



The Squeak Environment


Stéphane Ducasse

Stephane.Ducasse@inria.fr

<http://stephane.ducasse.free.fr>

License: CC-Attribution-ShareAlike 2.0

<http://creativecommons.org/licenses/by-sa/2.0/>



The image shows a summary of the Creative Commons Attribution-ShareAlike 2.0 license. It features the Creative Commons logo at the top, followed by the text 'Creative Commons Commons Deed Attribution-ShareAlike 2.0'. Below this, it lists the freedoms granted to users and the conditions for reuse. The conditions are represented by icons: a person in a circle for 'BY: Attribution' and a circular arrow for 'Share Alike'. The text explains that users must give credit to the original author and that any new works must be licensed under the same terms. A note at the bottom states that fair use and other rights are not affected, and it provides a link to the full legal code.

CC creative commons
COMMONS DEED

Attribution-ShareAlike 2.0

You are free:

- to copy, distribute, display, and perform the work
- to make derivative works
- to make commercial use of the work

Under the following conditions:

BY: **Attribution.** You must give the original author credit.

Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the [Legal Code \(the full license\)](#).

Smalltalk Run-Time Architecture

- Virtual Machine + Image + Changes and Sources

All the objects of the system
at a moment in time

IMAGE1.IM
IMAGE1.CHA

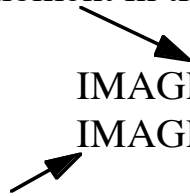
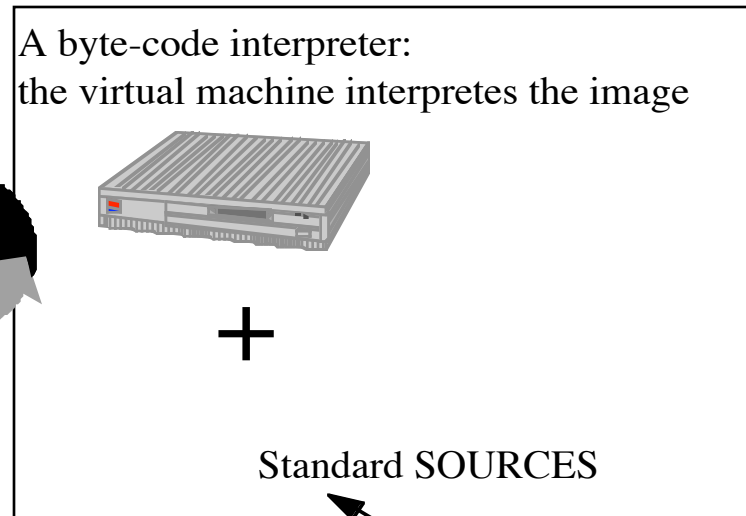
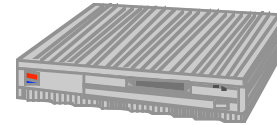


IMAGE2.IM
IMAGE2.CHA

One per user



A byte-code interpreter:
the virtual machine interpretes the image



+

Standard SOURCES

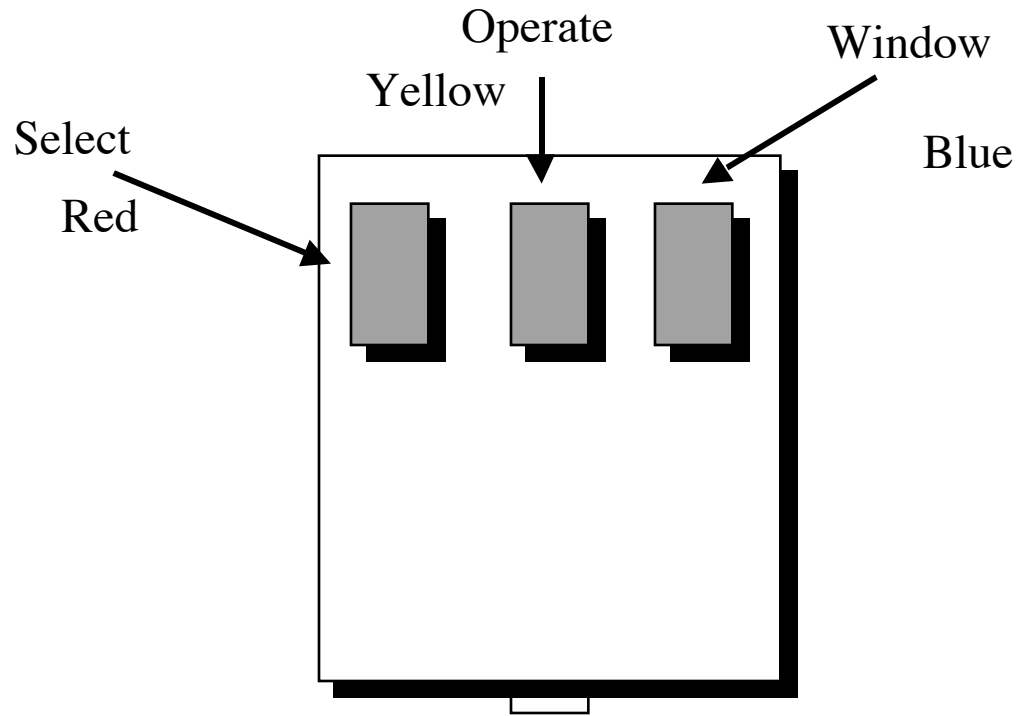
Shared by everybody

- Image = bytecodes
- Sources and changes = code (text)

Runtime Architecture

- The byte-code is in fact translated into native code by a just-in-time compiler.
- The source and the changes are not necessary for interpreting the byte-code, this is just for the development. Normally they are removed for deployment.
- An application can be delivered as some byte-code files that will be executed with a VM. The development image is stripped to remove the unnecessary development components.

