

1 SM's Release Notes

This is the fifth-and-a-half distribution of SM¹, but it might still have a few problems. As usual, the authors are actively using SM and it is pretty reliable. (If you don't agree send mail! You may be able to do this with the `gripe` command).

These Release Notes should include all the changes to SM since version 1.2, but do not detail all of the bug fixes.

¹ Why is it called SM? There is a package called Mongo with a similar command syntax that is in common use in Astronomy departments around the US. Mongo was used as a model for SM, and that's where the 'M' came from.

Changes Prior to Version 1.2

- Changes to `BOX` to write labels correctly. Two separate problems fixed (`LIMITS -5 5 0 0.2 TICKSIZE -1 0 0 0 BOX` should now work).
- A fix to stop x-axes having jiggles on some devices.
- Miscellaneous changes to the `imagen` driver.
- Changes to `stdgraph` support a new postscript entry.
- A new postscript `graphcap` entry, which should produce smaller files.
- Support for coloured lines, in drivers and `graphcap`.
- A Sunwindows driver, `dev sunwindows`, used with `gfxtool`.
- An additional variable syntax, `DEFINE name |`, to allow access to SM internal variables like `expand`.
- Restrictions on the length of macros that can be edited have been removed. In addition, the editor now starts on the first line of macros (not the header line). The history buffer is still 80 lines long.
- A new edit character `CONTROL-Q` to quote the next character.
- An interface to `fortran`, and documentation of the C calls that SM uses, to allow programmes to link to the SM libraries.
- An option to specify the number of lines on your terminal in the `TERMTYPE` command.
- A change to the default `LOCATION`, the better to fit x-axis labels.
- A shell escape (pass a command to the operating system), and a change directory command (`!` and `CHDIR`).
- A table of contents and index to the manual.

Changes from Version 1.2 to Version 2.0

- Fixed unix version of SM to accept redirected input. Who knows for VMS?
- Command line is read differently. The old behaviour (read one argument, treat it as a macro file, read macro file, strip path/suffix and attempt to execute) is preserved as the `-m` option. Otherwise, simply execute command line after the startup macro
- Added logfile option to command line (`-l logfile`), mostly for debugging
- Allowed `FOREACH var { list1 } { list2 }`, to allow keywords to appear in `list1`
- Fixed Interrupt handler to return control to user if requested
- Made spaces (and tabs) within quotes non-special, so they can appear as parts of single words
- New utility `get_grammar` written in C to extract grammar from yacc files
- Added support for `~` in `CHDIR`
- Made default verbosity 1 not 0
- Made `PRINT` page if printing to terminal
- The header lines written by `PRINT` now start with a `#`, so they can be read without using the `LINES` command
- Added commands `SAVE` and `RESTORE` to save and restore a complete session
- Replaced `SHOW` by a macro
- Fixed `READ OLD` to change `ERRORBAR` and `WINDOW` to new formats
- Allowed the use of `\` to continue long lines
- Columns of data may now be separated by commas as well as white space
- Removed all restrictions on the number of columns in a data file
- `HELP` gives the values of variables and vectors, as well as macros
- In the unlikely event of SM detecting an internal error (e.g. a segmentation violation) a `SAVE` file called `'panic.dmp'` is now written to your current temporary directory
- Strings in double quotes are now treated as words (e.g. `' '2.80 ' '`, note space following 0)
- `TERMCAP` variable is treated as the actual termcap
- Added explicit copyright statements to the code, to clarify its status as not being public domain, but the property of its authors
- Added `WHATIS` builtin function, to determine type of a string

- Added **OVERLOAD** to allow overloading of keywords, also added `set_overload` macro to set some of them up. “compatible” also makes use of this facility to redefine commands such as `points`
- Made `do i=1,10 { if($?$i) { ... } }` as expected
- Allowed TeX-style labels (if `$TeX_strings` is defined)
- Allowed `points` (of specified angle and `ptype`) and `lines` (of specified `ltype`) to appear in strings, if you are using TeX labels
- Allowed you to declare an error handling macro (called `error_handler`) that is called after `CONTROL-C` interrupts
- `RESTORE` can be used to read history files
- You can now specify lists of file/graph/termcap files to be searched in order, allowing private modifications string if it can't be opened as a file
- Changes to do with history and the editor:
 - Made `TERMTYPE` with a negative number of lines turn off cursor motion
 - Changed default bindings to make `ESC-q` `quote_next` (in addition to `CONTROL-Q`), and `CONTROL-}` an emergency `CONTROL-M`
 - Changed editor to use termcap
 - Changed editor to follow columns past a blank line
 - Made history ignore consecutive identical commands
 - You can now repeat a search to look for a command further back in the history list (use `^TAB`)
 - Changed paging of history list to allow you to search (use `?` while paging to find out how)
 - Changed terminal read routine, so `CONTROL-S/CONTROL-Q` and `typeahead` should now work at least under Unix. (À la unix: characters appear as you type. Refresh the screen (`CONTROL-L`) if it gets scribbled on)
 - The history system was deleting commands when the difference between the top and bottom numbers in use exceeded 80, rather than when there were more than 80 commands on the buffer. Fixed this, and took the opportunity to take the size of the history buffer from the `‘.sm’` variable `‘history’` (default : 80)
 - Added command `HISTORY -` to list history in backwards order
 - allowed you to move deleted words from line to line (`CONTROL-[h/CONTROL-[u` etc.)
 - made a `TERMCAP` environment (logical) variable override a termcap in `‘.sm’`

