

Run, or he's going to tell us about R again!

R Dev Day @ RSE Con25

R Dev Days are short events - usually over one day, or linked sessions over consecutive days - for novice and experienced contributors to work collaboratively on contributions to base R. These events have the support of the R Core Team and some will have R Core Developers participating directly.



People in orange are online

Shared notes	Topic	Project **(max 35 characters)**	Issue	GitHub URL	Status at end of de	Status note
	Documentation	Broken links to static help files	94	issue #94	Not assigned	
	Misc	Refine and test .editorconfig	104	issue #104	Not assigned	
	Documentation	Document VS Code and Positron settings	105	issue #105	Work in Pro...	
	Translation	Translating messages	2	issue #2	Assigned	Increased pt_BR tra
	Statistics/Graphics/Docu	**Improve the translations dashboard**	67	issue #67	Work in Pro...	Started transition to
	Misc	Review r-devel repos for best practices	51	issue #51	Work in Pro...	
	Documentation	Improve description of var.test for lm objects	45	issue #45	Patch submi...	
	I/O	Uniquely identify file connections	27	issue #27	Work in Pro...	updated patchfile
	Documentation	Fix braces NOTE on CRAN packages	110	issue #110	Assigned	Issue 110: Fix PR base mes Packages update
	Low-level	Add missing R public API pieces	108	issue #108	Work in Pro...	https://github.com/r-
	Misc	Improve manifest for R Dev Container		issue #	Not assigned	
	Low-level	Make `$` and `$<-`work for hash tables	115	issue #115	Patch submi...	
	Low-level	Add class to error from `setTimeLimit`	114	issue #114	Not assigned	
	I/O	Update last argument to substring	113	issue #113	Patch submi...	
	Translation	Combine Weblate and rOpenSci glossaries	112	issue #112	Not assigned	
	Translation	Add {pkgdown} website for {glossary}	111	issue #111	Closed fixed	
	Translation	Setup {precommit} on the repo	119	issue #119	Closed fixed	
	Translation	Add all contributors + github action	120	issue #120	Closed fixed	
	Translation	Test precommit without permissions on the repo	121	issue #121	Other (speci...	In review
	Translation	Test precommit with write permissions on the repo	122	issue #122	Other (speci...	PR in review
	Translation	Remove R CMD for now	123	issue #123	Closed fixed	
	Misc	Fix typo in rOpenSci's {allcontributors} pacl	124	issue #123	Other (speci...	Merged
	Documentation	Add info to bibliographic entries	116	issue #116	Work in Pro...	
	Documentation	Improve the CSS for the R manuals	26	issue #26	Closed fixed	In review
	Documentation	Improve the description of jitter(amount=)	39	issue #39	Closed fixed	
	Misc	improve summary.default() for character	117	issue #117	Work in Pro...	https://github.com/r-

HenrikBengtsson opened on Jul 11, 2024 Member

TL;DR

A connection "handle" might end up pointing to another connection risking corrupting data, e.g.

```
# Create an empty file
con1 <- file("1.txt", open = "w")
close(con1)

# Create another empty file
con2 <- file("2.txt", open = "w")

# Write to the first connect
# Expected: an "invalid connection" error
# Actual: "hello" is written to file "2.txt"
cat("hello\n", file = con1)
and there's a risk
close(con2)
```

After this, file `2.txt` will hold `hello\n`;

```
readLines("2.txt")
#> [1] "hello"
```

This is because connection handles are just integer indices that get reused, e.g.

```
> con1 <- file("1.txt", open = "w")
> close(con1)
> con2 <- file("2.txt", open = "w")
```

At this point `con1` points the same connection as `con2`;

```r
> str(con1)
 'file' int 3
 - attr(*, "conn_id")=<externalptr>

> str(con2)
 'file' int 3
 - attr(*, "conn_id")=<externalptr>

> con1
A connection with
description "2.txt"
class       "file"
mode        "w"
text        "text"
opened      "opened"
can read    "no"
can write   "yes"

> con2
A connection with
description "2.txt"
class       "file"
mode        "w"
text        "text"
opened      "opened"
can read    "no"
can write   "yes"
```

```
> showConnections(all = TRUE)
  description class      mode text   isopen   can read can write
0 "stdin"     "terminal" "r"  "text" "opened" "yes"    "no"
1 "stdout"    "terminal" "w"  "text" "opened" "no"     "yes"
2 "stderr"    "terminal" "w"  "text" "opened" "no"     "yes"
```

