

Pharo 13

<http://www.pharo.org>

ESUG 2025





Outline

- Pharo 13 remarkable achievements
- Pharo 14 under dev



Settings Browser

Find in settings

Code Browsing

General Code Diffs Code Formatting QualityAssistant Calypso Browser

Appearance

Code Browsing

Compiler

Network

System

Task it execution proc

Text Editing

Tools

General

Default Formatter

EFormatter

Enlumeur Pretty Printer

Max line length

70

Place comment close to expression

☒

Indent Style

tabulation

Number of spaces in indent

Indents for keywords

When block is unopened

Terminate Suspend Signal Priority Debug Inspect

 (80) DelaySemaphoreScheduler(DelayMicrosecondTicke
 (70) Callback queue
 (60) SDL2 Event loop
 (60) Low Space Watcher
 (50) Finalization Process
 (40) WatchDog (804c18fd-8bf1-0d00-a2bd-f7a602156ca9)
 (40) ThreadSafeTranscript worker: f4c04efd-8bf1-0d00-a2
 (40) Morphic UI Process
 (31) Calypso update
 (10) Idle Process

```

1 next
2   "Fetch the next item from queue.
   operation can complete"
3
4   | result |
5
6   [ | keepWaiting |
7     keepWaiting := false.
8     result := self nextIfNone: [ keepWaiting := true ].
9     keepWaiting ] whileTrue: [ self waitForNewItems ].
10
11   ^ result

```

Object Transcript

Hide Flash Stop Logger

Timestamp	Name	Contents
2025-05-20 11:34,616	Wrapper on ByteString	'this is a message only interesting from the logger domain'
2025-05-20 11:40,281	Wrapper on AlphaImageMorph	an AlphaImageMorph(390965760)
2025-05-20 11:42,757	Wrapper on AlphaImageMorph	an AlphaImageMorph(390965760)
2025-05-20 11:45,179	Wrapper on AlphaImageMorph	an AlphaImageMorph(390965760)

All logs Selected Log

a WrapperSignal (2025-05-20...

an AlphaImageMorph(390965760)

Raw Breakpoints Meta

Morph Layout Raw Breakpoints Meta

Variable	Value
self	2025-05-20T11:45:49.179+02:00 Wr
timestamp	2025-05-20T11:45:49.179+02:00
processId	Morphic UI Process
properties	a Dictionary [0 items] ()

Welcome

Welcome to Pharo 13



Welcome to Pharo, an immersive live programming environment.

Pharo is a pure object-oriented programming language and a powerful environment, focused on simplicity and immediate feedback (think IDE and OS rolled into one).

For more information, please visit here: <http://pharo.org>

Application Building

with

Spec 2.0

 K. De Hondt
 S. Ducasse
 with
 E. Lorenzano
 S. Jordan Montañó

self emit

Spec book is out!

Spec is the default UI building framework for Pharo. With Spec, developers focus on user interaction and widget layout. Spec is built around the Model View Presenter pattern. Presenters are responsible for **defining** the interaction logic of the application. They enforce the interaction of their subelements as well as domain objects.

With Spec, existing presenters or even complete UIs are **reused** and **configured** to form a new user interface. Spec is widget framework (GTK, Morphic, Toplo) agnostic. It makes desktop applications simple to build.

This book shows the basics of Spec in **Pharo 12** and how it enables reuse of user interface elements. It focuses on Spec 2.0 features such as new widget layouts, dynamic widget building and more advanced features such as transmissions or commands.

We will update it for P13/14

- A minimal email client revisiting all Spec 2.0 features.

Keepers of the lighthouse



Application Building with Spec 2.0

K. De Hondt and S. Ducasse with E. Lorenzano and S. Jordan-Montaño



Application
Building

with

Spec 2.0

K. De Hondt
S. Ducasse
with
E. Lorenzano
S. Jordan-Montaño

Fast-paced dedication

- JUST in the main Pharo repository 698 Pull Requests
- Closed 865 issues
- Contributions from more than 70 different contributors
- Other Pharo projects
 - <http://github.com/pharo-spec/NewTools>
 - <http://github.com/pharo-spec/Spec>
 - <http://github.com/pharo-vcs/Iceberg>
 - <https://github.com/pharo-graphics/Roassal>
 - <http://github.com/pillar-markup/Microdown>
 - <http://github.com/pharo-project/pharo-vm>

UX infrastructure changes

- All Spec Tools remember their size
 - This may lead to adaptation from your side
 - Forced to change the API
 - Check the documentation

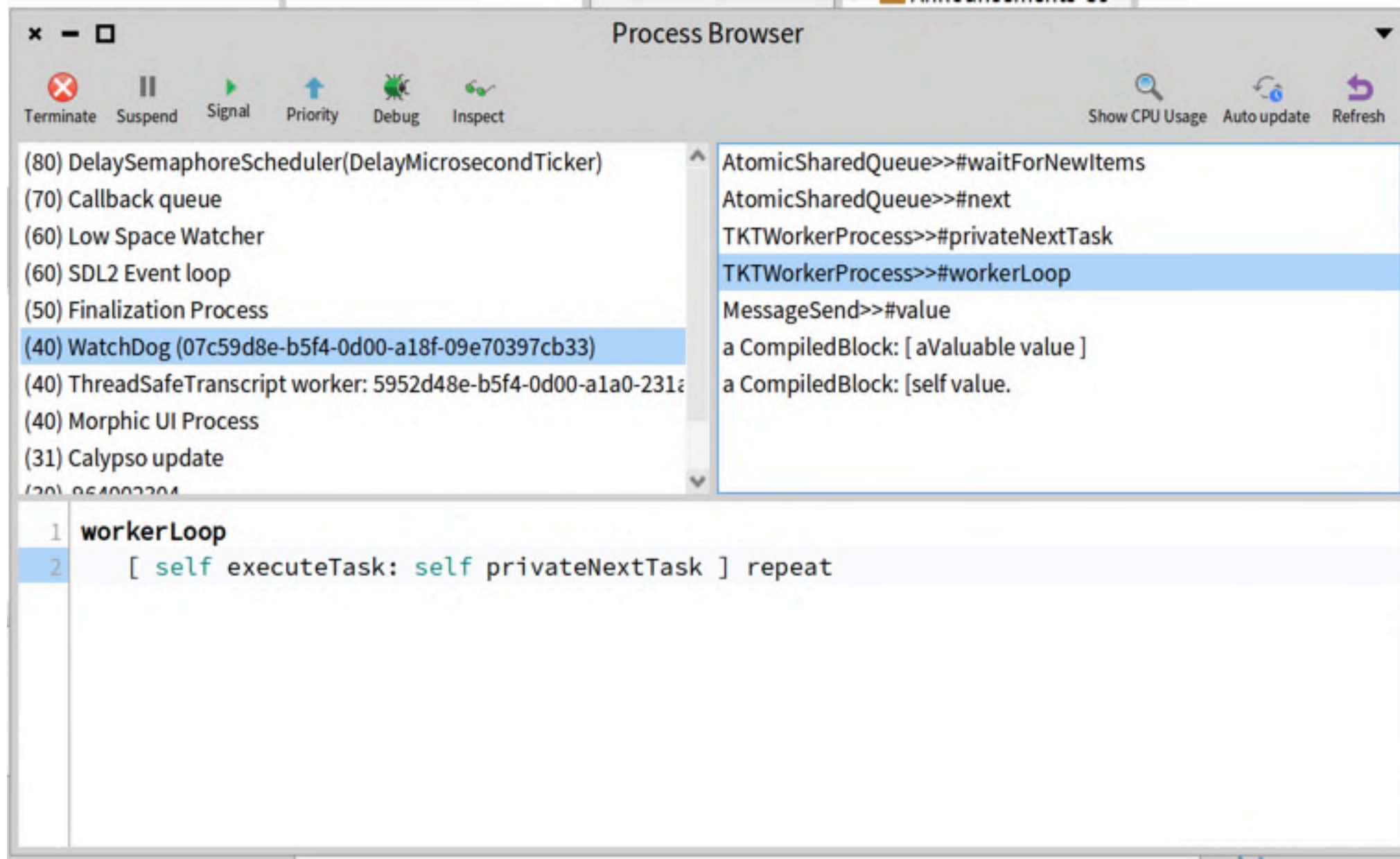
Better source navigation

- You can navigate using camelCase: andArrows:
- With a lot of nice tests
- Faster, more productive
- Tx Guille and Gabriel
- We should ask Guille to make a little video

