F-Script: Smalltalk Scripting for Mac OS X

Philippe Mougin

pmougin@acm.org

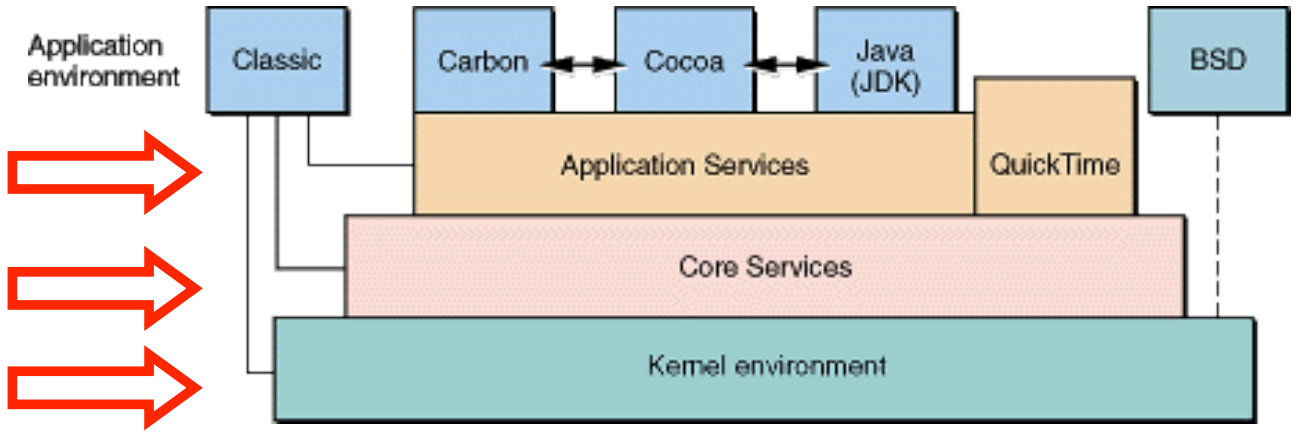
ESUG 2002 - DOUAI

F-Script in the Mac OS X Architecture

Quartz, OpenGL, multimedia, printing, event handling, low-level window management, etc.

Strings, run loops, collections, processes, resources, etc.

Multitasking, virtual memory, SMP, file systems, device drivers, networking, threading packages, etc.

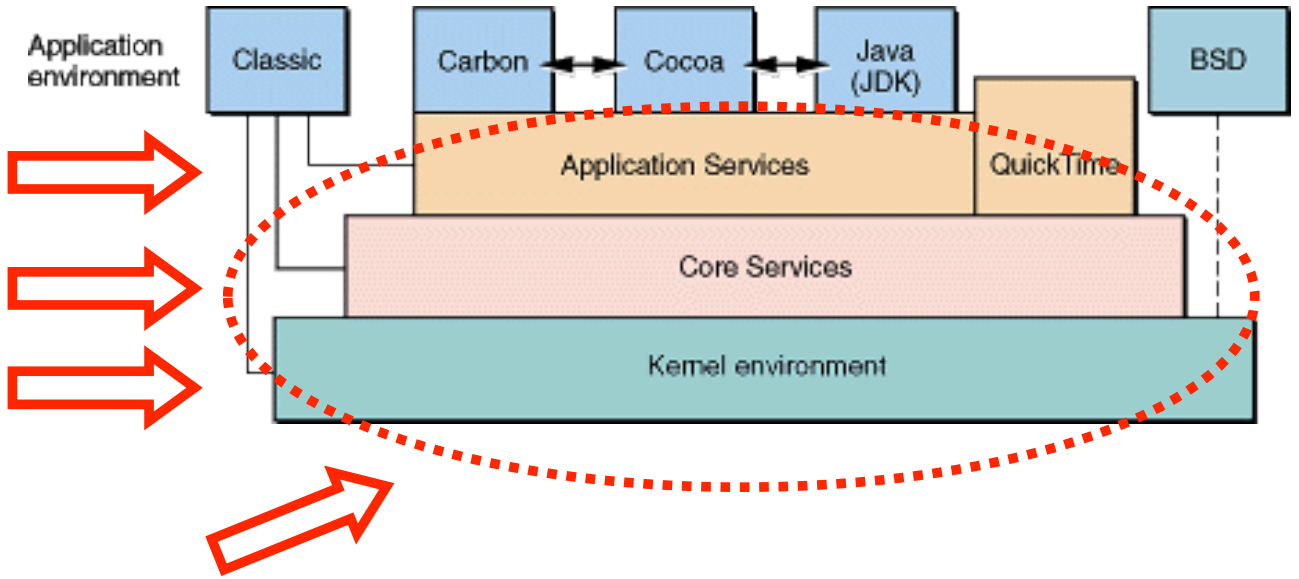


F-Script in the Mac OS X Architecture

Quartz, OpenGL, multimedia, printing, event handling, low-level window management, etc.

Strings, run loops, collections, processes, resources, etc.

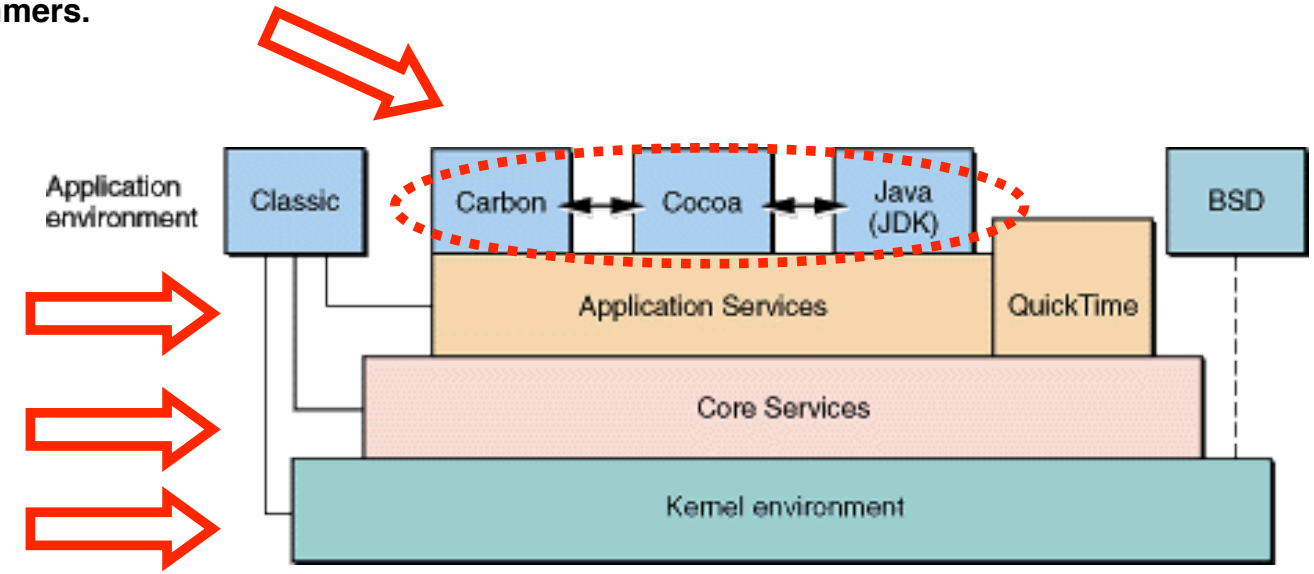
Multitasking, virtual memory, SMP, file systems, device drivers, networking, threading packages, etc.



Mac OS X is a UNIX operating system which is implemented (mostly) in C and which offers C-based APIs.

F-Script in the Mac OS X Architecture

Apple provides three high-level environments (tools, APIs) for programmers.



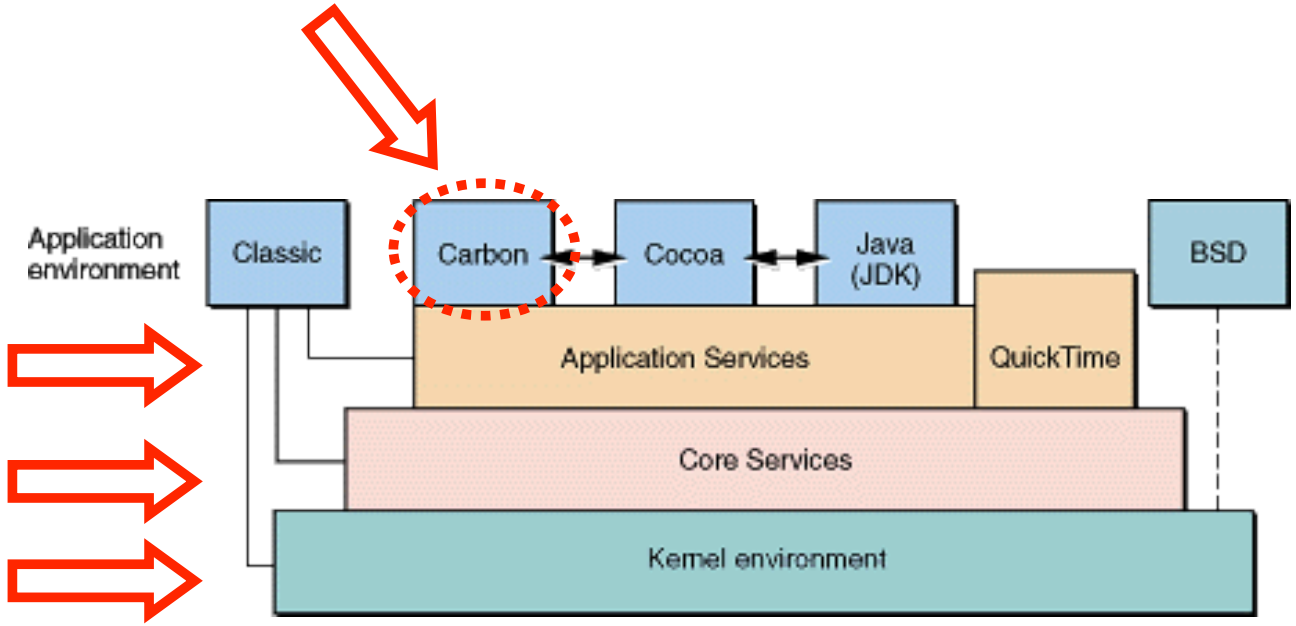
Quartz, OpenGL, multimedia, printing, event handling, low-level window management, etc.

Strings, run loops, collections, processes, resources, etc.

Multitasking, virtual memory, SMP, file systems, device drivers, networking, threading packages, etc.

F-Script in the Mac OS X Architecture

The main goal of Carbon is to provide toolbox-like APIs (i.e. old Mac OS APIs) for porting existing Mac OS applications, or creating new ones.



