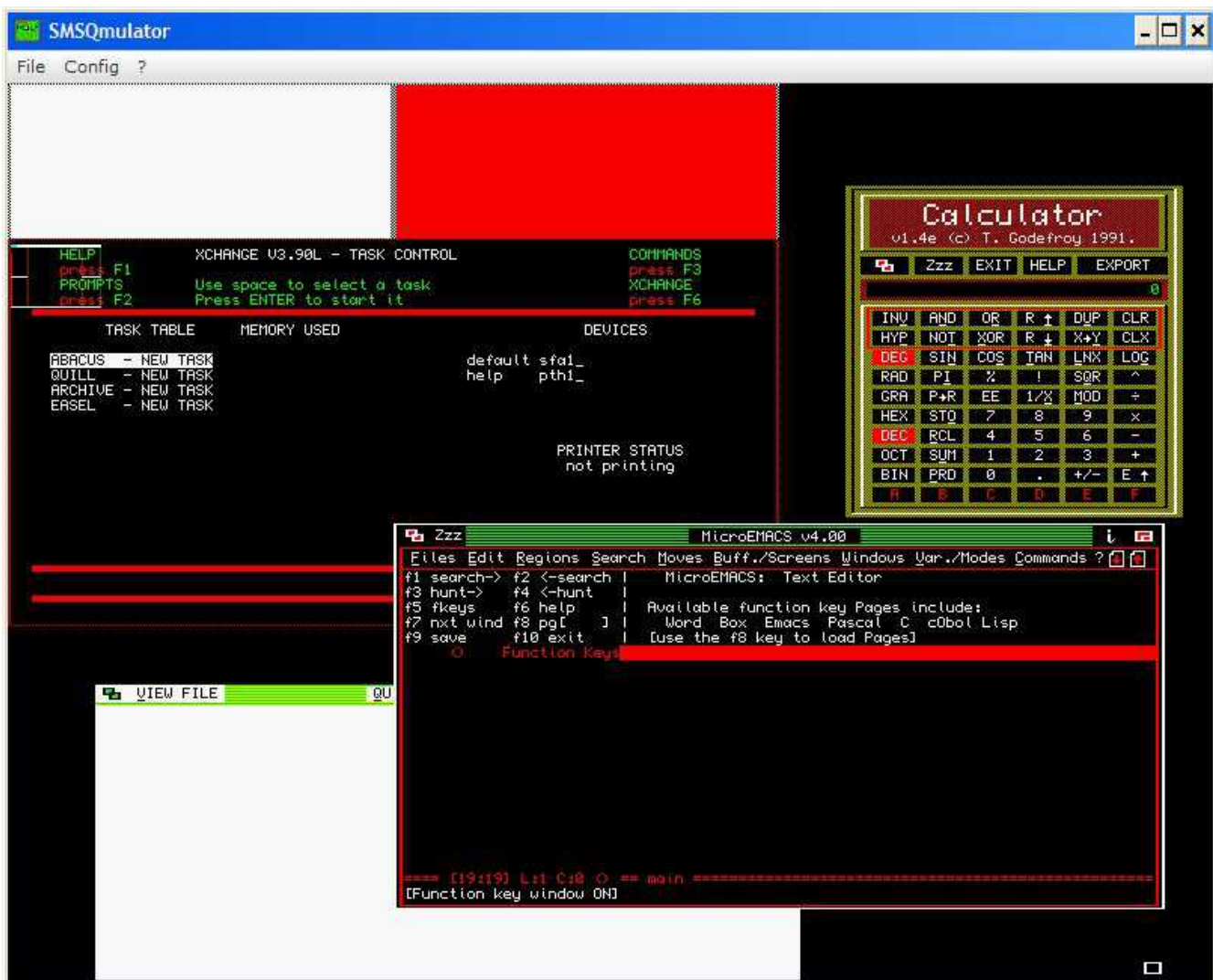


SMSQmulator

User Guide



By
Timothy Swenson

SMSQmulator User Guide

Copyright 2013
Timothy Swenson

Creative Commons License

Attribution – NonCommercial – ShareAlike 2.0

You are free:

- to copy, distribute, display, and perform the work
- to make derivative works

Under the following conditions:

Attribution
Noncommercial
Share Alike

- You must give the original author credit
- You may not use this work for commercial purposes
- If you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.