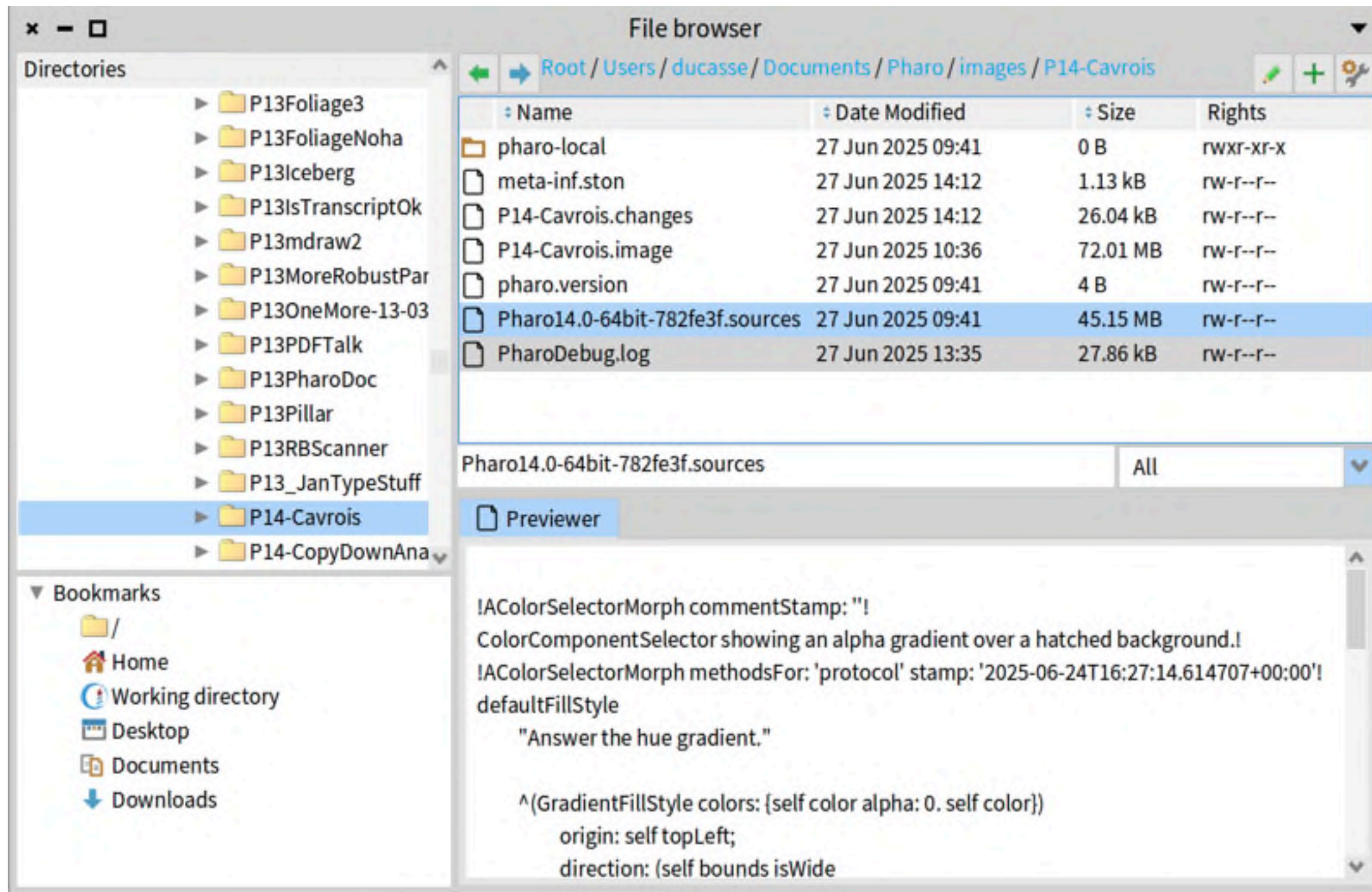
New Tools in Spec

- New Process Browser
- New Transcript
- New Finder
- New File Browser
- Preview
 - New Object Transcript
 - New Settings Browser

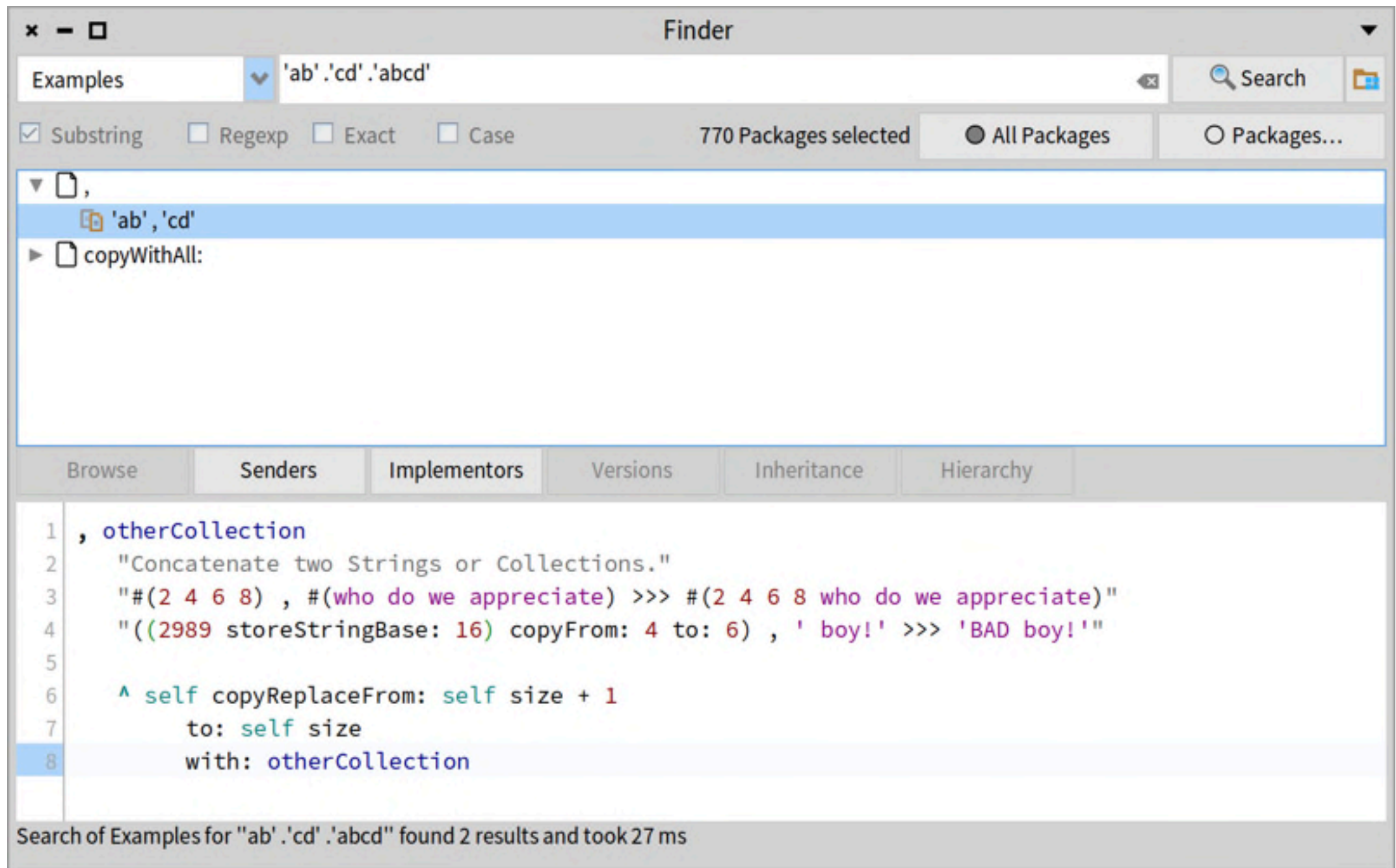
New Process Browser



New File List Browser



New Finder



New Transcript

×

▬

□

Transcript

▼

✕

Clear

🗑️

Reset

⬇️

Fetch

Hello Pharoers

[Preview] Object Transcript

The screenshot displays the 'Object Transcript' application window. At the top, there are window controls and three buttons: 'Hide' (with a green eye icon), 'Flush' (with a blue trash can icon), and 'Stop logger' (with a red stop icon). Below these is a table with three columns: 'Timestamp', 'Name', and 'Contents'. The first row shows a timestamp of '2025-06-27 14:21.906', a name of 'Wrapper on Array', and a content snippet '{Exception. Path. AbstractAnnouncemer'. Below the table are two tabs: 'All logs' and 'Selected Log'. The 'Selected Log' tab is active, showing two log entries: 'an Array [1 item] (2025-06-...)' and 'a WrapperSignal (2025-06-27...)', each with a set of icons (a red bug, a green bug, and a blue refresh icon). Below the log entries are two side-by-side panels. The left panel has tabs 'Items', 'Raw', and 'Meta', with 'Items' selected. It shows a list of items with a header ': Value' and a single entry '1 2025-06-27T14:21:17.906013+02:00 WrapperSig'. The right panel has tabs 'Raw' and 'Meta', with 'Raw' selected. It shows a tree structure of object properties with a header ': Variable' and ': Value'. The tree includes 'self', 'timestamp', 'processId', 'properties', and 'target'. At the bottom of each panel is a small table with a single row containing '1 self'.

Timestamp	Name	Contents
2025-06-27 14:21.906	Wrapper on Array	{Exception. Path. AbstractAnnouncemer

an Array [1 item] (2025-06-...) a WrapperSignal (2025-06-27...)

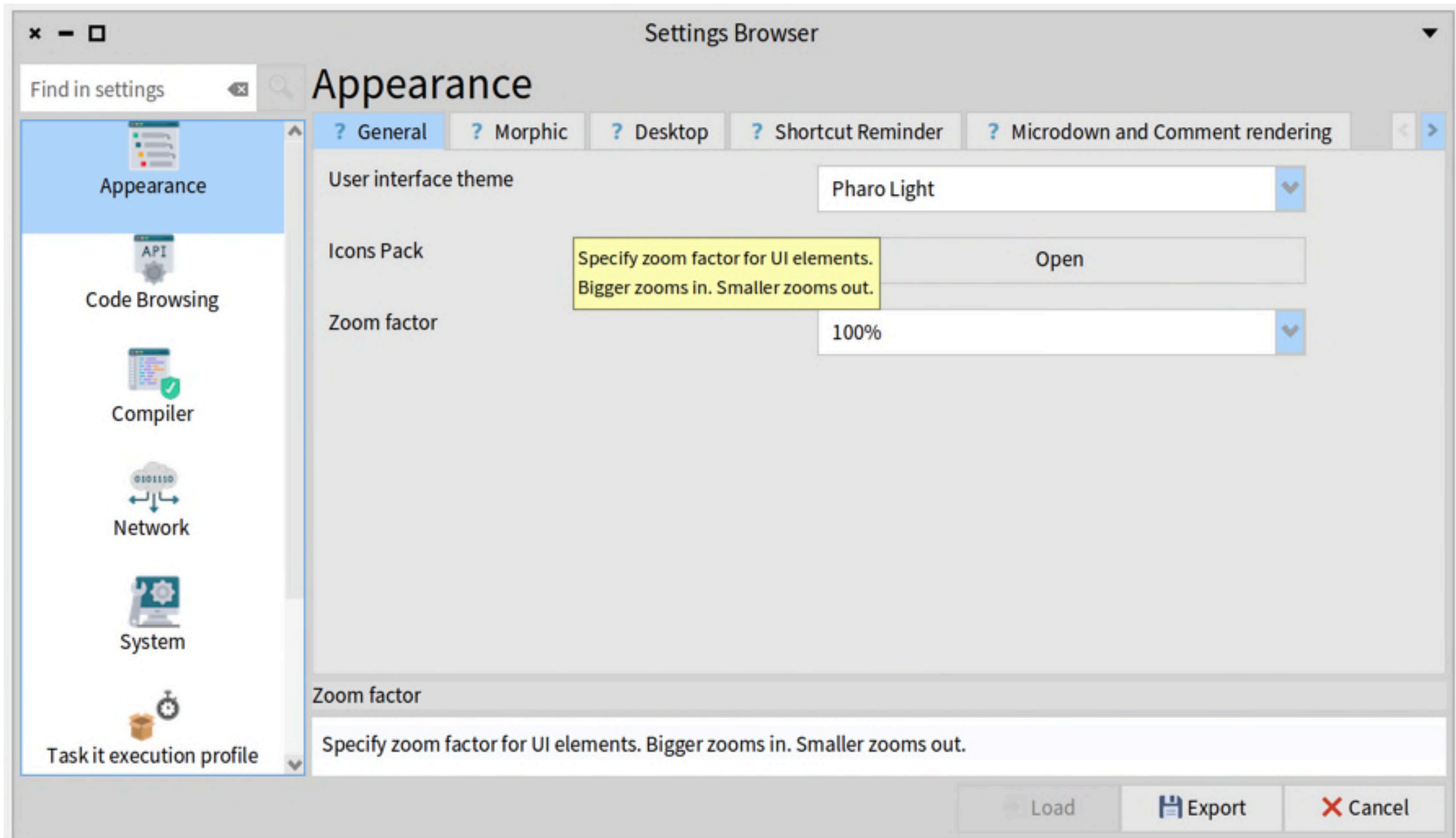
: Value
1 2025-06-27T14:21:17.906013+02:00 WrapperSig

