Answers to the Frequently Asked Questions about STk

Erick Gallesio July 1996

Contents

1 Introduction

The aim of this document is to answer to the frequently asked questions about STk. First versions of this document were written by Marc Furrer (furrer@di.epfl.ch)with the support of Erick Gallesio (eg@unice.fr)

Marc has not enough time to continue to maintain the FAQ up to date and he asked me to continue. Since I too have not a lot of time for keeping the FAQ alive, any help would be greatly apreciated.

The latest version of the FAQ is available at http://kaolin.unice.fr/FAQ/FAQ.html Text and Postscript versions of this document are also available at ftp://kaolin.unice.fr/pub/FAQ The HTML version is the recommended version of this document since all the links to other documents or softwares can be followed just by clicking on them.

This version of the document is very incomplete and if you think of questions that are appropriate for this FAQ, or would like to improve an answer, please send me e-mail.

1.1 Subject: What is STk

STk is a R4RS Scheme interpreter which can access to the Tk graphical package. It can be seen as the John Ousterhout's Tk package where the Tcl language has been replaced by Scheme (for more information on Scheme, Tcl and Tk, see the references below).

STk also provides an efficient object oriented system called STklos. STklos is an OO system with multi-inheritance, generic functions, multi-methods. It's implementation relies on a Meta Object Protocol (MOP) inspired from the Tiny -CLOS model.

Other features of STk are:

- Interpreter can be easily extended by adding new C modules.
- All the commands defined by the Tk toolkit are available to the STk interpreter (Tk commands are seen as a special type of objects by the interpreter).
- Widgets callbacks are Scheme closures
- Based on Tk4.1 (the last stable release of Tk)
- STk does not use the Tcl interpreter for Tk programming
- Tk variables (such are -textvariable) are reflected back into Scheme as Scheme variables.
- A set of STklos classes have been defined to manipulate Tk commands (menu, buttons, scales, canvas, canvas items) as STklos objects.
- New widgets can easily be built in STklos by composition of existing widgets.

1.2 Subject: Where do I get STk from

STk distribution is available on various sites. The official distribution site is http://kaolin.unice.fr/or ftp://kaolin.unice.fr/pub Distribution file names have the form `STk-x.y.tar.gz,` where `x` and `y` represent the version and the release of the package. You can also find here interim releases of STk. Intermediate releases are stored in file whose name have the form `STk-x.y.z.tar.gz` where `z` is the intermediate release number. Last stable release can be taken from http://kaolin.unice.fr/STk.tar.gzor ftp://kaolin.unice.fr/pub/STk.tar.gz

1.3 Subject: Are there distribution mirror sites

Stable releases of STk are also placed in ftp://sunsite.unc.edu/pub/Linux/devel/lang/lisp Since this site is eavily mirrored around the planet, you should easily find a stable release near you.

David Fox also maintain a mirror site of kaolin (daily updated) in the USA at the following URL ftp://cs.nyu.edu/pub/local/fox/stk

1.4 Subject: Which are the systems STk support

STk runs on a large variety of Unixes. It is known to run on:

- Sparc (SunOs 4.1.x & Solaris 2.{34})
- Dec 5xxx (Ultrix 4.2)
- SGI (IRIX 4.05, 5.1.1, 5.2)
- DEC Alpha 3000/400 (OSF-1 V1.3)
- RS6000 AIX 3.2.5
- HP 9000/735 (HP-UX 9.01)
- PC (Linux 1.0 -> 2.0)
- PC (FreeBSD 1.1)
- PC (SCO) Thanks to markd@grizzly.com
- PC (NetBSD-1.0)
- Sony WS (Sony NEWS, NEWSOS 4.2R)
- ...

1.5 Subject: STk and Windows

Starting with release 3.1, STk runs on 32 bits Microsoft Windows (only tested on Windows 95, but it should run on Windows NT). This version is based on the Tk 4.1 final release.

STk Win32 version is distributed in source and binary forms, since binary release seems more convenient on Windows.

Development of STk on this platform is not easy for me since it is not the system I use dayly. In particular, a lot of work need to be done (loading of packages compiled in DLL, sockets, ...). Help of a Win32 hacker would be greatly appreciated to achieve those tasks.

1.6 Subject: STk and MacOS

It is not intended to port STk on MAcOS, by lack of resource (and interest of myself :). However, port should be simple for a Mac hacker. I can provide help if necessary for this job.