For any reuse or distribution, you must make clear to others the license terms of this work. Any of these conditions can be waived if you get permission from the copyright holder. For fair use and other rights are in now way affected by the above.

Table of Contents

1.0 Introduction

- Systems Supported
- Components of SMSQmulator
- SMSQ/E Documentation
- SMSQmulator Performance

2.0 Installation

- Pre-requisites
- Downloading SMSQmulator
- Compiling SMSQ
- Unzipping SMSQmulator
- SMSQmulator Components

3.0 Starting SMSQmulator

- Creating a Desktop Shortcut
- Restarting SMSQmulator
- Linux Window Manager Keystroke Trapping

4.0 New Devices

- NFA and SFA Devices
- NFA Device
- SFA Device

5.0 Older Devices

- WIN Device
- FLP Device

6.0 New Device Keywords

- NFA_USE
- SFA_USE
- QXL_USE
- FLP_USE

7.0 Sounds

- BEEP
- SOUNDFILE
- SOUNDFILE2
- SOUNDFILE3
- KILLSOUND

8.0 System Clipboard and SCRAP

- JVASGET
- JVASPUT

9.0 Configuration

- SMSQ/E File
- Language
- NFA Drives
- SFA Drives
- QXL Drives
- FLP Drives
- Screen Size
- Color Mode
- Memory Size

Monitor Visible

Fast Mode

Cursor Blink Pause

Time Correction

Double Window Size

Hide the Menu Bar

Warnings

10.0 Using SMSQmulator

Making SMSQmulator Portable

Backuping up SMSQmulator

11.0 Upgrade

12.0 Support

Appendix I – SBASIC Keywords

1.0 Introduction

SMSQmulator is a java-based emulator of the SMSQ/E operation system, a follow-on operating system for the 68000-based Sinclair QL. Being written in Java, the emulator will work in a number of operating systems and platforms, including Windows, Linux, and Mac.

SMSQmulator is designed to be a portable SMSQ/E system that allows the running of most QL programs across many platforms. Since it supports SMSQ/E, all Pointer Environment programs will work on SMSQmulator.

SMSQmulator is not a QL or QDOS emulator, so it will not run some of the early QL programs that rely on certain memory locations of the QL. SMSQmulator is based on a M68000 emulator, so QDOS or SMSQ/E programs compiled for the Q40 or Q60, which as based in the 68040 and 68060 respectively, may not run. A lot of the programs hosted on www.q40.de, will not work with SMSQmulator, because they have been compiled specifically for the 68040 or 68060.

Systems Supported

Because SMSQmulator is written in the Java language, it will run on any system that has Java installed. SMSQmulator has been tested with Java 6 & 7, and on Windows XP, Windows 7, Windows 2008 R2, Mac OS X, Red Hat 6.1, CentOS 5, openSUSE 12.1 and Ubuntu 12.

Components of SMSQmulator

SMSQmulator is comprised of three separate parts, the SMSQ/E emulator, the 68000 emulator, and SMSQ/E operating system. The SMSQ/E emulator was written by Wolfgang Lenerz. The 68000 emulator was written by Tony Headford. SMSQ/E was written by Tony Tebby, with additional contributors.

SMSQE documentation

Full details on the use and operation of SMSQE are beyond the scope of this User Guide. More comprehensive documentation on SMSQE can be found on Dilwyn Jone's website www.dilwyn.me.uk in the "Documents" section, then "Replacement Manuals". There are two documents by Martin Head describing SMSQE for the QPC emulator. There is both a Concepts Guide and a Keywords Guide. SMSQmulator does not follow these documents exactly, but since both QPC and SMSQmulator use SMSQE, the documents are close enough.

SMSQmulator Performance

Using the Bogomips program as a method of measuring performance, on a AMD Athalon dual core 1.7 GHz system running Windows XP, SMSQmulator had a bogomips rating of 6.10. On the same system, Q-Emulator had a 39.41 bogomips rating. On an older Pentium 1.6 GHz system, SMSQmulator had a bogomips rating of 5.57.