Assignees
No one - Assign yo
Labels
C I/O RSE
Type
No type
Projects
No projects
Milestone
No milestone
Relationships
None yet
Development
Code
Create a branch for
Notifications
You're receiving no

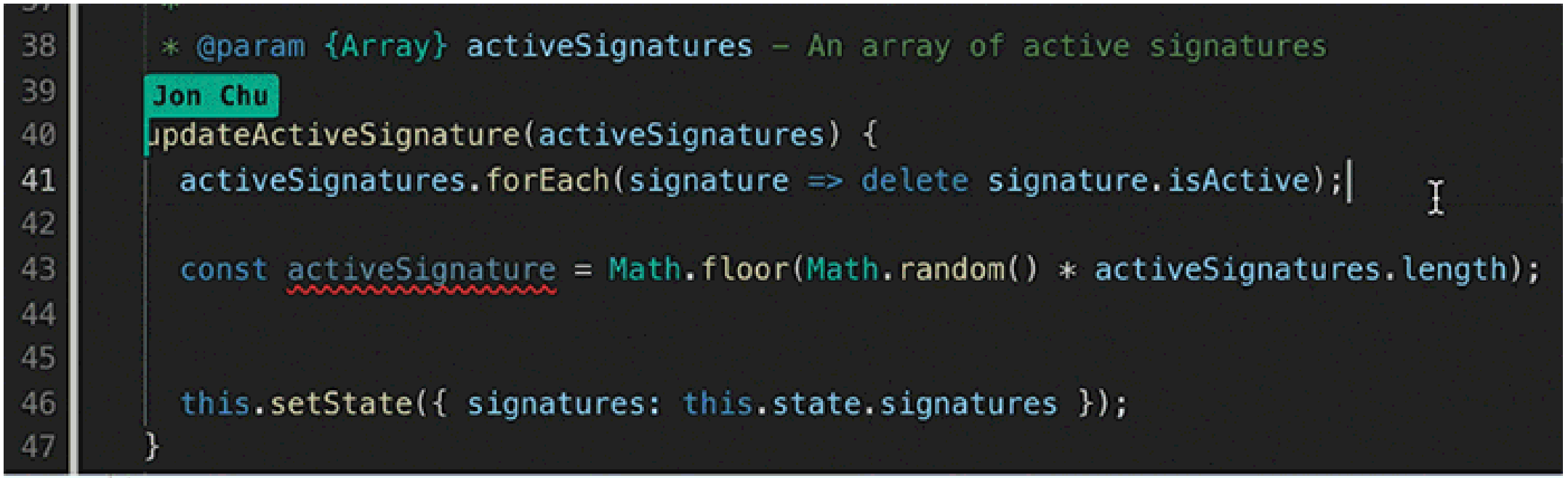
Internally, R connections are represented by an C-level `Rconn` struct. This structure has a field `void *id` that initated at zero and then incremented by one (= +4 bytes) for each new connection created (via `init_con()`).

Now, it turns out that the value of the unique `id` field is recorded in and exposed to us via (non-documented) attribute `conn_id` through the `void *ex_ptr` field (e.g. https://github.com/wch/r-source/blob/tags/R-3-5-1/src/main/connections.c#L5439-L5440 for file connections). For example,