: Variable	: Value
self	2025-06-27T14...
timestamp	2025-06-27T14...
processId	Morphic UI Pro
properties	a Dictionary [C
target	an Array [1,500

1	self
---	------

1	self
---	------

[Preview] Settings Browser



Spec SpEasyComponent*

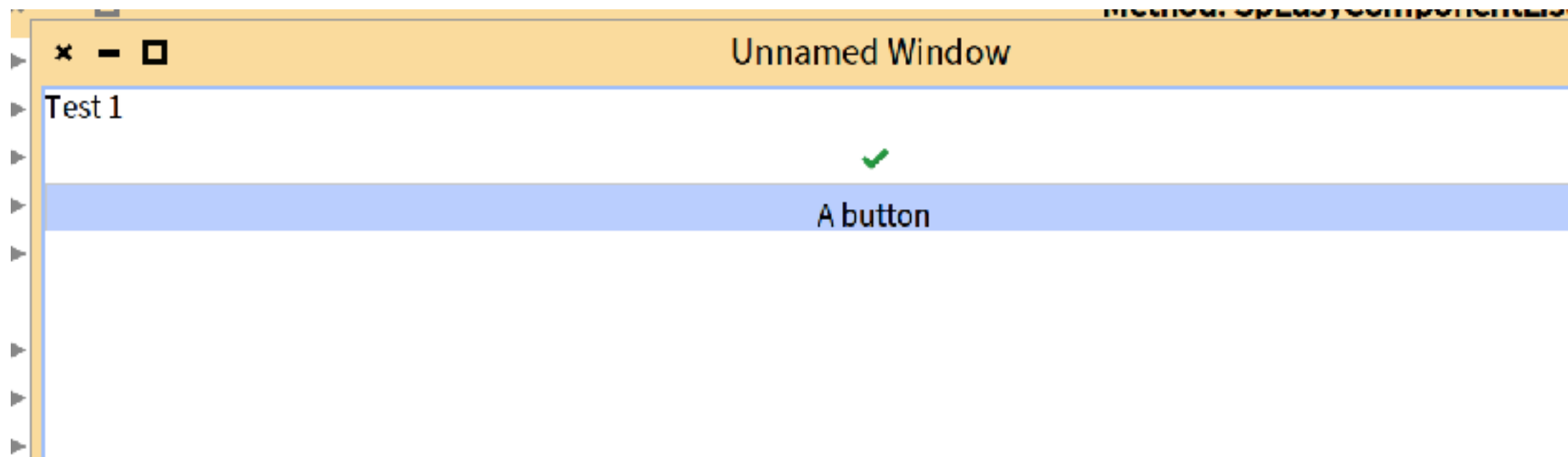
- New list, tables, and trees
- allow ***any*** presenter as their content












































Spec SpEasyComponent*

| list |

```
list := { SpLabelPresenter new label: 'Test 1'; yourself.  
         SpImagePresenter new image: (self iconNamed:  
         #smallOk); yourself.  
         SpButtonPresenter new label: 'A button'; yourself }.  
SpEasyComponentList Presenter new  
presenters: list;
```

open



 AbsolutePath (12)	 
 AbsolutePathStoredSetting (1)	  
 AbstractAnnouncementSubscription (14)	
 AbstractBinaryFileStream (42)	  
 AbstractCache (21)	  
 AbstractClassDescriptionAnnouncementTest (3)	
 AbstractClassDescriptionTest (4)	 
 AbstractDebugger (3)	  
 AbstractDelayTicker (7)	  
 AbstractDrTestsPresenter (27)	  
 AbstractEnumerationVisitor (6)	 
 AbstractEnumerationVisitorTest (4)	  
 AbstractEnvironmentTestCase (4)	 
 AbstractFileReference (171)	  
 AbstractFont (61)	
 AbstractFontFamily (6)	  
 AbstractFontFamilyMember (13)	  
 AbstractFontFamilyMemberTest (1)	

```

app := SpApplication new.
app addStyleSheetFromString: '.application [
    .label [
        .italic [ Font { #italic : true } ],
        .bold [ Font { #bold : true } ],
        .red [ Draw { #color : #red } ],
        .green [ Draw { #color : #green } ],
        .yellow [ Draw { #color :
#orange } ]
    ]
]'.
SpEasyListViewPresenter new
    application: app;
    rowPresenterClass: ComposedRowPresenter;
    items: Smalltalk allClassesAndTraits;
    open.