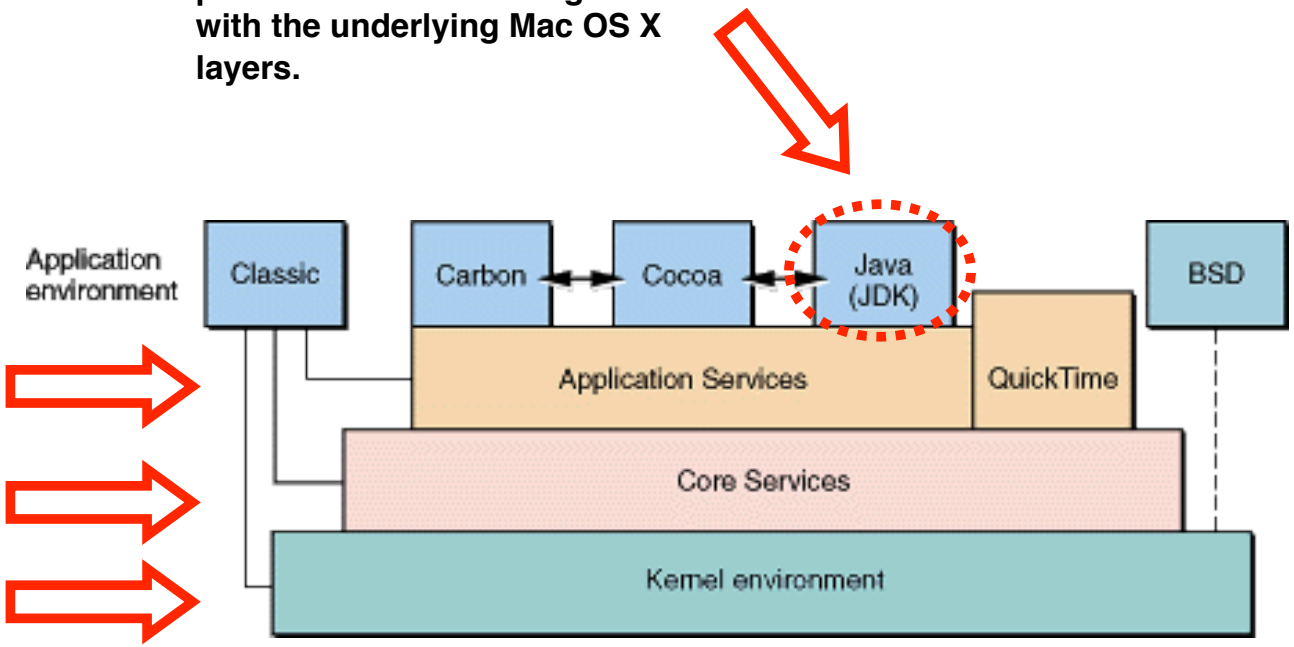
Quartz, OpenGL, multimedia, printing, event handling, low-level window management, etc.

Strings, run loops, collections, processes, resources, etc.

Multitasking, virtual memory, SMP, file systems, device drivers, networking, threading packages, etc.

F-Script in the Mac OS X Architecture

The Java environment supports both standards, cross-platform, Java frameworks and specific Mac OS X Java APIs. In both cases, the Java environment provides advanced integration with the underlying Mac OS X layers.



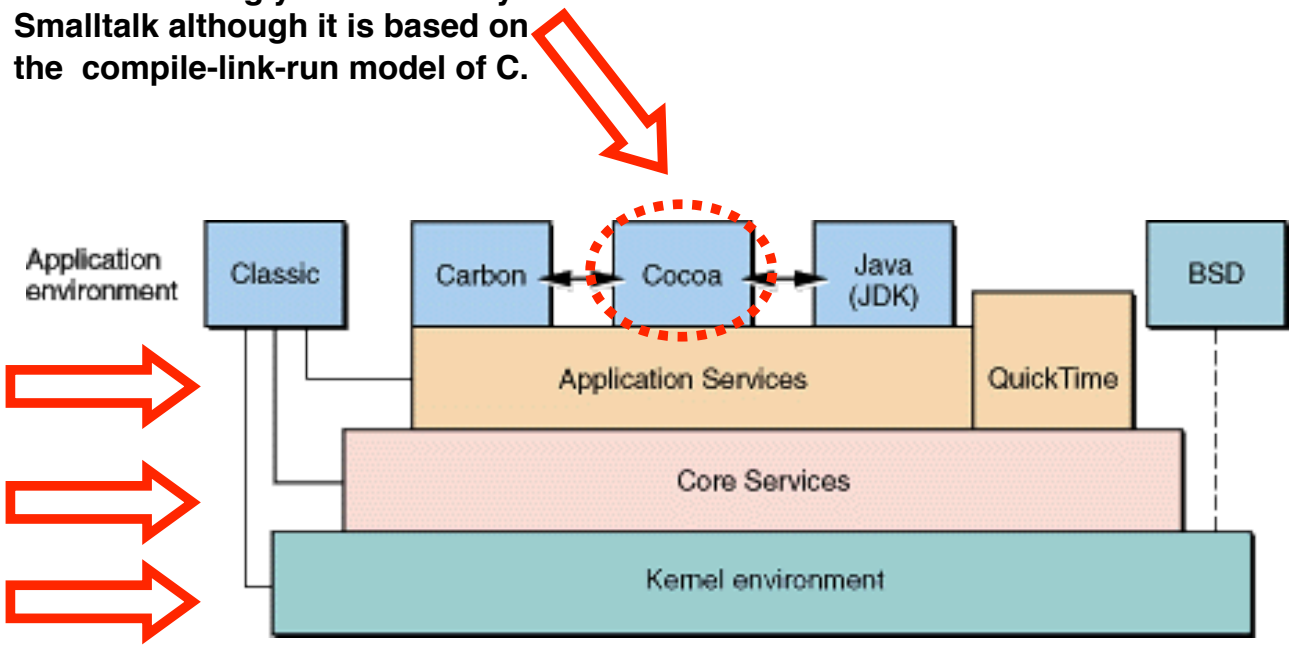
Quartz, OpenGL, multimedia, printing, event handling, low-level window management, etc.

Strings, run loops, collections, processes, resources, etc.

Multitasking, virtual memory, SMP, file systems, device drivers, networking, threading packages, etc.

F-Script in the Mac OS X Architecture

Cocoa provides a lightweight object-oriented extension of the C language, called Objective-C. It comes with various frameworks and tools.
Cocoa is strongly influenced by Smalltalk although it is based on the compile-link-run model of C.



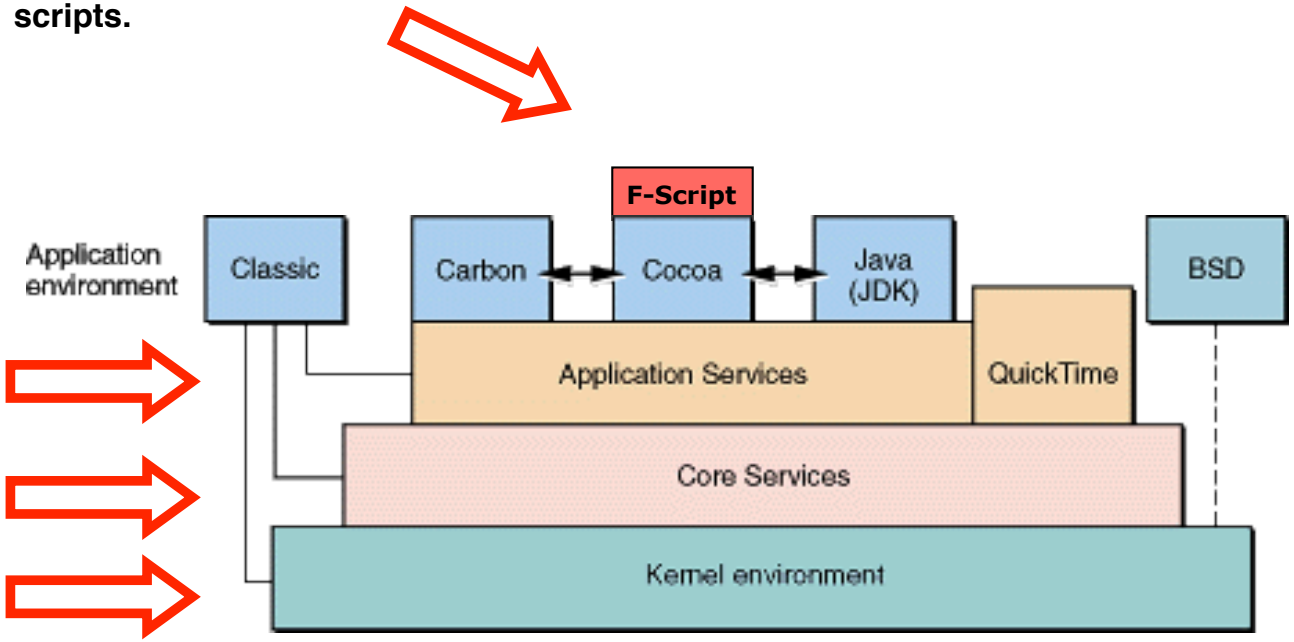
Quartz, OpenGL, multimedia, printing, event handling, low-level window management, etc.

Strings, run loops, collections, processes, resources, etc.

Multitasking, virtual memory, SMP, file systems, device drivers, networking, threading packages, etc.

F-Script in the Mac OS X Architecture

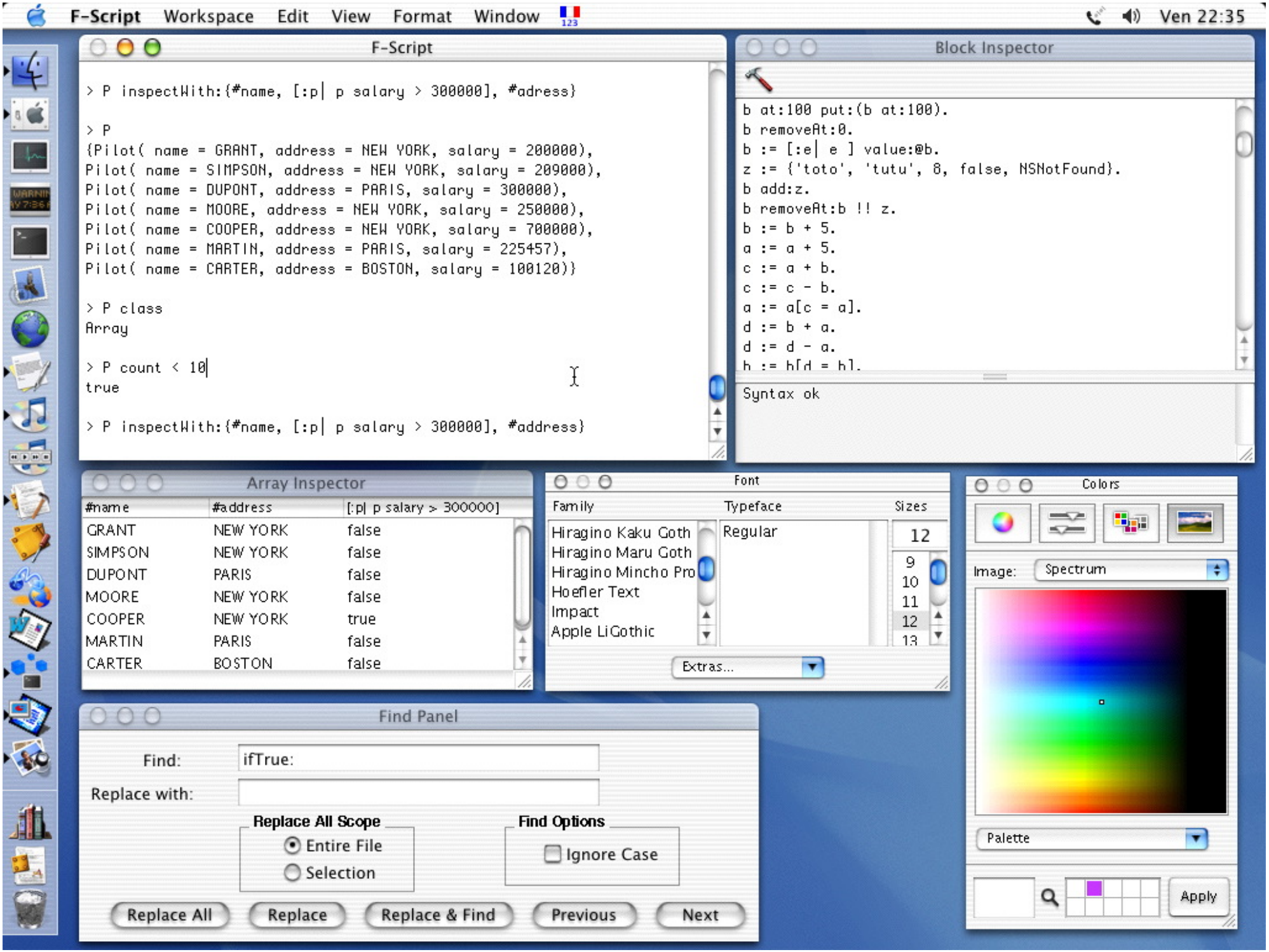
F-Script adds an interactive object-oriented environment to Cocoa. It lets you interactively manipulate Cocoa-based objects. It also lets you write scripts.



