Atorus Review Tool

Ben Arnst, Kyle Brennan,
Danny Halovanic, Alex Pham,
Austin Rush, Sebastian Tran

1001010101010100

1010010101001010

Table of Contents

The Client

- Atorus Research
- POC: Mike Stackhouse & Ashley Tarasiewicz
  - Clinical Analytics Solutions
  - OpenVal™ : open-source packages curated and catalogued
  - Development of in-house R packages
    - Tplyr - reports & pharmaRTF - text files

Purpose & Requirements

1
0
0
1
0
1
0
1
0
0
1
0
1
0
1
0

1
0
1
0
0
1
0
0
1
0
1
1
0
1
1
1

Purpose of the Atorus Review Tool

- To produce a code compliance tool within VS-Code for the R programming language
- Planned for use in clinical environments
- Helping users reach code compliance standards quicker

atorus

Project Requirements

1
0
0
1
0
1
0
1
0
0
1
0
1
0
1
0

<u>Major components for this project include:</u>

- Customizable Rule Layers
  - Base predefined rules
  - User-defined rules
  - Override/prioritize existing rules
- Real-Time Feedback
- In-Place Editing
- CI/CD Integration
- Package management

1
0
1
0
0
1
0
0
1
0
1
1
0
1
1
1

Features

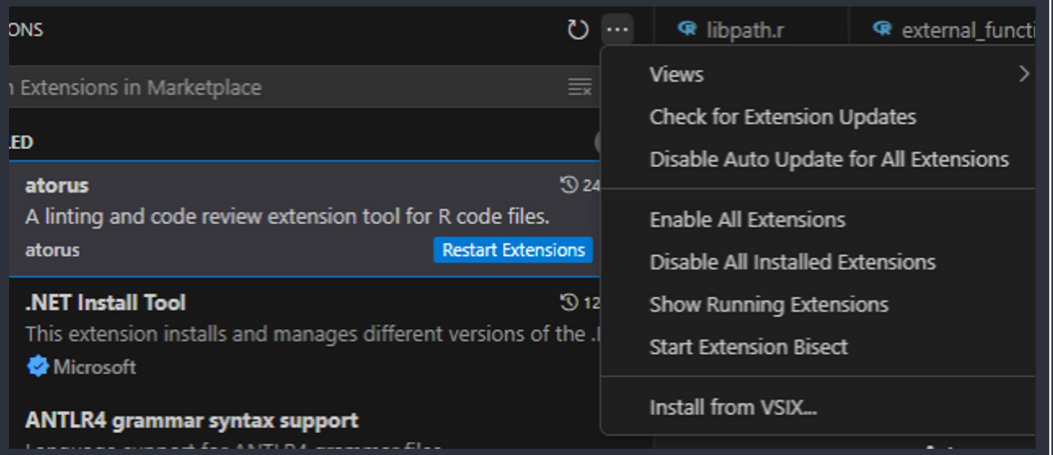
Description	Category	Status:
Check if all used packages are in the validated package list	Package Usage	Done!
Verify package versions match validated versions	Package Usage	Done!
Check if all used functions are approved	Function Usage	Done!
Check for usage of deprecated functions	Function Usage	Tabled
Check for unauthorized environment modifications	Environment	Tabled
Check for global variable assignments	Environment	IP

Features (cont.)

Description	Category	Status
Reassignment of library paths	Package Usage	Done!
Variable Assignment using ‘=’ instead of ‘<-’	Miscellaneous	Done!
Usage of non exported functions	Function Usage	Done!
Reassignment of functions using ‘::’ or ‘:::’	Function Usage	Done!
Variable assignment in function parameters	Function Usage	Done!