- Renumbered lines with every **HISTORY** command. Insert should now work
 - ESC-y will successively restore recently deleted lines, up to a current limit of 5.
- Changes to do with macros:
- Macros beginning **##** are not listed if **VERBOSE** is 0.
 - Added command **RETURN** to return from a macro, or to the prompt if encountered outside a macro. (Useful when fiddling with command scripts)
 - We now document that lines beginning **#** in macro files are comments, and are output for verbose > 0. Lines beginning **#** in data files now also comments, printed for verbose > 1, lines starting **!** are always printed. Note also that lines ending **** are logically continued to next line in data files, as well as to the command language.
 - Macros may be declared to have variable numbers of arguments
 - When editing macros, **RET** in the middle of a line splits the line at that point. The behaviour of the history editor is unchanged. We now delete blank lines as we go, so typing many **RET**'s will just get the 1 line, and it'll go away if you **CONTROL-P** or **CONTROL-N** without typing something
 - If a macro has a -ve number of args in **MACRO EDIT**, it is deleted
 - If **VERBOSE** is 3 only trace macros as they are executed, **VERBOSE 4** is the full trace
 - Added command **KEY** to define keyboard macros. The keys PF1 - PF4 may be specified in a terminal independent way.
 - Allowed you to redefine macro arguments inside macros, including ones that weren't declared. This allows you to use numbered variables as temporaries.
 - Allowed **MACRO name { DELETE }** to delete macros with an uppercase "DELETE"
 - Empty macros are deleted upon leaving the macro editor
- Changes to do with variables:
- Added more macros to utils, for instance **arrow**, **wilcoxon**, and **smirnov**
 - Changed **DEFINE var ?** to allow prompt string The value of the variable now consists of the complete line typed, rather than just one word. You are now offered as a default value the old value, if it is defined
 - Added **uxp** and **uyp** for user plot position to **DEFINE var | rule**.
 - Made variable names being defined appear on error traces
 - Added variable **\$date** with value of the current date and time
 - Allowed you to redefine macro arguments inside macros, including ones that weren't declared. This allows you to use numbered variables as temporaries.
- Changes to do with vectors:

- allowed `< ... >` as well as `{ ... }` for `PRINT` and `SORT`
- Added `SET DIMEN(WORD) = expr` to declare `WORD`, and initialise it to zero. Also a macro `declare` to do this for you
- `{ list }` is now an expression
- Allowed `READ < list >` as well as `READ { list }`
- Changed verbosity in vector arithmetic to report first error in a call for `VERBOSE > 0`, but only report further errors for `VERBOSE > 1`
- Added new operator `INT(expr)` to take integral part of an expression, can be used for rounding.
- Changed precedence of `CONCAT` to just below `'+'` and `'-'`
- allowed `< list >` as an expression
- Allowed `v1` and `v2` to both be scalars in `?:` statement
 - Changes to how plotting is done are as follows:
- Stopped `SM` changing between graph and alpha modes so often, flashing the screen
- Changed behaviour of `CTYPE` commands, and treatment of colour in `graphcap`. It is now possible to specify a private set of colours for use in drawing lines
- Added support for plotting to terminals other than the one that your are logged onto
- Changes to prevent plotting at above device resolution
- Added command `PTYPE { list }` to plot a user defined symbol at points
- Wherever the name of a vector is expected you can now use an expression in parentheses
- Added support for aspect ratios in drawing characters
- Allowed `DRAW (num num)` by analogy to `RELOCATE (num, num)`
- Fixed `ERRORBAR` to accept scalars for size of errors
- Redefined `IDENTIFICATION` as a macro
- Added `ILABEL 3` to `BOX`, to omit both ticks and labels, useful in conjunction with `AXIS`
- Added test in `BOX` for arguments to logarithmic axes out of range
- If `ANGLE` is non-zero, it is used for `XLABEL` and `YLABEL` (e.g. `ANGLE 360 YLABEL N`)
- Added `axis()`, `curs()`, `gflush()`, `plotsym()`, and `shade()` to `interface.c`
- Changed `ticksize` in `interface.c` to take real arguments, not integer
- `WINDOW` now ignores initial value of location and attempts to fill (almost) entire screen with little windows

