

[RStudio Support](#) > [Documentation](#) > [Package Development](#)

Developing Packages with RStudio

**Josh Paulson**

February 07, 2017 16:53

[Follow](#)

Developing Packages with RStudio

Overview

R packages are an ideal way to package and distribute R code and data for re-use by others. RStudio includes a variety of tools that make developing R packages easier and more productive, including:

- Build pane with package development commands and a view of build output and errors
- Build and Reload command that rebuilds the package and reloads it in a fresh R session
- R documentation tools including previewing, spell-checking, and [Roxygen](#) aware editing.
- Integration with [devtools](#) package development functions
- Support for [Rcpp](#) including syntax highlighting for C/C++ and gcc error navigation

Getting Started

Package Development Basics

If you aren't already familiar with the basics of R package development, the following links provide additional documentation and tutorials:

- [Writing R Extensions](#)
- [R Packages](#) (Hadley Wickham)
- [Creating R Packages: A Tutorial](#) (Friedrich Leisch)
- [Making an R Package](#) (R.M. Ripley)

Software Prerequisites

There are two main prerequisites for building R packages:

1. GNU software development tools including a C/C++ compiler; and
2. LaTeX for building R manuals and vignettes.

If you don't already have these tools installed on your system please consult the article on [Package Development Prerequisites](#) for additional details on how to install these dependencies.

Creating a New Package

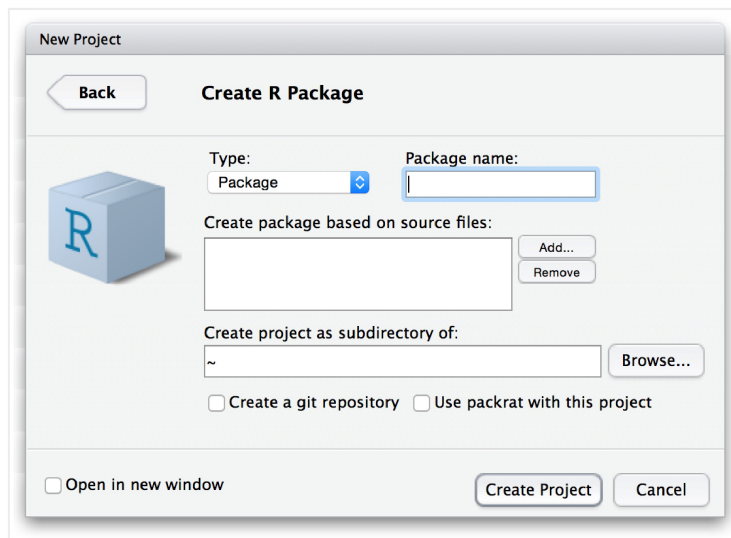
Recently viewed articles

[Resetting RStudio Desktop's State](#)

Related articles

[Building, Testing, and Distributing Packages](#)[Writing Package Documentation](#)[Customizing Package Build Options](#)[Version Control with Git and SVN](#)[Quick list of useful R packages](#)

To create a new package use the **Create Project** command (available on the Projects menu and on the global toolbar) and select the **New Directory** option. Then on the following screen specify the project type as **R Package**:



Note that if you have existing R scripts that you'd like to use as the basis for the new package you can specify them here and they'll be included in the new package.

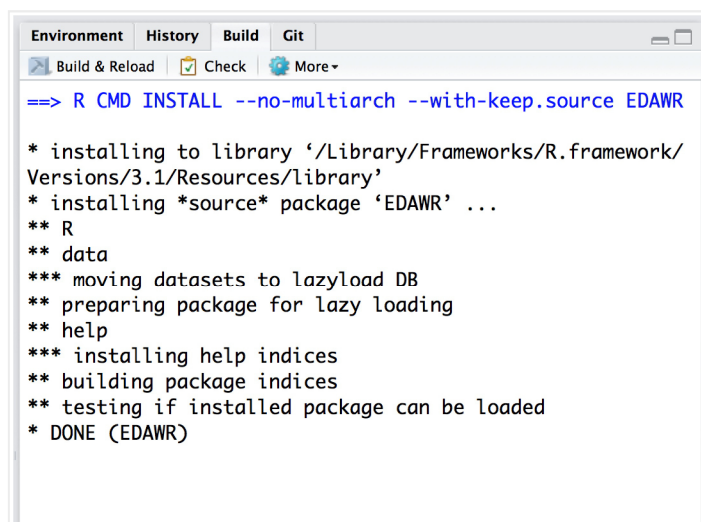
Working with an Existing Package

To enable RStudio's package development tools for an existing package you should do the following:

1. Create a new [RStudio Project](#) associated with the package's directory.
2. If the package DESCRIPTION file is located either in the project's root directory or at pkg/DESCRIPTION then it will be automatically discovered.
3. Alternatively, go to **Project Options : Build Tools**, select "Package" as the project build tools type, and then specify the the subdirectory containing the package's DESCRIPTION file.

Building a Package

To work with packages in RStudio you use the **Build** pane, which includes a variety of tools for building and testing packages. While iteratively developing a package in RStudio, you typically use the **Build and Reload** command to re-build the package and reload it in a fresh R session:



The Build and Reload command performs several steps in sequence to ensure a clean and correct result:

1. Unloads any existing version of the package (including shared libraries if necessary).
2. Builds and installs the package using `R CMD INSTALL`.
3. Restarts the underlying R session to ensure a clean environment for re-loading the package.
4. Reloads the package in the new R session by executing the `library` function.

Note that you can also execute **Build and Reload** using a keyboard shortcut (Ctrl+Shift+B) as well as configure RStudio to automatically save open source files prior to rebuilding.

Learning More

Once you've got a basic package built within RStudio you'll want to learn about the tools that can be used to test, document, and prepare packages for distribution. Please consult these articles for additional details:

- [Building, Testing, and Distributing Packages](#)
- [Writing Package Documentation](#)
- [Customizing Package Build Options](#)

Was this article helpful?
3 out of 3 found this helpful



Have more questions? [Submit a request](#)

Comments

 Powered by Zendesk