

---

# **r2readthedocs**

***Release 0.0.1***

**Mark Padgham**

**Oct 23, 2023**



## INTRODUCTION:

1 Examples

3



This main document is in ‘reStructuredText’ (“rst”) format, which differs in many ways from standard markdown commonly used in R packages. `rst` is richer and more powerful than markdown. The remainder of this main document demonstrates some of the features, with links to additional `rst` documentation to help you get started. The definitive argument for the benefits of `rst` over markdown is the [official language format documentation](#), which starts with a very clear explanation of the [benefits](#).



## EXAMPLES

All of the following are defined within the `docs/index.rst` file. Here is some coloured text which demonstrates how raw HTML commands can be incorporated. The following are examples of `rst` “admonitions”:

---

**Note:** Here is a note

**Warning:** With a warning inside the note

---

### See also:

The full list of ‘[restructuredtext](#)’ directives or a similar list of [admonitions](#).

This is a line of centered text

- |               |               |                |
|---------------|---------------|----------------|
| • and here is | • short items | • displayed    |
| • A list of   | • that are    | • in 3 columns |

The remainder of this document shows three tables of contents for the main README (under “Introduction”), and the vignettes and R directories of a package. These can be restructured any way you like by changing the main `docs/index.rst` file. The contents of this file – and indeed the contents of any [readthedocs](#) file – can be viewed by clicking *View page source* at the top left of any page.

## 1.1 r2readthedocs

Convert R package documentation to a ‘[readthedocs](#)’ website.

### 1.1.1 Why?

Two compelling reasons:

1. R packages are not necessarily self-contained, yet much of the surrounding infrastructure, from CRAN to automatic documentation generators such as [pkgdown](#), presume a package to be a self-contained unit. Any projects requiring greater flexibility must employ alternative documentation systems, for which ‘[readthedocs](#)’ (RTD) offers an extremely simple and highly flexible way to embed or enhance package documentation within or alongside almost anything else.

2. ‘readthedocs’ is based on a markup language called ‘reStructuredText’ (.rst), which is more flexible and powerful than the markdown generally used in R packages. Alternatively, yet another form called ‘Markedly Structured Text’ (MyST) can be used instead. The possibility of readily swapping between different markdown parsing systems provide much greater flexibility than available in convention RMarkdown parsers.

### 1.1.2 Installation

readthedocs requires three python packages which can be installed via your usual python installation method, for example,

```
pip install sphinx myst_parser sphinx_rtd_theme
```

This R package can then be installed by running one of the following lines:

```
remotes::install_github ("ropenscilabs/r2readthedocs")
pak::pkg_install ("ropenscilabs/r2readthedocs")
```

The package can then loaded for use with

```
library (r2readthedocs)
```

### 1.1.3 How?

For those unfamiliar with ‘readthedocs’ (RTD), we recommend using the `rtd_dummy_pkg()` function provided with this package, which generates a “skeleton” of a package which can be used to generate an RTD site.

```
path <- rtd_dummy_pkg ()
```

Then either using that path to the dummy package, or a local path to a package of your choice, generate an RTD website by running:

```
r2readthedocs (path)
```

The RTD website will be automatically opened in your default browser. Most of the content is automatically generated straight from the package documentation, as for a [pkgdown](#) website. The primary difference you’ll immediate notice is that the front page is not the package’s `README.md` document. There are two reasons for this:

1. While markdown pages can be readily inserted into .rst pages, this is not possible for the main page, which must be encoded entirely in .rst form.
2. This package may easily be used to document several packages, or a project of which an actual R package is just one component. Leaving the main page largely unstructured enables ready adaptation to almost any configuration.

The main file controlling the site’s appearance is `index.rst`, located in a sub-directory of the local path named “docs” - in R terms, at `file.path(path, "docs", "index.rst")`.

### 1.1.4 Further Information

An RTD site is primarily determined by a few simple files. The `r2readthedocs()` function initially inserts these files, and then generates the entire site, creating a "docs" sub-directory with a large number of files. The `rtd_clean()` function can be used to remove all files automatically generated by RTD itself, reducing the files down to the primary set of files controlling the site's structure and appearance. These are:

1. `index.rst` responsible for the structure of the main page of the site.
2. `conf.py` A python script for configuring aspects of the site.
3. `requirements.txt` containing a list of required python packages, which may be extended as desired.
4. `make.bat` and `Makefile` which are responsible for the main build system, and should **not** be modified. Type `make` in the local "docs" directory to see a full list of `make` options.

The main RTD documentation files can be re-generated at any time with `rtd_build()`.

## 1.2 r2readthedocs-package

r2readthedocs: Convert R Package Documentation to a 'readthedocs' Website

### 1.2.1 Description

Convert R package documentation to a 'readthedocs' website.

### 1.2.2 Seealso

Useful links:

- <https://github.com/ropenscilabs/r2readthedocs>
- Report bugs at <https://github.com/ropenscilabs/r2readthedocs/issues>

### 1.2.3 Author

Maintainer : Mark Padgham [mark.padgham@email.com](mailto:mark.padgham@email.com)

## 1.3 r2readthedocs

Convert package documentation to readthedocs format

### 1.3.1 Description

Convert package documentation to readthedocs format

### 1.3.2 Usage

```
r2readthedocs(path = here::here(), dev = FALSE, open = TRUE)
```

### 1.3.3 Arguments

Argument	Description
path	Path to local R package with documentation to be converted
dev	If TRUE , include function documentation of all dependency packages in site.
open	If TRUE , open the documentation site in default browser.

### 1.3.4 Value

TRUE (invisibly) if documentation successfully converted.

## 1.4 rtd\_build

Build the RTD site

### 1.4.1 Description

Build the RTD site

### 1.4.2 Usage

```
rtd_build(path = ".", dev = FALSE)
```

### 1.4.3 Arguments

Argument	Description
path	Path to local R package with documentation to be converted
dev	If TRUE , include function documentation of all dependency packages in site.

## 1.5 rtd\_clean

Clean the RTD \_build directory

### 1.5.1 Description

Clean the RTD \_build directory

### 1.5.2 Usage

```
rtd_clean(path = ".", full = FALSE)
```

### 1.5.3 Arguments

Argument	Description
path	Path to local R package with documentation to be converted
full	If TRUE, also clean R package files, including vignettes and help files. Use <a href="#">r2readthedocs</a> to re-generate these.

## 1.6 rtd\_dummy\_pkg

Make a dummy package for testing 'r2readthedocs' functions

### 1.6.1 Description

Make a dummy package for testing 'r2readthedocs' functions

### 1.6.2 Usage

```
rtd_dummy_pkg(base_dir = tempdir(), pkg_name = "demo")
```

### 1.6.3 Arguments

Argument	Description
base_dir	Base directory in which package is to be constructed (as a sub-directory)
pkg_name	Name of package, which will become a sub-directory of base_dir .

## 1.6.4 Value

Path to root directory of resultant package

## 1.7 rtd\_open

Open the main RTD 'html' page in default browser

### 1.7.1 Description

Open the main RTD 'html' page in default browser

### 1.7.2 Usage

```
rtd_open(path = ".")
```

### 1.7.3 Arguments

Argument	Description
path	Path to local R package with documentation to be converted