Quartz, OpenGL, multimedia, printing, event handling, low-level window management, etc.

Strings, run loops, collections, processes, resources, etc.

Multitasking, virtual memory, SMP, file systems, device drivers, networking, threading packages, etc.


F-Script Workspace Edit View Format Window
Ven 22:35
F-Script
> P inspectWith:{#name, [:p| p salary > 300000], #adress}
> P
{Pilot(name = GRANT, address = NEW YORK, salary = 200000),
Pilot(name = SIMPSON, address = NEW YORK, salary = 209000),
Pilot(name = DUPONT, address = PARIS, salary = 300000),
Pilot(name = MOORE, address = NEW YORK, salary = 250000),
Pilot(name = COOPER, address = NEW YORK, salary = 700000),
Pilot(name = MARTIN, address = PARIS, salary = 225457),
Pilot(name = CARTER, address = BOSTON, salary = 100120)}
> P class
Array
> P count < 10
true
> P inspectWith:{#name, [:p| p salary > 300000], #address}
Block Inspector
b at:100 put:(b at:100).
b removeAt:0.
b := [:e| e] value:@b.
z := {'toto', 'tutu', 8, false, NSNotFound}.
b add:z.
b removeAt:b !! z.
b := b + 5.
a := a + 5.
c := a + b.
c := c - b.
a := a[c = a].
d := b + a.
d := d - a.
Syntax ok
Array Inspector
#name #address [:p| p salary > 300000]
GRANT NEW YORK false
SIMPSON NEW YORK false
DUPONT PARIS false
MOORE NEW YORK false
COOPER NEW YORK true
MARTIN PARIS false
CARTER BOSTON false
Font
Family Typeface Sizes
Hiragino Kaku Goth
Hiragino Maru Goth
Hiragino Mincho Pro
Hoefler Text
Impact
Apple LiGothic
Regular
12
9
10
11
12
13
Extras...
Colors
Image: Spectrum
Palette
Apply
Find Panel
Find: ifTrue:
Replace with:
Replace All Scope
Entire File
Selection
Find Options
Ignore Case
Replace All
Replace
Replace & Find
Previous
Next

Lightweight interactive & scripting layer

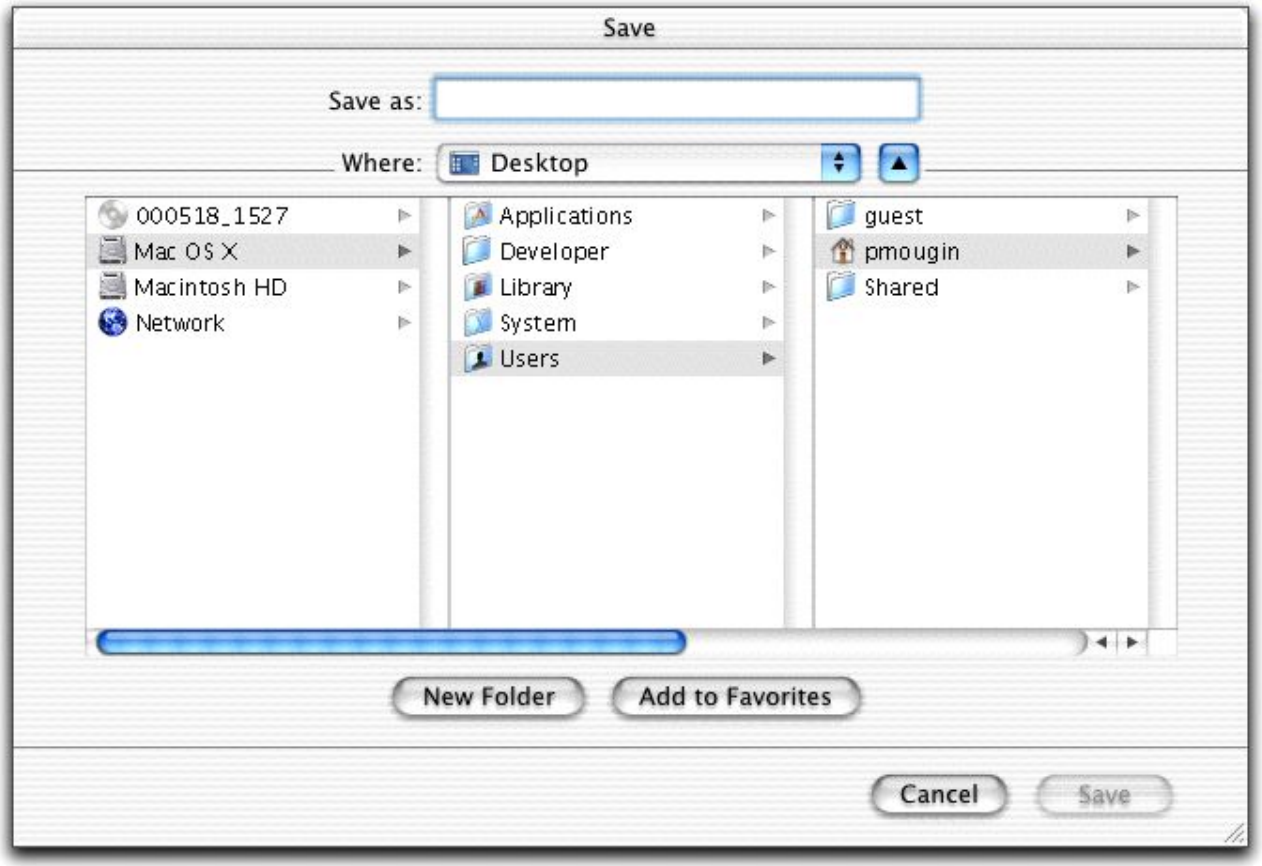
- **F-Script does not replace Mac OS X development tools, frameworks and object run-time: developers continue to use existing tools & the Objective-C language to develop new classes.**
- **F-Script adds a lightweight interactive and scripting layer to the Cocoa stack.**
- **Emphasis is on interactive high-level and user-friendly object manipulation rather than class development.**

```
> 7 arcCos

error: receiver of message "arcCos" must be a number between -1 and 1
```

Lightweight interactive & scripting layer

- **Emphasis is on interactive high-level and user-friendly object manipulation rather than class development.**

```
> myObject := [:a :b| a+b]
> myObject save
```



Lightweight interactive & scripting layer

- **Emphasis is on interactive high-level and user-friendly object manipulation rather than class development.**

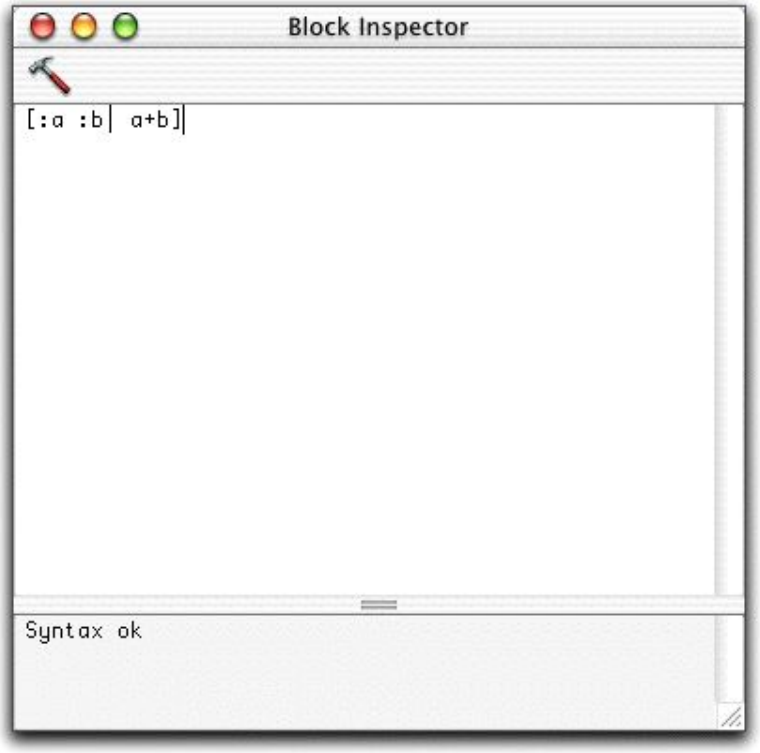
```
> myObject vend:'foo'
```

myObject **is now registered in the Mac OS X distributed object system, under the public name 'foo'. Other applications can connect to it and use it.**

```
> 'foo' connect

a proxy for [:a :b| a+b]
```

Lightweight interactive & scripting layer

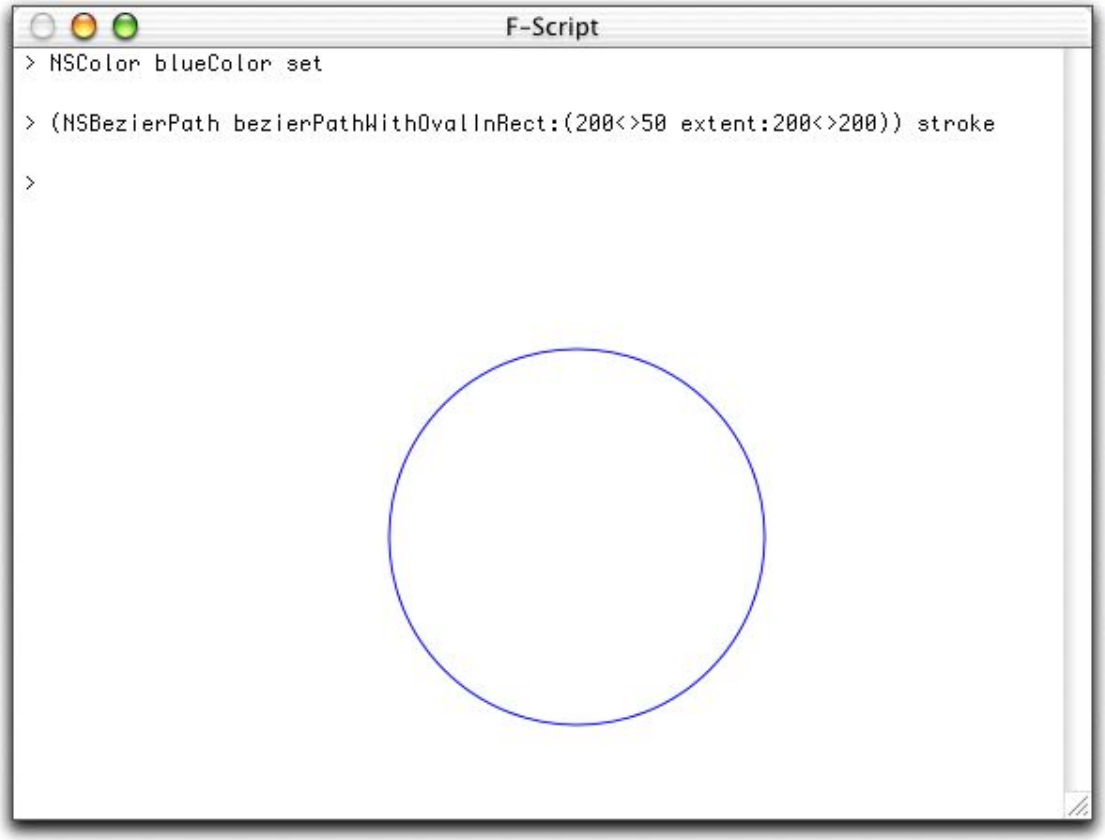
- **Emphasis is on interactive high-level and user-friendly object manipulation rather than class development.**



Block inspector allowing in-place editing of block code.