```

```

defaultLayout
    ^ SpBoxLayout newLeftToRight
        spacing: 5;
        add: iconPresenter expand: false;
        add: labelPresenter expand: false;
        add: countPresenter;
        add: actionsPresenter expand: false;
        yourself

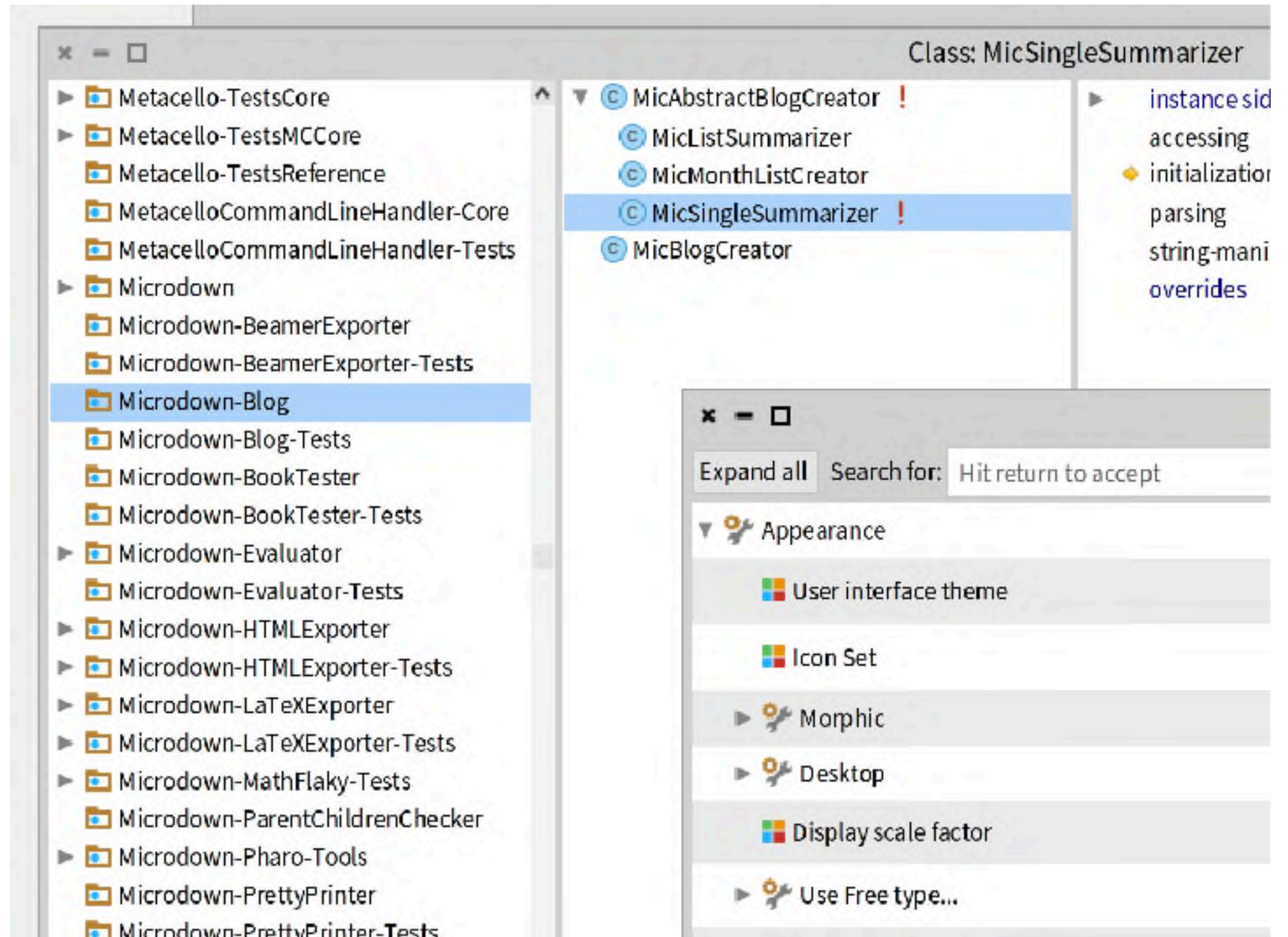
initializePresenters
    iconPresenter := self newImage.
    labelPresenter := self newLink.
    countPresenter := self newLabel
        addStyle: 'dim'; yourself.
    actionsPresenter := self newPresenter
        layout: SpBoxLayout newLeftToRight;
        yourself.

updatePresenter
    | actions |
    self model ifNil: [ ^ self ].
    iconPresenter image: (self iconNamed: self model
systemIconName).
    labelPresenter label: self model name.
    countPresenter label: '(', self model numberOfMethods
asString, ')'.
    self model numberOfMethods > 5 ifTrue:
[ countPresenter addStyle: 'green' ].
    self model numberOfMethods > 8 ifTrue:
[ countPresenter addStyle: 'yellow' ].
    self model numberOfMethods > 15 ifTrue:
[ countPresenter addStyle: 'red' ].
    actionsPresenter layout removeAll.
    actions := self fakeActions.
    3 atRandom timesRepeat: [
        | action |
        action := actions atRandom.
        actionsPresenter layout add: (self perform:
action).
        actions := actions copyWithout: action ]

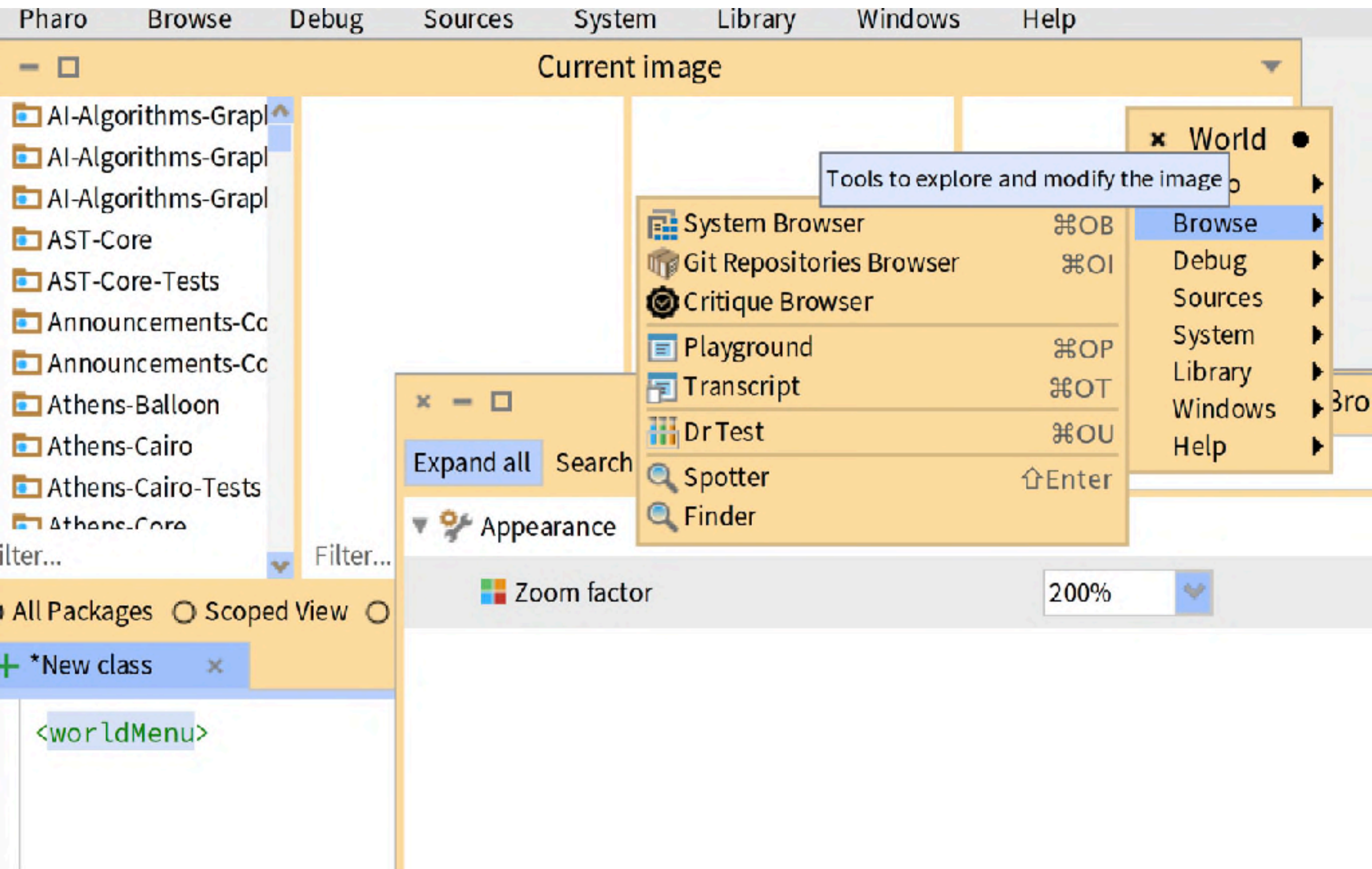
```

Nice looking fonts

- On retina
- Tx a lot Kris



Polished Zoomable UI



Organic window manager

- Place your tools
 - Snapshot
 - Edit tool behavior
-
- When you open new tools they get placed automatically

PharoBrowseDebugSourcesSystemLibraryWindowsHelp

Current image

AI-Algorithms-GraphAI-Algorithms-GraphAI-Algorithms-GraphAST-CoreAST-Core-TestsAnnouncements-CoAnnouncements-CoAthens-BalloonAthens-CairoAthens-Cairo-TestsAthens-Core

All PackagesScoped ViewProjects

+ New class

Transcript show: 'Hello Pharoers!'

Current image

AI-Algorithms-GraphAI-Algorithms-GraphAI-Algorithms-GraphAST-CoreAST-Core-TestsAnnouncements-CoAnnouncements-CoAthens-BalloonAthens-CairoAthens-Cairo-TestsAthens-Core

Filter...

Filter...

All PackagesScoped ViewProjects

+ New class

Object << #MyClass
slots: {};
package: ''

Playground

Do it allPublishBindingsVersionsPages

1

Line: 1:1

+L

Dr Tests - Tests Runner

Tests RunnerMinimize

Packages (1 selected): Tests Cases (5 selected)

AI-Algorithms-GraphAST-Core-TestsAnnouncements-CoAthens-Cairo-TestsAthens-Core-TestsBeacon-Core-Tests

AthensAffineTAthensProjectAthensTransfAthensAffirAthensProj

Filter...Filter...

Grouped by type of result

Results

Errors (0)

Failures (0)

Skipped tests (0)

▶ Passing tests (60)

ExpectedFailures (0)

Unexpected passed tests (0)

Run Tests

2023-06-27 14:50: Tests finished.

Repositories

Fetch all

SettingsAdd

ProjectsSystem

Repositories

Status

Branch

CozyUp to datemaster

PastelsUp to datemaster

Filter...