SMSQmulator is known to take up a lot of CPU, so don't be alarmed when running it. Experience with SMSQmulator has shown that system performance is not really impacted and other applications are

fairly responsive when SMSQmulator is running. See Section 8 on Configuration for ways to configure system performance.

2.0 Installation

The installation of SMSQmulator is fairly simple and only requires unzipping the distribution archive file. Once unzipped, the next step is to tell SMSQmulator the location of the drive devices.

Pre-requisites

A system running SMSQmulator must have a version of Java installed. Java can be downloaded from Oracle/Sun at www.java.com. SMSQmulator has been tested with the latest versions of Java, versions 6 & 7.

Downloading SMSQmulator

SMSQmulator can be downloaded from the following link:

<http://www.wlenerz.com/SMSQmulator/>

There is a version of SMSQmulator for Java 6 and Java 7. Please download the version that is right for the version of Java that you are running. To tell what version of Java you are running, open a Command Window (in Windows) or a shell (in Linux) and run the command "java -version". The second number is the version number (ie. 1.7 is version 7). There were some issues with Java and Linux that required a separate version.

The source code for SMSQmulator can be downloaded from the same web page.

Unzipping SMSQmulator

Using any number of unzipping programs, either graphical or command-line based, unzip the SMSQmulator.zip file into a directory. It is suggested that the files be unzipped into a directory called 'SMSQmulator'.

SMSQmulator Directories

Inside the SMSQmulator directory are a number of files and a directory:

SMSQmulator.jar	- The emulator
SMSQmulator.win	- Example QXL.WIN file
SMSQE	- SMSQE binary
README.pdf	- SMSQmulator Manual in PDF format
README.odt	- SMSQmulator Manual in OpenOffice format.
UserGuide.pdf	- This User Guide
lib	- Library directory

3.0 Starting SMSQmulator

SMSQmulator.jar is the emulator file. For most operating systems, just use a graphical file manager to double click on SMSQmulator.jar. Because a .jar file is associated with Java, Java will execute and run the jar file.

Some versions of Linux may not associate java with a .jar file. Java can be executed from the command line:

```
java -jar smsqmulator.jar
```

Another workaround is to create a wrapper shell script that runs the emulator. The script can be put into /usr/local/bin where it can be executed from any directory.

```
#!/bin/sh
cd /home/user/SMSQmulator/
java -jar "SMSQmulator.jar"
```

SMSQmulator will start with the default QL screen size of 512x256. The example SMSQmulator.win file has a number of example programs. SMSQmulator executes a BOOT program from this drive and starts a demo program.

Creating a Desktop Shortcut

For Windows, right click on the Desktop, select New -> Shortcut. Click on Browse and browse to the location of SMSQmulator, into the dist directory and select SMSQmulator.jar. Edit the name of the shortcut and remove the .jar extension. Once the shortcut is on your desktop, you can run SMSQmulator from there instead of having to use Windows Explorer to browse to it.

For Linux, the process is very similar, at least for Lubuntu with the OpenBox window manager.

BOOT file

Which ever device has been set to WIN1_, if there is a file called BOOT, it will be executed as soon as the emulator is started (just like the example loaded on SMSQmulator.win).

Restarting the SMSQmulator

If you need to reset or restart SMSQmulator for any reason, select File -> Reset. Under some systems, the keyboard combination of CTRL-ALT-SHIFT-TAB will reset SMSQmulator.

Linux Window Manager Keystroke Trapping

Some key sequences that are used by SMSQ/E are trapped by the Linux window manager. With Lubuntu, the Ctrl-Alt-Right is trapped and used to switch between desktops. Each window manager has a configuration file that can be edited to remove these key stroke trappings and let them pass to

SMSQmulator. With Lubuntu with the OpenBox the file is:
`~/.config/openbox/lubuntu-rc.xml`

4.0 New Devices

SMSQmulator provides 2 new devices that access the native file system, Configuration of these devices is detailed in the Configuration section.