Another way to run STk on a Mac consists to run STk under Mach Ten. If I have well understood, Mach Ten is a Unix (Mach-) like system running on top of MacOS. STk compiles fine on this system.

1.7 Subject: Getting more information about Scheme

The R4RS document

- R4RS is the document which fully describes the Scheme Programming Language, it can be found in the Scheme repository at the following URL ftp://ftp.cs.indiana.edu/pub/scheme-repository/doc
- Aubrey Jaffer has also translated this document in HTML. A version of this document can be browsed at http://www-swiss.ai.mit.edu/ jaffer/r4rs_toc.html

The Scheme Repository

The main site where you can find (many) informations about Scheme is located in the University of Indiana. The Scheme repository is maintained by David Eby. The repository currently consists of the following areas:

- Lots of scheme code meant for benchmarking, library/support, research, education, and fun.
- On-line documents: Machine readable standards documents, standards proposals, various Scheme-related tech reports, conference papers, mail archives, etc.
- Most of the publicly available Scheme Implementations.
- Material designed primarily for instruction.
- Freely-distributable promotional or demonstration material for Scheme-related products.
- Utilities (e.g., Schemeweb, SLaTeX).
- Extraneous stuff, extensions, etc.

You can access the Scheme repository at:

- ftp://ftp.cs.indiana.edu/pub/scheme-repository
- http://www.cs.indiana.edu/scheme-repository/SRhome.html

The Scheme Repository is mirrored in Europe at:

- ftp://ftp.inria.fr/lang/Scheme
- ftp://faui80.informatik.uni-erlangen.de/pub/scheme/yorku

- ftp://ftp.informatik.uni-muenchen.de/pub/comp/programming/ languages/scheme/scheme-repository

Usenet newsgroup and other addresses

There is a usenet newsgroup about the Scheme Programming language: comp.lang.scheme

Miscelleanous

Following addresses contains also interesting material about the Scheme language

- The Scheme FAQ http://www.cs.cmu.edu:8001/Web/Groups/AI/html/faqs/lang/scheme/top.htmlheme/scheme-repository
- The Scheme Home Page at MIT http://www-swiss.ai.mit.edu/scheme-home.html
- The Scheme Underground web page: http://www.ai.mit.edu/projects/su/su.html

1.8 Subject: Differences with R4RS

Symbols

STk symbol syntax has been augmented to allow case significant symbols. Many global symbols have been added to the global environment, see Appendix B of the **STk** Reference Manual.

Types

STk implements all the types defined as mandatory in R4RS. However, complex numbers and rational numbers (which are defined but not required in R4RS) are not implemented. The lack of these types implies that some functions of R4RS are not defined.

Some types which are not defined in R4RS are implemented in STk. Those types are listed below:

- input string port type
- output string port type
- keyword type
- Tk command type (to use the Tk widgets)
- environment type (environments are first class in STk)
- macro type
- address type (obsolete now)
- hash table type
- process type (launch and communication with subprocess)
- socket type (TCP/UDP sockets)
- regexp type (access to tcl regexp)

Procedures

The following procedures are required by R4RS and are not implemented in the STk interpreter:

- transcript-off
- transcript-on

These procedure can be easily simulated with various Unix tools such as *script* or *fep*.

The following procedures are not implemented in the STk interpreter whereas they are defined in R4RS (but not required). They are all related to complex or rational numbers:

- numerator
- denominator
- rationalize
- make-rectangular
- make-polar
- real-part
- imag-part
- magnitude
- angle

This is more detailed in the Appendix B of the STk Reference Manual.

1.9 Subject: Compatibility with SLIB

Aubrey Jaffer maintains a package called **SLIB** which is a portable Scheme library which provides compatibility and utility functions for all standard Scheme implementations. To use this package, you have just to type

```
(require "slib")
```

and follow the instructions given in the library to use a particular package.