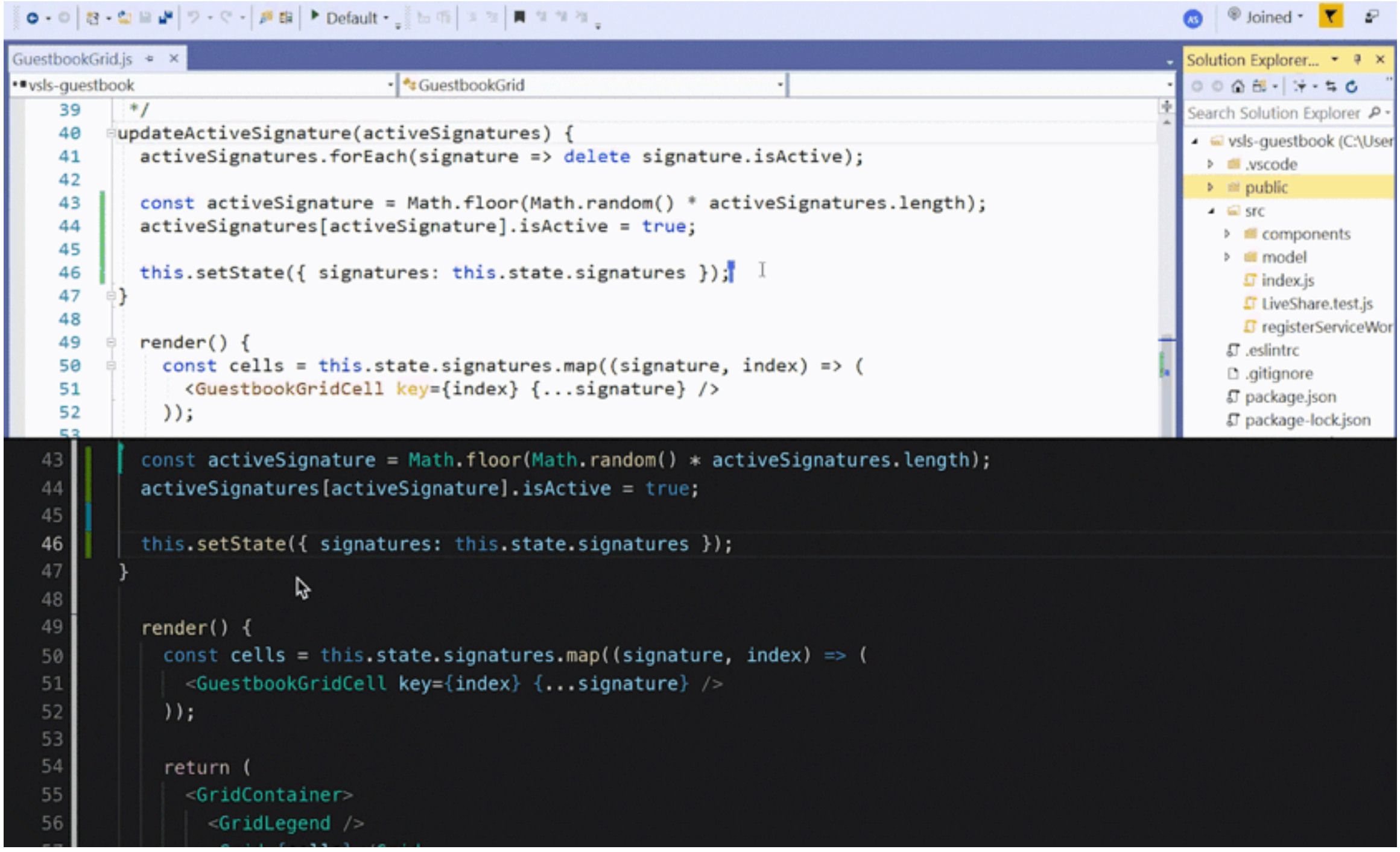
```
> str(fh)
 'file' int 3
 - attr(*, "conn_id")=<externalptr>
> attr(fh, "conn_id")
<pointer: 0x373>  ## <== unique 'id' value

> str(fh2)
 'file' int 3
 - attr(*, "conn_id")=<externalptr>
> attr(fh2, "conn_id")
<pointer: 0x37c>  ## <== unique 'id' value
```

VS Code Live Share




```
37  *
38  * @param {Array} activeSignatures - An array of active signatures
39  */
40 updateActiveSignature(activeSignatures) {                    Amanda Silver
41   activeSignatures.forEach(signature => delete signature.isActive);
42
43   const activeSignature = Math.floor(Math.random() * activeSignatures.length)
44
45
46   this.setState({ signatures: this.state.signatures });
47 }
48
```


Default
Joined
GuestbookGrid.js
vsls-guestbook
GuestbookGrid
Solution Explorer...
Search Solution Explorer
vsls-guestbook (C:\User
.vscode
public
src
components
model
index.js
LiveShare.test.js
registerServiceWor
.eslintrc
.gitignore
package.json
package-lock.json
*/
updateActiveSignature(activeSignatures) {
activeSignatures.forEach(signature => delete signature.isActive);
const activeSignature = Math.floor(Math.random() * activeSignatures.length);
activeSignatures[activeSignature].isActive = true;
this.setState({ signatures: this.state.signatures });
}
render() {
const cells = this.state.signatures.map((signature, index) => (
<GuestbookGridCell key={index} {...signature} />
));
return (
<GridContainer>
<GridLegend />