Lightweight interactive & scripting layer

- **Emphasis is on interactive high-level and user-friendly object manipulation rather than class development.**



User-friendly access to core Mac OS X technologies (here: drawing a blue circle with Quartz).

High level object manipulation

- **Emphasis is on interactive high-level and user-friendly object manipulation rather than class development.**

One of the main weakness of object technology: high-level manipulation of data.

Weakness of object technology

- **Years ago relational supporters argued that object-oriented databases were a twenty-year step backward.**
- **They were right !!!**
- **Sure, object technology provides a high-level modeling approach. But think about how low-level, object technology is compared to relational algebra when it comes to manipulating whole sets of data.**
- **This is a big challenge!**

Weakness of object technology

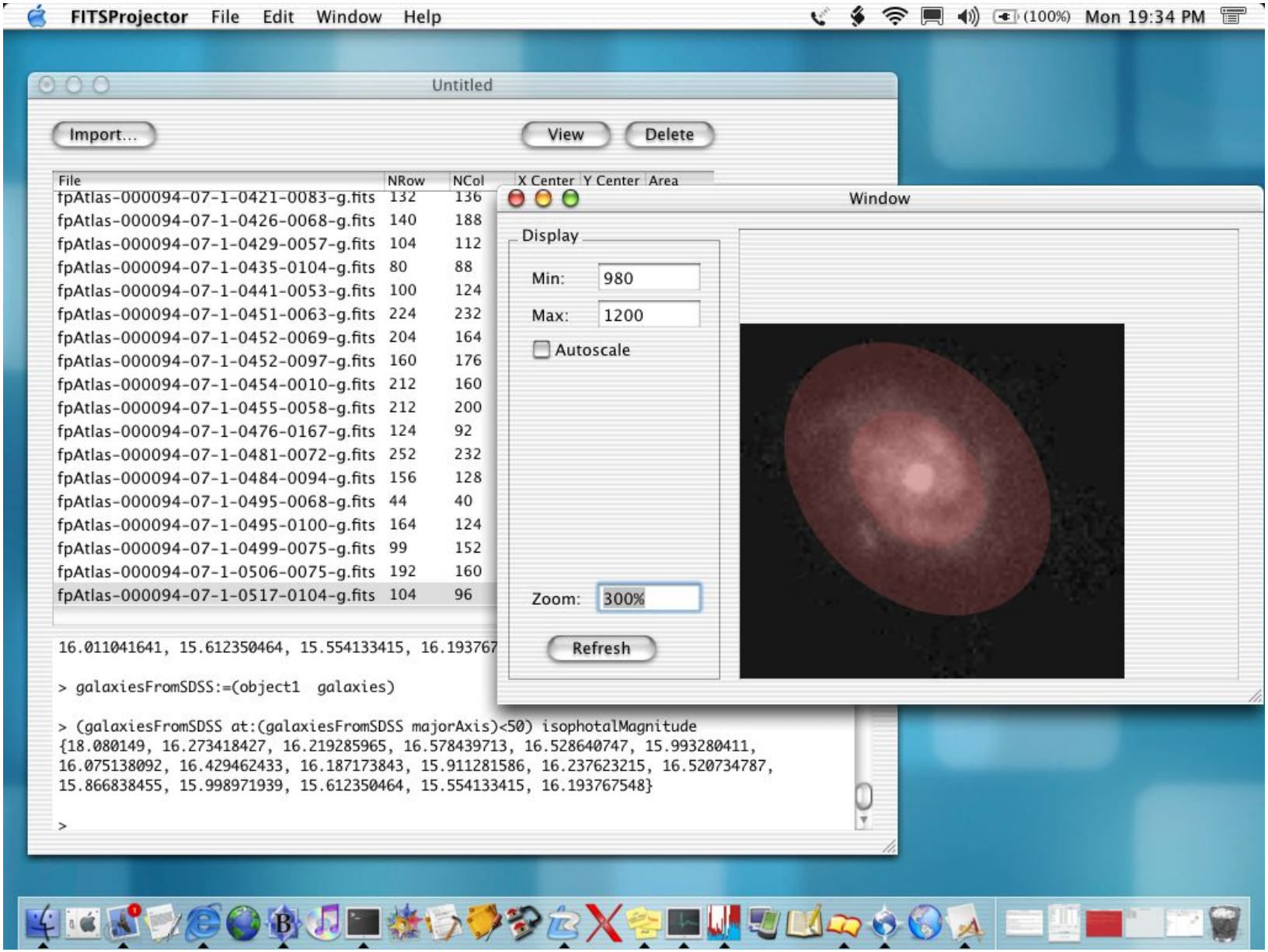
- **This criticism prompted a major enhancement of object technologies: the development of object query languages.**
- **This development provided a solution to the lack of high-level features found in traditional object languages.**
- **So far, results are mixed.**
- **Object query languages have not yet made their way into mainstream object languages.**
- **In their most recent incarnations, object query languages adopt quite a low profile. They are merely used as an interface to an underlying database, and not as a general means of manipulating objects (e.g. JDO Query Language, EJB Query Language, Gemstone etc.). "The query is executed in the databse, not in the VM".**

High-level object manipulation

- **We want something with the power of relational algebra built into our programming language!**
- **Array programming principles (cf. Ken Iverson's APL) to the rescue.**

Smalltalk extended into an object query language allowing synthetic expression of object manipulations.

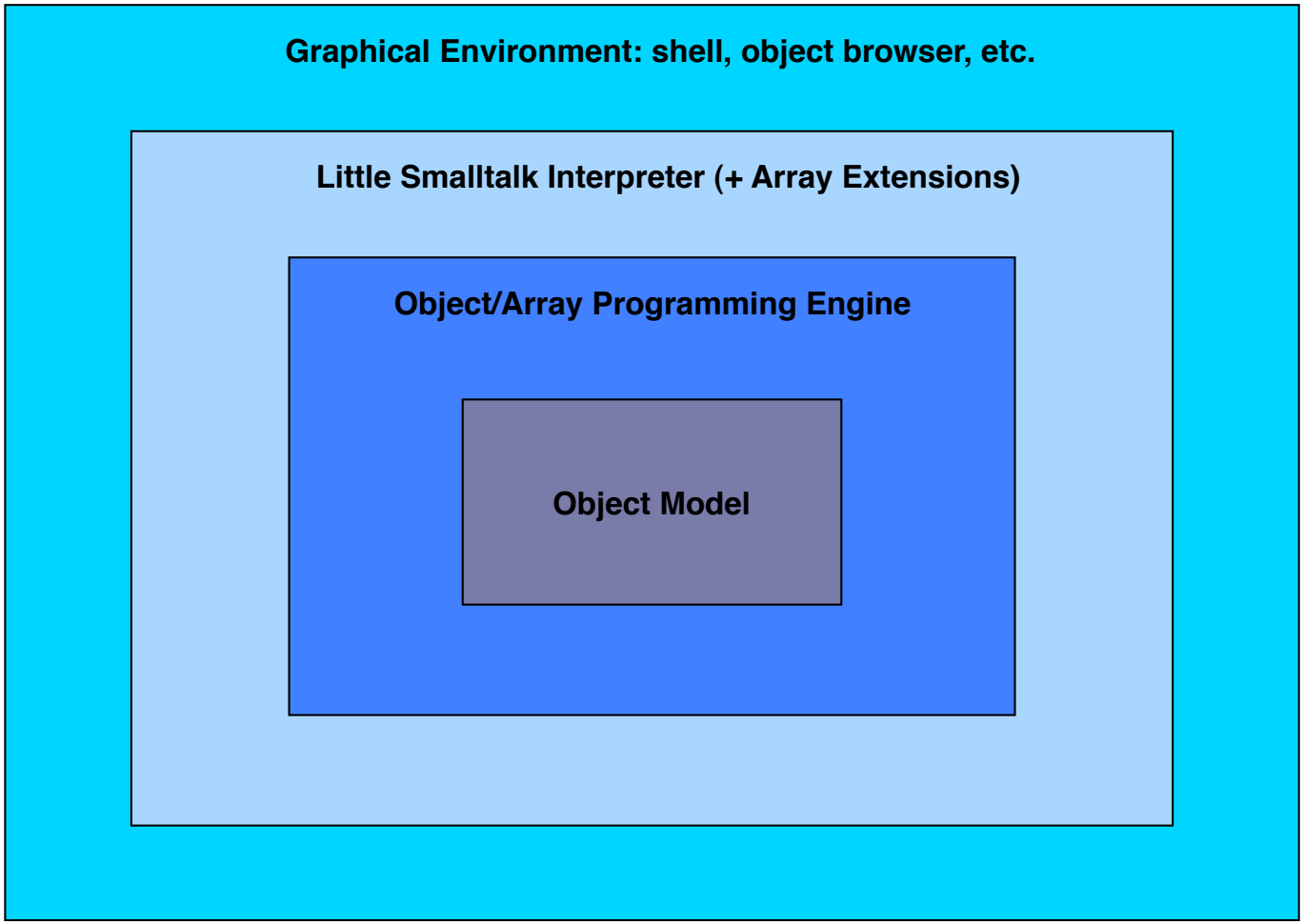
- A new message send paradigm. The classic Smalltalk message construct becomes a special case of a more general messaging system.
- A small kernel of high-level operators for object collection manipulation: Compression, Reduction, Join, Transposition, etc. (implemented as methods).
- Aim was to minimize extensions needed to include query language level capacities in Smalltalk. From a technical point of view, can easily be adapted to existing Smalltalk implementations.



"F-Script lets one easily do pretty complex data mining to drill down through samples of thousands of galaxies distributed throughout a very large parameter space."

Prof. Roberto Abraham - Dept. of Astronomy & Astrophysics - University of Toronto.