Mouse Semantics



World Menu and Open Menu

World

- previous project
- jump to project...
- save project on file...
- load project from file...
- can't undo
- restore display (r)
- open...
- windows...
- changes...
- help...
- appearance...
- do...
- objects (o)
- new morph...
- authoring tools...
- playfield options...
- flaps...
- projects...
- print PS to file...
- debug...
- save
- save as...
- save as new version
- save and quit
- quit

open...

- browser (b)
- package browser
- workspace (k)
- file list
- file...
- transcript (t)
- method finder
- message names (W)
- simple change sorter
- dual change sorter
- email reader
- web browser
- IRC chat
- mvc project
- morphic project

Browsing a class

The screenshot shows the 'System Browser: Monster' window. The left pane lists various packages, with 'Sesame Street' selected. The middle pane shows the class hierarchy, with 'Monster' selected. The right pane shows the class's methods: 'askForCookie', 'complaintAbout:', 'eat:', 'initializeToStandAlone', and 'isEmpty'. Below the panes are buttons for 'browse', 'senders', 'implementors', 'versions', 'inheritance', 'hierarchy', 'inst vars', 'class vars', and 'col'. The bottom pane displays the following code:

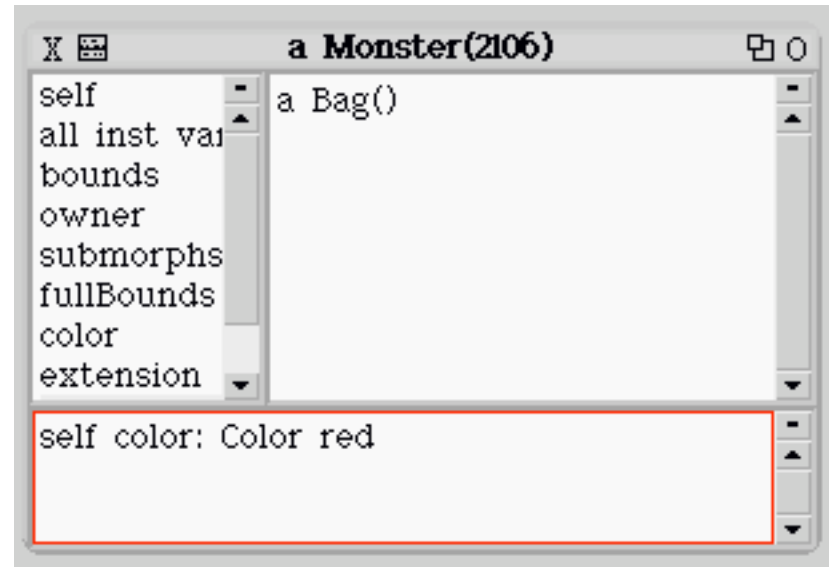
```
Morph subclass: *Monster
  instanceVariableNames: 'tummy '
  classVariableNames: ''
  poolDictionaries: ''
  category: 'Sesame Street'
```

Browsing methods

The screenshot shows the 'System Browser: Monster' window. The left pane lists a hierarchy of packages, with 'Sesame Street' selected. The middle pane shows the class 'Monster' (highlighted in red) under the 'CookieMonster' package. Below the class name are buttons for 'instance', '?', and 'class'. The right pane lists methods: 'askForCookie', 'complaintAbout:', 'eat:', 'initializeToStandAlone', and 'isEmpty'. The 'eat:' method is selected. Below the browser is a toolbar with buttons: 'browse', 'senders', 'implementors', 'versions', 'inheritance', 'hierarchy', 'inst vars', 'class vars', and 'col'. The main area displays the code for the 'eat:' method: `eat: someltem` followed by `tummy add: someltem`.