- WINDOW can be used to draw boxes that touch
- Changed playback to allow you to optionally specify the line or range of lines to be played back, and similar changes to hcopy.
- If the two limits given for an axis in the LIMITS command are the same, the limits for that axis are not changed.
- When using the RANGE command, LIMITS now attempts to centre the axes on the median of the data.
- Can now shade inside a region, or shade a histogram (see SHADE).
- Changed fonts somewhat. Old characters in old order available as old_fonts.bin (in '.sm' file), added Old English font (\o), changing font sizes inside strings (\digit), and a number of new characters.
- Modified filecap to understand FITS data files for contouring.
- In a label, (e.g. PUTLABEL), @\ is printed as a real \.
- Added ASPECT command, to allow you to override SM's choice of aspect ratio (e.g. to position labels for a plot that will finally appear on a laser printer)
- Fixed WINDOW to allow boxes to touch. The macros wind and bo take touching into account when labelling axes.
- made histograms draw 'flanges' at ends (as had been done historically)
- Changes to stdgraph:
 - Changed stdgraph to accept %T for 12-bit encoded Tektronix (Also changed graphcap to reflect this: xr, yr, XY, TB, SC)
 - Removed RT capability from graphcap, and changed stdgraph to use ch and cw (the height and width of the hardware characters)
 - Added DT to graphcap to deal with funny requirements of devices
 - added support for raster devices (e.g. printronix). These actually use a combination of graphcap and a stand-alone rasteriser
 - Added support for area fill in graphcap for PTYPE n 3
- Changes to drivers
 - The postscript drivers are now available landscape, postscript, or square (and work over a network for Unix users)
 - The sunwindows driver now supports a cursor
 - CONTROL-C interrupts now work even if you are using sunwindows
 - Mouse buttons work with the X10 cursor

- There is a new SunView driver (device sunview) that creates its own graphics window.
Bug fixes: (only the ones most visible to users are noted)
- Fixed bug with PTYPE n 3 for angle != 0
- Fixed bug in CONTOUR, to do with 2 lines in one plot cell; previously always assumed lines SW-NE
- Fixed bug in CONTOUR, that could lose last row/column of contour plot
- Fixed occasional bad axis labels
- Fixed rounding error in drawing ticks
- Fixed contouring fortran files
- Fixed precedences of logical and and or
- Fixed bug in CONTOUR to do with point at (or beyond) right hand size of image
- Miscellaneous postscript problems fixed

Real, Inconsistent, Changes from V2.0 to V2.1

- The programme's name has been changed to SM, that the default directory name is now 'sm', and that the '.smongo' file is now called '.sm'.
- Typing 'make install' now installs the macro and help libraries. The destinations of some files is now different – see the main Makefile for details.
- The compiled graphcap entry format has been changed (fixed); you must rerun 'compile_g'.
- The command SET name = macro(vector) has been changed. Previously only one argument was permitted, but it could be an expression, and the result was returned by assigning to \$1 in a rather confusing way. Now many arguments are permitted (but they can't be expressions), and the result is returned as \$0. See SET in the manual for more details.
- the SET DIMEN(vec) = expr statement has been altered, so the right-hand expression must be a number (with an optional suffix to denote a string-valued vector).
- The way you specify limits for logarithmic axes to AXIS is now consistent with the way you specify them to BOX.
- The 8th bit of graphcap entries is no longer stripped. To include an ascii null (\000) in an entry specify \377, a \377 may be written \377\377.
- You must now include the library utils.a in addition to plotsub.a and devices.a when linking to SM as a graphics library.

Changes from Version 2.0 to Version 2.1

Note that the programme's name has been changed to SM, that the default directory name is now 'sm', and that the '.smongo' file is now called '.sm'.

Changes to do with making SM:

- Compilation options are now in a file src/options.h rather than in the Makefiles.
- The compiled graphcap entry format has been changed; you must rerun 'compile_g'.

Changes to do with history and the editor:

- Syntax errors now report the macro in which they occur. Beware that this may be incorrect, if the macro has already been fully scanned.
- If VERBOSE is 1 or more, report the line that arithmetical errors occur on (e.g. mismatched dimensions, lg(0)).
- You can now choose your history character (default: ^). This may be useful for T_EX strings, I use '.
- You can go to the start or end of a macro/the history list with ESC-< and ESC->. Use ESC-g to go to a line number.
- You can no longer type a space after the history character to insert it into a line – use quote_next instead (e.g. ESCq^).
- Lines starting with a # in READ EDIT files are comments
- In '.sm' files comments run from a # to the end of the line.
- The various DELETE HISTORY commands now have siblings that omit the HISTORY and do *not* themselves appear on the history list.
- The way that DO, FOREACH, IF, and MACRO command lists are treated has been fixed. If VERBOSE is 5 or more such lists will be printed as they are scanned.

Changes to do with macros:

- The command APROPOS lists all macros whose names or initial comments match a given pattern.

Changes to do with variables:

- Variables given in <> or {} are no longer tokenised, so spaces will no longer appear in funny places.

Changes to do with vectors:

- You can now specify a printf-style format string to PRINT
- *The command SET name = macro(vector) has been changed.* Previously only one argument was permitted, but it could be an expression, and the result was returned by assigning to \$1 in a rather confusing way. Now many arguments are permitted (but they can't be expressions), and the result is returned as \$0. See SET in the manual for more details.
- It's possible to FFT vectors (actually pairs of vectors for the real and complex parts).
- String valued vectors have been added to SM. They may be read from files (either by the row or by the column), printed, assigned to, tested for (in-)equality, and used to set the PTYPE. In the latter case the strings are used to label the points.
- the SET DIMEN(vec) = expr statement has been altered, so the right-hand expression must be a number (with an optional suffix to denote a string-valued vector).
- You can now define variables from IMAGE files (currently only supported for FITS), using DEFINE variable IMAGE
- The expression vector[expr] has been generalised to allow expr to be a vector, so the result is now a vector of the same dimension as expr.
- You can convert an expression into a string using STRING(expr).
- It is now possible to specify that IMAGE files have no header, in which case you'll be prompted for the values of nx and ny (file_type no_header).