F-Script Architecture



F-Script Object Model & Frameworks

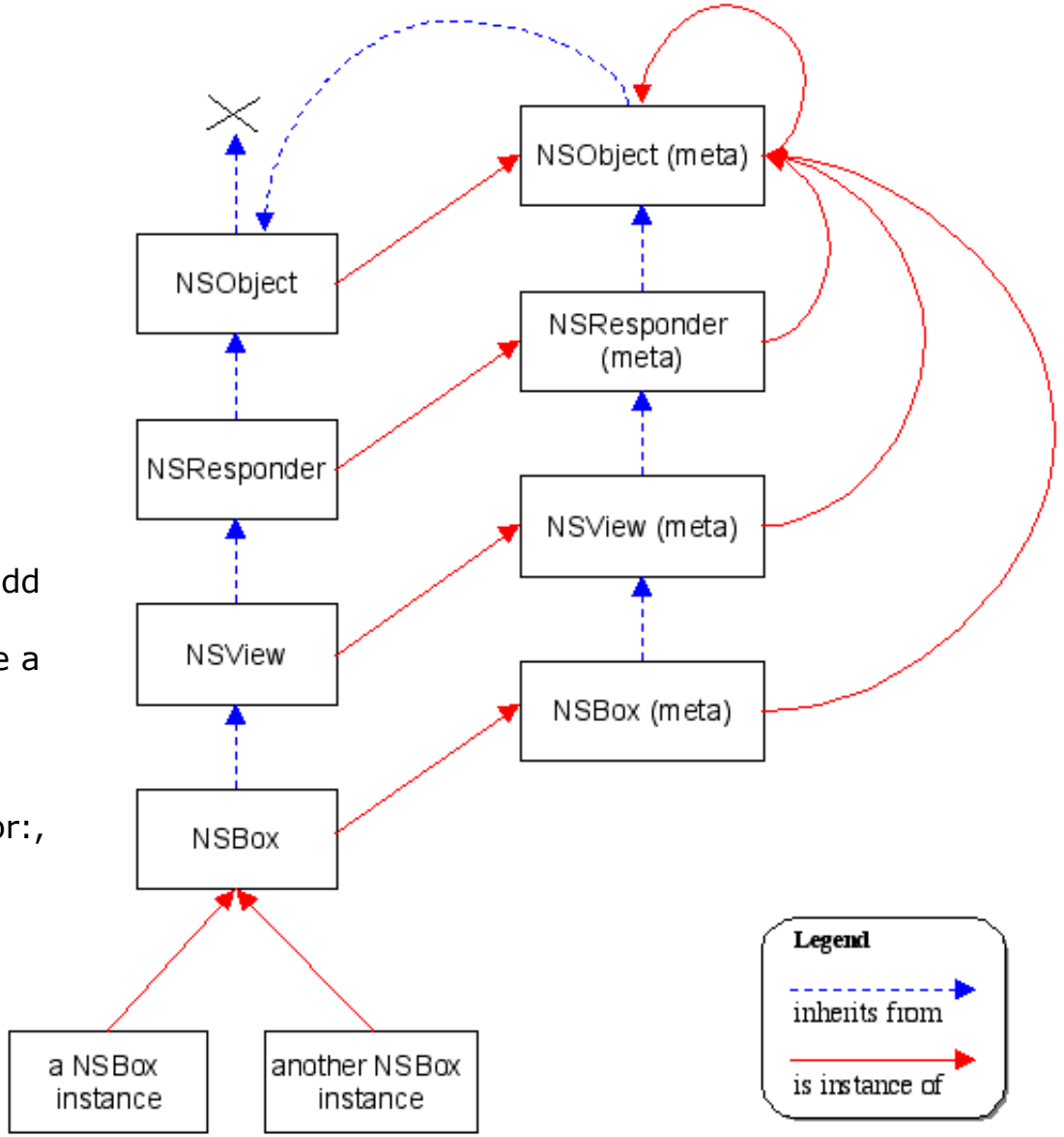
- **F-Script is based on Smalltalk syntax and concepts, but instead of using Smalltalk's object model and common Smalltalk frameworks, it use the Cocoa object model and frameworks.**

- **Thanks to the integration with Cocoa, F-Script gains many advanced features, which can be directly accessed by the F-Script user, "for free":**

  - **Sophisticated GUI framework (Windows, Views, Event Model, Text Layout, etc.),**
  - **Unicode support,**
  - **Access to the Quartz features (Mac OS X graphic subsystem),**
  - **Distributed Objects,**
  - **Object Persistence,**
  - **Integration with Interface Builder,**
  - **Networking,**
  - **File System interaction,**
  - **UNIX abstractions: tasks, pipes, etc,**
  - **Collection classes,**
  - **Etc.**

F-Script/Cocoa Object Model Example

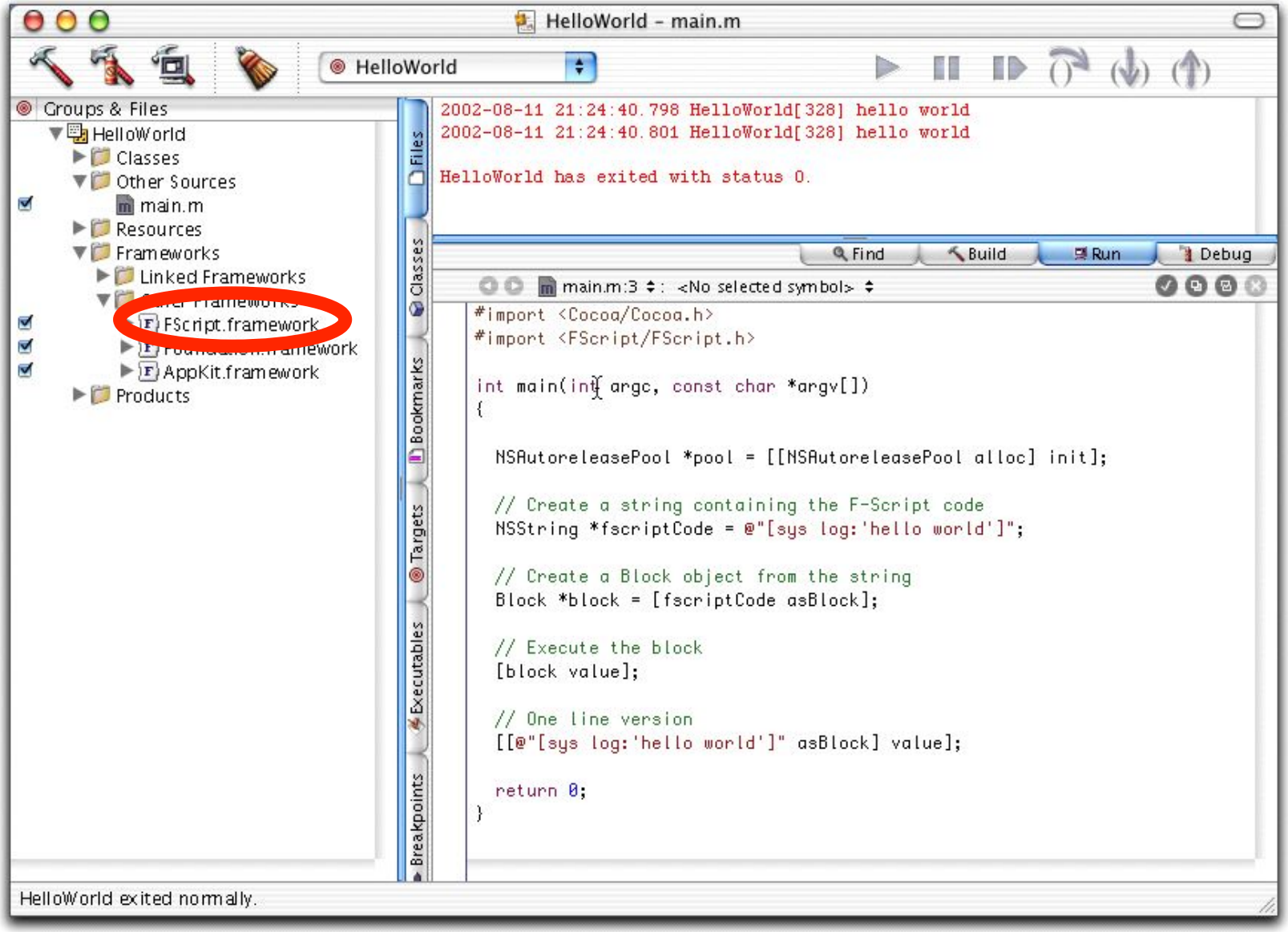
- Derived from the Smalltalk object model.
- Smalltalk's keyword syntax.
- Class and meta-classes are objects.
- Single inheritance. Support for protocols.
- Fully dynamic. Static typing optional.
- Flexible: the notion of "category" lets you add methods to classes you have not produced. The "poseAsClass:" method lets you replace a class by another one a runtime, etc.
- isKindOfClass:, respondsToSelector:, performSelector:, doesNotRecognizeSelector:, etc.



F-Script is embeddable

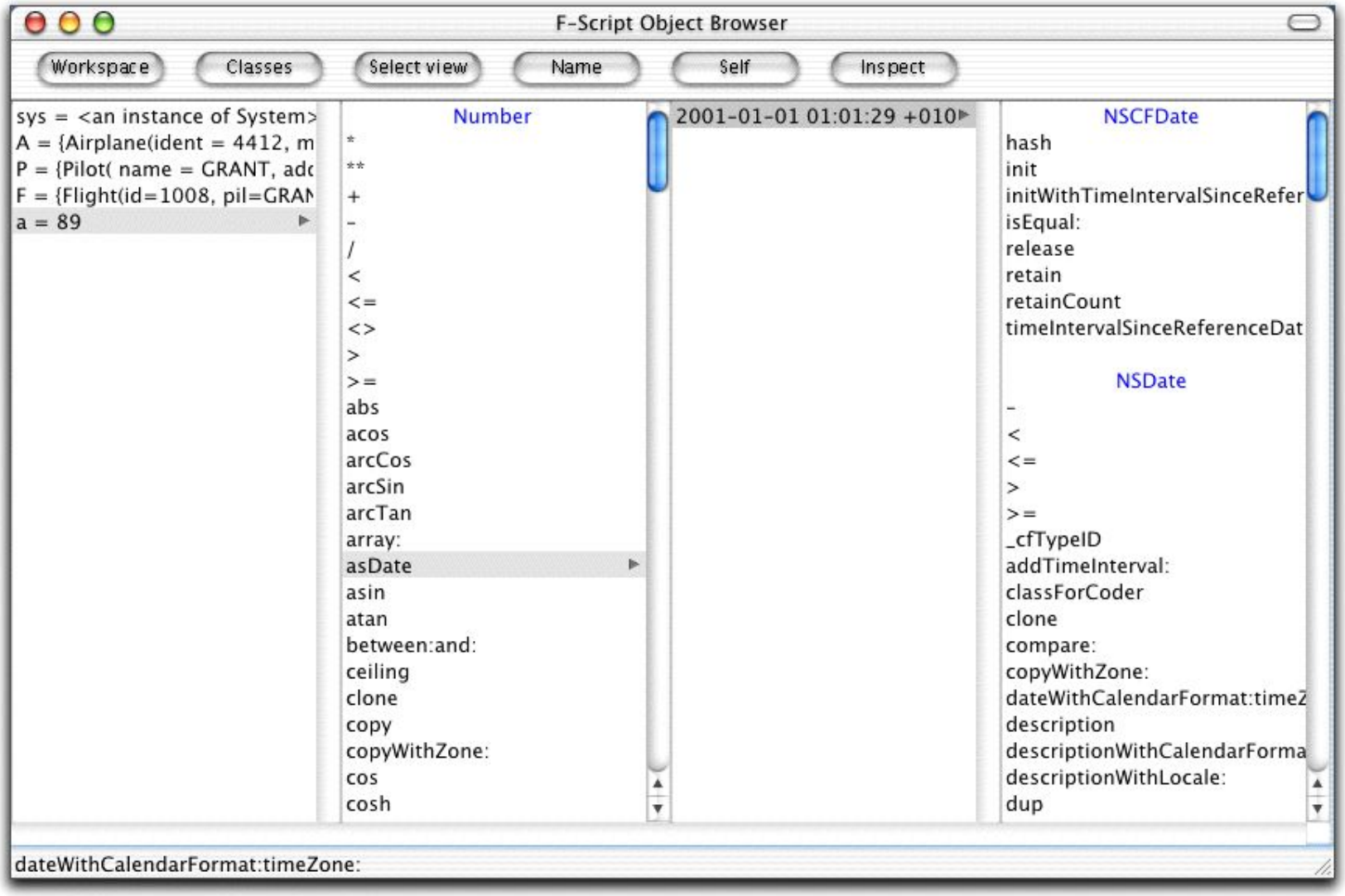
- **In addition to a stand-alone application, F-Script comes as an embeddable Mac OS X framework.**
- **This means you can use an F-Script interpreter (or several ones) in your own applications.**
- **F-Script components are accessed through an Objective-C API.**

