

# SELF 4.1

## Read This First

Welcome to SELF 4.1. This document describes this release and takes you through the steps needed to install and start using SELF. I hope that even more of you enjoy using SELF now that it also runs on the Macintosh. Good luck!

Past members of the SELF Group were:

Ole Agesen  
Lars Bak  
Craig Chambers  
Bay-Wei Chang  
Urs Hölzle  
Elgin Lee  
Ole Lehrmann Madsen  
John Maloney  
Randall B. Smith  
David Ungar  
Mario Wolczko

Thanks to MetroWerks for their excellent programming environment and to Apple Computer. Finally, it gives me great pleasure to thank Sun Microsystems Laboratories and its director (and Sun CTO) Greg Papadopoulos and Neil Wilhelm for their support.

Copyright (c) 1999, Sun Microsystems, Inc. and Stanford University. All Rights Reserved.

Sun Microsystems, Inc.  
901 San Antonio Road  
Palo Alto, CA 94303 USA

**RESTRICTED RIGHTS LEGEND:** Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 (Oct. 1988) and FAR 52.227-19(c) (June 1987).

**SOFTWARE LICENSE:** The software described in this manual may be used internally, modified, copied and distributed to third parties, provided each copy of the software contains both the copyright notice set forth above and the disclaimer below.

**DISCLAIMER:** Sun Microsystems, Inc. makes no representations about the suitability of this software for any purpose. It is provided to you "AS IS", without express or implied warranties of any kind. Sun Microsystems, Inc. disclaims all implied warranties of merchantability, fitness for a particular purpose and non-infringement of third party rights. Sun Microsystems, Inc.'s liability for claims relating to the software shall be limited to the amount, if any of the fees paid by you for the software. In no event will Sun Microsystems, Inc. be liable for any special, indirect, incidental, consequential or punitive damages in connection with or arising out of this license (including loss of profits, use, data, or other economic advantage), however it arises, whether for breach of warranty or in tort, even if Sun Microsystems, Inc. has been advised of the possibility of such damage.

# 1 The SELF Project

SELF is an object-oriented programming system. It includes a programming language, a collection of objects defined in the SELF language, and a programming environment built in SELF for writing SELF programs. The language and environment attempt to present objects to the programmer and user in as direct and physical a way as possible. The system uses the prototype-based style of object construction.

The first version of the SELF language was designed in 1986 by David Ungar and Randy Smith [US87]. The very first implementation of SELF was an experimental interpreter written in the Smalltalk system by Craig Chambers, Elgin Lee, and Martin Rinard in early 1987. The SELF project started in earnest in the summer of 1987 with the implementation of a custom memory system and a SELF compiler. By the summer of 1988, the first efficient implementation was completed [CU89] [CUL89] [Lee88], and two public releases, 1.0. and 1.1 took place thereafter.

In early 1991 the SELF project moved to its present home, Sun Microsystems Laboratories Inc., and another release, version 2.0, took place in Spring 1992. The first compiler was retired in version 2.0, and new ideas led to the development of a second-generation compiler [CU90][CU91]. Version 2.0 also included new language mechanisms for multiple inheritance [CUC91] [UCC91].

A further release, 3.0, took place in January 1993, which included an experimental graphical user interface [CU93] to test version user interface ideas, a much-simplified inheritance mechanism (most of the features added in 2.0 were removed!), and a third-generation compiler [HCU91].

Release 4.0 contains an entirely new user interface and programming environment designed for “serious” programming.<sup>†</sup> This environment enables the programmer to create and modify objects entirely within the environment, and then save the object into files for distribution purposes. You no longer have to edit source files using an external editor (indeed, it is discouraged). The metaphor used to present an object to the user is that of an *outliner*, allowing the user to view varying levels of detail. Also included in the environment is a graphical debugger, and tools for navigation through the system.