Installation

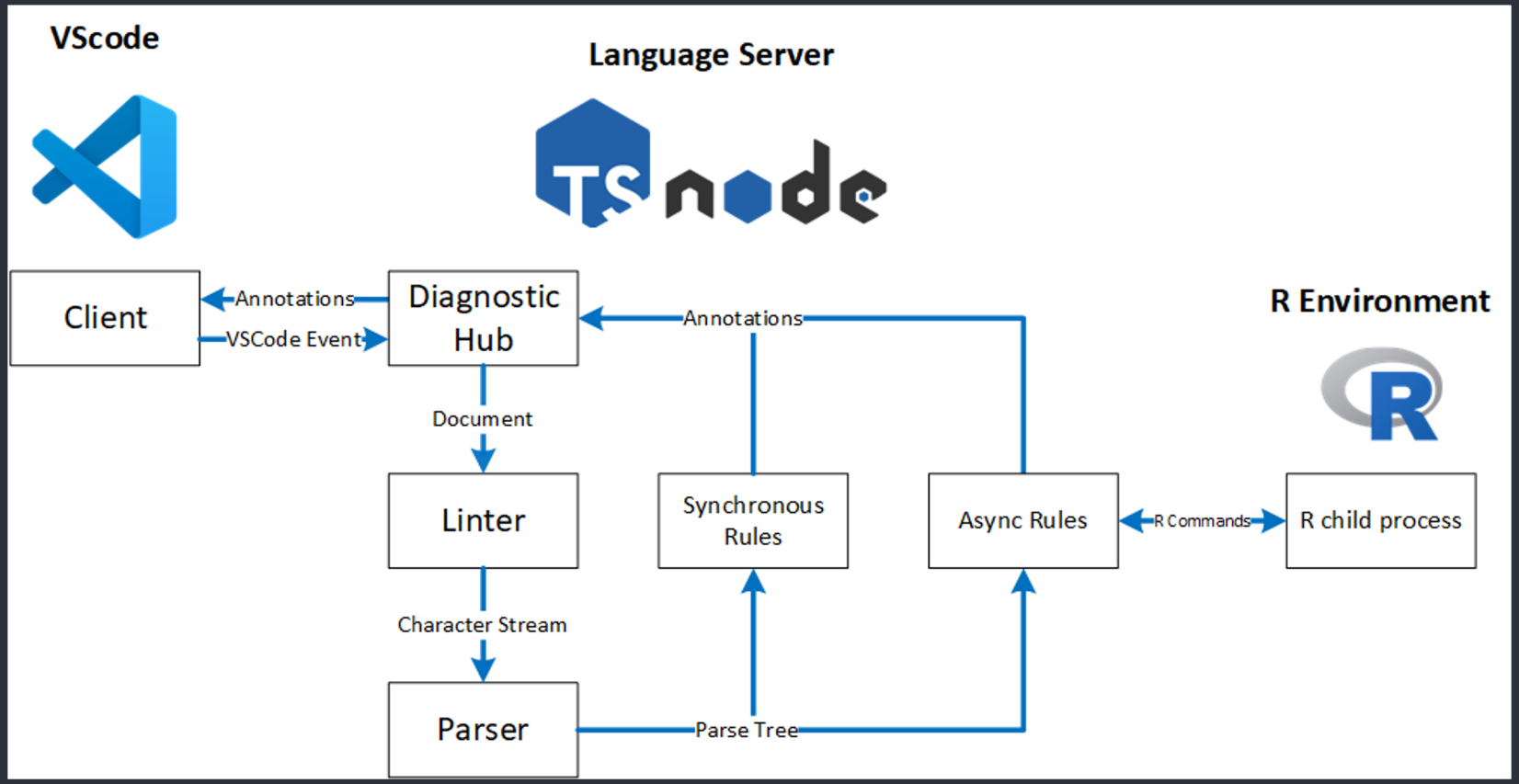
- Install R and extension as .VSIX
- Add R installation path to PATH environment variable
- Click … in Extension Sidebar Window
- Install from VSIX and select extension



System Architecture

1
0
0
1
0
1
0
1
0
0
1
0
1
0
1
0

1
0
1
0
0
1
0
0
1
0
1
1
0
1
1
1


VScode
Language Server
Client
Annotations
VSCode Event
Diagnostic Hub
Annotations
R Environment
Document
Linter
Synchronous Rules
Async Rules
R Commands
R child process
Character Stream
Parser
Parse Tree