F-Script is embeddable

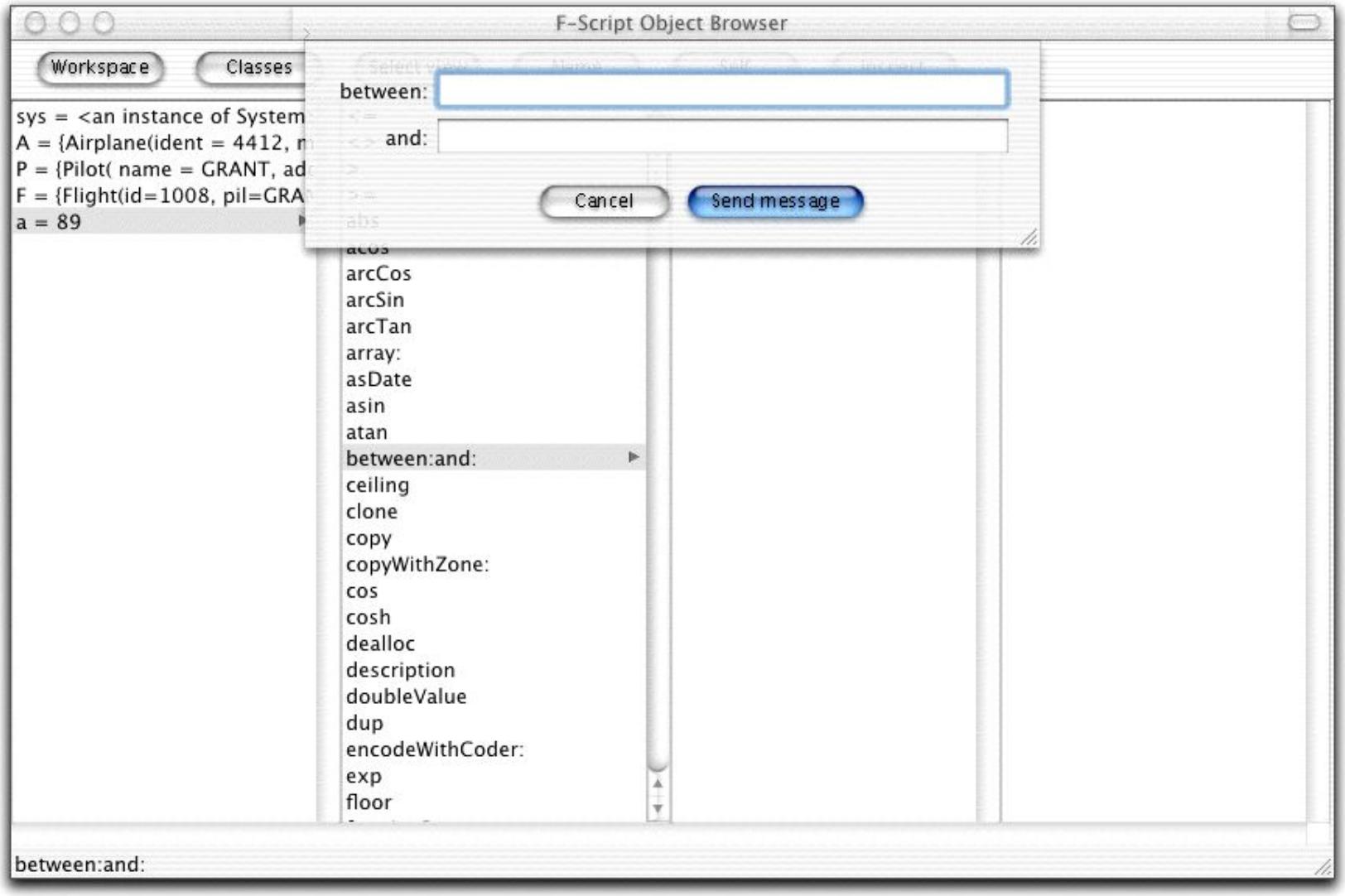


Developing with Project Builder and the F-Script framework

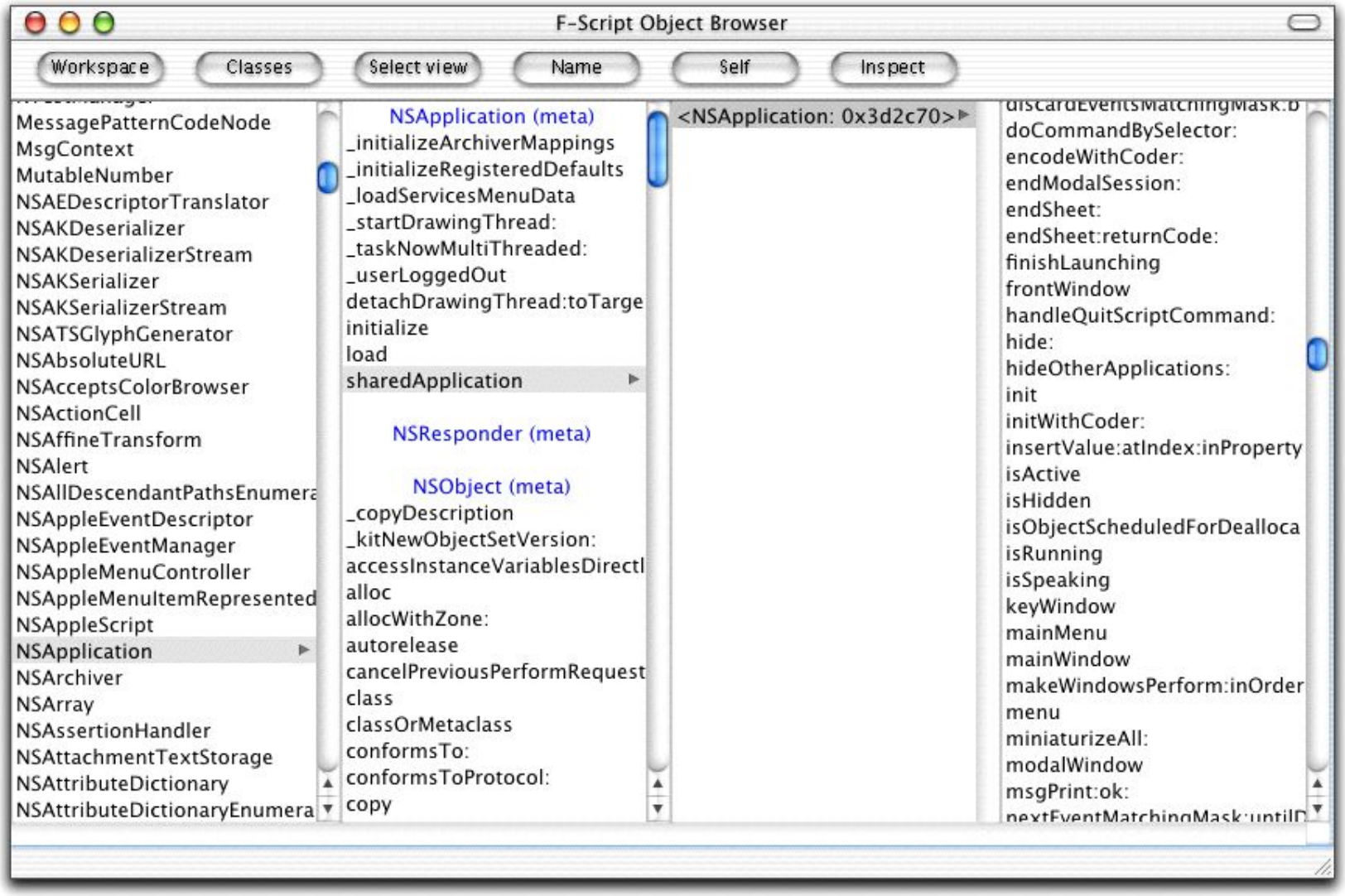
Object browser: browsing the workspace



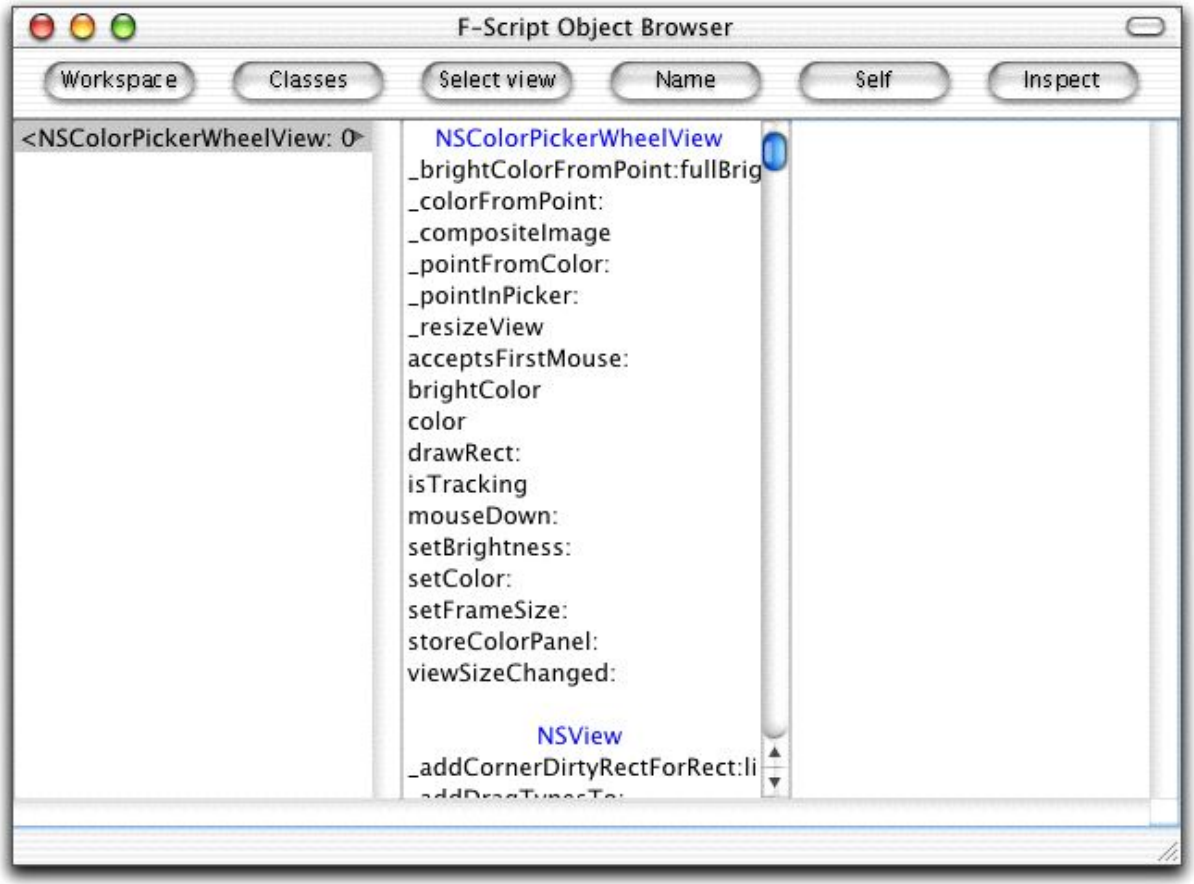
Object browser asks for arguments

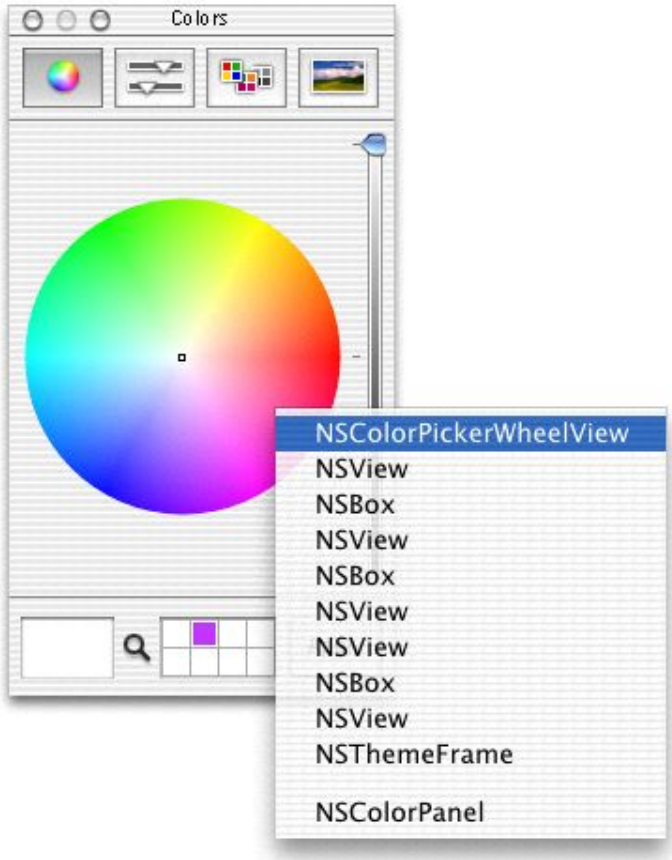


Browsing classes and meta-classes

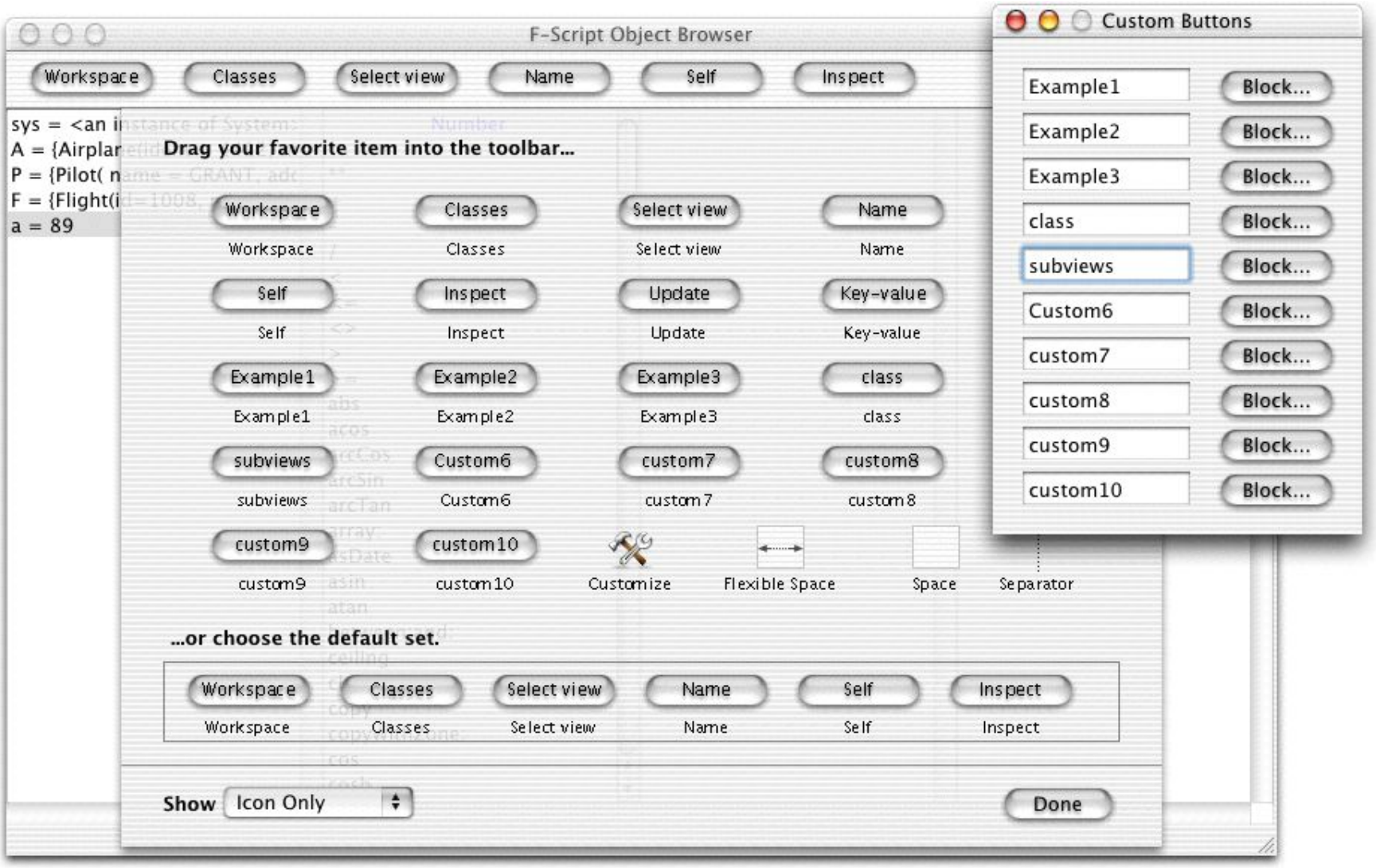


On-Screen selection of widgets

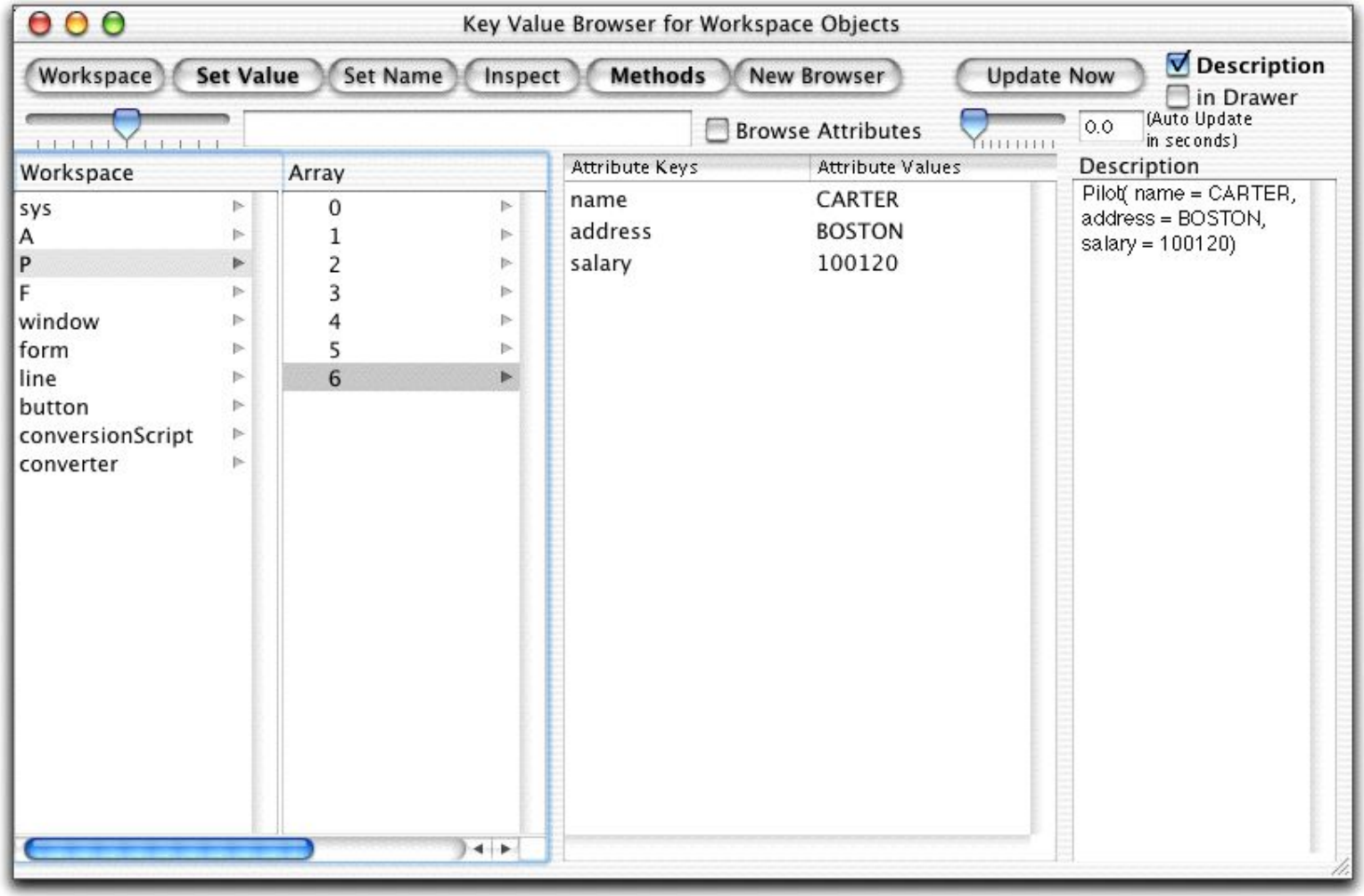




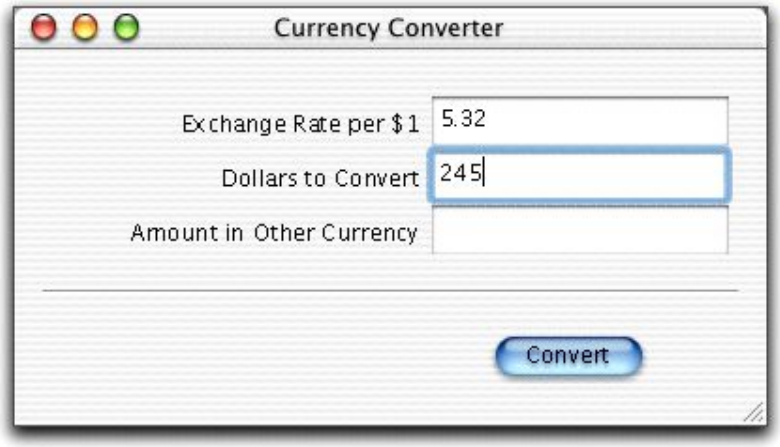
Customizing the object browser



The key-value browser *(by Joerg Garbers)*

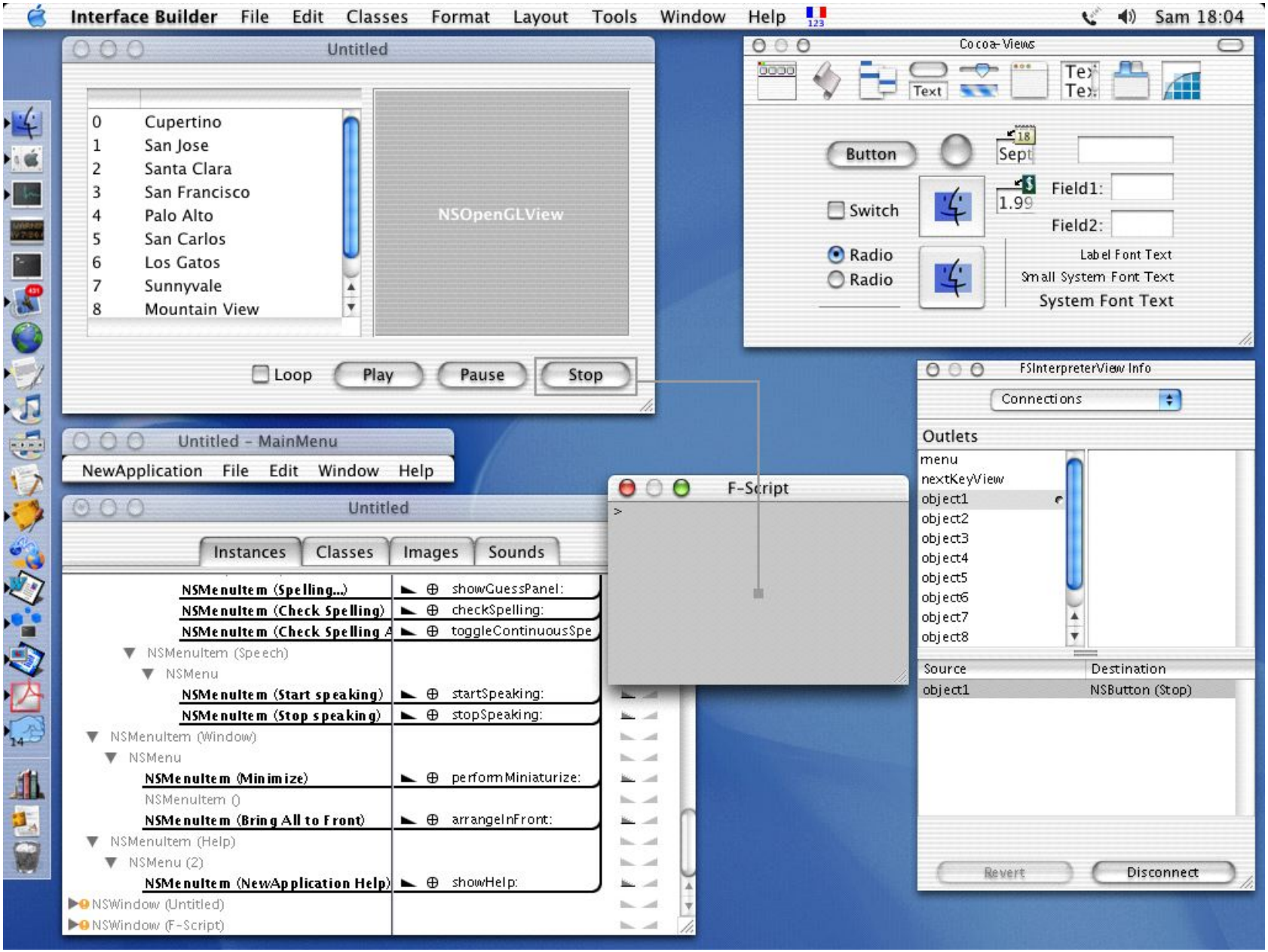


Using Mac OS X frameworks from F-Script



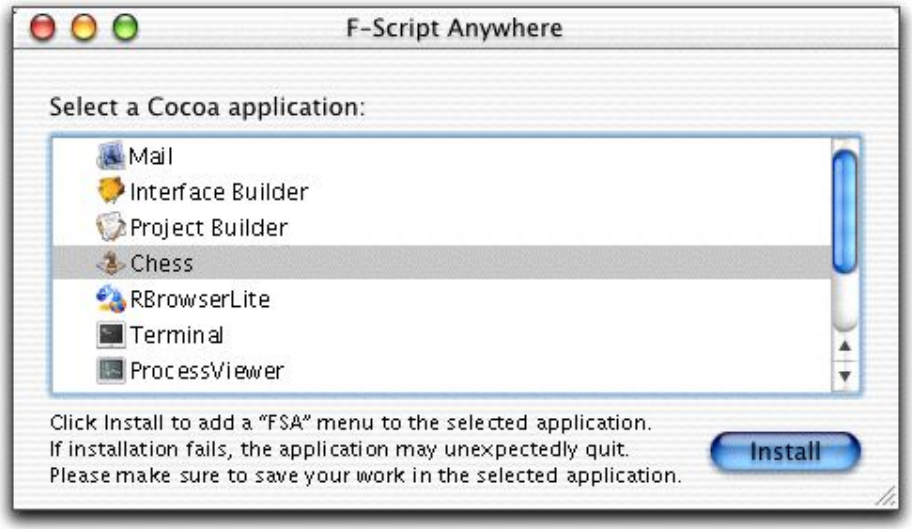

```
[:title | |window conversionScript form button line|
window := NSWindow alloc initWithContentRect:(125<>513 extent:400<>200)