2 Contributed Packages

Small (in size) contributions are included in the release. Among them we can cite:

- *STk-wtour*: A didactic demo to learn **STk** written by Suresh Srinivas (ssriniva@cs.indiana.edu)
- *Stetris :* A tetris game written by Harvey J. Stein (hjstein@math.huji.ac.il)

- *Queens*: The famous 8 queens problem by Grant Edwards (grante@winternet.com)
- *Ttt*: A 3-D Tic-Tac-Toe by Edin "Dino" Hodzic ehodzic@scu.edu

Complex contributions are kept in the ftp://kaolin.unice.fr/pub/Contribdirectory. This directory contains also old code which should probably be updated to work with current STk release. Among the packages which can be found in this directory are:

- *EnvDraw*: is a environment diagramming package which draws environment diagrams as taught in Abelson and Sussman's *Structure and Interpretation of Computer Languages*. It was written as an instructional tool for the CS61A course at the University of California at Berkeley. The environment diagrammer is a metacircular evaluator which draws procedures, environments, and box and pointer diagrams along with all the accompanying symbols and mutation. It includes a box and pointer diagrammer which handles circular list structures, cons cell mutation, and also will watch for modification of any symbols known to be pointing to drawn cells. The author of this very neat package is Josh MacDonald jmacd@po.EECS.Berkeley.EDU
- *WWW* is a HTML browser which greatly enhances the hbrowse demo which is part of STk distribution. You should look at this package. Author: Harvey J. Stein hjstein@math.huji.ac.il
- *Nanocad* is chemistery CAD program which allows you to create small molecules and to rotate them. The author of this package is Will Ware wware@world.std.com
- ...

Some packages are not in the `kaolin Contrib` directory:

- A preemptive thread package can be found at ftp://ftp.cs.uoregon.edu/pub/lth/tasking-alpha2.tar.gz The uthor of this package is Lars Thomas Hansen lth@cs.uoregon.edu
- A low level interface to the Postgres 95 Database can be found at http://www.webbuild.com/~rtoy/pgstk/http://www.webbuild.com/~rtoy/pgstk/ .
- *Tabula rasa* is a system written by David Fox which use the STk interpreter (but not Tk).This package can be found at ?????. (TO BE COMPLETED)
- spice package from Fritz.Heinrichmeyer@fernuni-hagen.de (I cannot find the reference)
- ...

3 STk tips

TO BE COMPLETED

4 STklos tips

TO BE COMPLETED

5 Getting more information about Tk

There are two reference book about Tk (of course, you better not read the chapters about Tcl ;-):

- "Tcl and the Tk Toolkit" by John Ousterhout (the author of the Tcl/Tk package), ISBN 0-201-63337-X, Addison-Wesley, April 1994.
- "Practical Programming in Tcl and Tk" by Brent Welch, ISBN 0-13-182007-9, Prentice Hall, 1995.

There is also a Usenet newsgroup for Tcl/Tk: comp.lang.tcl A FAQ, for Tcl/Tk is regulary posted in comp.lang.tcl.annonce

6 Miscelleanous

6.1 Subject: The STk Mailing List

There is a mailing list for STk located on `kaolin.unice.fr`. The intent of this mailing list is to permit to STk users to share experiences, expose problems, submit ideas and everything which you find interesting (and which is related to STk).

To *subscribe* to the mailing list, simply send a message with the word `"subscribe"` in the `Subject:` field of you mail. Mail must be sent to the following address: `stk-request@kaolin.unice.fr`.

To *unsubscribe* from the mailing list, send a mail at previous e-mail address with the word `"unsubscribe"` in the `Subject:` field.

For more information on the mailing list management, send a message with the word `"help"` in the `Subject:` field of your mail. In particular, it is possible to find all the messages which have already been sent on the STk mailing list.

Subscription/un-subscription/information requests are processed automatically without human intervention. If something goes wrong, send a mail to eg@unice.fr

Once you have properly subscribe to the mailing list,

- you can send your messages about STk to stk@kaolin.unice.fr
- you will receive all the messages of the mailing list to the e-mail address you used when you subscribed to the list.

The mailing list is archived at:

- http://kaolin.unice.fr/mlist/Hypermail/index.htmlin hypertext format, or
- ftp://kaolin.unice.fr/pub/MailingList/index.htmlas raw text.

6.2 Subject: Reporting a Bug

When you find a bug in STk, please send its description to the following address stk-bugs@kaolin.unice.fr. Don't forget to indicate the STk version you use, the architecture and the system release on which you run the interpreter. STk version and architecture can be found by using the `version` and `machine-type` Scheme primitives. If possible, try to find a small program which exhibit the bug. Even better, if you have a patch, plese send it at the previous address.

6.3 Subject: Using STk from Emacs

There are several ways to call `STk` from emacs. One simple way consists to add the following lines in your file `$HOME/.emacs:`

TO BE COMPLETED (with ILISP)