Changes to how plotting is done are as follows:

- The values of TICKSIZE now make a difference for logarithmic axes.
- The way you specify limits for logarithmic axes to AXIS is now consistent with the way you specify them to BOX. *This Is A Change!*
- GRID now takes an optional second argument, specifying only an x or y grid
- LTYPE ERASE allows you to erase some lines in a plot without erasing everything. This is only supported by some devices
- ANGLE and EXPAND can now be given as vectors, to specify different angles and expansions for each point plotted
- DOT will now work with user-defined ptypes
- You can specify a PTYPE of a vector of strings.
- The order of the default CTYPEs has been changed; you can now restore the default using the macro reset_ctype.

Changes to labels:

- T_EX strings now makes A_a^b put both sub- and super- scripts on the same symbol (use A_a{^b if that isn't what you want)
- A new bold (over-struck) `\bf` font is available.
- If you define the variable `default_font` it will be used as your default font for labels (including for axes). You can define it in your `.sm` file if you so desire.
- The position of the symbol for the constellation Taurus has been moved; this will not affect TeX_strings users. An Old English full stop is now a full stop.
- 'T_EX definitions' are now specified in a file (and put into fonts.bin). This will not affect users, but does allow a SM administrator to add new definitions.
- You can use `\def` to define your own T_EX definitions.
- You can specify more than 9 sides in T_EX 'point' or 'apoint'
- The 'T_EX' commands `\raise` and `\kern` are now available to move parts of strings about.
- T_EX-labels can now be put into variables without difficulty.

Changes to stdgraph:

- A new encoder command `'str'` prompts you, then reads a character from the keyboard.
- You can specify graphcap entries (or parts of graphcap entries) on the command line.
- The 8th bit of graphcap entries is no longer stripped. To include an ascii null (`\000`) in an entry specify `\377`, a `\377` may be written `\377\377`.

Changes to drivers

- You can now specify options like `-Ws` to the sunview driver, use `-WH` for a list.
- There is a new driver for a Unix-PC.
- There is a new X11 driver (try `DEVICE x11 -help` for a list of options).
- If you write your own drivers, the declarations of some functions have been changed to return void.

Changes to Callable SM

- You must now include the library `utils.a` in addition to `plotsub.a` and `devices.a` when linking to SM as a graphics library.
- The library names have been changed by pre-pending a 'lib', so you can now say either 'libdevices.a' or '-ldevices'. This allows the `-L` option to `ld` to work (under Unix).

- If you compiled SM with an ANSI compiler, you should include the file "`sm_declare.h`" at the top of any C code that calls SM functions.

Bug fixes: (only the ones most visible to users are noted)

- You can now define arbitrarily big vectors with the cursor.
- LOCATION now works with non-touching WINDOWS.

Changes from Version 2.1 to Version 2.2

Changes to do with making SM:

- Typing ‘make install’ now installs the macro and help libraries. The destinations of some files is now different – see the main Makefile for details.
- There is now a search path for ‘.sm’ files; you may want to prepare a system-wide one (see `SMPATH` in the manual).

Changes to do invoking SM

- SM now searches a path looking for .sm files; you can override this by setting the `SMPATH` environment (VMS: logical) variable.
- The command line option `-u name` makes SM treat ‘~’ as `name`’s home directory when looking for ‘.sm’ files.
- There is a new command line option `-h` to print a help message.
- SM now supports a search path for help files.

Changes to do with .sm files:

- Entries in .sm files with ‘@’ in the first column mean that the entry isn’t present, and that SM shouldn’t search the rest of the path.
- Entries in .sm files with ‘+’ in the first column direct SM to keep looking for more entries, whose values are concatenated with this one.

Changes to do with history and the editor:

- There is now a macro `input_new` analogous to the Mongo “input” command, but reading a file of SM commands. Not recommended.
- There are two new commands `CONTROL-R` and `CONTROL-S` (or `ESC-s`) to do searches in macros or history. The use of `^-TAB` is no longer supported.
- There is a new command `ESC-CONTROL-D` to delete the current line; this includes deleting it from the history list.
- If you set the variable `remember_history_line` the history editor will remember where you are in the history list, so a `CONTROL-P` will get the command before the last one that you retrieved.
- Negative history numbers are now taken relative to the current command, so `-1` is the previous command.
- `APROPOS` will (on most systems) search the help files as well as the macro headers.

- When reading from a file (or a pipe) SM will behave as if you gave the `-S` option.
- Key bindings may now contain ASCII NUL's.
- Shell escapes (with `!`) now look for a variable `$SHELL` to decide which shell to use; the Bourne shell is still the default. They also now reset terminal modes before the shell command, and set them after.
- There is a new variable `exit_status` which is the return code from a shell escape.
- You can now use `CONTROL-V` and `ESC-v` to move around the history list as well as within macros.

Changes to do with files:

- If you define the variable `noclobber` (maybe in your `.sm` file) `MACRO WRITE` and `PRINT` won't overwrite existing files.

