

SOP RPACT CRAN Release General

© RPACT GmbH, Germany

SOP-RPACT-005

Version 1.0.0

Contents

1 Standard Operating Procedure for The Release of R Packages Developed by RPACT to CRAN	3
2 Preconditions	3
3 General Preparation Steps Before Release	4
3.1 Spell Check	4
3.2 Run All Unit Tests	4
3.3 Ensure Access to Unit Test Resources (If Needed)	4
3.4 Ensure All Unit Tests Are Uploaded (If Applicable)	4
3.5 Calculate and Check Test Coverage	4
3.6 Run Package-Specific Internal Validation Checks	4
3.7 Cross-Platform Checks on RHub	5
3.8 Recreate and Check All Vignettes	5
3.9 Push Source Code to GitHub and Review CI/CD Results	5
4 Submit Package to CRAN	5
4.1 Run <code>devtools:::release()</code>	5
5 Post-Release Activities (One Week After Release)	5
5.1 Take a Screenshot of CRAN Check Results	5
5.2 Save Screenshot and Update Validation Documentation	5
5.3 Confirmation of Execution and Review	6

RPACT GmbH

Am Rodenkathen 11
23611 Sereetz
Germany
www.rpact.com

Copyright © 2025 RPACT GmbH. All rights reserved.

1 Standard Operating Procedure for The Release of R Packages Developed by RPACT to CRAN

This Standard Operating Procedure (SOP) describes the required steps to prepare, validate, and release any R package developed by RPACT to the Comprehensive R Archive Network (CRAN). It applies to all R packages maintained by RPACT, hereafter referred to as .

The purpose of this SOP is to ensure high quality, reproducibility, regulatory suitability, and compliance with CRAN submission guidelines.

SOP-ID	SOP-RPACT-005
Title	SOP RPACT CRAN Release General
Description	Standard operating procedure for releasing a new R package version developed by RPACT with publication on CRAN
Author	Friedrich Pahlke
Creation date	2025-11-21
Version	1.0.7
Date of modification	2025-11-21

2 Preconditions

Before initiating the CRAN release process, the following preconditions must be met:

1. Check open GitHub issues

All known bugs and critical issues reported on
<https://github.com/rpact-com/<package-name>/issues>
must be resolved and closed.

2. Check CRAN package check results (if applicable)

If already exists on CRAN, ensure all results show
OK or *NOTE*:
https://cran.r-project.org/web/checks/check_results_<package-name>.html.

3. Overall package check time

R CMD check must complete in under **10 minutes**.

4. Test coverage

Test coverage must be **> 80%**.
The Codecov page for the package should be checked, if applicable:
<https://app.codecov.io/gh/rpact-com/<package-name>/?branch=main>.

3 General Preparation Steps Before Release

3.1 Spell Check

Run a spell check across all documentation:

```
devtools::spell_check()
```

3.2 Run All Unit Tests

Verify that all unit tests pass:

```
devtools::test()
```

3.3 Ensure Access to Unit Test Resources (If Needed)

If the package requires external credentials, test data, or configuration files used during unit testing, verify that:

- All required files are available
- All required credentials are present in the testing environment

3.4 Ensure All Unit Tests Are Uploaded (If Applicable)

If unit tests or test data are stored externally (e.g., FTP, S3, private server), ensure that the resources for the current version

<current-version>

are available and correct.

3.5 Calculate and Check Test Coverage

Generate a coverage report and ensure the threshold is met:

```
covr::package_coverage()
```

3.6 Run Package-Specific Internal Validation Checks

If provides an additional internal validation function , run them before release.

Example:

```
<package-name>::<test-package-function>()
```

3.7 Cross-Platform Checks on RHub

Test the package on multiple platforms:

```
rhub::check_for_cran()
```

3.8 Recreate and Check All Vignettes

Ensure the documentation builds correctly:

1. Open the Quarto project (if available)
2. Click: **Build** → **Render Website**

3.9 Push Source Code to GitHub and Review CI/CD Results

1. Push the latest changes to the development branch.
2. Create a pull request.
3. Verify that all GitHub Actions pipelines in
<https://github.com/rpact-com/<package-name>/actions>
pass successfully.
4. Merge into **main** only when the pipeline completes without issues.

4 Submit Package to CRAN

4.1 Run `devtools::release()`

Submit the package using:

```
devtools::release()
```

5 Post-Release Activities (One Week After Release)

5.1 Take a Screenshot of CRAN Check Results

After one week, visit:

https://cran.r-project.org/web/checks/check_results_<package-name>.html

Create a screenshot of all platform results.

5.2 Save Screenshot and Update Validation Documentation

Save the screenshot under:

`<validation-results-folder>/CRAN-Package-Check-Results.png`

Update the validation documentation for all clients.

5.3 Confirmation of Execution and Review

By signing below, I confirm that all steps outlined in this SOP have been fully executed and reviewed. I certify that the procedures were followed as described, and that the results have been verified for accuracy and compliance with the relevant standards.

Name	Signature	Date
------	-----------	------
