shiny and rmarkdown R packages: Regulatory Compliance and Validation Issues

A Guidance Document for the use of affiliated R packages in Regulated Clinical Trial Environments

September 2020

RStudio PBC
250 Northern Ave
Boston, MA USA 02210

Tel: (+1) 844 448 1212
Email: info@rstudio.com

1. Purpose and Introduction

The purpose of this document is to demonstrate that the shiny and rmarkdown R packages[10], when used in a qualified fashion, can support the appropriate regulatory requirements for validated systems, thus ensuring that resulting electronic records are “trustworthy, reliable and generally equivalent to paper records.” For R, please see the document below:

R: Regulatory Compliance and Validation Issues
A Guidance Document for the Use of R in Regulated Clinical Trial Environments [8]

What qualifies as a record?

Validation guidance is a result of regulation, most notably the US Food and Drug Administration’s 21 CFR Part 11[2]. This regulation was originally written to apply to health records. The definition of a record has subsequently been clarified and extended.

A key consideration for companies subject to this regulation is determining the extent to which analytic software systems such as R, and the outputs they produce (web applications, documents, presentations, etc.) constitute records or derived records subject to Part 11 compliance[3].

RStudio, following the work of the R core foundation and the R Validation Hub, does not consider the outputs created, in whole or in part, by R and R packages as records directly subject to compliance regulations[8]. However, these outputs, and as a result, the software used to create them, should follow the spirit of the regulation and strive to be trustworthy and reliable. Risk-based monitoring was first documented by the FDA in the reflection paper and guidance in 2011. Following the interpretation of FDA guidance, RStudio recommends that organizations consider a risk-based approach to the use of R and R packages[1]. This document outlines important considerations to the risk of using a set of R packages affiliated with RStudio, PBC. Separate from these packages are RStudio’s products, which are developed using a controlled process that consists of distinct development phases as outlined here:

RStudio: Regulatory Compliance and Validation Issues
A Guidance Document for the Use of RStudio Professional Products in Regulated Clinical Trial Environments [11]

Risk-Based Approach and Open Source Software

Before covering the specific details of the Software Development Cycle and 21 CFR Part 11 compliance functions related to these affiliated packages, it is worth quickly noting the role of a risk-based approach in open source software. Noting that the use of analytic software is often complex, the FDA has clarified[2] that organizations should take a risk-based approach aimed at ensuring analysis is trustworthy and reliable. Each organization will adopt its own standards and definitions for risk.

This document is intended to provide a reasonable consensus position on the part of RStudio relative to the use of packages (for the shiny and rmarkdown ecosystems) within regulated environments and to provide a common foundation for people to meet their own internal standard operating procedures, documentation requirements, and regulatory obligations. Risk assessment and management are possible. **More so, it is our view that at a much deeper and fundamental level, open source software fulfills the role of a "trustworthy and reliable" system** far beyond any closed-source, proprietary software. Specifically, open source software is always available, to those interested, to be inspected and reviewed. By its very nature, the availability of open source software is not subject to the rise or fall of specific corporations. Nor is its use, review, and improvement subject to the economic means of the user. As a result, the outputs and methods of open source software are more amenable to being shared, more open to challenges and improvements, and significantly more repeatable and reproducible.

What is an R package?

R packages are extensions of the base R language for loading code, data, and documentation[14]. R packages exist as components in an ecosystem of software used together for analysis. In addition to this document, we advise that users refer to:

Regulatory Guidance for the R Language[8]

Regulatory Guidance for the Tidyverse, Tidymodels, r-lib, and gt R Packages

Regulatory Guidance for the RStudio Professional Software[11]

R Validation Hub[1]

The R Validation Hub resource strives to provide organizations with a risk-based approach to validating R add-on packages that are not expressly addressed in this document or other guidance documents[1]. The R Validation Hub is supported by The R Consortium, Inc., a group organized under an open source governance and foundation model to support the worldwide community of users, maintainers, and developers of R software[4]. As of August 2020, Its members include organizations such as Roche/Genentech, Microsoft, Google, Merck, Janssen Pharmaceutica, and more leading institutions and companies dedicated to the use, development, and growth of R. Such efforts include supporting the R in Pharma conference as

well as the R Validation Hub.