Changes to do with macros:

- If a macro is declared to have 100 or more arguments the last argument will extend to the end of the line.
- Tabs in macro definitions are now expanded to spaces when read; you can set the tab width with the `.sm` variable `tabsize`.
- Macros used as functions can now be used in expressions.
- In a file of macros, lines starting `##` are only printed if `VERBOSE` is two or more.

Changes to do with variables:

- If a `"DEFINE var :"` fails, if `var` is capitalised SM will now look for it from the environment.
- The way that 'internal' variables (those defined as `DEFINE name |`) work is changed. If you *don't* define them they will always have the correct value — see the section on `DEFINE` for further explanation.
- You can get at the size of an image with `DEFINE NX IMAGE` and `DEFINE NY IMAGE` (note that `NX` and `NY` are capitalised).
- There are two new 'internal' variables `$nx` and `$ny` which give the current size of the screen in hardware-specific units such as pixels or dots, and `$cheight` and `$cwidth` which give the current (average) size of a character in plot units.
- There is a new special form of variable expansion `$(expr)` which expands to the result of `expr` (taken to be a scalar).

Changes to do with vectors:

- You can now specify names for user ctypes, e.g. `CTYPE={ puce khaki vermilion }`.
- You can make the `PRINT` command append to the output file, rather than simply overwriting it.
- In response to overwhelming public pressure, an element of a vector may once more be specified wherever a number is allowed (e.g. in a `RELOCATE` command).
- You can now write a number as `1.234d56` if you must.
- There is a new operator `ATAN2` that takes two arguments and returns an answer in the correct quadrant.
- `STRING` will now (by default) produce nicely formatted strings, similar to those used to label axes.
- You can now declare an image (with `IMAGE (nx, ny)`), and fill it with `SET IMAGE(x,y) = z`.
- There are some more string operators, similar to those in the new Unix `awk`, `INDEX`, `SPRINTF`, `STRLEN`, and `SUBSTR`.
- You can determine how long a string is (in device coordinates) with `LENGTH`.
- You can specify your own format statement for `SM` to read lines of data with: `READ 'format' { vectors }`.

Changes to how plotting is done are as follows:

- The `hcopy` and `hmacro` macros will now work even if the printer name is omitted (using the value of `PRINTER` from `‘.sm’` or the environment instead).
- ‘Dotted’ lines (`ltype 1`) are now really dotted, not short lines.
- You can now start a new page for hardcopy devices with the new command `PAGE`.
- The new command `ABORT` closes the current device without making a hardcopy.

Changes to labels:

- `SM`’s “`TEX`” macros can now take arguments
- There is a new pseudo-macro `\move` to move a group around, but leave the current position unchanged.
- There are new pseudo-macros `depth`, `height`, and `\width` to find the dimensions of a group.
- The height and depth information is used to place axis labels more intelligently. This will only work if you draw the axes before the axis labels, which seems not unreasonable.

- LABEL will now draw a string with its **baseline** at the starting point. If you want the old behaviour use PUTLABEL 6.

Changes to stdgraph:

- :RA: has been retired in favour of the much more general :DV=device:.
- The new :TC: entry is like :tc: but doesn't start scanning graphcap again. This can be used to modify standard entries.
- Comments are now permitted within graphcap entries.
- SY entries can now use "\$..." to prompt the user for a string; the "\$..." will be replaced by the reply.
- SY entries now support graphcap variables, so that you can say things like


```

      BIN = /usr/local/bin
      device|some device:\
          ...
      :SY=${BIN}/rasterise -r $0 $F $1:
      
```

Changes to drivers:

- The imagen driver now supports portrait, landscape, and square plots.
- There is a new driver (`sgi`) for Silicon Graphics's Irises, which should also work on IBM's RS/6000.

Changes to Callable SM:

- The device call now returns 0 for success, -1 for failure.
- You can now compile SM to prefix external names with "sm_" (or any other string of your choice).
- There is a new call `redraw` that should be used on window systems that need some help redrawing their windows (e.g. SGI's GL, or X11 on machines like RS/6000 that don't support backing stores).

Changes to 2-D SM:

- If you specify `nx` or `ny` to be negative they will be taken to be the negation of the actual size of the image, rather than an offset in the file.
- If you specify `nx` or `ny` as *string* values (e.g. `:nx=10:ny=100:` they are taken to be the actual size of the image (just like the previous item).
- You can get at the size of an image with `DEFINE NX IMAGE` and `DEFINE NY IMAGE` (note that `NX` and `NY` are capitalised).

- You can now declare an image (with `IMAGE (nx, ny)`), and fill it with `SET IMAGE(x,y) = z`.
- You can define any (2-letter) name in `filecap` and retrieve the value at the specified offset with `DEFINE nn IMAGE`.
- In FITS files, `CRPIX`, `CRVAL`, and `CDELTA` are now interpreted correctly.

Bug fixes: (only the ones most visible to users are noted)

- The problems with calling SM routines have been fixed.

