Beagle Smalltalk Overview

David Buck

# Introduction

Beagle Smalltalk is an implementation of the Smalltalk programming language. Smalltalk is a dynamic programming language that lets you explore ideas without long waits to compile or run your code.

# Launching Beagle Smalltalk

In Windows, launch:

StartupBeagleSmalltalk.bat

In Linux, open a shell and cd to the BeagleSmalltalk directory. From there, run:

./beagle beagle.im  
Then open the web browser on BeagleUI.html

If it’s successful, you’ll see something like this:

A screenshot of a computer

AI-generated content may be incorrect.

The message “Connected!!” means that Smalltalk is running.

# The Screen Buttons

The buttons on the screen allow quick access to Smalltalk tools and to low-level tools that can be used if the window-based tools don’t work or are inappropriate.

## Workspace

The Workspace button opens the Workspace window. The workspace is a window that allows you to execute Smalltalk code. You can select text in the workspace and use the pop-up menu to execute the code.

* Do It – run the selected text without printing the result
* Print It – run the selected text and print the result in the workspace
* Inspect It – run the selected text and open the inspector on the result
* Debug It – run the selected text with a halt to open the debugger before the code starts
* Load… prompts for a text file to load into the workspace
* Save… saves the contents of the workspace into a text file

A screenshot of a computer

AI-generated content may be incorrect.

## System Browser

The System Browser is a window that can show you Smalltalk code stored in the system and allows you to change and add code. The top of the browser has four panes.

A screenshot of a computer

AI-generated content may be incorrect.

* Kits
  + The first pane shows kits. Kits are groupings of code that contain classes and extension methods of other classes. A kit can require other kits which must be loaded before this one can be loaded.
* Classes
  + The second pane shows classes. Classes represent object types such as Integer or String.
* Protocols
  + The third pane shows method protocols. A protocol is just a convenient named grouping of methods to allow you to quickly find methods.
* Methods
  + Methods are the actual Smalltalk code associated with classes.

The text pane at the bottom of the browser shows the Smalltalk code of the selected method.

## Senders…

The Senders button prompts for the name of a method and opens a browser on all methods that send a message with that name.

## Implementers…

The Implementers button prompts for the name of a method and opens a browser on all methods with that name.

## Class References…

The Class References button prompts for the name of a class and opens a browser on all methods that reference that class.

## Fileout Sources

Fileout Sources writes code for all kits into files in the kit folder.

## Filein Sources

Load the kits from the kits folder.

## Garbage Collect

Perform a global garbage collection.

## Save Image

Overwrite the beagle.im file with a snapshot of the current Smalltalk environment. This operation will first close all windows.

## Close

Close the connection to the image but keep the image running.

## Shut Down

Shutdown the image.

## Do It

Do It is a tool that helps if the regular user interface isn’t working for some reason. It allows you to type Smalltalk code in the upper text box and execute it.

## Print It

Print It allows you to type Smalltalk code in the upper text box and execute it showing the result in the lower text box.

## Show String

Show String is the same as Print It except that it prints the plain string instead of a quoted string if the result is a string.

## File In

File In allows you to enter text in the upper text box which it will compile and load into the Smalltalk image. This is handy if your Smalltalk crashes and you want to recover your changes from the beagle.cha file. Scroll to the bottom then search backwards for “saveImage”. Everything afterward is code you changed since the last save.

For methods, copy from the line that says “methodsFor:” to the end of the method including two ! symbols. For example,

! TestcaseWindow methodsFor: 'updating' !

allTestcases

| collection |

collection := OrderedCollection new.

self suites do: [:eachSuite | collection addAll: eachSuite allTestcases].

^collection

! !

Paste this into the top text area and press File In to load that method into Smalltalk.

For classes, take the class definition and add a ! at the end

Testcase subclassNamed: #ObjectTests

instVarNames: #()

classVarNames: #()

classInstVarNames: #()

environment: Object systemDictionary

kitName: 'Tests' !

Again, paste this into the upper text area and press File In.