Scope of this Guidance

This document applies to the shiny and rmarkdown R packages maintained by RStudio. These two packages have unique characteristics and considerations compared to the majority of R packages. As opposed to containing statistical routines or analysis methods, these two packages are both *generative*, enabling users to create their own classes of outputs. Shiny enables users to create web applications. R Markdown allows users to create documents, reports, presentations, and a variety of other output types. Both packages are foundational to a large cohort of additional packages specific to the sub-domain of R they enable, for example, there are many packages that extend shiny. This document discusses the principles that guide the development of these packages and gives considerations for the use of each. However, the packages exist in an ecosystem of dependencies and reverse dependencies, it is incumbent on each user of the packages at any moment in time to qualify their installation - ensuring a complete and compatible cohort of packages and related software.

This document is NOT in any fashion, applicable to any other R-related software or add-on packages. It is important to note that there is a significant obligation on the part of the end-user's organization to define, create, implement and enforce R installation, validation, and utilization related Standard Operating Procedures (SOPs). The details and content of any such SOPs are beyond the scope of this document. This document also does not provide guidance for software or artifacts derived from these foundational packages. For example, a web application developed with the shiny package may require its own validation considerations.

This document is not intended to be prescriptive, does not render a legal opinion, and does not confer or impart any binding or other legal obligation. It should be utilized by the reader and his or her organization as one component in the process of making informed decisions as to how best to meet relevant obligations within their own professional working environment.

RStudio, Inc. makes no warranties, expressed or implied, in this document.

2 Software Development Life Cycle (SDLC)

2.1 Operational Overview

The shiny and markdown R packages in these organizations follow a common development lifecycle[10]. The size of the R user community provides for an extensive review of source code and testing in enterprise settings and all having full access to the source code, enables a superior ability to anticipate and verify the performance and the results produced by the packages discussed within this document.

2.2 Source Code Management

The source code is managed using Git, a widely-used open source version control software. The source is stored in public Git repositories and made available through Github, a Microsoft Affiliate that hosts Git repositories. Specifically:

- R Markdown Source Code: https://github.com/rstudio/rmarkdown
- Shiny Source Code: https://github.com/rstudio/shiny

2.3 Testing and Validation

The packages are tested through a combination of unit tests, CRAN checks[15], and integration tests. Unit tests are written to cover specific functions and features provided by each package. The test suites are run in an automated fashion, and the test results, as well as the test coverage (the amount of code tested by the package unit tests), are publicly displayed. CRAN checks, also run automatically, provide a thorough test of whether the package source code can be built and installed across a variety of operating systems. Any package accepted on CRAN must pass a series of automated tests that enforce the CRAN submission policies[15]. These checks also account for checks for consistency in function definition and documentation. The results of these tests are available publically for each package. Tests are executed automatically when any changes are proposed to the code, documentation, tests, or metadata in the package. These tests are run across multiple versions of R and multiple operating systems. Finally, packages released on CRAN undergo a test for compatibility with other dependent and reverse dependent packages.

Additionally, the shiny package undergoes a comprehensive integration test with each release, consisting of applying the package update to a wide variety of derived shiny applications and testing for any changes in functionality. These tests take advantage of the automated tooling available for testing web applications built using shiny[12].

2.4 Release Cycle

Packages are developed incrementally, following the best practices of the Git version control system. At specific points in time, a package will be released by incrementing the package version number, tagging the commit as a release in Git, archiving the new sources, and submitting the package to CRAN.

For each release, comprehensive release notes are maintained within the repository, and a summary of the major changes and features in a release are documented in a news changelog[9]. Releases tend to be accompanied by a summary of the major changes and features in a blog post (see examples for shiny and rmarkdown)[9].

2.5 Availability of Current and Historical Archive Versions

The source code for all packages, including every revision to this source code, is maintained in GitHub, a distributed service that provides free and public access to the projects' Git repositories (see section 2.2).