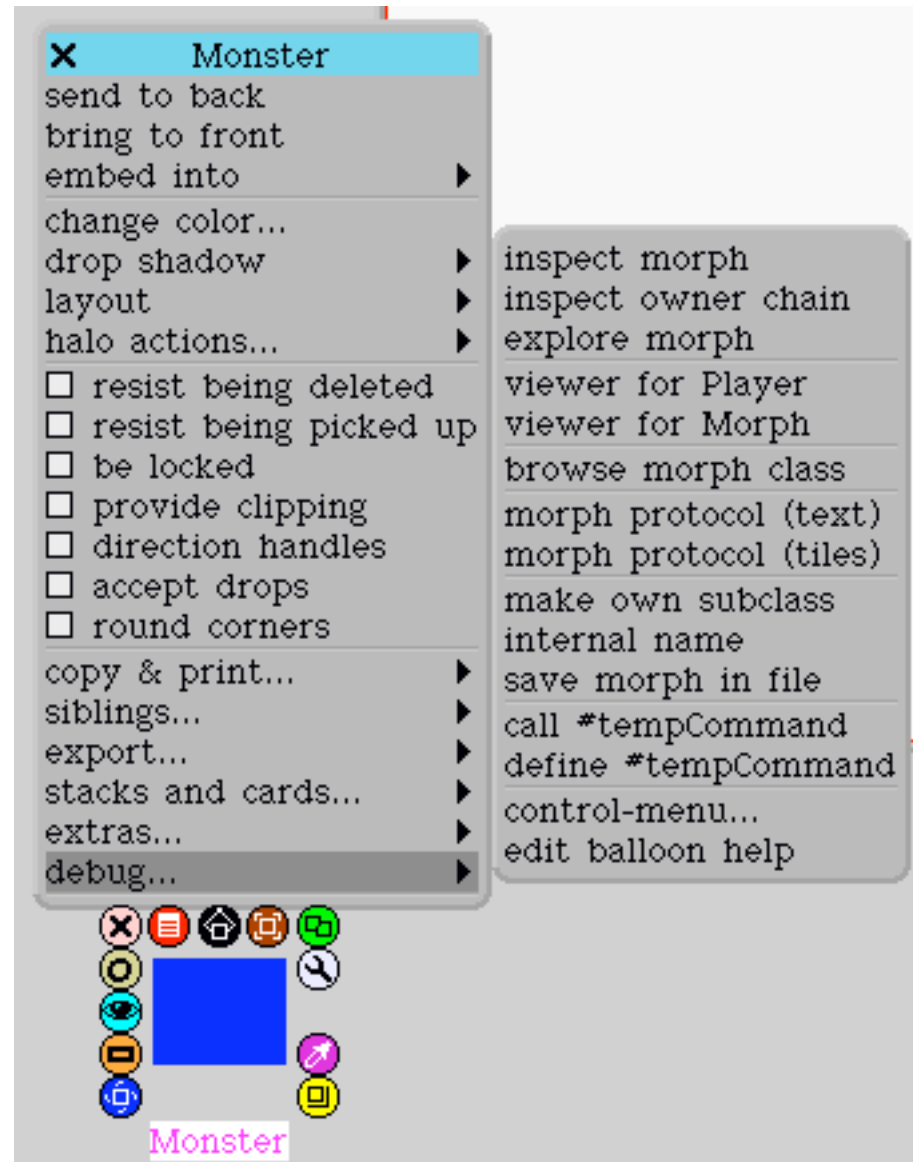
Inspector

- To look inside objects
- Violates encapsulation!!!
- Monster new inspect

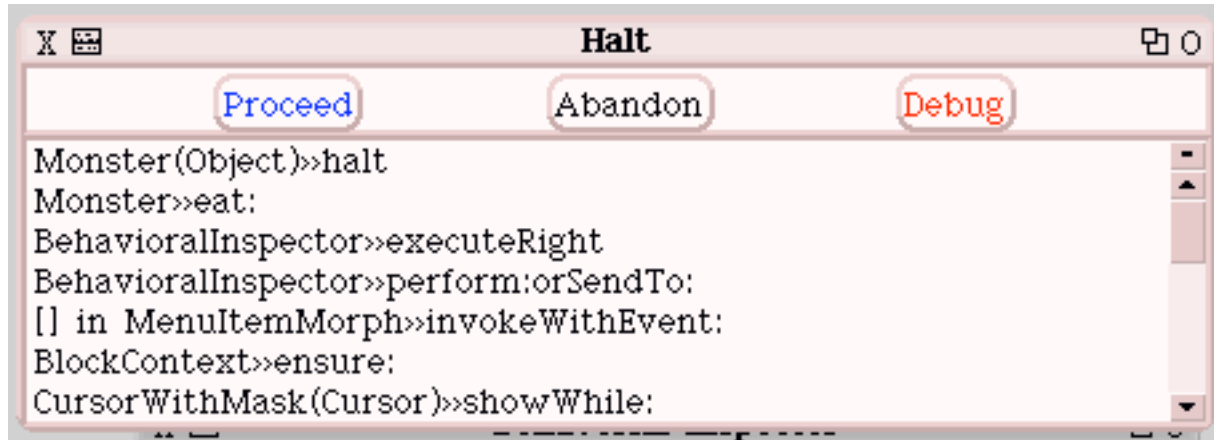


Direct Manipulation

- Bring the halo
- Experiment



Debugger?



Debugger !!!

The screenshot shows a debugger window titled "Halt". The main area displays a stack trace with the following entries from top to bottom:

- Monster(Object)>>halt
- Monster>>eat:
- BehavioralInspector>>executeRight
- BehavioralInspector>>perform:orSendTo:
- [] in MenuItemMorph>>invokeWithEvent:
- BlockContext>>ensure:
- CursorWithMask(Cursor)>>showWhile:

Below the stack trace is a control bar with buttons: Proceed, Restart, Send, Step, Through, Full Stack, Where, and Browse.

The "eat: C" frame is expanded, showing the following code:

```
tummy add: C
```