                   styleMask:NSTitledWindowMask+NSClosableWindowMask+NSMiniaturizableWindowMask+NSResizableWindowMask
                   backing:NSBackingStoreBuffered
                   defer:true.

conversionScript := [(form cellAtIndex:2) setStringValue:(form cellAtIndex:0)
floatValue * (form cellAtIndex:1) floatValue. form selectTextAtIndex:0].

form := (NSForm alloc initWithFrame:(60<>90 extent:320<>85)) autorelease.
form addEntry:@{'Exchange Rate per $1', 'Dollars to Convert', 'Amount in Other Currency'}.
form setAutosizesCells:true; setTarget:conversionScript; setAction:#value.

button := (NSButton alloc initWithFrame:(250<>20 extent:90<>30)) autorelease.
button setBezelStyle:NSRoundedBezelStyle; setTitle:'Convert'; setKeyEquivalent:'\r'.
button setTarget:conversionScript; setAction:#value.

line := (NSBox alloc initWithFrame:(15<>70 extent:370<>2)) autorelease.

window contentView addSubview:@{form, button, line}.
window setTitle:title; orderFront:nil.
]
```

Interface Builder and F-Script

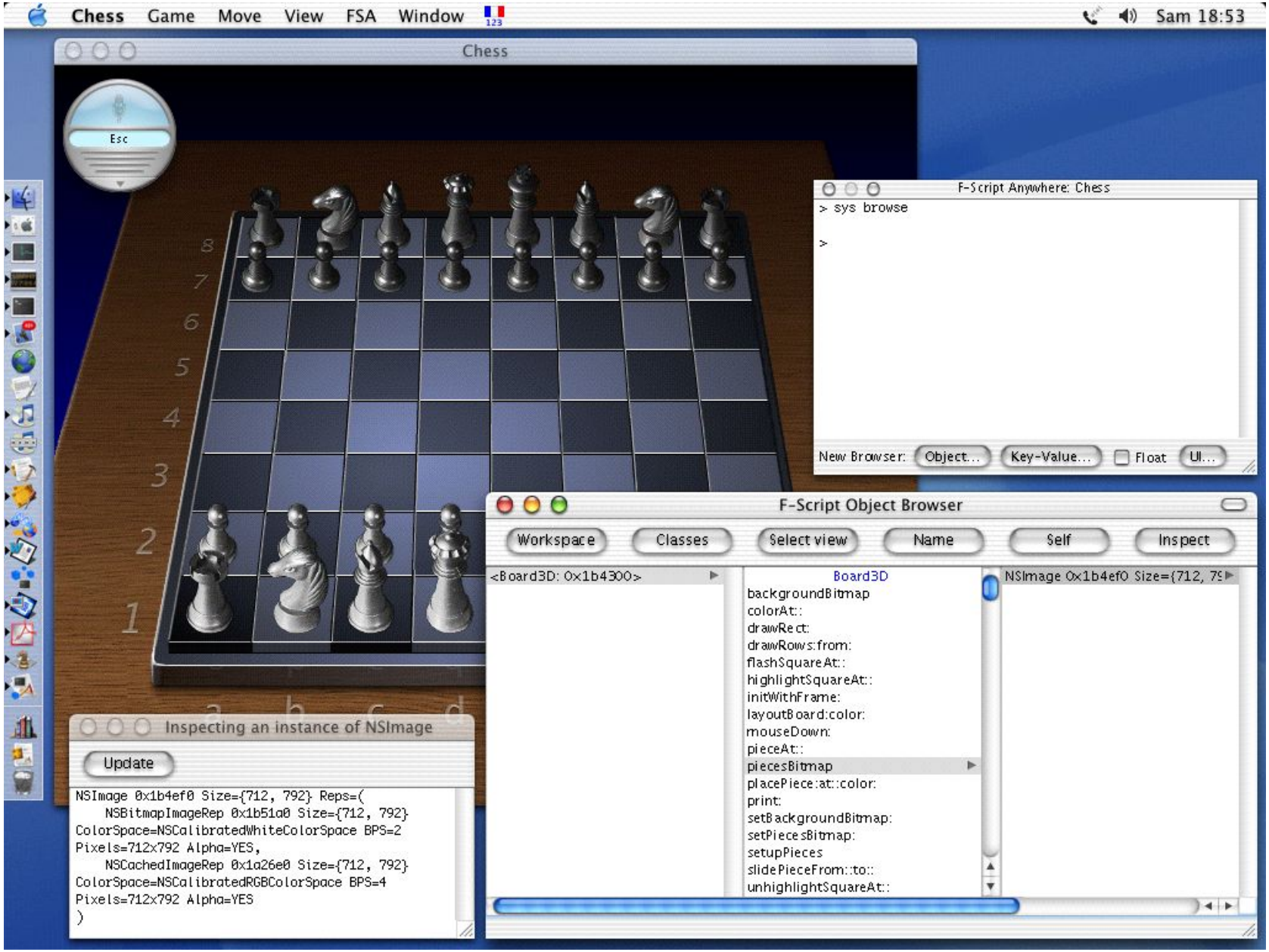


F-Script Anywhere

- **F-Script Anywhere, developed by Nicholas Riley, allows you to inject a complete F-Script environment into any running Cocoa application - even third-party applications!**



F-Script Anywhere inside Chess



Download & Resources

- **Binaries for Mac OS X**
- **Source code**
- **User guide with complete reference documentation**
- **Articles published by O'Reilly Network**
- **Paper from APL 2000, documenting the high-level extensions**
- **Additional tools, including F-Script Anywhere**
- **Mailing List**

www.fscript.org

Thanks !

See you at ESUG 2003 !