

## 1. Installation Instructions

The following lists installation instructions for those systems to which Version 3 of Little Smalltalk has been ported at present. Note that installation involves the creation of two programs. The first, called ``initial'', is run once to create the initial object image (usually a file called ``systemIm-age"). The second program is the smalltalk interpreter. To run smalltalk, both these files must be accessible. Systems that use the supplied Makefile run initial automatically; in some other systems you may need to do this manually.

If you receive the distribution on Mac or IBM disks and you want to run the system under Unix you must ``undo" some of the changes described below.

### 1.1. Atari

I've been told (no first hand experience) that the code works on the atari. I've set up a minimal description in env.h - could somebody tell me if the atari supports prototypes, signals, or some of the other features?

You do have to make the 'rb' changes described for the IBM PC (below), however you keep the rm instruction instead of DEL, and change the editor to whatever your system has (memacs?).

### 1.2. Gnu C Compiler (Sequent Balance)

If at all possible, Use the Gnu C Compiler. I have found the code to be much smaller (up to 1/3 smaller) and much faster (up to twice as fast). So far this has been used on the Sequent Balance system.

Note that these sources support old style prototypes, as are used in Lightspeed C and Turbo C, and not the newer ANSI prototypes as are used in the gcc compiler. So do not define PROTO when using the gcc compiler

### 1.3. HP-UX

Simply say ``make sysvttty" to make a version with the tty interface. (As of yet, I don't have access to a system v system with an X-window interface, so I can't test that code).

### 1.4. IBM PC / Turbo C compiler

Note: If you receive the sources on 5.25" disks containing both source and executable, the following changes have already been made to the system.

Define the symbol TURBOC at the beginning of the file env.h.

Edit the file file.st, changing the command used to delete files from rm to del (notice the space following the del):

```
delete  
    ('del ', name) unixCommand
```

In the file file.st change the mode on the command to save images from w to wb.

```
savelImage: name
  scheduler critical: [
    " first get rid of our own process "
    scheduler removeProcess: scheduler currentProcess.
  ]
File new;
  name: name;
  open: 'wb';
  savelImage;
  close ]
```

In a similar manner change the mode on the file open in the initialize method in file tty.st to use wb instead of w.

```
initialize
  " initialize the initial object image "
  self createGlobals.
File new;
  name: 'systemImage';
  open: 'wb';
  savelImage;
  close.
```

And also in tty.st change the editor from vi to me (or whatever your favorite editor happens to be).

```
editor <- 'me'.
```

Because of segmentation limits it is not possible to have an object table any larger than 6500 objects (the current default). This value is set by a define found in memory.h

```
# define ObjectTableMax 6500
```

Compile in the compact mode (small code, large data).

## 1.5. Macintosh Lightspeed C

NOTE: If you get the distribution on 3.5" MAC Disks the source code changes described below have probably already been made for you.

The mac distrubtion disk contains the following.

(a) A folder called ``C Sources" that contains (naturally) all the C sources.

(b) A folder called ``ST Sources" that contains (also naturally) all the Smalltalk sources, plus an application called ``initial" that can be used to create or recreate the initial object image. To make changes to the image, simply edit the appropriate smalltalk files, run initial, and move the file ``systemImage" to the appropriate location.

(c) Two Lightspeed C projects called ``TextEdit" and ``Stdwin", containing code taken from Guido van Rossums Standard Windows package.

(d) A file called ``systemImage", which is the output of the application from part (b)

(e) An application called ``st", which is the smalltalk interpreter.

(f) A folder called ``misc" that contains various different files, such as documentation and other things.

It is only necessary to recompile if you make changes to the C source. If you make changes to the Smalltalk source you only need to rerun the application called ``initial" contained in the ``ST Sources" folder.

If you get the sources from some other location (say off the net), you must make the following alterations. Change the mode on the file open in the saveImage command (in file.st) and in the initialize command (file stdwin.st). Define the symbol LIGHTC at the beginning of the file env.h (See instructions for the IBM PC above for a fuller explanation).