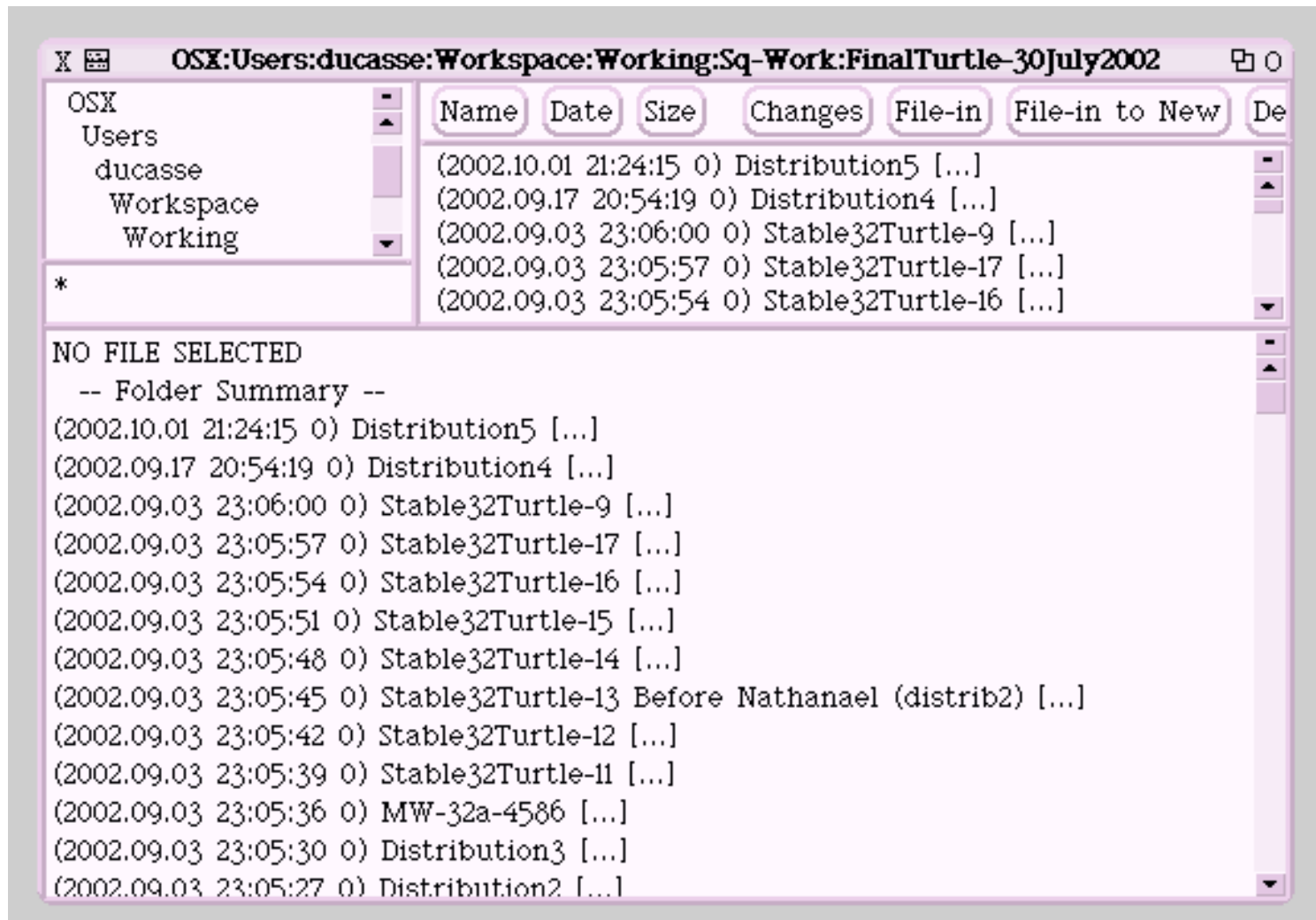
At the bottom, there are two variable inspectors. The left one shows the state of the object:

- bounds
- owner
- submorphs
- fullBounds
- color
- extension
- tummy

The right inspector shows the state of the context:

- thisContext
- all temp vars
- C

FileList



The screenshot shows a window titled "OSX:Users:ducasse:Workspace:Working:Sql-Work:FinalTurtle-30July2002". The window contains a sidebar on the left with a tree view showing the path: OSX > Users > ducasse > Workspace > Working. The main area displays a table of files with columns for Name, Date, Size, Changes, File-in, File-in to New, and De. The table lists several files, including Distribution5, Distribution4, Stable32Turtle-9, Stable32Turtle-17, and Stable32Turtle-16. Below the table, there is a section titled "NO FILE SELECTED" followed by a "Folder Summary" listing the same files as the table.

Name	Date	Size	Changes	File-in	File-in to New	De
(2002.10.01 21:24:15 0) Distribution5 [...]	(2002.10.01 21:24:15 0)					
(2002.09.17 20:54:19 0) Distribution4 [...]	(2002.09.17 20:54:19 0)					
(2002.09.03 23:06:00 0) Stable32Turtle-9 [...]	(2002.09.03 23:06:00 0)					
(2002.09.03 23:05:57 0) Stable32Turtle-17 [...]	(2002.09.03 23:05:57 0)					
(2002.09.03 23:05:54 0) Stable32Turtle-16 [...]	(2002.09.03 23:05:54 0)					

NO FILE SELECTED

-- Folder Summary --

- (2002.10.01 21:24:15 0) Distribution5 [...]
- (2002.09.17 20:54:19 0) Distribution4 [...]
- (2002.09.03 23:06:00 0) Stable32Turtle-9 [...]
- (2002.09.03 23:05:57 0) Stable32Turtle-17 [...]
- (2002.09.03 23:05:54 0) Stable32Turtle-16 [...]
- (2002.09.03 23:05:51 0) Stable32Turtle-15 [...]
- (2002.09.03 23:05:48 0) Stable32Turtle-14 [...]
- (2002.09.03 23:05:45 0) Stable32Turtle-13 Before Nathanael (distrib2) [...]
- (2002.09.03 23:05:42 0) Stable32Turtle-12 [...]
- (2002.09.03 23:05:39 0) Stable32Turtle-11 [...]
- (2002.09.03 23:05:36 0) MW-32a-4586 [...]
- (2002.09.03 23:05:30 0) Distribution3 [...]
- (2002.09.03 23:05:27 0) Distribution2 [...]

