org/docman/view.php/1/34/gforge_manual.pdf.
- C. Michael Pilato, Ben Collins-Sussman, and Brian W. Fitzpatrick. *Version Control with Subversion*. O'Reilly, 2004. Full book available online at <http://svnbook.red-bean.com/>.
- R Development Core Team. *Writing R Extensions*. R Foundation for Statistical Computing, Vienna, Austria, 2006. URL <http://www.R-project.org>. ISBN 3-900051-11-9.