RStudio®

Open source & enterprise-ready professional software for R

RStudio Package Manager

The Problem

R's package ecosystem is awesome, providing users access to more than 12,000 packages. The package community is vibrant and active, with more than 300 weekly updates on CRAN, R's primary package repository. R users also have access to packages developed on GitHub, BioConductor, and often local packages internal to the enterprise.

However, the majority of R's package ecosystem was built and designed for open source users. In the enterprise, managing access to R packages can be problematic. General purpose repository managers don't work with R, and creating a package repository by hand is error-prone and time consuming. This leads many organizations to lock down access or enable full access — both approaches leaving something to be desired. Locking down access to packages installed by IT leaves R users frustrated and forces IT admins to manage change control. Open access is often impossible and prevents standardization and consistency across an organization.

These pain points will surface as different symptoms to various stakeholders:

Jr. Analyst

Just hired, but learned R in college.

Has to
DISCOVER
tribal knowledge

Sr. Analyst

Built functions to connect to the company's database.

Wants to
SHARE
code without emailing files

Statistician

Submitted valued trial results to the FDA.

Needs to
REPRODUCE
results, without worrying about packages

IT / Ops

Receives install and upgrade requests.

Goal is to
MAINTAIN
servers without breaking R code.

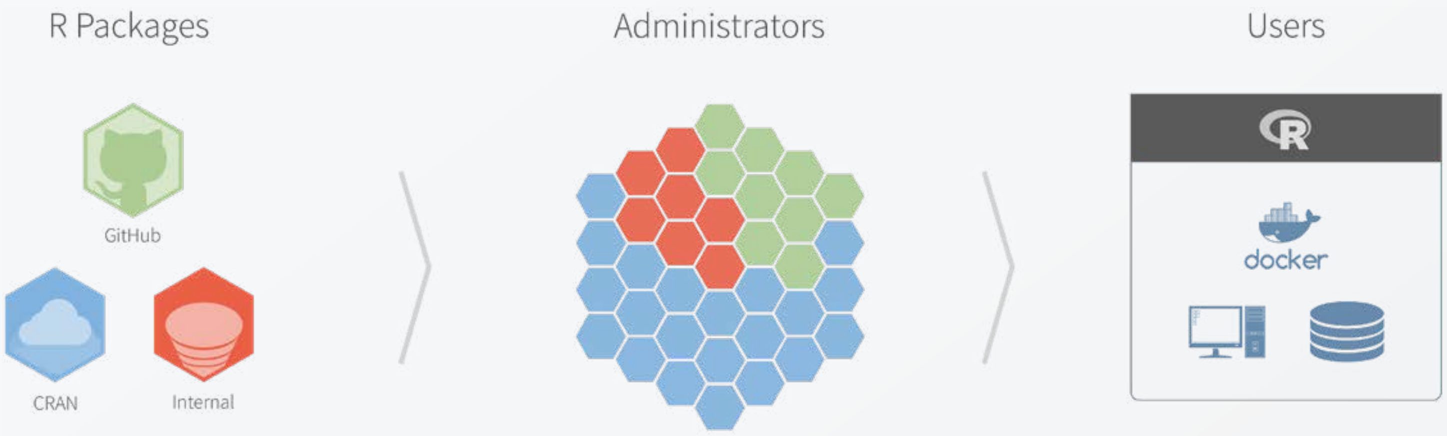
InfoSec

Enforces legal and risk policies.

Requires
SECURE
downloads that meet standards

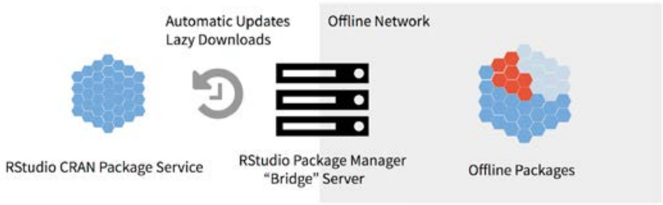
The Solution

RStudio Package Manager solves these problems by creating a central location for R packages. Administrators specify which packages are approved and can distinguish different groups like "production" or "bleeding edge users". RStudio Package Manager serves packages in an R-compliant interface optimized for stability, discovery, storage, and speed. RStudio Package Manager handles updates and versioning, ensuring that users can go back in time. R users discover packages, request additions, and share their own packages, all while using the R tools they already know like RStudio or `install.packages()`.

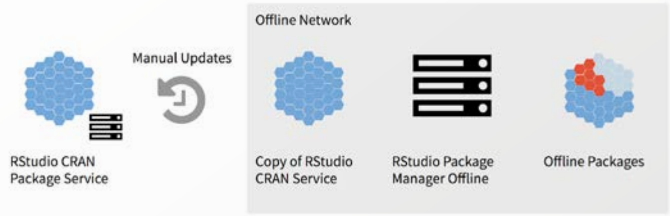


The Devil is in the Details

RStudio Package Manager is distributed as a Linux service and includes a web interface and a command line utility. Administrators install RStudio Package Manager on a server that has access to the online RStudio CRAN service and the organization's intranet.


Automatic Updates
Lazy Downloads
Offline Network
RStudio CRAN Package Service
RStudio Package Manager "Bridge" Server
Offline Packages

Alternatively, RStudio Package Manager can be used in a completely air-gapped environment, though manual syncs are required.


Offline Network
Manual Updates
RStudio CRAN Package Service
Copy of RStudio CRAN Service
RStudio Package Manager Offline
Offline Packages

Administrators define **what** packages are available to users (sources) and **how** they are organized (repos). For example, an administrator might define a "local" source with a handful of internal packages and a "CRAN" source that accesses RStudio's CRAN service. The two sources can be grouped into a "Production" repository that users access with a single url.

Under the hood, RStudio Package Manager stores metadata about each package in a database. Repositories are created virtually and are fully R-compliant. Virtual repositories optimize on-disk storage; packages can be presented in different repositories without redundantly storing them on disk. Packages from CRAN can be lazily fetched on-demand or eagerly downloaded all at once. RStudio's CRAN service ensures metadata is updated regularly, and allows system administrators to define how often to present updates to users.

Any change to a virtual repository is versioned, allowing users to roll back in time. In addition, each repository also versions individual packages and serves an R-compliant archive. Together, these two methods provide a variety of ways for R users to manage change control across their projects.

In the future, RStudio Package Manager will provide native support for accessing administrator-approved GitHub repositories and continuously deploying internal Git repositories that store local packages. Today, the flexibility of the command line interface enables admins to write their own scripts that customize RStudio Package Manager to meet the organization's specific goals.

Once RStudio Package Manager is setup, users can search for repositories and packages through a web interface. To install packages from RStudio Package Manager, administrators or users configure the repository options in RStudio. Once configured, users can access RStudio Package Manager through all their familiar tools including base R's `install.packages()`, packrat, and devtools.

About RStudio

RStudio® makes data analysis with R easier and provides powerful tools for sharing reports, dashboards, and interactive Shiny® applications with your entire enterprise.