The released and archived versions of each package are maintained on CRAN, an extensive network of repository mirrors. Archive package versions can be found via the CRAN Package Archive. RStudio maintains a CRAN mirror with current and archived packages at https://cran.rstudio.com. The RStudio mirror uses Amazon Cloudfront to maintain copies of CRAN on servers all over the globe. These copies are updated off of the main CRAN mirror in Austria once per day. RStudio created this mirror to provide a consistently fast option around the world, a reliable option for users, and to provide a rich source of data about R and package usage. In addition, RStudio maintains a history of the CRAN mirror, accessible at https://packagemanager.rstudio.com.

The packages are leased using public licenses that are not subject to commercial control, ensuring they are fundamentally available in a greater sense than any commercial software.

2.6 Maintenance, Support, and Retirement

RStudio understands support and maintenance to encompass a wide range of activities, corresponding to the open nature of the package development and use. Each package provides extensive documentation, both within the package for specific functions, but also outside the package to document best practices, examples, and common usage patterns:

- R Markdown: https://rmarkdown.rstudio.com
- Shiny: https://shiny.rstudio.com

The websites provide articles, galleries of examples, and extensive video and written documentation. Numerous supporting materials exist for learners. The rmarkdown package includes a freely available book that serves as a definitive resource guide and another freely available book that contains practical examples[6][5].

A community forum affiliated with RStudio provides an opportunity for direct Q&A with categories dedicated to both shiny and R Markdown. Specific issues related to the software, such as requests for new features or identification of bugs, can be submitted and tracked on the relevant package GitHub site. For example, the shiny and rmarkdown issue boards[10].

2.7 Qualified Personnel

All development of shiny and rmarkdown occurs through open contributions to the packages source code. Each contribution can be specifically enumerated using the Git version control system. For convenience, the history of contributors is available for each package. For example, the contributors to the rmarkdown package as well as the shiny package. Contributions are peer reviewed in the open, and contributions are considered in light of the testing and style guides previously documented[10].

While a wide range of contributors have enabled the success of these packages, a core set of primary contributors employed by RStudio have relevant professional qualifications such as advanced degrees in related subject matter, peer reviewed publications, conference talks, and/or industry experience. Many members of RStudio hold Ph.D. and/or Master's degrees from accredited academic institutions and have published extensively in peer reviewed journals. Several have written books on statistical computing technologies and applications using the packages detailed in this document.

2.8 Physical and Logical Security

The physical and logical security of the package source code is handled through GitHub's disaster recovery plan and security standards.

2.9 Disaster Recovery

The disaster recovery of the package source code is handled through GitHub's disaster recovery plan.

3 21 CFR Part 11 Compliance Functionality

3.1 Overview

The United States regulation known as Title 21 CFR Part 11, or the "Electronic Records; Electronic Signatures" rule[3], provides information about what constitutes trustworthy and reliable electronic records and electronic signatures. FDA industry guidance for the use of electronic health record data in clinical investigations is here[13]. RStudio continues to monitor FDA regulations and guidelines that pertain to packages covered in this document. People can use RStudio products and packages to build data collection, analysis, and other systems that can be used in compliance with Part 11. Compliance with this regulation ultimately depends on

how a package is installed and used, how users are trained, and other factors. Users need to use packages according to the system requirements, install it according to the installation instructions, and use it via the user documentation. Users should refer to the predicate rule or consult the FDA or its guidance documents to determine whether packages are in compliance with regulatory expectations.

shiny and rmarkdown are R packages that extend the base R language. As a result, these packages inherit the CFR Part 11 compliance guidelines and interpretation documented for the base R language[8]. For this reason, we recommend all compliance considerations for R packages begin with the interpretation and compliance guidelines for R itself.

However, RStudio recognizes that these packages are often used in novel applications that go beyond the "performance of calculations or creation of graphics". Indeed, both shiny and rmarkdown provide frameworks for extending the R programming language to create entire application suites. As such, we provide guidance for how CFR compliance can be approached in the practical application of these packages. However, it is important to note that the R packages often provide only a piece of the final product; they are often combined with other systems and tools for distribution and use. Organizations should take care to validate the entire system, not just individual components. We provide additional guidance for the use of RStudio's professional products which, in conjunction with the R packages, provide a holistic system[11].