Message Names Finder

The screenshot shows a software interface titled "Message names containing 'match:'". It features a search input field with "match:" entered. Below the search field is a list of message names, with "match:" highlighted. To the right of the list is a preview pane showing the details for the selected "match:" message, including "Parser match:", "PositionableStream match:", and "String match:". At the bottom of the interface are several tabs: "browse", "senders", "implementors", "versions", "inheritance", "hierarchy", "inst vars", "class vars", and "col". Below the tabs is a text area displaying several test cases for the "match:" message, such as "'foo*baz' match: 'foo23bazo' false" and "'foo' match: 'Foo' true".

```
Message names containing "match:"
```

Search Parser match:
PositionableStream match:
String match:

- encodeMatch:distance:
- findMatch:lastLength:lastMatch:chainLength
- howManyMatch:
- match:**
- match:fields:
- match:inContext:
- reorderParametersToMatch:
- scaleToMatch:
- startingAt:match:startingAt:
- sunitMatch:

browse senders implementors versions inheritance hierarchy inst vars class vars col

```
'foo*baz' match: 'foo23bazo' false  
'foo' match: 'Foo' true  
'foo*baz*zort' match: 'foobazort' false  
'foo*baz*zort' match: 'foobazzort' false  
*foo*zort' match: 'afoo3zortthenfoo3zort' true  
*foo*zort' match: 'afoodezortorfoo3zort' true  
"
```

The image shows a screenshot of the IntelliJ IDEA IDE. Two windows are open:

- Selector Browser:** This window is in the foreground. The top pane contains a list of selectors, with `'*b' match: 'ab' --> true` selected. The bottom pane contains instructions: "Type a fragment of a selector in the top pane. Accept it." and "Or, use an example to find a method in the system. Type receiver, args, and answer in the". Below the instructions are dropdown menus for "instance", "?", "class", "system primitives", and "Celeste".
- Collections-Text:** This window is in the background. It displays the documentation for the `match: text` method. The text includes:
 - "Answer whether text matches the pattern in this string.
 - Matching ignores upper/lower case differences.
 - Where this string contains *, text may contain any character.
 - Where this string contains *, text may contain any sequence of characters."
 Below the text is a code snippet:


```

↑ self
  startingAt: 1
  match: text
  startingAt: 1"
      
```

Methods in ChangeSets + Versions

The image shows two overlapping windows from an IDE. The top window, titled "Methods in Change Set Sesame", lists several methods for the class "Monster":

- askForCookie {asking}
- complaintAbout: {asking}
- eat: {behavior}** (highlighted in red)
- initializeToStandAlone {initialization}
- isEmpty {behavior}

Below the list are buttons for "browse", "senders", "implementors", "versions", "inheritance", "hierarchy", "inst vars", "class vars", and "col". The "eat: someItem" method is selected, and its implementation is visible in the main editor area:

```
tummy add: someItem
```

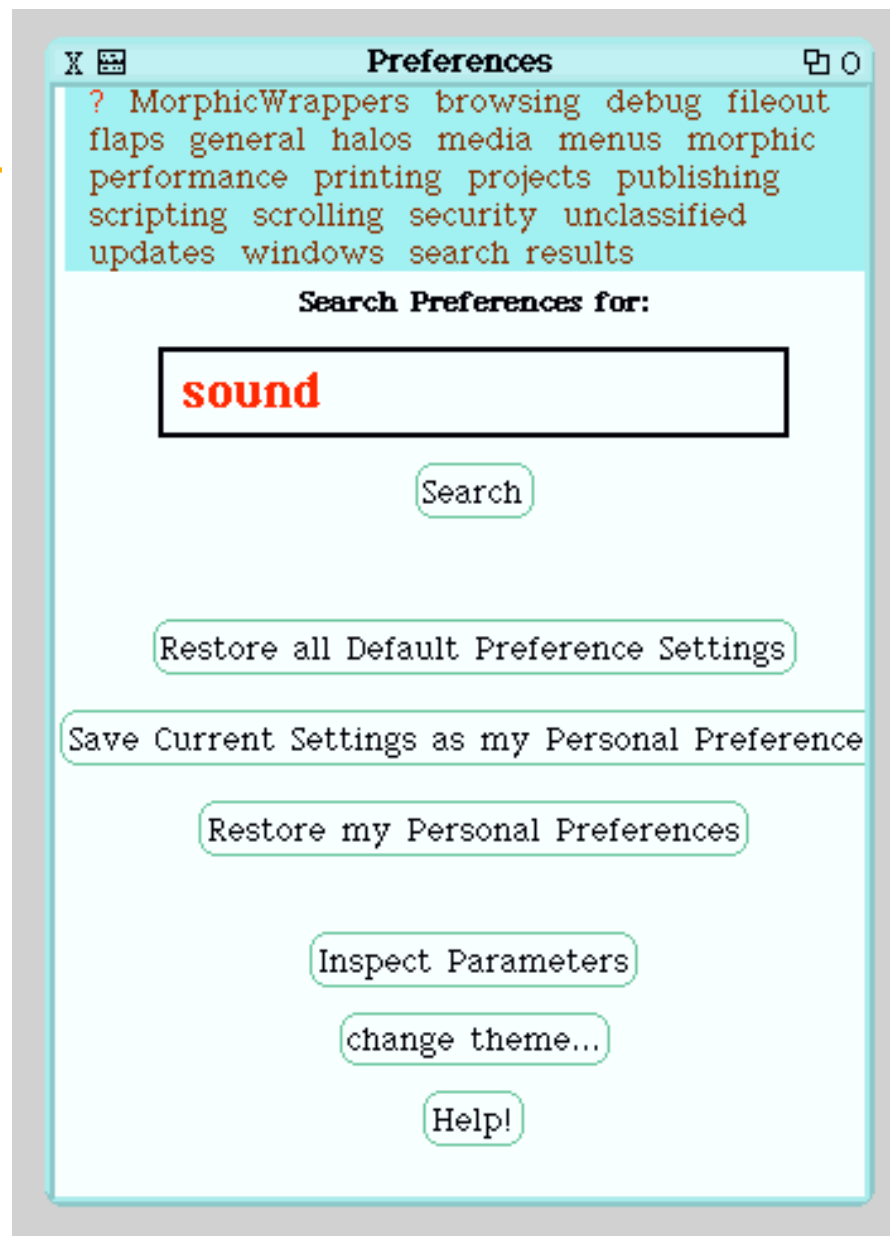
The bottom window, titled "Recent versions of eat:", shows a list of recent changes to the "eat:" method:

- sd 10/11/2002 21:55 Monster eat: (highlighted in red)
- sd 10/11/2002 21:31 Monster eat:

Buttons for "compare to current", "revert", "remove from changes", "help", "diffs", and "P" are visible below the list. The selected version's implementation is shown in a separate pane:

```
eat: someItem  
  
eat: someItem  
.  
tummy add: someItem
```


Preferences



SUnit

Workspace

TestRunner open

SUnit Camp Smalltalk 3.0 Test Runner

REFRESH

TurtleTest

TemporaryToInstanceVariableTest

AddInstanceVariableTest

Array (simple)

RUN

RUN ALL

Fail

6 run, 0 passed, 0 failed, 6 errors

TemporaryToInstanceVariableTest>testConditions

TemporaryToInstanceVariableTest>testNonExistantName

TemporaryToInstanceVariableTest>testPrintRefactoringManager

TemporaryToInstanceVariableTest>testRedefinedTemporary

TemporaryToInstanceVariableTest>testTemporaryToInstanceVariable

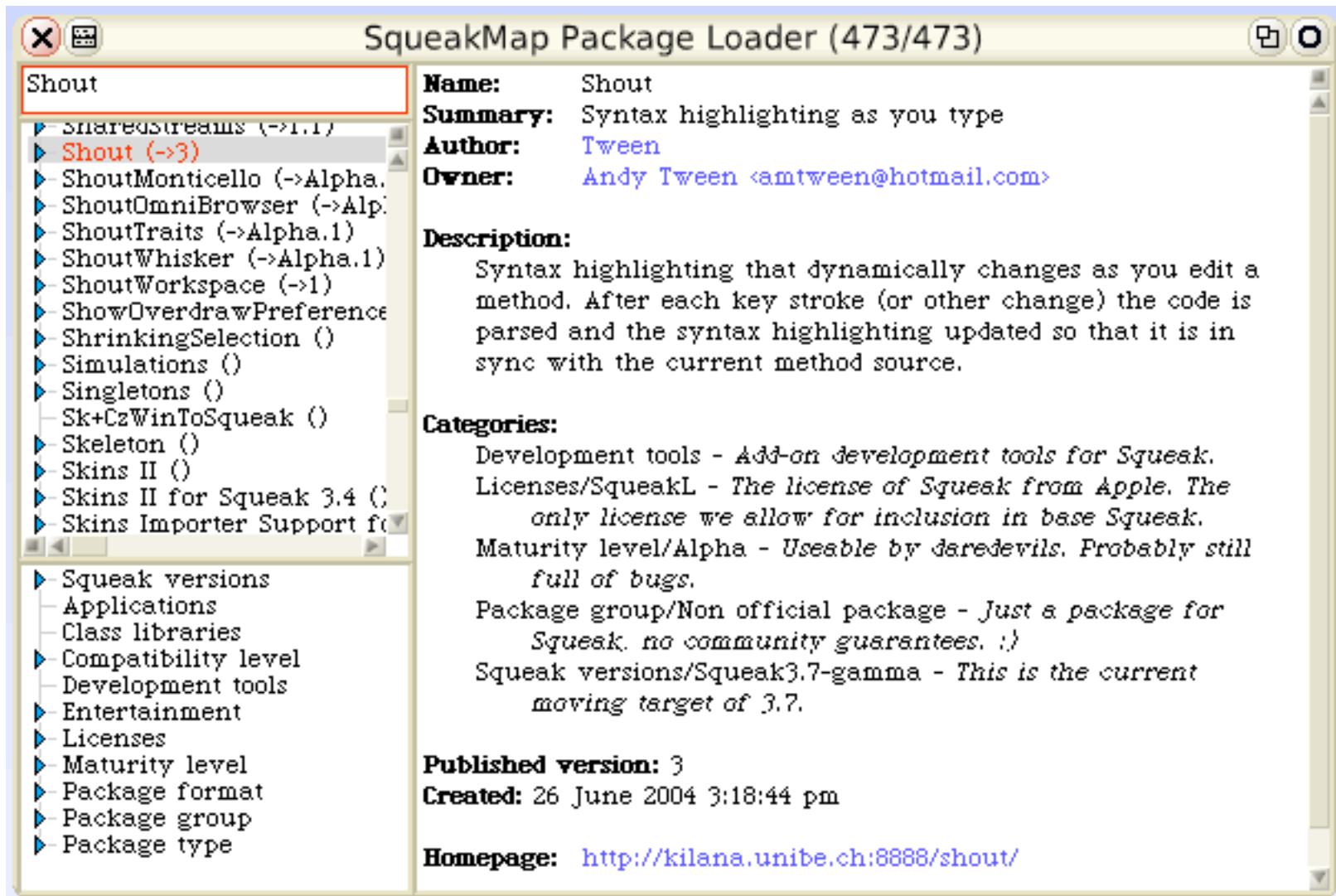
TemporaryToInstanceVariableTest>testing

Transcript is Slooooooow

```
Transcript
----SNAPSHOT----*(12 October 2002 4:00:23 pm) Squeak3.2.image
40238726007709377354370243392300398571937486421071463254379991042993851239862902059204
42084869694048004799886101971960586316668729948085589013238296699445909974245040870737
599188236277271887325197795059509952761208749754624970436014182780946464962910563938874
378864873371191810458257836478499770124766328898359557354325131853239584630755574091142
624174743493475534286465766116677973966688202912073791438537195882498081268678383745597
31746136085379534524221586593201928090878297308431392844403281231558611036976801357304216
16874760967587134831202547858932076716913244842623613141250878020800026168315102734182797
77047846358681701643650241536913982812648102130927612448963599287051149649754199093422215
6683257208082133318611681155361583654698404670897560290095053761647584772842188967964624
101516076505010010000100511010700105005000101010000555660010015000070600075013783761530
```

```
Workspace
Transcript show: (999 factorial) printString
999 factorial / 1000 factorial
-> (1/1000)
194545257223
501537808898
873617459926
516284225540
652682272807
794659551311
840244324384
4977401141334
9949022222176
5901641921068
9353143970137
764456000007
```

Squeak Map Package Loader



The screenshot shows a window titled "SqueakMap Package Loader (473/473)". On the left is a tree view of packages, with "Shout" selected. On the right, the details for the "Shout" package are displayed.