To compile you need guido van rossums Standard Windows package. Follow his instructions to create the stdwin and textedit projects (these are already on the distribution disk). To make the initial program, create a project ``initialProj" with segments as follows. In the first segment place MacTraps. In the second segment place Stdwin. In the third place TextEdit. In the forth place the Unix library files math, stdio, storage, strings and unix. In the fifth place the sources filein.c, initial.c, interp.c, memory.c, names.c, news.c, primitives.c, unixio.c and winprims.c. In the sixth and final segment place lex.c and parser.c. To create the st program use the same structure, substituting st.c for initial.c. You must check the ``separate STRS" option on both projects.

Make sure when you run the initial object that all the smalltalk sources are in the current directory; it does not complain if it can't open a file, it simply goes on. Also when you fileIn a file, the file must be in the current directory.

The Mac version uses the windowing interface. It is currently very fragile. (A few known bugs; can't restore from saved image files, output sometimes goes wrong places, output often doesn't appear until you click the mouse).

[ It would be nice if clicking on an image file would start the smalltalk application. If anybody knows how to make Lightspeed C do this, let me know. Thanks ].

## 1.6. Sequent Balance

Say ``make bsdtty" to make a tty interface system.

## 1.7. TekTronix 4315, Green Hills C Compiler

Say ``make bsdtty" to make a tty interface system. Say ``make bsdx11" to make an x-windows interface system (still somewhat buggy).

## 1.8. VAX / VMS

Since VMS doesn't understand Unix Makefiles, the distribution tape supplies a command file you can use. First define the symbol VMS near the beginning of the file env.h, then execute the command file called vms.com. This makes a version using the tty interface. A VMS version using the X-windows interface has not been created yet.

## 2. Test Cases

Once you have a running system; the following can be used to run the standard test cases. First load the file test.st. If you are using the windowing interface select the fileIn menu item and the file ``test.st" (from the ST Sources folder), if you are using the tty interface use the following command

File new; fileIn: 'test.st'

Then give the command to run all test cases.

Test new all

Messages will be displayed as test cases are performed, and if any test cases fail.

### 3. The Standard Window Package

There is an experimental windows style interface based on Guido van Rossum's standard window package. This permits the system to work on top of X-windows, as well as the macintosh. Information on standard windows can be obtained directly from guido at [guido@mcvax.uucp](mailto:guido@mcvax.uucp), or [mcvax!guido](mailto:mcvax!guido), or possibly [gvr@src.dec.com](mailto:gvr@src.dec.com). His paper mail address is Guido van Rossum, Center for Mathematics and Computer Science, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands. Sources for the standard window package are not included on the Little Smalltalk distribution, but they are available public domain by ftp from DEC SRC, [machine.gatekeeper.dec.com](http://machine.gatekeeper.dec.com) (address [128.45.9.52]). The subdirectory is `pub/stdwin`. Contact guido for more details.

To make the projects for the macintosh version, follow guidos instructions. For other versions, make a file `stdw.o` by linking together all of guidos sources for your particular system. Here is a makefile for the X11 version, for example.

```
#
# X11 version of stdwins
#
x11 = caret.o draw.o font.o menu.o timer.o cutbuffer.o error.o general.o scroll.o
window.o dialog.o event.o llevant.o system.o alfa = bind.o draw.o event.o keymap.o
measure.o menu.o scroll.o stdwin.o syswin.o gen = askfile.o perror.o
textedit = editwin.o textdbg.o textedit.o textlow.o textbrk.o tools = endian.o getopt.o
glob.o monospace.o strdup.o swap.o x11files = ${x11} ${gen} ${textedit} ${tools}

stdw.o: ${x11files}
    ld -r -o stdw.o ${x11files}
```

I emphasize this interface is very fragile.

### 4. Possible Changes

There are a couple of easy changes you may want to make at your site. The default editor is `vi` (indicated by the value of the global variable set in the routine `createGlobals` in either `tty.st` or `stdwin.st`); this can be changed to any other editor you like. The system also prints the current object count prior to asking for commands from the user. This can be eliminated by removing the primitive `<2>` from the method `initialize`, class `Scheduler`, file `tty.st`.