NAME

stk, snow – A Scheme interpreter using the Tk toolkit

SYNOPSIS

stk [*options*] [*arg arg ...*]
snow [*options*] [*arg arg ...*]

OPTIONS

stk interpreter accepts several command-line options which may be abbreviated, as long as the abbreviation is unambiguous (e.g. **-h**, **-he**, **-hel** are three possibe abbreviations for the **-help** option).

–colormap new Specifies that the window should have a new private colormap instead of using the default colormap for the screen.

–display *display* Display (and screen) on which to display window.

–file *fileName* Read commands from *fileName* rather than standard input. The last element in *fileName* will be used as the title for the application and name of its interpreter for **send** commands (unless overridden by the **–name** option).

–geometry *geometry*
Initial geometry to use for window.

–name *name* Use *name* as the title to be displayed in the window, and as the name of the interpreter for **send** commands.

–sync Execute all X server commands synchronously, so that errors are reported immediately. This will result in much slower execution, but it is useful for debugging.

–visual *visual* Specifies the visual to use for the window.*Visual* may be one of the following: *best*, *directcolor*, *grayscale*, *greyscale*, *pseudocolor*, *staticcolor*, *staticgray*, *staticgrey*, *truecolor*, or *default*

–no-tk Don't initialize the Tk toolkit

–load *fileName* Evaluate expressions contained in *fileName* before reading expressions from standard input.

–image *fileName* Restore the state saved in *fileName* by the *dump* primitive (Note: For now, *dump* works only on SunOS 4.1.x, Linux 1.x (a.out format) and FreeBSD).

–cells *number* Set the default size for the heap to *number* cells. The given number is also the amount of cells used when extending the heap. Default value is 20000.

–interactive Tell the interpreter that it is used interactively (even if it doesn't think so).

–help Print a summary of the command-line options and exit.

– – Pass all remaining arguments through to the script's **argv** variable without interpreting them. This provides a mechanism for passing arguments such as **–name** to a script instead of having **wish** interpret them.

DESCRIPTION

Stk is a Scheme R4RS interpreter which provide a simple access to the X11 Tk toolkit. If the **–no-tk** option is provided to the interpreter, the Tk library is not initialized and no main window is created. If **stk** is invoked with no **–f** option then it reads Scheme expressions interactively from standard input. It will continue processing commands until all windows have been deleted or until end-of-file is reached on standard input.

Snow is a *light* version of the **stk** interpreter which does not provide support for the Tk toolkit. This interpreter does not recognize the options **−display**, **−geometry**, **−sync -colormap**, **-visual** and **−no-tk** which are meaningless without Tk. This interpreter is called, rather than the standard one, when the shell DISPLAY variable is not initialized.

If the **−file** option is provided to Tk, then **stk** reads Scheme forms from the file named in the **−file** option. These forms will normally create an interactive interface consisting of one or more widgets. When the end of the command file is reached, **stk** will continue to respond to X events until all windows have been deleted.

Note: The **-file** exits for compatibility reasons and can be eventually ommited. In there is no **-file** option used, the first argument in the command line is taken as the name of the script file to execute. To launch an interactive script with an argument, you have to use the **--** option.

The **−interactive** option forces the interpreter in interactive mode. In this mode, standard output and standard error are unbuffered. Use this option when you launch **stk** from **emacs**.

VARIABLES

Following Scheme variables are set when **Stk** starts:

argc Contains a count of the number of *arg* arguments (0 if none), not including the options described above.

argv Contains a Scheme list whose elements are the *arg* arguments (not including the options described above), in order, or an empty list if there are no *arg* arguments.

program-name Contains *fileName* if it was provided (in a **-file** option or as first argument of the command line). Otherwise, contains the name by which the interpreter was invoked.

SCRIPT FILES

If you create a STk script in a file whose first line is

```
#!/usr/local/bin/stk -f
```

then you can invoke the script file directly from your shell if you mark it as executable. This assumes that **stk** has been installed in the default location in /usr/local/bin; if it's installed somewhere else then you'll have to modify the above line to match. Note that **-f** which was necessary with pre-4.0 versions of **STk** is no more necessary and can be omitted.

ENVIRONMENT VARIABLES

stk uses the following shell variables:

STK_LIBRARY This variable indicates where the library files are located. This variable allows to overload the default value of the Scheme variable **stk-library** which is automatically calculated by the interpreter.(i.e. *stk* or *snow*).

STK_LOAD_PATH
This variable serves to initialize the **load-path** Scheme variable. This variable is a list of paths to try when in a load command.

STK_HELP_PATH
This variable serves to initialize the **help-path** Scheme variable. It must contain a list of documentation directories. This variable is used by the help procedure.

STK_IMAGE_PATH

This variable serves to initialize the **image-path** Scheme variable. It must contain a list of images directories. This variable is used by the make-image procedure.

FILES

./init.stk

$STK_LIBRARY/STk/init.stk

When **stk** starts running, it tries to load the file *init.stk* in the current directory. If this file is not present, it tries to load this file in the **STk_LIBRARY** directory (the value of **STk_LIBRARY** is automatically computed to be in a sibling directory of the interpreter executable.

~/.stkrc Standard *init.stk* file tries to load the file *~/.stkrc*. This file can be used to store functions definitions or variables settings that you want to be executed at each interpreter invocation.

SEE ALSO

wish(1)