- A new version of the *transporter*, a facility for saving objects structure into files, has been used to modularize the system. The programming environment presents an interface to the module system which allows for straightforward categorization of objects and slots into modules, and the mostly-automatic saving of modules into files. Handwritten source files have almost completely disappeared.
- The environment has been constructed using a new, flexible and extensible user interface construction kit, based on *morphs*. Morphs are flexible and general-purpose user interface components. An extensive collection of ready-built morphs is provided in the system, together with facilities to inspect, modify and save them to files. We believe the morph-based substrate provides an unprecedented degree of openness and flexibility in user interface construction. A novel feature of this user interface substrate is that all applications built using it can be shared among multiple users.

---

<sup>†</sup> We do expect, however, that even serious programmers will have fun using Self.

- Two medium-sized applications are included which use the new user interface: a Web browser, and a Smalltalk emulator.
- Significant engineering has been done on the Virtual Machine to reduce the memory footprint and enhance memory management. For example, a 4.0 system containing a comparable collection of objects to that in the 3.0 release requires 40% less heap space. A SELF-level interface to the memory system is now available that enables SELF code to be notified when heap space is running low, and to expand the heap.

The privacy syntax has been removed; in the previous release it was accepted but privacy was not enforced. The concept of privacy still exists, and is visible in the user interface, but is supported entirely through the annotation system.

SELF is the result of a research project and is therefore an experimental system. We believe, however, that the system is stable enough to be used by a larger community, giving others the opportunity to explore SELF.

## 2 This release

Self 4.1 adds an alpha-quality Macintosh port to Self 4.0. Thus, it adds three major parts:

- a portability framework for the virtual machine
- a PowerPC/MacOS port of the simple compiler (the NIC), and
- a portability framework for UI2 and a port of UI2 to the MacOS.

Although it has many rough edges, we hope this release (done as a spare-time project) will benefit two user communities: Macintosh users can now experiment with prototype-based exploratory programming, and those who wish to port the Self VM can benefit from the portability framework.

SELF currently runs on SPARC-based Sun workstations using Solaris 2.3 or later, and Power PC Macintoshes. I am not sure which Macintoshes it will run on; I have run it under MacOS 8.6 on a 266 Mhz G3 Powerbook and a 300 Mhz G3 Powerbook. The SPARC compiler is an improved version of the one used in 3.0 [Hoe94] [HU94]. The Macintosh includes only the simple, nonoptimizing compiler.

A mailing list exists for discussion of Self and related topics. To subscribe to the list, send a request to [self-interest-request@self.sml.com](mailto:self-interest-request@self.sml.com). The list itself can be reached at [self-interest@self.sml.com](mailto:self-interest@self.sml.com).

### 2.1 Bug reports

Bug reports can be sent to [david.ungar@sun.com](mailto:david.ungar@sun.com). Please include an exact description of the problem and a program or snapshot that can reproduce the bug. Before reporting a bug, check that it is not already in the list of known bugs (see file `help.text`).

## 3 Documentation

There are several items of documentation available:

1. This document, which includes installation instructions.
2. An on-line tutorial which teaches you about various aspects of SELF.
3. A guide to the programming environment introduced in this release. This describes the basics of how to use the system to create, debug, modify and export SELF applications.
4. A Programmer's Reference Manual which describes the SELF language, the core objects of the SELF world, discusses SELF style, and describes the SELF Virtual Machine. This document is changed only slightly from the 3.0 release.
5. A guide to the user interface framework introduced in this release. Before embarking on a non-trivial application using the SELF user interface framework, you should read this document.
6. A document describing the structure editing facilities implemented by Ole Lehrmann Madsen, and their implementation.
7. A document describing the Smalltalk emulator.

We recommend that you complete reading this document, install the system, and try the on-line tutorial. Thereafter, read parts of the remaining documentation as required.

