

# Setting up a dev environment

Jim Hester  
RStudio

 @jimhester

 @jimhester\_

<https://whattheyforgot.org/setup-an-r-dev-environment.html>

Types of R packages 

Source packages 

Binary packages 

# Binary packages

MacOS / Windows only

Pre-compiled on CRAN

Only R needed to install

Source packages 

compiled on users' machines

Extra tools needed to install

If  uses C/C++/Fortran

# Windows system prep

**Rtools** - Jeroen Ooms

**NOT** an R package

GNU compiler collection (**gcc**)

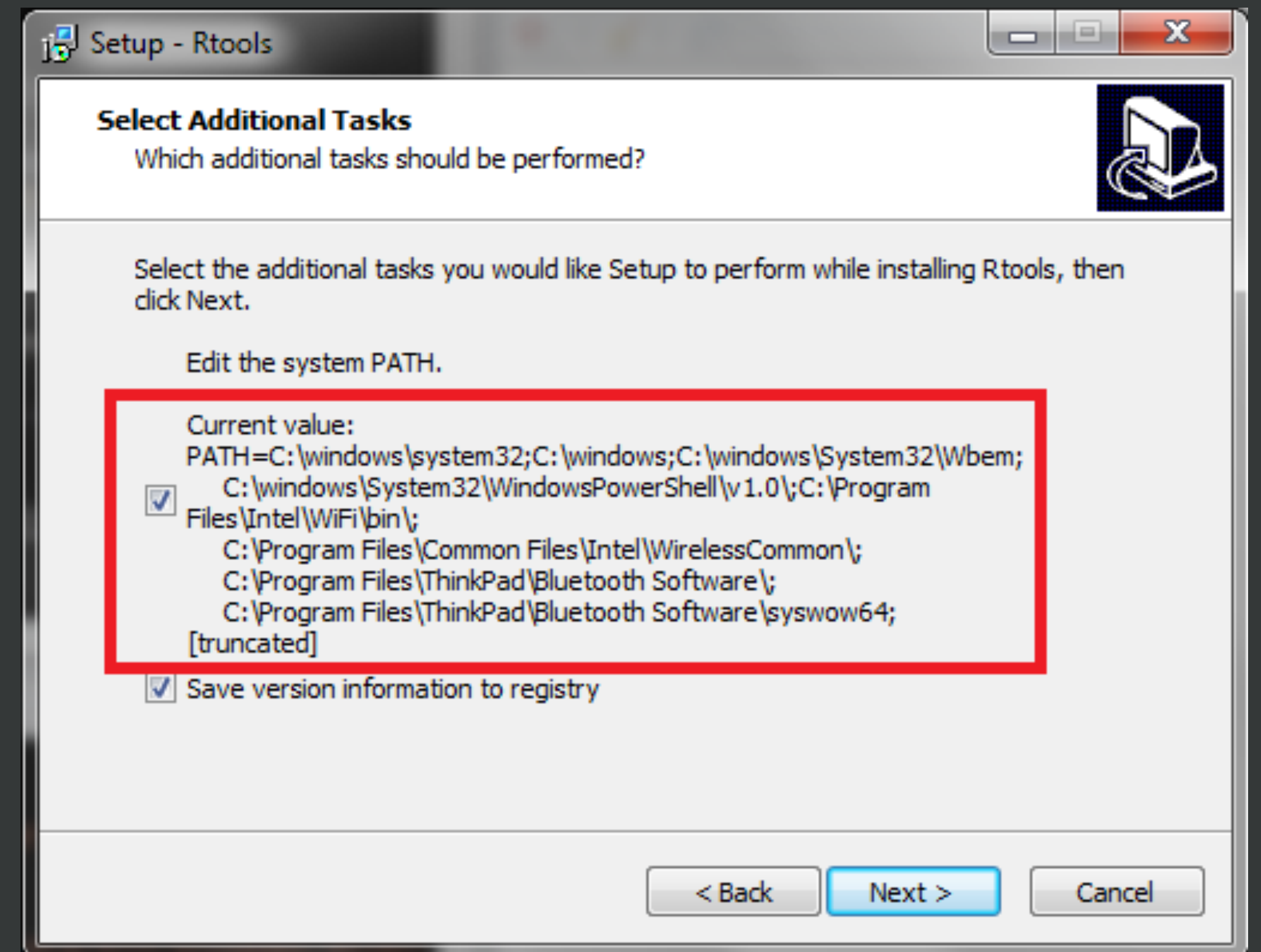
Minimalist GNU for Windows (**MinGW**)

# Rtools installation - Windows

<http://cran.r-project.org/bin/windows/Rtools>

**Uncheck** edit system PATH

**check** Save version information to registry



# Xcode command line tools - macOS

Apple provides

Command line approach

`xcode-select --install`

Mac App store

Xcode



# Verification of prep success

```
install.packages("devtools")
```

```
devtools::has_devel()
```

# What about Homebrew?

Avoid

```
brew install r
```

Use

```
brew cask install r-app
```

# Why?

No package binaries

More **time** to install packages

More **errors** to debug

# What about Conda?

## Support for R users

|             | Linux | macOS | Windows |
|-------------|-------|-------|---------|
| r           | 382   | 381   | 375     |
| conda-forge | 1139  | 1141  | 1102    |
| bioconda    | 112   | 103   | NA      |

source: Conda with R - John Blischak

# What about Conda?

**Use** `conda install r-ggplot2`

**Don't use** `install.packages("ggplot2")`

Why not `install.packages()`?

`Lose` reproducibility

Lots of `installation issues` with compiled packages

# Conda or not?

Fewer package binaries

1,200 on conda vs 13,000+ on CRAN

Can't easily install source packages

General recommendation

**Avoid** conda