Changes from Version 2.2.0 to Version 2.3.0

- When reading from a file (or a pipe) SM won't trap CONTROL-C
- Commands are now put on the history list even if you run with `-s` or `-S`.
- The history editor is now active when you have entered a partial command (e.g. `macro hello {`). There is a different prompt, which you can change by defining `$prompt2`.
- The primary and secondary prompts can be set by specifying `prompt` or `prompt2` in your `.sm` file.
- In a graphcap SY string, if a positional parameter is called (e.g.) `$(3)` instead of `$3` there's no complaint if it is missing.
- You can now say `!expr` just like in C
- Two new 'number' operators in T_EX strings, `\multiply` and `\divide`. Both multiply (or divide) by the argument/1000.
- You no longer need the parentheses in `vec[(1+2)]`. This might be thought of as a bug fix.
- A side effect of the previous item is that whereas `define v vec[0]` used to be accidentally legal, you must now say `define v (vec[0])`.
- LWEIGHT is now a floating point number
- The graphcap encoder now has limited support for floating point numbers; specifically you can use `'%e'`, `'$f'`, or `'%g'` to format the bottom of the stack, and `'l'` to convert the bottom of the stack to int.
- New `'l'` variables `swidth`, `sheight`, and `sdepth` give the (pixel) size of the last string drawn.
- `PUTLABEL 0 str` doesn't draw anything, but does set the string dimensions (and any T_EX definitions are remembered).
- Commands that draw strings now leave the plot pointer on the baseline at the right of the last character drawn.
- You can give `AXIS` vectors of values where you want the small and big ticks, thereby finally relieving us of the need to make `TICKSIZE` infinitely complicated. You can even provide a string-valued vector to specify the labels that you want written.
- There are two new SM variables `x_gutter` and `y_gutter` that are used to modify the spacing between boxes left by the `WINDOW` command.
- You can now specify a font size along with a type in `default_font`, e.g. `\ti\bf\it\2`.
- `WRITE` can now append to a file by saying `WRITE + filename`.

- SUM is now implemented using a double precision intermediate value.
- There is a new operator RANDOM to generate uniformly distributed random numbers. The seed can be set with SET RANDOM.
- Any combination of string- and arithmetic-vectors is now allowed in SORT.
- If you define \$save_read_ptr, the current READ pointer is saved, allowing faster sequential access to a file.
- You can specify a range of columns to be used for reading numerical vectors.
- Macro files may contain left-justified comments within macros which are not included in the macro, but processed at read time.
- The hidden surface code (VIEWPOINT and SURFACE) now works. The C is based on a set of SM macros written by Gabor Toth.
- There are a set of SM macros for use with the new surface commands; say load surfaces if you want to use them.
- If the environment file specified by -f filename on the command line starts with a /, no search path is used. Note that this means that you can't inherit entries from other '.sm' files.
- Argument parsing for DEVICE commands now understands quotes, e.g. DEVICE x11 -title \"Hello World\".
- You can now call the SM parser from your own code. There is also a function to declare your arrays to SM as vectors. Examples have been added to the 'callable' directory.
- If VERBOSE is one or more, you'll be warned if the axis labels are expected to overlap the ticks (and a solution is suggested).
- If a macro isn't found, the macro macro_error_handler is called with 2 arguments; the first is NOT_FOUND and the second the name of the macro. If the variable missing_macro_continue is defined (maybe in your '.sm' file) the behaviour is unaltered.
- There is a new command USER ABORT that forces SM to think that a syntax error has occurred.
- If you retrieve that last history command with ^\$ or ^^ you will always get the last command, even if it doesn't appear on the history list; using the arrow keys or ^P still gets you the previous remembered command.
- There are macros to support greyscale plots (glevels and greylevel)
- The vms_var filecap entry has been changed, and so has the documentation on how to write out image files under VMS Fortran. For variable recordtype files, you must specify recordtype='variable', and not give a record length. This seems to work for VMS 5.x, but Robert has only been reluctantly persuaded that my "bug fix" in

this case is any more than sleight of hand, so you are free to complain vociferously (to monger@mcmaster.ca) if this causes problems.

- SM now supports metafiles via the pseudo-device META and the command META READ file.

Changes from Version 2.3.0 to Version 2.4

- (This one is actually a change for 2.3.0). You can no longer choose your own prefix for SM graphics calls.¹
- You can specify a pattern to restrict the output of the LIST DEVICE command.
- You can say IMAGE CURSOR WORD WORD WORD to get intensities into a vector.
- An argument of 4 to BOX stops that axis from being drawn.
- VERSION is now implemented as a macro, printing \$version.
- The algorithm for choosing axis labels now checks that there is enough room for the labels that it has chosen and acts accordingly.
- Searches are now by default case *in*-sensitive. You can still use \$case-fold-search to change this behaviour.
- You can specify :tc=...: or :TC=...: as part of a DEVICE command.
- You can set the maximum size of string-valued vector elements in 'options.h'.
- Removed restrictions on number of vectors in READ and PRINT commands.
- Added atof to convert string-valued vectors to arithmetic.
- LIST DEFINE | will list internal variables (such as \$fx1).
- You can use READ ! to continue reading past lines that don't match what SM expects (either the wrong number or types of columns, or a mismatch with the format)
- There is a new format %n that returns the number of fields correctly matched in the line being READ.
- You can now specify the minimum and maximum number of arguments that a macro expects (but see the change notes (see [\[Change 2.4\]](#), page [\[undefined\]](#)) for a slight inconsistency). Note that you cannot declare macros that expect zero — n arguments with a **macro name nargs {...}** command.
- Most of the behaviour that is influenced by values in your '.sm' file will now look at a variable of the same name rather than the '.sm' file directly (they are defined from '.sm' when SM is started).
- There is now support for a wider range of fonts; the changes were introduced to support Japanese. Details in the fonts appendix (see section "Fonts" in *The SM Manual*).