Name: Shout

Summary: Syntax highlighting as you type

Author: Tween

Owner: Andy Tween <amtween@hotmail.com>

Description:
Syntax highlighting that dynamically changes as you edit a method. After each key stroke (or other change) the code is parsed and the syntax highlighting updated so that it is in sync with the current method source.

Categories:
Development tools - *Add-on development tools for Squeak.*
Licenses/SqueakL - *The license of Squeak from Apple. The only license we allow for inclusion in base Squeak.*
Maturity level/Alpha - *Useable by daredevils. Probably still full of bugs.*
Package group/Non official package - *Just a package for Squeak, no community guarantees. :)*
Squeak versions/Squeak3.7-gamma - *This is the current moving target of 3.7.*

Published version: 3

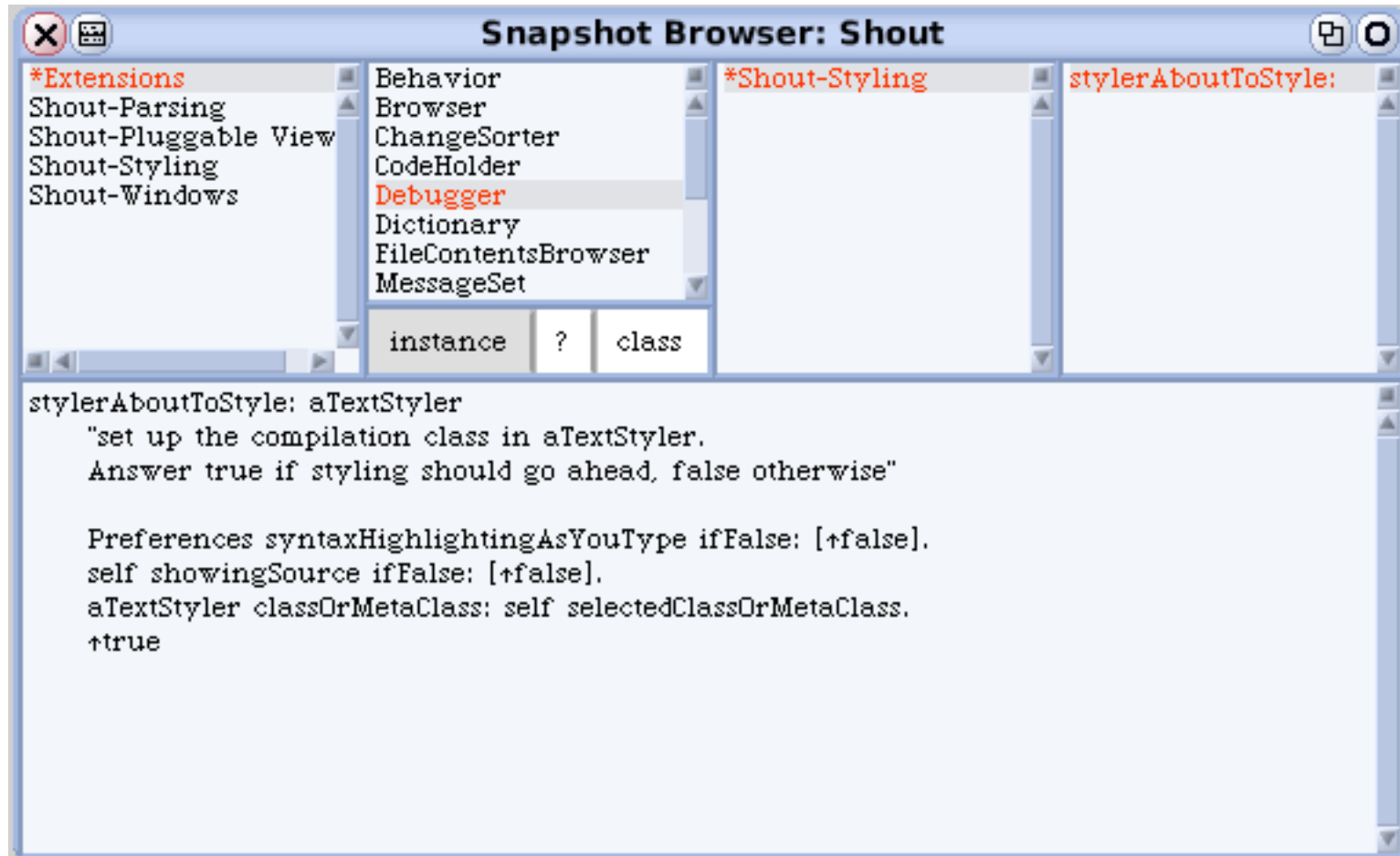
Created: 26 June 2004 3:18:44 pm

Homepage: <http://kilana.unibe.ch:8888/shout/>

Monticello



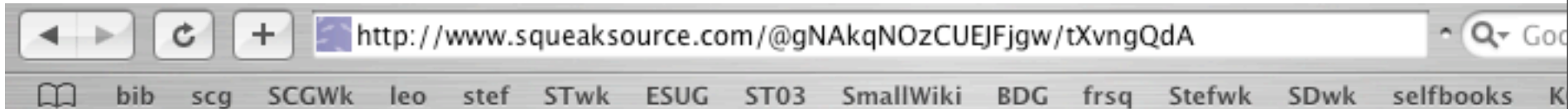
Monticello Change Browser



Repository



SqueakSource



SqueakSource

[Home](#) [Projects](#) [Members](#) [Help](#)