Additionally, as mentioned in the introduction, a key consideration is whether the artifacts created using shiny and rmarkdown constitute CFR 11 records. Following the base R guidance, RStudio understands that these outputs are typically not records, and the majority of applications derived from these packages are not tools designed for the creation, maintenance, modification, or deletion of CFR 11 records. As such, CFR 11 does not typically apply directly. Quoting the base R guidance:

R's design and development are focused on reporting, by enabling leading edge statistical analysis and presentation, rather than on data management tasks as illustrated by transaction/data processing and related functionality.

In the following sections, the term record means an electronic record that is interpreted to fall within the remit of Part 11 as defined in FDA Guidance for Industry Part 11, Electronic Records; Electronic Signatures– Scope and Application (2003).

R is not intended to create, maintain, modify or delete Part 11 relevant records but to perform calculations and draw graphics[7].

Organizations using shiny or rmarkdown to create applications that are directly related to the maintenance of CFR 11 records should take care to qualify and validate those systems in their entirety.

3.2 11.10(b) The ability to generate accurate and complete copies of *records* in both human readable and electronic form suitable for inspection, review, and copying

The shiny package allows users to create web applications that are composed of standard web components such as HTML, JavaScript, and CSS. These web applications can then be combined with a web server and a client browser to generate a human readable, interactive application. The Shiny package is tested to ensure a wide range of compatibility with these components, especially client browsers.

The rmarkdown package is designed to support literate programming, allowing the results of computation to be included alongside prose in a variety of formatted and human readable outputs including RTF, Word, PDF files, HTML files, and presentations. These outputs are typically viewed through a client application such as a web browser or PDF viewer. The outputs are designed to be compatible with a wide variety of nearly ubiquitous client tools.

3.3 11.10(c) Protection of records to enable their accurate and ready retrieval throughout the records retention period

The shiny package creates interactive applications that require both a client browser and an active server running the user code. As such, to enable the ready retrieval of such applications, an organization must use the shiny package in conjunction with a web server designed to host these applications. RStudio provides guidance for RStudio Connect which is one such web server[11].

The rmarkdown package creates a variety of output formats. The resulting outputs are typically stand-alone files or directories of files that can then be stored and retrieved through a validated system of record for files.

3.4 11.10(d) Limiting system access to authorized users

The development of code that uses the shiny and rmarkdown R packages occurs through the use of the R language and as such the guidance for the base R language applies:

R is an application that runs within the hosting computer environment, which must provide user access controls at hardware and/or operating system levels. The requirement for this section is typically met via system level functionality and is based on user roles, object level security and related security policies[8].

However, RStudio also recognizes that once written, code that utilizes the shiny and rmarkdown

packages is often deployed. In these cases, the deployed system must provide these access controls.

RStudio provides guidance for RStudio's professional products which implement these authorization constraints in both development and deployment of code using the shiny and rmarkdown frameworks[11].

3.5 11.10(e) Use of secure, computer-generated, time-stamped audit trails to independently record the data and time of operator entries and actions that create, modify, or delete electronic records. Record changes shall not obscure previously recorded information. Such audit trail documentation shall be retained for a period at least as long as that required for the subject electronic records and shall be available for agency review and copying

The shiny package allows users to create interactive web applications. It is incumbent upon the creators of these applications to incorporate logging that generates such time-stamped audit trails.

The rmarkdown package automatically outputs time-stamped logs anytime the package is used to render content. It is incumbent upon the organization to store and/or supplement these default logs.

RStudio provides guidance for RStudio's professional products that implement additional audit and storage capabilities for both the development and deployment of code using the shiny and rmarkdown frameworks[11].

3.6 11.10(f) Use of operational system checks to enforce permitted sequencing of steps and events, as appropriate

Following the base R guidance, RStudio understands this item to mean that effective user technology, processes, and interfaces must be in place to reduce errors made by an operator. Both the shiny and rmarkdown packages are built to include error handling and provide functions for users to capture, diagnose, and customize the behavior of outputs in scenarios where errors occur, as outlined in this document[14].

3.7 11.10(g) Use of authority checks to ensure that only authorized individuals can use the system, electronically sign a record, access the operation or computer system input or output device, alter a record, or perform the operation at hand

Following the base R guidance, RStudio refers to the recommendations in 11.10(d) that understand authority checks here as intimately related to system authorization.