VS Code Extension Client

1
0
0
1
0
1
0
1
0
0
1
0
1
0
1
0

- Provides UI for rule states
- UI is written with HTML/CSS
- Easily scalable

1
0
1
0
0
1
0
0
1
0
1
1
0
1
1
1

Language Server

- Written with Typescript + NodeJS
- Integrates very well with VSCode API
- Great package ecosystem
- Project structure provided

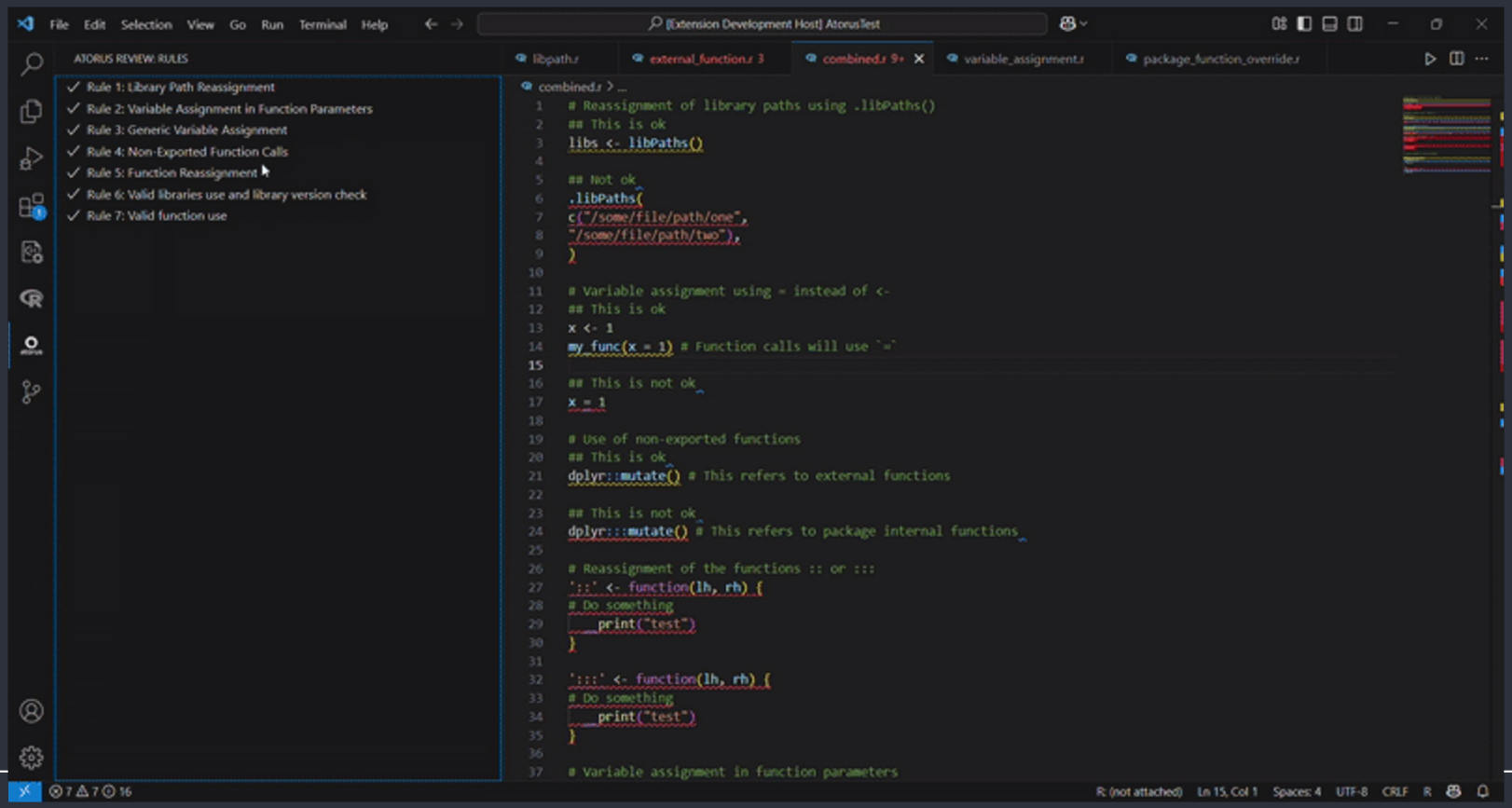
1
0
1
0
0
1
0
0
1
0
1
1
0
1
1
1

Completed Requirements

1
0
0
1
0
1
0
1
0
0
1
0
1
0
1
0

1
0
1
0
0
1
0
0
1
0
1
1
0
1
1
1

Rule Customizability



Valid Library Check

Checks if used package is part of OpenVal

```
3x2x2 factorial.R > ...
1
2
3
```

Library Version Check

Checks if installed package version matches OpenVal's

3x2x2 factorial.R > ...

```
1
2
3
4
```

Approved Function Check

```
this function has likely not been validated atorus-review-tool
View Problem (Alt+F8)    Quick Fix... (Ctrl+.)    Fix using Copilot (Ctrl+I)
                                                                    + press