NFA and SFA Devices

SMSQmulator introduces two new file devices, NFA(Native File Access) and SFA (SMSQE File Access). Both of these devices are links to the underlying file system. Both devices allow you to read and write files from the underlying file system, but the SFA devices saves the QDOS header information to the file, necessary for executables. The NFA devices does not add any header information to the files. To import or export files from SMSQmulator to the native file system, NFA is the right device. To save files to be used only in SMSQmulator, esp. executables, then SFA is the right device. SMSQmulator supports up to 8 NFA and SFA devices. They are denoted as SFA1_, SFA2_, etc, and NFA1_, NFA2_, etc.

The names of the NFA and SFA devices may be changed to another name, like FLP, MDV, WIN, etc. This is done via the Config toolbar menu. The new name must be exactly 3 characters with no numbers. For this change to take affect, SMSQmulator must be reset. Once reset, the new name will work, but the old name will not. If SFA is changed to WIN, then win1_ will work, but sfa1_ will result in “not found.” If the name is changed, to change it back, either clear out the new name, or replace it with the original name.

Each SFA and NFA links to a directory on the native file system. The directory can reside on any drive or device that the native file system is capable of reading. This can be a network device via CIFS or NFS, an external device like a USB drive or SD card, or a simple directory on the local hard drive.

Depending on the native file system, the file names used in NFA and SFA devices are case sensitive. On other words, the file SFA1_Test_doc can be different from the file SFA1_TEST_doc. This does go against the SMSQE convention of the file system being case insensitive. It is possible to limit this by making the SFA and NFA devices semi-case independent. See the Configuration section for this.

Here are a number of rules about the SFA and NFA devices:

1. Do not mix the directories for NFA and SFA devices

It is vitally important that any SFA or NFA device do not share the same directory, meaning that NFA1_ and SFA1_ must never point to the same directory. These devices are not designed for using the same directory and data loss is very possible of such a thing were to happen. Doing otherwise in entirely **not supported, not recommended**, and something that should **not even be considered**.

2. Avoid accented characters in NFA or SFA file or directory names

SMSQmulator has issues with accented characters when communicating with the native file system.

3. File names must be valid for the native file system

Some SMSQ/E file names that would be valid under a pure SMSQ/E system are not valid in the native file system. The biggest example is the use of the asterisk or star “*” in a file name. Valid under SMSQ/E rules, but not valid for Windows or most Linux file systems.

4. NFA and SFA drives do not support direct sector access

Direct sector access (e.g. win1_*d2d) will not work with SFA and NFA directories. SMSQmulator is only allowed access to the native file system at the file level.

5. Do not create NFA or SFA directories with underscores “_” in them

Since SMSQ/E used an underscore to define a directory, having a native file system directory with an underscore in it will confuse the NFA and SFA device drivers.

6. Making a directory may result in files changing names

If a file exists in a directory that also has the same name as a new sub-directory, the file will be moved to the new directory, thereby changing its name. If there is a file test_doc in a directory, and the command “MAKE_DIR test” is executed, then test_doc will be moved into the test directory and be called test->test_doc or test_test_doc.

7. Do not FORMAT NFA or SFA devices

Since the NFA and SFA devices are links to the native file system, there is no need to FORMAT the NFA or SFA devices. If tried, the error “not implemented” will be displayed.

NFA Device

When saving files to the NFA device, the QDOS file header will not be saved to the file. This means that any executable saved to an NFA device will not longer work. Since SBASIC programs are text files, it is possible to save and load SBASIC programs on the NFA device. If you need to export a file from SMSQmulator to the native file system, say, to send to a printer, then the NFA device is the one to use.

SFA Device

The SFA device adds the QDOS header to the beginning of any saved file. Only those files saved by the SFA device will be seen in the directory. If a file is moved to the directory via the native file system, it will not be seen by the SFA device. The file header saved with the file is standard QDOS file header, except that an additional identification long word “SFA0” is added to the front of the header.

5.0 Old Devices

SMSQmulator supports two older devices, WIN and FLP.

WIN Device

The WIN device is how the emulator reads and writes to standard QXL.win files. There are 8 WIN devices, WIN1_ to WIN2_. Each WIN device points to a different QXL.win file. Like the NFA and SFA devices, the device name for the WIN device can be changed.