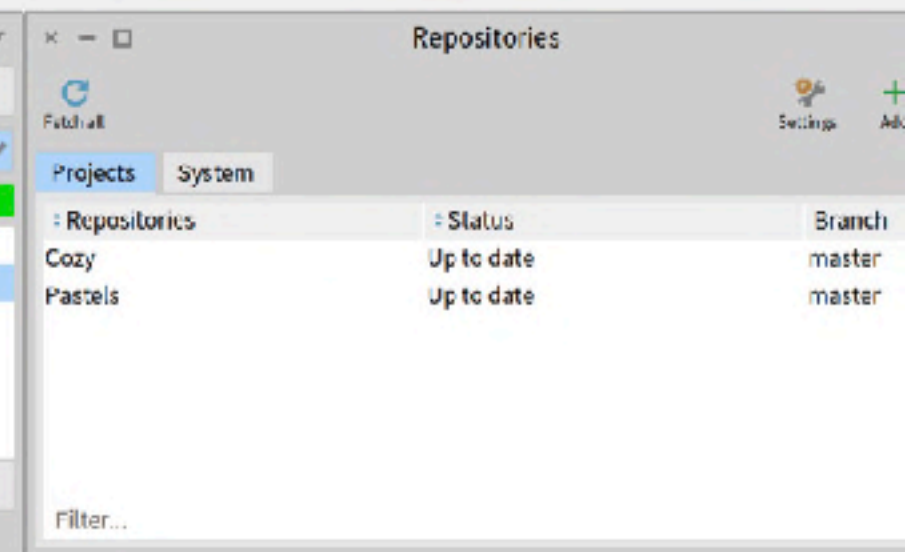
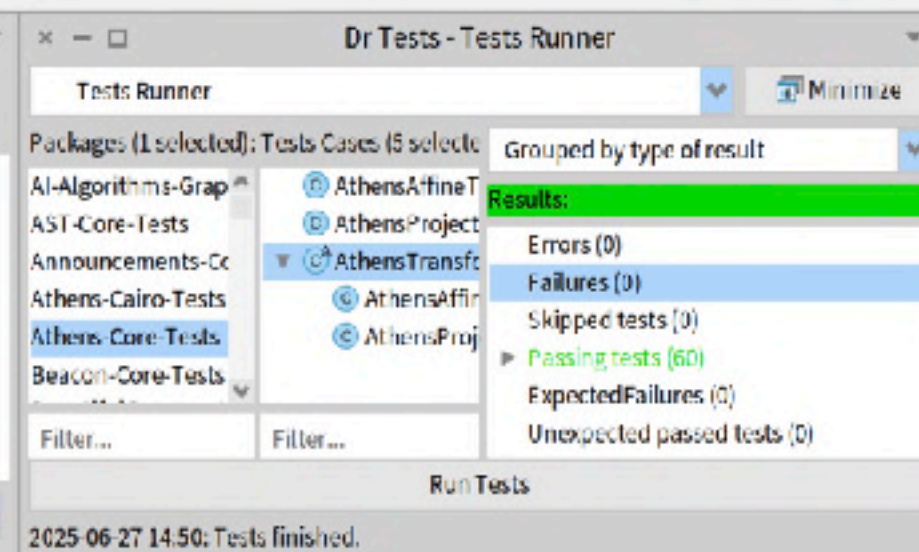
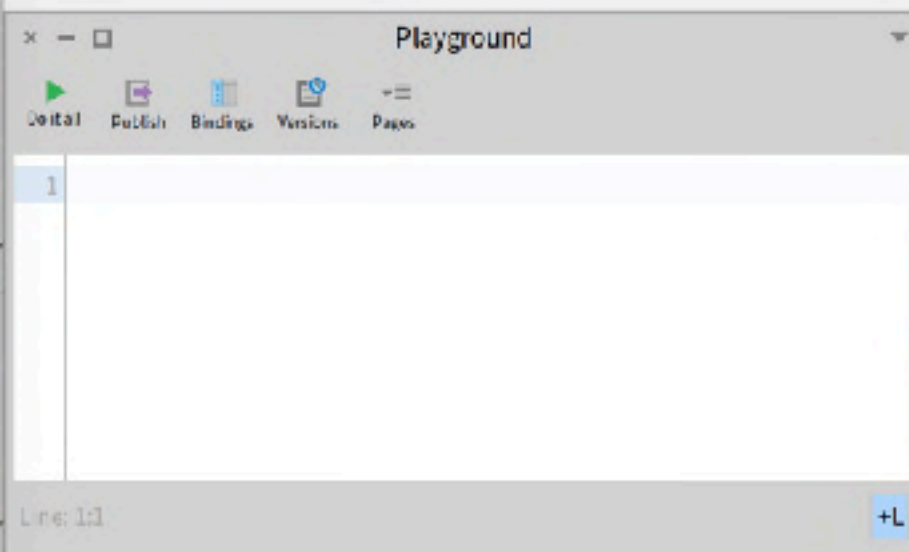
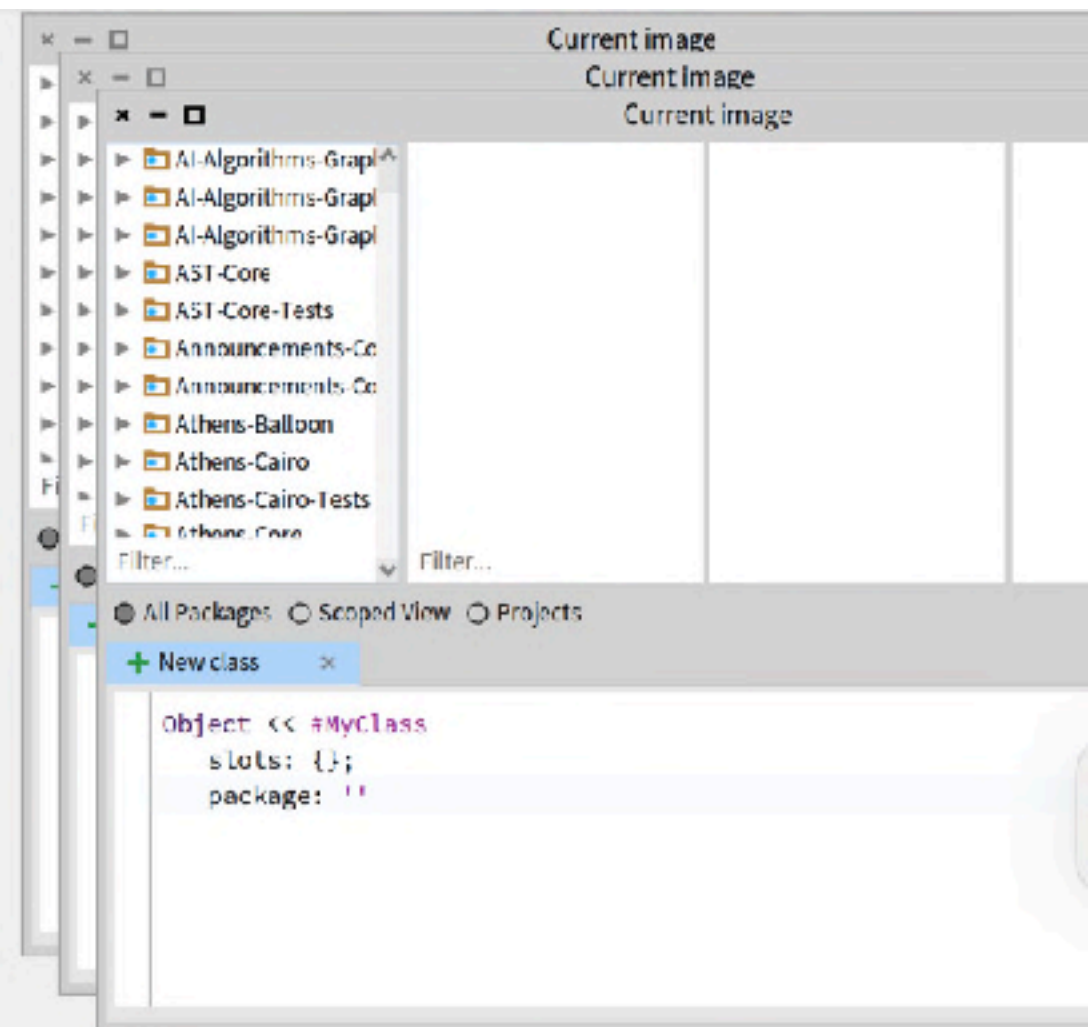
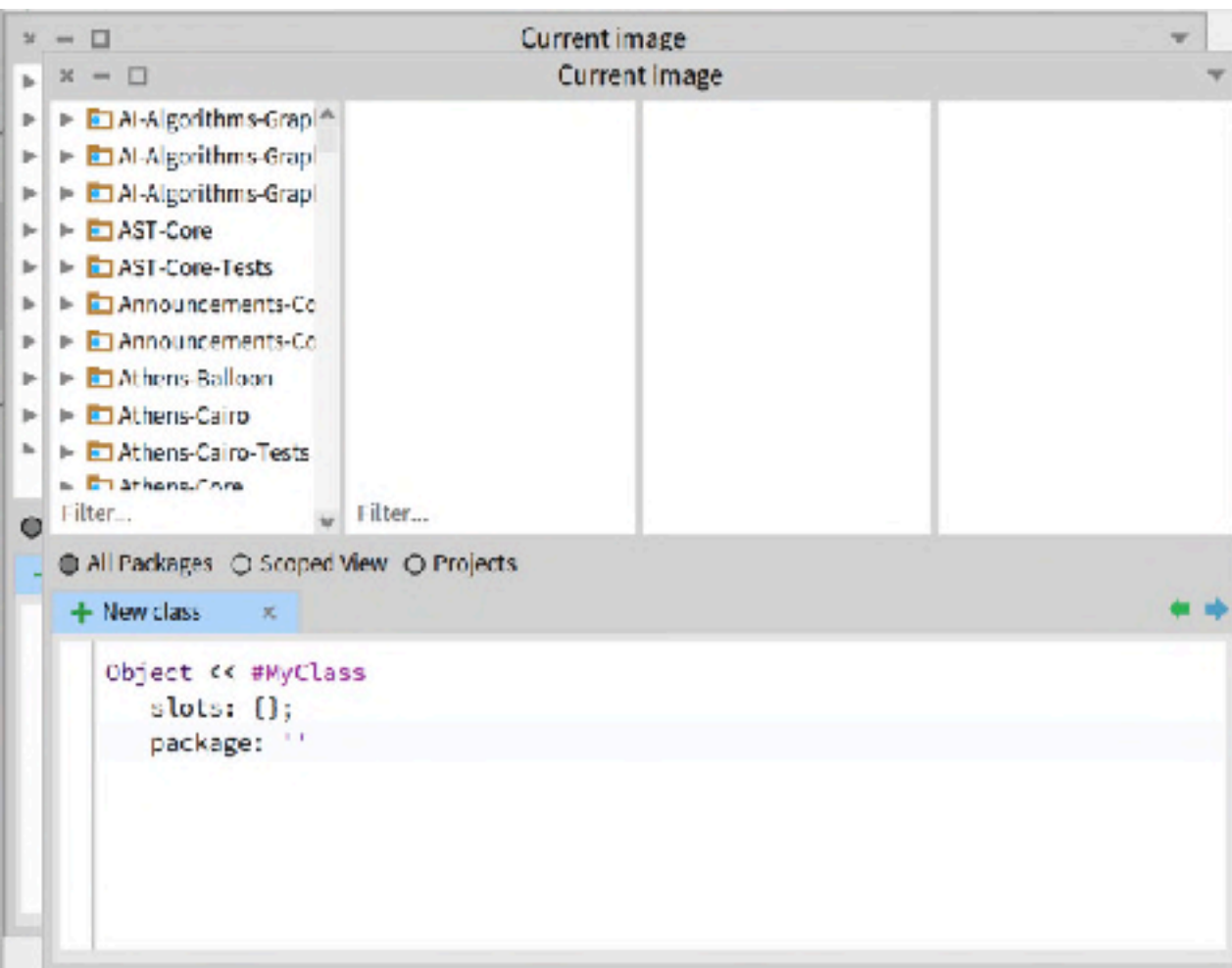
ClyFullBrowserMorph - Stacking Strategy