¹ Did you know that sun's fortran i/o package had a function called cursor in it? You see why we had to do this, I imagine. There are lots of functions in standard packages that have names that we use.

- Shading with a solid colour on hardware that supports area fills is now done using the hardware support; this currently means X11 and postscript.
- The `-f` command line option can now be used to name the `.sm` file instead of specifying a full path.
- Startup files can be called `.sm` or `sm.rc`; the latter is looked for first on non-unix machines.
- You can now specify the slant and condensation of fonts with `\slant` and `\condense`.
- Support for reading tables, in particular FITS binary tables.
- The `WINDOW` command can handle ranges of windows (e.g. `WINDOW 1 4 1 2:3`) (n.b. for a while this was written as `WINDOW 1 4 1 2.3`).
- You can now create vectors and variables with local (actually nested) scope with the commands `DEFINE var LOCAL` and `SET vec LOCAL`.
- `PRINT` now supports the integer formats `'d'`, `'o'`, and `'x'`.
- Added colour to the raster devices
- Added new devices to generate `pbm`, `ppm`, and `gif` files.
- Added a compile-time option to use `doubles` rather than `floats` for vectors.
- Understand hex integers (of the form `0x...`).
- The X11 driver now puts up the current cursor position when in cursor mode; this can be disabled with the `-nocurswindow` flag to the `DEVICE x11` command.
- `SM` now supports the syntax `FOREACH var name { ... }` to iterate over each element of a vector `name`.
- You can access the help string associated with a vector `name` as the string-valued expression `HELP(name)`; useful in labels as `$(HELP(name))`.
- (This isn't really new, but it's little known). The devices `tpic` and `gpic` can be used to generate figures that `TEX` can pass to `dvips` and `xdvi`.
- You can now use `%o` and `%x` formats in `READ 'format' ...` commands.
- You can now declare a macro `abort_handler` to trap `USER ABORTs`.
- Added command `DITHER` to do Floyd-Steinberg dithering of images.
- Added expressions `DO(s_expr, s_expr)` and `DO(s_expr, s_expr, s_expr)` to allow implicit `DO` loops within expressions.
- You can now extract values from images by index as well as by coordinate; furthermore you can specify entire rows or columns (e.g. `set r=image[*,0]` to read the entire first row of an image).

- You can refer to an entire image as `image[*,*]` (e.g. `set image[*,*] = 2*image[*,*]`).
- Commands like "list device" now work even if device is overloaded.
- Added "list table" to list columns in (fits binary) tables
- When reading tables, you can now say "read table { name[0-4] }" to read an array from the table into (in this case) 5 vectors.
- Added "while { expr } { commands }".
- Added break to escape from WHILE loops.
- Added modulus operator, %.
- Added bitwise operators, | and &.
- You can define the variable `print_noheader` to stop the PRINT command printing the vectors' names.
- Expressions are now allowed among the arguments of macros used as functions
- The default is now to save the read pointer (see READ).
- Added incomplete Gamma functions (useful for, e.g., χ^2 probabilities)
- SM is now able to open an arbitrary number of X11 devices simultaneously (e.g. `dev x11 -dev 1`).
- You can request a private colour map for X11 devices; allocating (e.g.) 256 colours should work with this option.
- The connect function for callable SM is now `sm_connect` (`sm_conn` is still supported for backwards compatibility).
- Added `sm_ctype_i` and `sm_set_ctypes` to manipulate colour maps from callable SM.

Inconsistent Changes from Version 2.3.0 to Version 2.4

- The changes to macros with variable numbers of arguments means that you'll have to declare macros with zero to one arguments as having 01 arguments; the old form (11) means between one and one arguments.
- Italic fonts are now *not* toggled off again by a second `\it` in T_EX mode; rather you must use explicit grouping with `{}`.
- In callable SM, `sm_window` now expects 6 arguments, not 4.
- The macro `is_set` now does what it says; it doesn't do an implicit WHATIS. You probably want to use `is_macro` or `is_vector` rather than `is_set`.
- The command `"set image(expr,expr) = expr"` is now deprecated; please say `"set image[expr,expr] = expr"`.
- The command to use ranges of windows is of the form `"WINDOW 1 4 1 2:3"`, but for a while it was `"WINDOW 1 4 1 2.3"` (it was changed as `"WINDOW 1 12 1 8.10"` is indistinguishable from `"WINDOW 1 12 1 8.1"`).
- For devices with strange aspect ratios, points are now expanded to preserve shape rather than compressed (this is now consistent with the behaviour of characters). Axis ticks are also suitable scaled.

Changes from Version 2.4.0 to Version 2.4.1

- Added a new SM device TK for creating Tk widgets to display SM plots. See also tkSM in the callable directory.
- Added experimental driver for linux virtual console windows; can corrupt video memory and require a reboot (not, I claim, an SM bug). See RHL for details.
- The variables `angle`, `expand`, and `ptype` now acquire an extra value when vector-valued (namely, the dimension of the vector).