WIN drives can be formatted with the normal FORMAT command. The command has two variations:

FORMAT winX_size (e.g. FORMAT win2_50)

FORMAT winX_size_label (e.g. FORMAT win2_50_Data)

The size parameter must be present is the size of the QXL.win file in megabytes (MB). In the example, both drives are 50 Megs each. Size must not be lower than 1 and no greater than 2000. The label is the same as a disk label in QDOS and is shown using WDIR. The label must not be more than 20 characters. Before the FORMAT command can be run, the WIN device must be pointed to a non-existing QXL.win file. As an example, when creating a QXL.win file called DATA.WIN of size 20 Megs and have it as WIN2_, do the following:

1. In the Configuration Menu, point WIN2 to a file called DATA.WIN (can be in lower case).
2. Issue the command, FORMAT WIN2_20_DATA

FLP Device

SMSQmulator does not support directly Floppy access. Disk image files made from a floppy disk can be read by SMSQmulator. SMSQmulator supports both 702K and 1.4MB floppy disk images. SMSQmulator does not support writing to a floppy image.

The FLP device is how the emulator reads floppy image files. There are 8 FLP devices, FLP1_ to FLP2_. Each FLP device points to a different floppy image file. Like the NFA and SFA devices, the device name for the FLP device can be changed.

Creating Floppy Image Files

A floppy image file is nothing more than a raw read of a floppy and that data sent to a file. The most common tool to do this with is the Unix command 'dd'. This command is part of all Linux distributions. For Windows, the command is available via Cygwin, a Unix-like environment for Windows and is available at www.cygwin.com. There are other Windows programs that function the same as dd, so the user is not limited to just Cygwin.

For Windows users, with Cygwin installed, start up the Cygwin shell. The floppy drive must first be mounted to be access by Cygwin. The a: drive will be mounted to the normal location of the floppy in Linux. The following commands do this:

```
mount -b ../a: /dev/fd0
```

With either Windows/Cygwin or Linux, use the following dd command to read the data from the floppy and save it in a file called "floppy.img":

```
dd if=/dev/fd0 of=floppy.img
```

For Linux, if using a USB floppy, the drive shows up as an additional hard drive. Use the command 'fdisk -l' to see the additional hard drives. If you only have one hard drive, the floppy will show up as /dev/sdb. The above commands can be used for both 720K and 1.4MB floppies. With a 720K floppy, you should see output like this:

```
1440+0 records in
1440+0 record out
```

For a 1.4MB floppy, the value will be 2880.

For Windows, the program QLtools comes with a program called QDC (Quintessential Disk Cloner) that will create a floppy image file.

6.0 New Device Keywords

Four new device keywords are introduced in SMSQmulator for the new and old devices:

```
FLP_USE
NFA_USE
WIN_USE
SFA_USE
```

All four keywords work in the same manner, but they each address a different device. Each keyword has two forms.

NFA_USE name\$

This redefines the name of the device to the same as name\$, eg.

```
NFA_USE win           - Device NFA1...8 will be called win1...8
```

NFA_USE drive,directory\$

This sets the NFA device to the specified directory. This is the same as setting it via the configuration menu. This setting is only temporary and does not change the configuration file.

```
NFA_USE 3,"C:\SMSQmulator\nfa3\" - sets NFA3 to the indicated directory
```

SFA_USE name\$

This redefines the name of the device to the same as name\$, eg.

SFA_USE win - Device sfa1...8 will be called win1...8

SFA_USE drive,directory\$

This sets the SFA device to the specified directory. This is the same as setting it via the configuration menu. With QXL_USE instead of a directory, a file name is used. This setting is only temporary and does not change the configuration file.

SFA_USE 3,"[C:\SMSQmulator](#)\sfa3\" - sets SFA3 to the indicated directory

WIN_USE name\$

This redefines the name of the device to the same as name\$, eg.

WIN_USE FLP - Device win1...8 will be called flp1...8

WIN_USE drive,file\$

This sets the WIN device to the specified QXL.win file. This is the same as setting it via the configuration menu. This setting is only temporary and does not change the configuration file.

WIN_USE 3,"[C:\SMSQmulator](#)\qxl3.win" - sets win3_ to the indicated directory

FLP_USE name\$

This redefines the name of the device to the same as name\$, eg.