ClyFullBrowserMorph - Stacking Strategy

StPlaygroundPresenter - Stacking Strategy

DrTests - Stacking Strategy

IceTipRepositoriesBrowser - Stacking Strategy

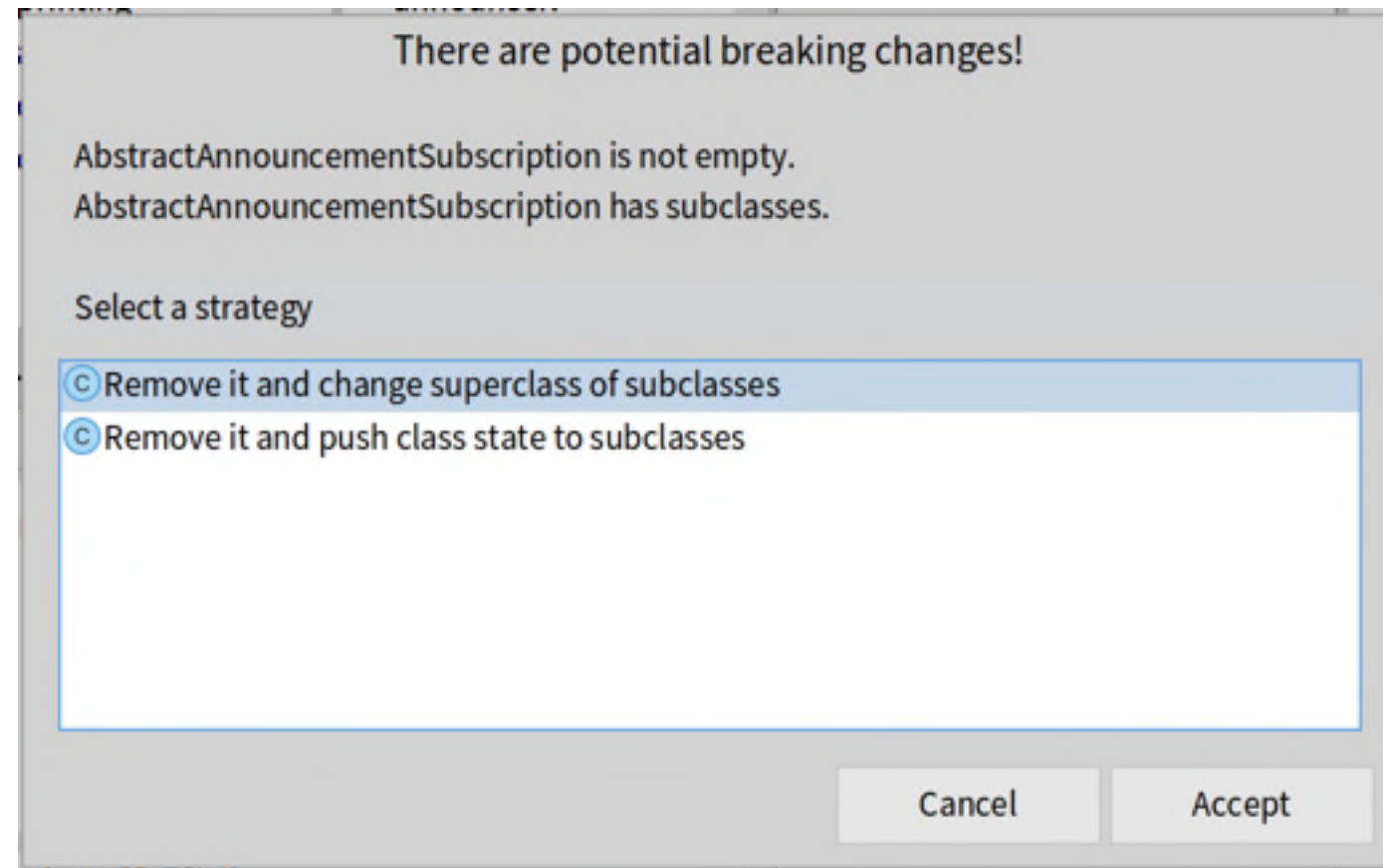


Halt if: / Halt once

- Faster
- More robust implementation
- Do not rely on AST cache anymore
- Introduces other helpers... check the Halt API

More UX for refactorings

- To understand the point
 - You remove a class
 - But it has references
 - But it has instance variables
 - But it has methods
- During the interaction the driver proposes various choices to you



One refactoring driver

- Can drive
 - different refactorings
 - different transformations
- This requires a fine grained precondition protocol

Debugger UX

- When coding in the debugger
- Everybody wants something else for protocol
- So now
 - by default the same protocol than in superclass
 - or not classified (no pop up plague)
- Or you define where you want



Instance of Array did not understand #pharo

Stack

UndefinedObject>>Dolt

[OpalCompiler-Core] OCReceiverDoltSemanticScope (OCDoltSemanticScope)>>evaluateDolt:

[OpalCompiler-Core] OpalCompiler>>evaluate

[Spec2-Code] SpCodePresenter>>[self announcer announce: (SpCodeWillBeEvaluatedAnnounceme



Create method

Problem

Instance of Array did not understand #pharo

Class or Trait

{ } Array
F { } Array
{ } ArrayedCollection
{ } SequenceableCollection
{ } Collection
Ⓢ Object
Ⓢ ProtoObject

1 Array new pharo

an UndefinedObject (nil)



Type	Variable	Value
implicit	Ⓢ self	nil
implicit	⌈ stackTop	#emptyStack
implicit	Ⓢ thisContext	UndefinedObject>>Dolt
implicit	⚡ Exception	Instance of Array did not understand #pharo
implicit	{ } DNU receiver	an Array [0 items] ()

An iteration on debugPoints

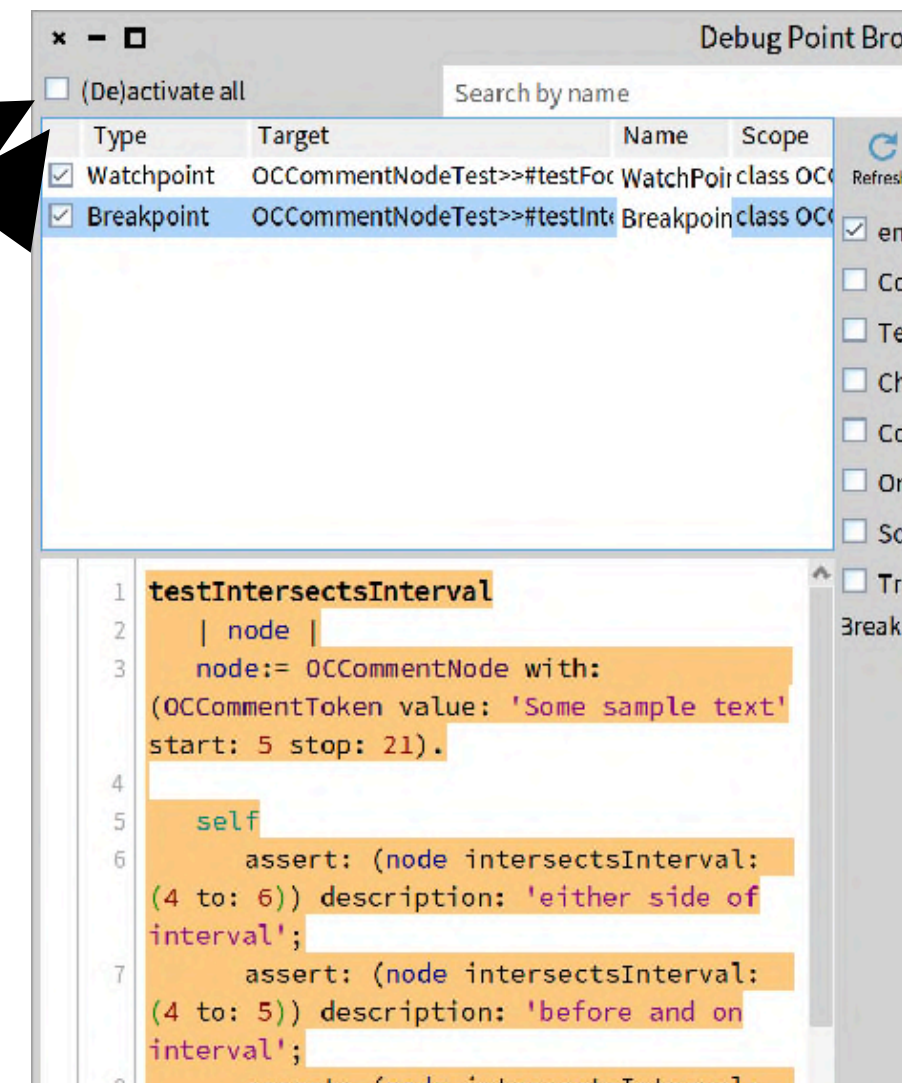
The screenshot shows the 'Debug Point Browser' window. At the top, there is a search bar labeled 'Search by name' and a checkbox for '(De)activate all'. Below this is a table with columns: Type, Target, Name, and Scope. The table contains two entries: a Watchpoint and a Breakpoint, both targeting 'OCCommentNodeTest>>#testFor' and 'OCCommentNodeTest>>#testInt' respectively. To the right of the table are 'Refresh' and 'Remove' buttons. Below the table is a list of checkboxes for various debug point settings: 'enabled: (de)activates debug point', 'Condition: Hit when the condition evaluates to true', 'Test Environment Only: Hits only when executing tests', 'Chain: Each debug point is hit once in sequential order', 'Counter: Tracks how many times the debug point was reached', 'Once: Deactivates debug point after one hit', 'Script: Executes a script at each hit', and 'Transcript: Logs to transcript at each hit'. At the bottom of the window is a code editor showing the following code:

```
1 testIntersectsInterval
2   | node |
3   node:= OCCommentNode with:
4   (OCCommentToken value: 'Some sample text'
5   start: 5 stop: 21).
6   self
7   assert: (node intersectsInterval:
8   (4 to: 6)) description: 'either side of
9   interval';
10  assert: (node intersectsInterval:
11  (4 to: 5)) description: 'before and on
12  interval';
13  assert: (node intersectsInterval:
14  (5 to: 6)) description: 'and and after
15  interval';
16  assert: (node intersectsInterval:
```

DebugPoints during TDD

No need to manually remove the halts when you want to run all your tests

- you set up your halts
- when you want to run all the tests
 - you just disable your halts in one click
- run the tests/commit
- then you reenable your lovely halts



Many many cleanings

Many of the little annoyances that once they are fixed you believe that they were there day one.