Changes from Version 2.4.1 to Version 2.4.2

- The CURSOR command now always updates the plot position
- The command CURSOR READ updates the plot position, i.e. it RELOCATEs to the current cursor position.
- When using the cursor to define vectors, a 'p' is now equivalent to 'e' 'q', i.e. it defines a point and exits.

Changes from Version 2.4.2 to Version 2.4.4

- Macros are now automatically forced to end in space or newline, which means that local variables/vectors don't softly and suddenly vanish at the end of (some) macros.
- When reading a table, if you specify the name of an array field without specifying the desired element, it's now taken to be 0, and a 0 is appended to the field name.

Changes from Version 2.4.2 to Version 2.4.4

- Added TABLE N command to select Nth HDU of a table; for fits binary tables this caches the read pointer and is *much* more efficient than using TABLE N filename.
- Added WRITE IMAGE to write images to disk (useful for binary i/o of vectors)
- Added IMAGE DEFINE to define keywords for images written to disk
- Added an ability to run SM from within emacs, e.g. sending the current buffer to be executed; load sm-mode and type M-x `describe-mode` for details.

Changes from Version 2.4.4 to Version 2.4.5

No record exists of these changes; see the changes in 2.4.6.

Changes from Version 2.4.4 to Version 2.4.6

- Added LTYPE EXPAND to rescale the pattern of dots and dashes used in drawing lines
- Added SET WORD[expr, expr] = expr to efficiently set parts of vectors.

Changes from Version 2.4.6 to Version 2.4.7

- Allow FITS binary tables following image HDUs

Changes from Version 2.4.7 to Version 2.4.8

- Advertised an SM bug page at <http://www.astro.princeton.edu/cgi-bin/gnatsweb.pl>.
- Support for FITS files with IMAGE extension (`IMAGE n filename`)
- Support for integer valued vectors. *Watch out!* This is a major change deep down in SM's insides and may well have broken something. Please reports any problems ASAP.
- Added << and >> operators.
- Allow `set_opts` to control byte swapping
- Added (bad) CMYK support to DEVICE postfile to make ApJ happy (I hope).
- Added `$cwd` to track current directory
- Added "`DEFINE name < str > [INT, INT]`" command

Changes from Version 2.4.8 to Version 2.4.20

- Support `IMAGE n file [... : ...]` to read parts of images.
- Added `"STRING<WORD>"` to convert text that contains single quotes to SM strings.
- Support e.g. `READ " { x y z }` to read x, y, and z from the first 3 columns of a file.
- Integer vectors in a `SORT { ... }` list are no longer converted to float (unless they're being sorted on)
- Integer vectors may be specified as e.g. `READ x 1.d` as well as `1.i`
- The `xr` and `yr` fields in `graphcap` entries may now be given as `$n` to expand to an argument to the `DEVICE` command.
- A new option (setting `ICLOCK`'s 0x8 bit) is now supported in `Axis` to draw axis labels on the opposite side from the ticks.
- You can also choose `AXIS`'s `ICLOCK` value to totally suppress ticks, but not labels.
- Modify history editor to save last line you modified at bottom of history list (avoiding losing changed by accidental uparrow-downarrow pair).
- Added script `make_tags` that makes emacs-stype `TAGS` files from sets of SM macros.
- You can now say `MACRO name "02 " { ... }` to declare a macro with 0 — 2 arguments (previously only possible using `MACRO READ`). Clumsy, but better than nothing.
- Added internal variable `label_offset` to control how far axis labels are drawn from their axes.
- Added support for writing (FITS binary) tables: `WRITE TABLE`.
- Added short-circuiting logical operators (in scalar contexts) `AND` and `OR`; e.g. `if($?foo AND $foo == 1) { echo OK }`
- Allow `SORT(expr)` in expressions; e.g. `set x = sort(y)`
- Allow `set x = y if(scalar)`

Changes from Version 2.4.20 to Version 2.4.23+

- Allow type qualifiers (e.g. `.s`) in `READ '' {}`; e.g. `READ '' { x s . s i . d }`;
- Allow `\colour{}` in `TeX_strings`; also `\ctype` and `\color`
- In `READ '' { vector list }` you can specify columns to be skipped as `.`, e.g. `READ '' { a . . d }`.
- You can now request `SURFACE` to draw summits that poke above/below the z-axis limits.
- Integer vectors in a `SORT { ... }` list are no longer converted to float even if they're being sorted on
- All sorts are now done by `qsort`; heap sorts no longer make sense with the size of vectors people are using.

Inconsistent Changes from Version 2.4 to Version ??

- The `CURSOR` and `CURSOR vec vec` commands now only print a help string if `verbose` is greater than zero.
- Because macros are now forced to end in space or newline, an extra space may appear in your output. End macro with an explicit `#` to avoid this.
- Complain (if `VERBOSE` is 1 or more), but allow incorrect `PRINT` formats. E.g. `PRINT '%d' { vec }` for a floating point vector (use `'%.0f'` instead)
- Fix bug in postscript driver whereby the positions of points/lines could drift after many many closely spaced dots. Actually, it's arguably a bug in postscript whereby sequences of `currentpoint moveto` accumulate rounding error.
- Post v2.4.13 I increased `CHARMAX` from 160 to 1000; this caused SM to crash under Mac OS/X. Reverted (only for macs) in sm2.4.22