FLP_USE win - Device FLP1...8 will be called win1...8

FLP_USE drive,file\$

This sets the FLP device to the specified floppy image file. This is the same as setting it via the configuration menu. This setting is only temporary and does not change the configuration file.

FLP_USE 3,"[C:\SMSQmulator](#)\floppy3.img" - sets FLP3 to the indicated directory

7.0 Sounds

SMSQmulator does support sound, but due to issues with the Java sound driver, the sound may not sound as nice as expected. SMSQmulator supports both the BEEP command and the Sampled Sound System, similar to what is used on the Q40/60.

BEEP

The BEEP command works like the normal QL BEEP command, except that the last three arguments, wrap, fuzzy and random, are not implemented and if used, are ignored. The rest of the arguments are supported.

BEEP duration, pitch, pitch_2, grad_x, grad_y

The corresponding BEEPING command has not been implemented.

SOUNDFILE

Loads and plays a sound file. The sound file is

SOUNDFILE "filename",[rep]

The "filename" must be on a local or fast device, as the sound file is read in as it is playing. A slow device will cause the sound to be interrupted. Rep is the number of repetitions. SOUNDFILE will not return to the command line until the entire sound file is played.

SOUNDFILE2

This command functions the same as SOUNDFILE, but the sound is played by a new job and once the sound starts, the command line is returned. If the job that owns the SOUNDFILE2 job is removed, then the SOUNDFILE2 job is also removed.

SOUNDFILE2 "filename",[rep]

SOUNDFILE3

This command functions the same as SOUNDFILE2, in that the sound is played by a different job, but if the parent job is removed, the SOUNDFILE3 job will not be removed.

SOUNDFILE3 "filename",[rep]

KILLSOUND

This command stops any sounds being played by any of the above commands. If a SOUNDFILE job is removed, due to an internal buffer, the sound may still play for a number of seconds. KILLSOUND when run after removing a job, will stop this from happening. The command will remove the first SOUNDFILE job that it finds. It is not recommended to have more than one SOUNDFILE job running at any one time (plus having more than one sound playing at once will not sound good).

8.0 System Clipboard and SCRAP

Each operating system has the concept of the clipboard, where text or graphics is temporarily stored and later can be inserted or "pasted" into another application. For SMSQ/E, the Menu Extensions has the SCRAP thing that functions exactly the same. A number of new keywords have been added to SMSQ/E to link the system clipboard with SCRAP. The Menu Extensions must be loaded for these commands to work.

JVASGET

Gets the contents of the clipboard and puts it into the SCRAP. Any text in the SCRAP is overwritten

JVASPUT

Gets the contents of the SCRAP and puts it into the clipboard. Any text in the clipboard is overwritten.

See the Menu Extensions for more information on using the SCRAP and linking them to Hot Keys for ease of use.

9.0 Configuration

The default configuration of SMSQmulator is no configuration. SMSQmulator does not know where the SMSQ/E binary file is, does not know where any NFA, SFA, or QXL devices are, etc.

The configuration of SMSQmulator is stored in a configuration file called SMSQmulator.ini, located in your home directory. For Windows this will be C:\Documents and Settings\[user] where [user] is either owner or your user name. For Linux it is /home/[user], where [user] is your user name.

Configuration can be done by editing the SMSQmulator.ini file or through the configuration menu in the emulator. It is recommended that the initial configuration be done via the menu until the format of the configuration file is understood.

All configuration options are under the Config toolbar menu, except the SMSQ/E file, which is under the File toolbar menu.



Language

SMSQmulator supports English, German (Deutsch), Spanish (Espanol) and French (Francais). Use Config -> Language to select that language that you want for SMSQmulator. The change takes affect immediately. This setting does not change the language of SMSQ/E itself. This must be set from the SMSQ/E command line with the LANG_USE command.

NFA Drives

NFA drives have the following options:

Set Dirs for NFA Drives

Select this option and a pop-up window will appear letting you define up to 8 NFA drive directories. The name of the directory can be typed into the window or click on the button next to the text window to bring up a browse window.