- We cleaned again a lot of little issues
- We improved UX
 - like remember the last searched value in the find/replace

VM side of things

- Async IO using epoll on unixes
- Faster byte array / string comparisons
- Improve Windows support for non ASCII filenames
- FreeBSD support
- Update SDL2 version in OSX (Intel & Apple)
- Minimal MacOS version required raised to MacOS 11 and above

Toplo is coming

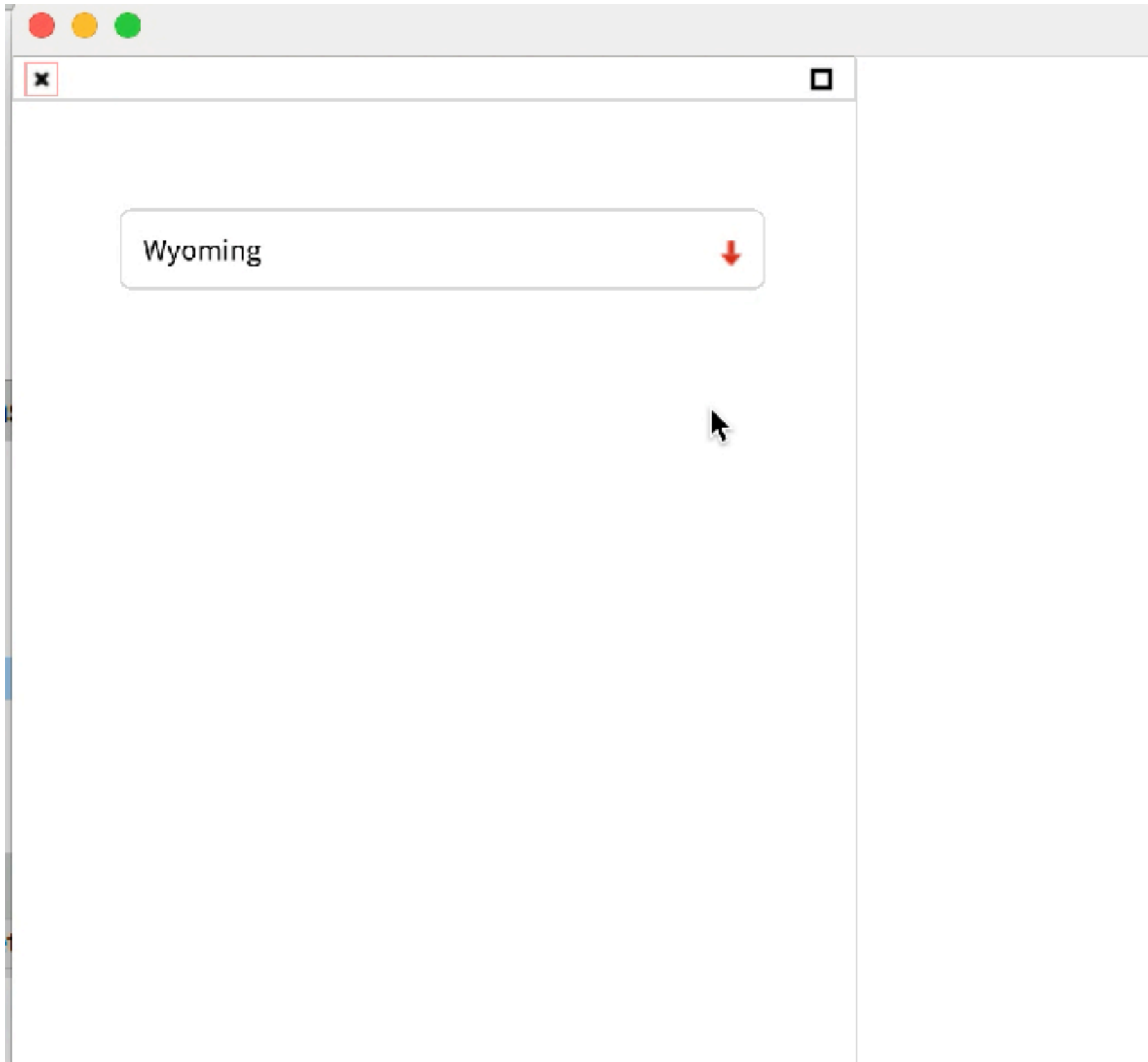
Yes it takes time to make it good

But we are getting there and it is great!

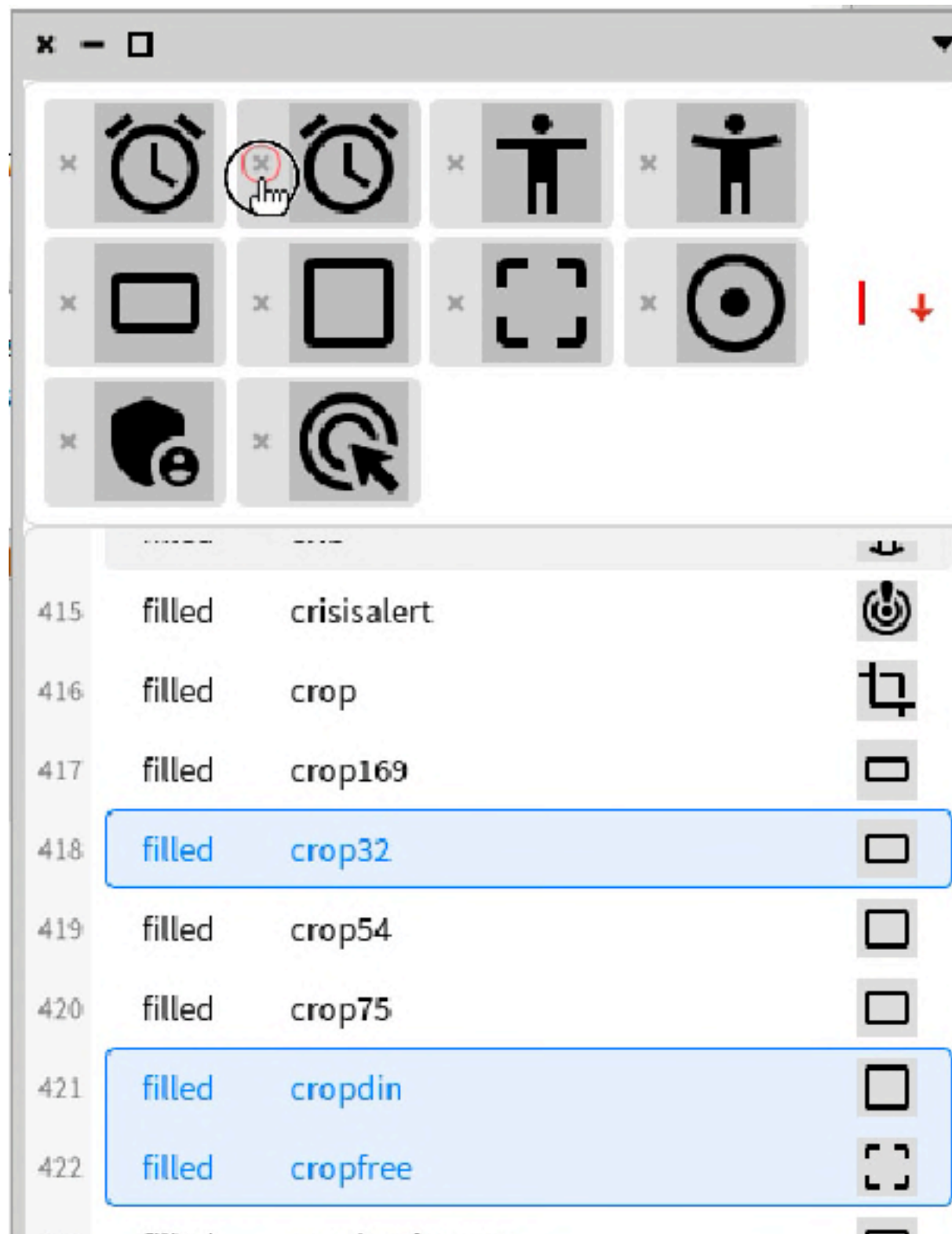
Toplo

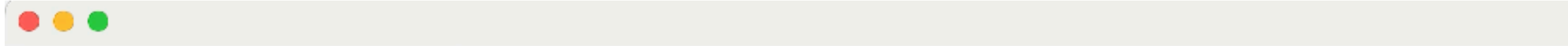
- New widget library on TOP of bLOck
- Sponsored by Thales (with deployed products in 2023)
- Default skins are based on <https://ant.design/>
- Spec back-end is making progress
- Currently
 - All widgets except Table
 - Skin *second iteration*

Combo box with groups



Select with icons

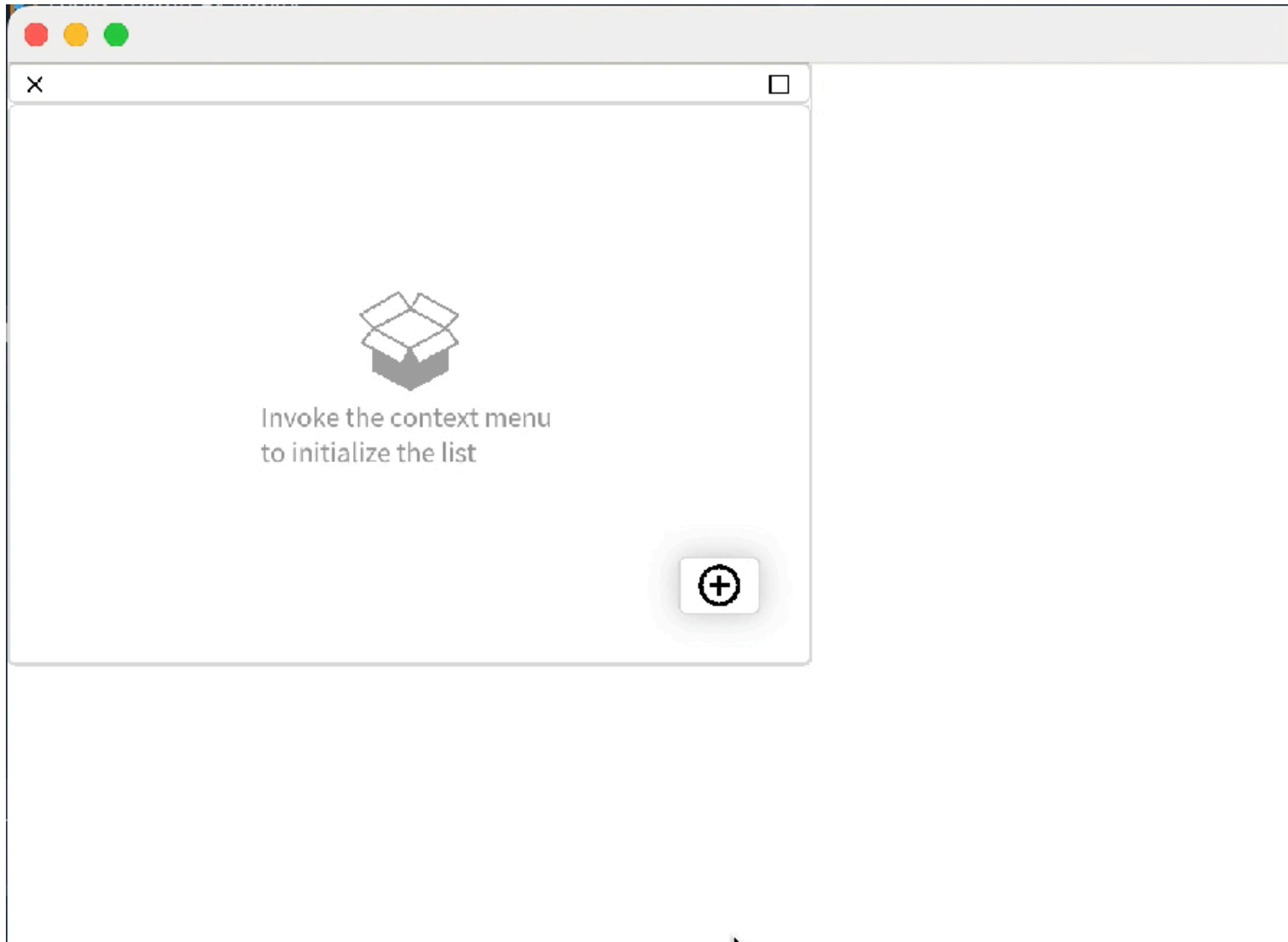




- Vertical ☒
- Wrapping ☒
- Uniform element width ☐
- Scrolling ☒
- Auto-distribution ☐
- Animate selection transition ☐