This section inherits the same guidance found in the base R documentation[7].

3.8 11.10(h) Use of device (e.g., terminal) checks to determine, as appropriate, the validity of the source of data input or operational instruction

RStudio understands these checks to be warranted in the case where shiny or rmarkdown are used as a primary-source data management or data entry system. In these cases, it is incumbent on users to implement these quality checks.

Specifically, in the context of shiny applications that perform data management or entry, patterns exist for ensuring the validity of a user 11.10(d) as well as quality checks for entered data. However, the shiny package does not provide for these capabilities directly itself.

3.9 11.10(j) The establishment of, and adherence to, written policies that hold individuals accountable and responsible for actions initiated under their electronic signatures, in order to deter record and signature falsification

Neither the shiny nor rmarkdown packages provide functions for electronic signatures. In cases where applications or outputs derived from these packages are determined, through the application of relevant predicate rules, to require an electronic signature, it would be incumbent on the organization to establish policies and integrate with a 3rd party system.

3.4 11.10(k) Use of appropriate controls over systems documentation

RStudio understands this item to mean that there must be control over who can change and access documentation for these packages. As documented in section 2, the packages provide for specific contribution guidelines *including contributions to documentation*, and core documentation is maintained and governed in the same version control systems as the source code.

Section 11.30 Controls for Open Systems - the system shall employ procedures and controls designed to ensure the authenticity, integrity and as appropriate the confidentiality of electronic records from the point of their creation to the point of their receipt. Additional measures such as document encryption and use of appropriate digital signature standards to ensure, as necessary under the circumstances record authenticity, integrity and confidentiality

This section inherits the same guidance found in the base R guidance[8].

Key References

1. A Risk-Based Approach for Assessing R Package Accuracy within a Validated Infrastructure. *Nicholls, A. et al.* 2020-01-23. https://www.pharmar.org/white-paper/

2. Guidance for Industry: Part 11, Electronic Records; Electronic Signatures -- Scope and Application. 2003-08. https://www.fda.gov/media/75414/download

3. Part 11, Electronic Records; Electronic Signatures - Scope and Application Guidance for Industry. 2003-08-01. https://www.fda.gov/regulatory-information/search-fda-guidance-documents/part-11-electronic-records-electronic-signatures-scope-and-application

4. R Consortium About Page. Accessed 2020-09-25. https://www.r-consortium.org/about

5. R Markdown Cookbook. *Xie, Y. Dervieux, C. Riederer, E.* 2020-09-01. https://bookdown.org/yihui/rmarkdown-cookbook/

6. R Markdown: The Definitive Guide. *Xie, Y. Grolemund, G.* 2020-09-24 https://bookdown.org/yihui/rmarkdown/

7. R Packages. *Wickham, H. Bryan, J.* Accessed 2020-09-25. https://r-pkgs.org/

8. R: Regulatory Compliance and Validation Issues A Guidance Document for the Use of R in Regulated Clinical Trial Environments. *R Foundation for Statistical Computing.* 2020-03-2018. https://www.r-project.org/doc/R-FDA.pdf

9. RStudio Blog Package News Category. Accessed 2020-09-25. https://blog.rstudio.com/categories/packages

10. RStudio Github Organization Containing shiny and rmarkdown Repositories. Accessed 2020-09-25. https://github.com/rstudio

11. RStudio: Regulatory Compliance and Validation Issues. *RStudio, Inc.* 2019-06-26. https://rstudio.com/wp-content/uploads/2019/06/rstudio_compliance_validation.pdf

12. Shinytest Homepage. Accessed 2020-09-25. https://rstudio.github.io/shinytest/

13. Use of Electronic Health Record Data in Clinical Observations: Guidance for Industry. 2018-07. https://www.fda.gov/media/97567/download

14. Write error messages for your UI with validate. *Cheng, J. Grolemund, G.* 2018-06-28. https://shiny.rstudio.com/articles/validation.html

15. Writing R Extensions, Version 4.0.2. *R Core Team.* 2020-06-02. *https://cran.r-project.org/doc/manuals/r-release/R-exts.pdf*