

TESTING

August 16, 2025 at 04:57 PM

Object Browser Testing Guide

Testing the New Object Browser Feature

The object browser is implemented in the feature/object-browser branch and provides vim-peekaboo style R workspace inspection.

Quick Start Testing

1. **Switch to feature branch:**

```
git checkout feature/object-browser
```

2. **Open test file:**

```
vim test_files/object_browser_demo.R
```

3. **Start R terminal:**

- Press <LocalLeader>r (usually \r)

4. **Create test objects:**

- Execute the demo script with <CR> on various lines
- Or select all and execute with visual mode + <CR>

5. **Open object browser:**

- Press <LocalLeader>" (usually \")
- Should see right panel with object list

Expected Behavior

Browser Window:

- Opens as 40-column vertical split on right
- Shows numbered list of R objects with types

- Objects display as: 1. object_name (type dimensions)
- Clean, uncluttered interface

Navigation:

- **Number keys 1-9**: Quick inspect specific objects
- **Arrow keys + <CR>**: Navigate and inspect at cursor
- **ESC**: Return to object list (from detail view)
- **q**: Close browser entirely

Object Inspection:

- **Data frames**: Structure + head() preview
- **Vectors**: First/last elements if long
- **Models**: Summary statistics
- **Lists**: Nested structure display
- **Functions**: Class and basic info

Test Cases

1. Basic Functionality

```
# Create simple objects
x <- 1:10
df <- data.frame(a = 1:5, b = letters[1:5])

# Test browser:
# - Press <LocalLeader>"
# - Should see both objects listed
# - Press 1 to inspect x
# - Press 2 to inspect df
```

2. Navigation Flow

```
# Test navigation:
# - Open browser with <LocalLeader>"
# - Press 1 (should show object details)
# - Press ESC (should return to object list)
# - Press q (should close browser)
# - Press <LocalLeader>" again (should reopen)
```

3. Different Object Types

```
# Create variety:
vec <- 1:100                # Long vector
mat <- matrix(1:12, 3, 4)   # Matrix
lst <- list(a = 1:3, b = "hi") # List
mod <- lm(mpg ~ wt, mtcars)  # Model

# Test each type shows appropriate details
```

4. Error Handling

```
# Test edge cases:
# - Open browser with no R terminal (should show error)
# - Open browser with empty workspace
# - Try browser in non-R file (should show error)
```

Performance Testing

Large Workspace:

```
# Create many objects:
for(i in 1:20) {
  assign(paste0("obj_", i), rnorm(100))
}

# Browser should:
# - Load quickly
# - Show all objects
# - Navigate smoothly
```

Complex Objects:

```
# Create complex nested structures:
complex <- list(
  data = data.frame(matrix(rnorm(10000), 100, 100)),
  models = list(
    lm1 = lm(mpg ~ wt, mtcars),
    lm2 = lm(mpg ~ hp, mtcars)
  ),
  metadata = list(
    created = Sys.time(),
```

```

    version = R.version.string
  )
)

# Should handle gracefully without hanging

```

Integration Testing

Multi-Terminal Workflow:

```

# Test with multiple R files:
# 1. Open file1.R, create objects, open browser
# 2. Open file2.R in new split, create different objects
# 3. Each browser should show only its buffer's objects

```

Existing Functionality:

```

# Verify browser doesn't break existing features:
# - <CR> smart submission still works
# - <LocalLeader>h/s/d object inspection still works
# - Chunk navigation in Rmd files still works

```

Troubleshooting

Common Issues:

1. **Browser doesn't open:**
 - Check you're in R file (set ft=r)
 - Verify R terminal exists (<LocalLeader>r)
 - Try Ex command: :RObjectBrowser
2. **Empty object list:**
 - Check R workspace has objects (ls() in R)
 - Verify terminal is active
 - Wait a moment and try refreshing
3. **Navigation not working:**
 - Ensure cursor is in browser window
 - Try number keys 1-9
 - Check buffer-local mappings are active

Debug Commands:

```

" Check if function exists:
:echo exists('*s:RObjectBrowser')

```

```
" Check terminal association:
:RShowTerminal
```

```
" Manual browser call:
:RObjectBrowser
```

Success Criteria

[CHECKMARK] **Feature is ready for merge when:**

- Browser opens instantly with <LocalLeader>"
- Shows all workspace objects with correct types
- Number key navigation works smoothly
- ESC/q close behavior works properly
- No conflicts with existing functionality
- Works across different R file types (R, Rmd, Qmd)
- Handles empty workspace gracefully
- Error messages are clear and helpful

Rollback Plan

If issues are found:

```
# Return to master:
```

```
git checkout master
```

```
# Delete feature branch:
```

```
git branch -D feature/object-browser
```

```
# Or keep for later fixes:
```

```
git checkout feature/object-browser
```

```
# Make fixes and re-test
```

The object browser represents a significant enhancement to zzzvim-R's capabilities, bringing modern IDE-style object inspection to the Vim environment while maintaining the plugin's lightweight, terminal-based philosophy.