- 1: Alabama
- 2: Alaska
- 3: Arizona
- 4: Arkansas
- 5: California
- 6: Colorado
- 7: Connecticut
- 8: Delaware
- 9: Florida
- 10: Georgia
- 11: Hawaii
- 12: Idaho
- 13: Illinois Indiana
- 14: Iowa
- 15: Kansas
- 16: Kentucky
- 17: Louisiana
- 18: Maine
- 19: Maryland
- 20: Massachusetts
- 21: Michigan
- 22: Minnesota
- 23: Mississippi
- 24: Missouri
- 25: Montana
- 26: Nebraska

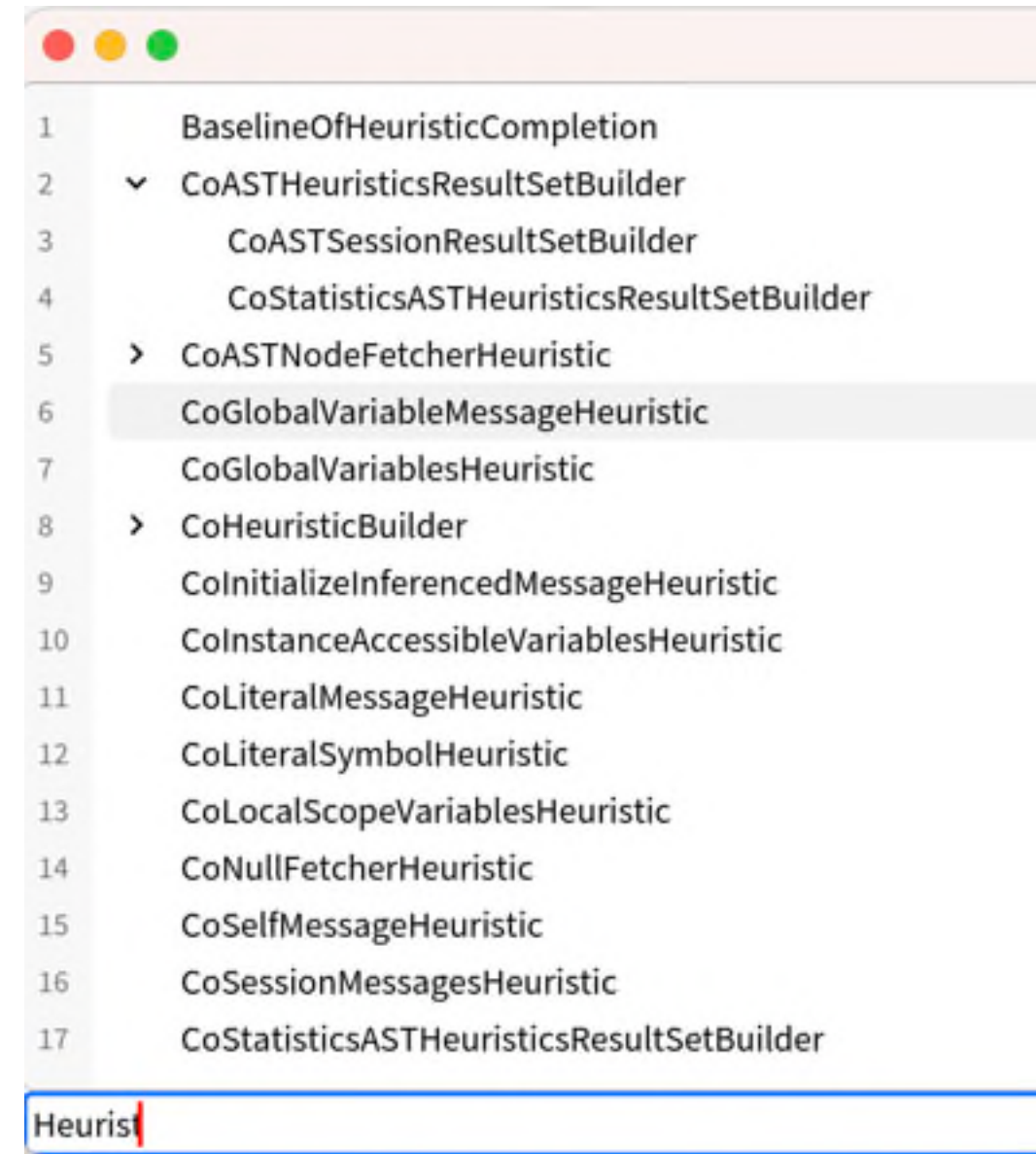
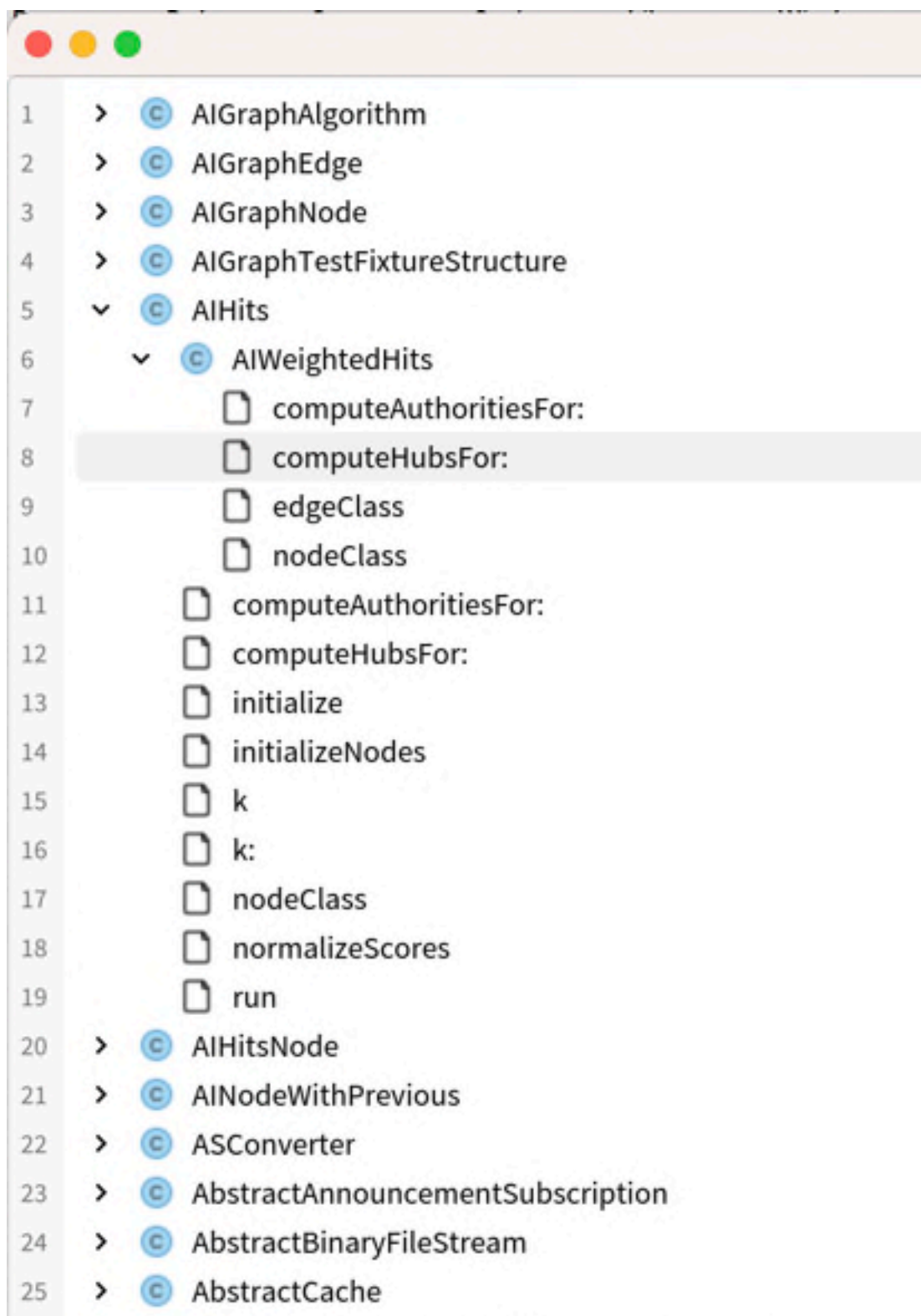
Floating buttons



Accordeons / Trees



Trees



Goals for P14



Works under way

- Bloc/Toplo integration
 - Spec Toplo-backend
 - This implies to cut off our dependencies to Morphic
- Better file support for Windows
- Working on new compiler (named Armor) to support first-class modules
 - new static description for debugging API
- Better refactoring effort continued

Works under way

- Calypso on Steroids
- New Message Browser
- New version of Cavois with cells
- Revisiting completion
 - Package level completion
 - Package Dependency
 - History (soon)

Works under way - the VM side

- Better Slang
- Fundamental work in the code zone design
- More work on Dropal (Pharo compiler with design of a real compiler - ssa, basic blocks, ...)
- Evaluation of static method call
 - Support for writing primitives from within Pharo (like in Bee)

Looking for an engineer

Talk to our Guishe guillermo.polito@inria.fr

**This is more than 10 years I dreamed about
another UI stack and we are there with a
gorgeous one**

The future is promising

PharO