Actions

[Syndication](#)

[Register Member](#)

[Register Project](#)

Authentication

[Login](#)

Home

Welcome to SqueakSource, the smart Monticello code-repository for Squeak. To get started, you need to create a personal account and create any number of projects to host on our server. You'll immediately receive the necessary permissions to create and manage your account, projects and versions. Detailed instructions are found on the [Help](#) page.

This service is brought to you for free: it is hosted on a Solaris server that is backed up daily to ensure that you have proper backups and note that we cannot give any accessibility guarantee. If you wish you can set up your own SqueakSource server. It is self-hosted and can be downloaded from [our website](#).

Please report any problems or suggestions to the [Squeak Mailing-List](#). Enjoy!

Statistics

Members (123)

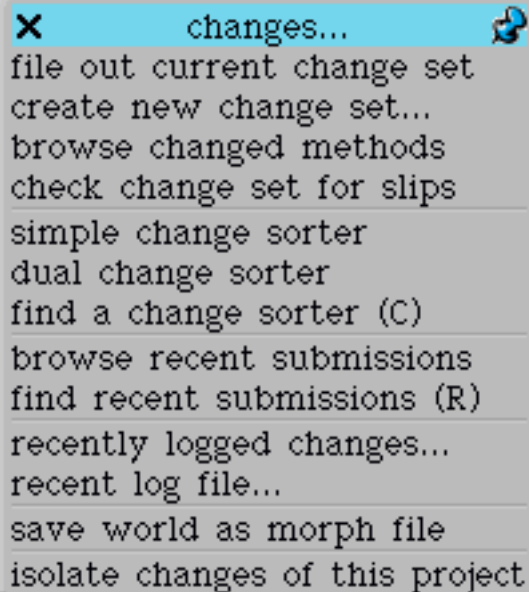
Recently Joined: [Jeff Sparkes](#), [Bernd Eckardt](#), [Luc Fabresse](#), ...


Most Active: [Lex Spoon](#), [Hernán Tylim](#), [Samuel Tardieu](#), ...

Projects (111)

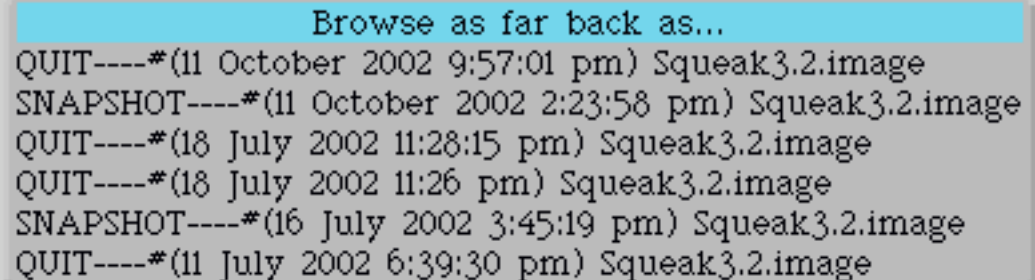
Changes Menu...Recently logged files

- Everything you do is recorded



changes... 

- file out current change set
- create new change set...
- browse changed methods
- check change set for slips
- simple change sorter
- dual change sorter
- find a change sorter (C)
- browse recent submissions
- find recent submissions (R)
- recently logged changes...
- recent log file...
- save world as morph file
- isolate changes of this project



Browse as far back as...

- QUIT----*(11 October 2002 9:57:01 pm) Squeak3.2.image
- SNAPSHOT----*(11 October 2002 2:23:58 pm) Squeak3.2.image
- QUIT----*(18 July 2002 11:28:15 pm) Squeak3.2.image
- QUIT----*(18 July 2002 11:26 pm) Squeak3.2.image
- SNAPSHOT----*(16 July 2002 3:45:19 pm) Squeak3.2.image
- QUIT----*(11 July 2002 6:39:30 pm) Squeak3.2.image

ChangeSorter: to sort your changes

Change Set

make changes go to me (m)

new change set... (n)

find...(f)

show category... (s)

select change set...

rename change set (r)

file out (o)

mail to list

browse methods (b)

browse change set (B)

copy all to other side (c)

submerge into other side

subtract other side

add preamble (p)

add postscript...

category functions...

destroy change set (x)

more...

Changes go to "Sesame"

MW-base	CookieMonster	AniTurtle	AniTurtle
4917dupNavBar-sw	Monster	Monitor	AniTurtle class
Sesame		ExtendedTurtleBeha	
More About Sound		TurtleforMW	
Squeak in 3D		TurtleBrowsers	
Squeak and the Inte		TurtleEnvironment	
askForCookie			
complaintAbout:			
eat:			
initializeToStandAlone			
isEmpty			
askForCookie			
↑ FillInTheBlank request: 'Give me cookie!!! (please)'			

Cool Packages to Load

Shout

Refactoring Browser

eCompletion

Change your Mind

Everything you do is recorded !!
So try and learn how to recover your code

You are smart so

Experiment,

Learn for you, browse,

educate your taste (not all Squeak code is good)

Be aggressive, *****all***** the code is there