anovatable=aov1(y ~ carbon + press + speed + carbon*press + carbon*speed
anovatable
```

Generic Variable Assignment

Checks if '=' is used for variable assignment

3x2x2 factorial.R > ...

```
1

2

3
```

Internal Package Function call

Checks if Internal package function is called or not. (Functions with ::: can be unsafe).

3x2x2 factorial.R > ...

```
library(dplyr)
dplyr::mutate()
dplyr:::mutate()
```

Internal package function calls are not allowed atorus-review-tool

Modifying Library Path

Checks if string is hard coded or not in libPath function.

3x2x2 factorial.R > ...

```
lib <-
```

Variable Assignment inside Function Parameters

```
13  data
14
15
16  par(mfrow=c(2,2))
17  plot.design(data)
```

Do not assign variable in function parameters atorus-review-tool

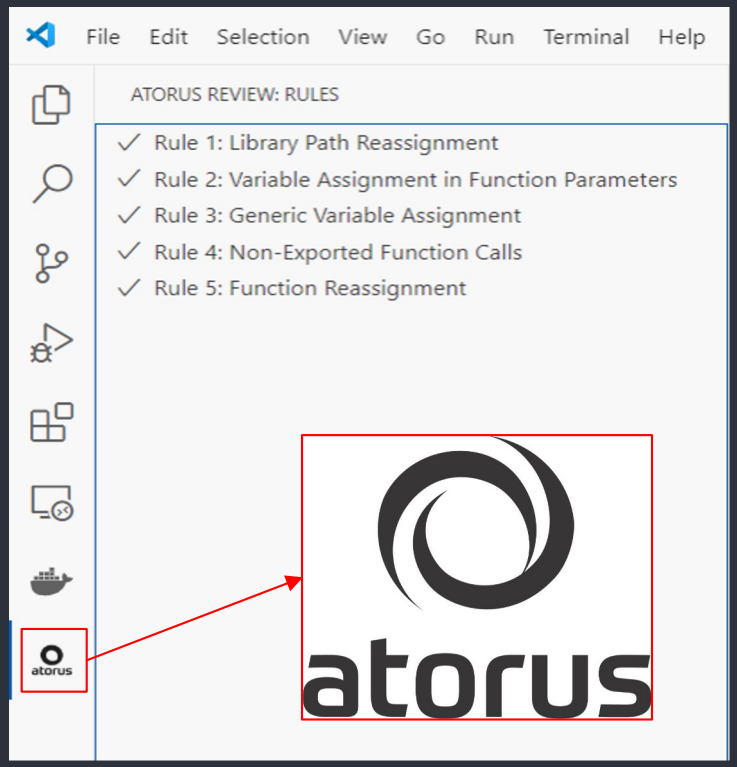
View Problem (Alt+F8) Quick Fix... (Ctrl+.) Fix using Copilot (Ctrl+I)