All manuals are PostScript files (with the first page first) that can be viewed on-line using a Post Script viewer program such as pageview or printed on a PostScript printer. If your printer deposits pages face up, you may wish to run the PostScript files through the psrev program to reverse the page order before printing.

## 4 How to install SELF

The SELF system can be obtained by ftp or http. Installation consists of three steps: getting the archive files, performing the site installation, and setting up the working environment for an individual user.

### 4.1 System requirements

To run SELF you will need a SPARC-based Sun computer or clone running SunOS 4.1.X or Solaris 2.3 or 2.4. To use the programming environment you will need to run X Windows version 11 or OpenWindows on an 8-bit or deeper color display<sup>†</sup>. The X server need not reside on the same host as SELF.

To run the distributed system (including on-line tutorial, Web browser and Smalltalk emulator) well, the SELF system requires a host machine with 48 Mb of RAM or more. Further, the system places high demands on the CPU. We do not recommend its use on anything slower than a Sparc-

---

<sup>†</sup> Although the system will work on a gray-scale display, the color choices have not been tested and distinct colors may map to similar shades of gray. Also, the documentation assumes you are using color.

Station-10. In particular, user interaction on a SparcStation-1 or 2 is unlikely to be acceptable; a SparcStation-5, 10 or 20 is fine.

The user interface makes substantial demands of the X server. A graphics accelerator (such as a GX card) improves the responsiveness of the user interface significantly, and therefore we recommend that you use one if possible.

## 4.2 Getting the system

To get the SELF system, use a Web browser at URL `http://self.sml.com` or anonymous ftp to retrieve the files from `self.sml.com`. The required files reside in the directory `/pub/self-4.0`.

Archive file	Contents
README	Describes the contents of the directory.
<code>readThisFirst.ps</code>	This document.
<code>Install</code>	Shell script to unpack the SELF distribution.
<code>Self-4.0.sunos.tar.Z</code>	The SELF system for SunOS 4.1.X.
<code>Self-4.0.solaris.tar.Z</code>	The SELF system for Solaris 2.3/2.4.

Retrieve `Install`, and one or more of the `tar` files as appropriate for your host machine(s).

There are other, optional files available, containing the source to the Virtual Machine, filed-out versions of the SELF modules, and code for interfacing to external libraries. These are not required to run the system, but are also available by ftp. Further details on how to get these files can be found in the documentation available with the system (specifically, in the Programmer's Reference Manual).

## 4.3 Unpacking the archive files

Prior to running `Install`, you should be sure there is sufficient disk space. The disk space required for the unpacked system is ~30 Mb (in addition to the `tar` file).

Type in the following to execute the `Install` script, substituting the name of the archive file that you have retrieved.

```
% chmod +x Install
% ./Install Self-archive-file-name
```

This will unpack the archive file. The archive file can be deleted when the installation is complete. Also included with Self are utilities for image conversion, which are required by the Self Web browser. These are unpacked into directories called `alchemy` and `pbmplus`. If you do not intend to use the Web browser, you may remove these. If you do use them, be sure to read the copyright and licensing policies that are included with them.

## 5 How to start and stop SELF

You are now ready to run SELF. Make sure that the SELF Virtual Machine (the program named `Self`) is in your command search path. You should be running OpenWindows or X, and `$DISPLAY` should refer to a suitable X display (see the requirements above).

To start SELF, type “Snapshot” at the UNIX prompt:

```
% Snapshot
```

```
        Welcome to the Self system!
```

```
Copyright 1992, 1993, 1994, 1995 Sun Microsystems, Inc. and
Stanford University.
```

```
See the LICENSE file for license information.
```

```
Type _Credits for full credits.
```

```
VM version: 4.0.5
```

```
Server started.
```

```
"Self 1"
```

If all is well, the system will start the programming environment and display the on-line tutorial.

To exit the system, you can use the ‘quit’ item from the middle-button background menu from within the programming environment.

If you have any problems when using the system, refer to the help file in the documentation (`help.txt`).