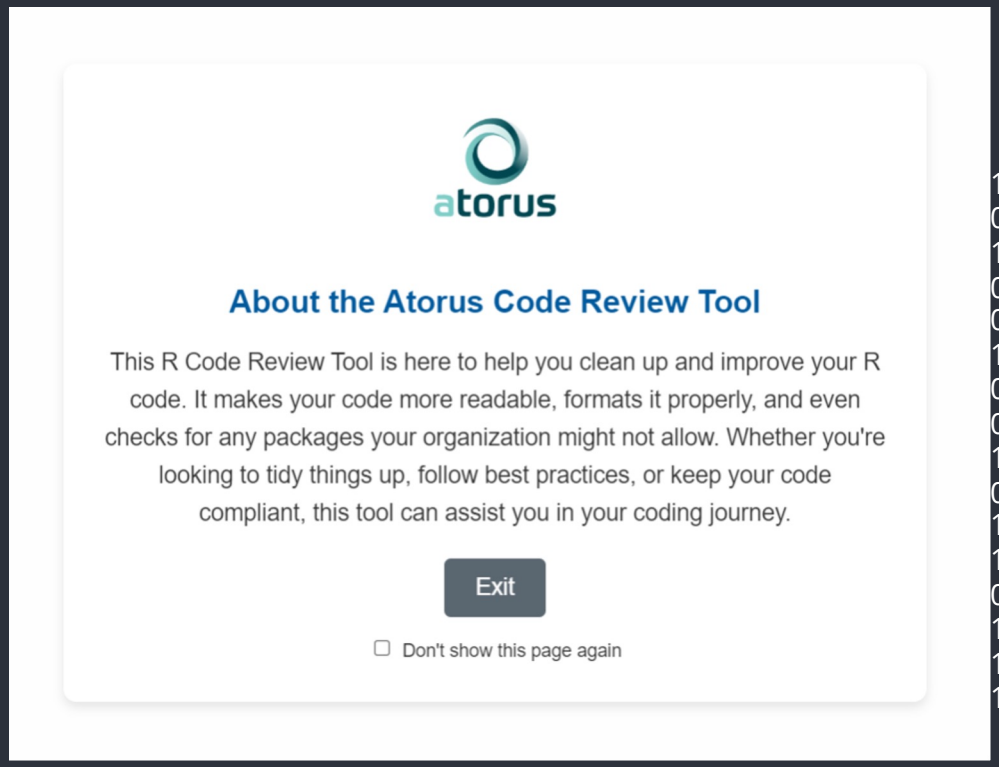
Overriding Infix Operator

Checks if infix operator is overridden or not.

```
Do not override :: or ::: infix operator atorus-review-tool
View Problem (Alt+F8)    Quick Fix... (Ctrl+.)    Fix using Copilot (Ctrl+I)
'::' <- function(a, b)
```

Extension User Interface





1
0
1
0
0
1
0
0
1
0
1
1
0
1
1
1

Client Satisfaction

1
0
0
1
0
1
0
1
0
0
1
0
1
0
1
0

- Demo for clients
  - Installation, explanation of framework, etc.
- “Impressed...further than we thought you’d get...” - Mike
- Questions about internal user testing

1
0
1
0
0
1
0
0
1
0
1
1
0
1
1
1

Future Development

- Deprecated functions check
- Environment change log
- R studio implementation

What We Learned

- Design considerations
  - From scratch vs using preexisting code
- Required learning many concepts on the fly
- Developing requirements with a client
- Being flexible with scheduling and meetings
- Team communication and collaboration

1
0
0
1
0
1
0
1
0
0
1
0
1
0
1
0

Questions?

1
0
1
0
0
1
0
0
1
0
1
1
0
1
1
1