NFA file name changes

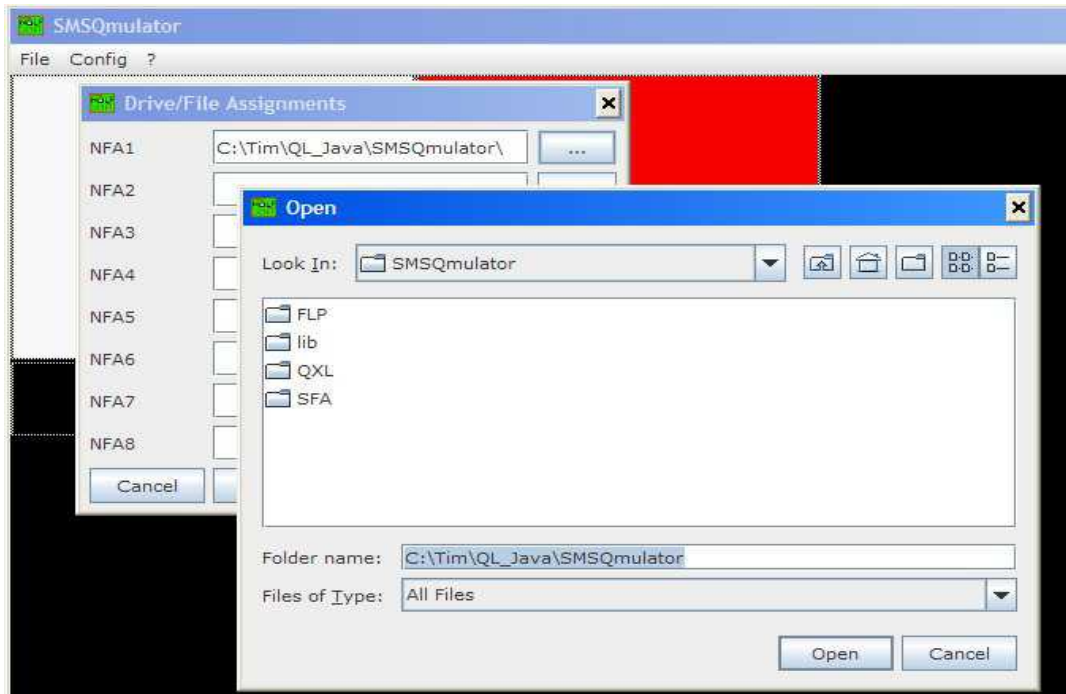
This option has the following possible settings:

- Leave as they are
- Set all file name to upper case
- Set all file name to lower case

The first option is the default and will allow for upper and lower case in file names. The other two options will convert any file names for the NFA device to either upper case or lower case. If the setting is lower case, then the file NFA1_FILE will be converted to NFA1_file before it is saved. For the upper case setting, the behavior is just the opposite.

NFA USE Name

This option sets the NFA_USE name to something else, such as WIN, FLP, MDV, etc. This setting will remain after reboots (unlike the NFA_USE keyword). For this setting to take effect, SMSQmulator needs to be either reset or restarted.



SFA Drives

SFA drives have the following options:

Set Dirs for SFA Drives

Select this option and a pop-up window will appear letting you define up to 8 SFA drive directories. The name of the directory can be typed into the window or click on the button next to the text window to bring up a browse window.

SFA file name changes

This option has the following possible settings:

- Leave as they are
- Set all file name to upper case
- Set all file name to lower case

The first option is the default and will allow for upper and lower case in file names. The other two options will convert any file names for the NFA device to either upper case or lower case. If the setting is lower case, then the file SFA1_FILE will be converted to SFA1_file before it is saved. For the upper case setting, the behavior is just the opposite.

SFA USE Name

This option sets the SFA_USE name to something else, such as WIN, FLP, MDV, etc. This setting will remain after reboots (unlike the SFA_USE keyword). For this setting to take effect, SMSQmulator needs to be either reset or restarted.

WIN Drives

WIN drives have the following options:

Set Files for WIN Drives

Select this option and a pop-up window will appear letting you define up to 8 QXL.win files. The name of the file can be typed into the window or click on the button next to the text window to bring up a browse window.

WIN Use Name

This option sets the WIN_USE name to something else, such as FLP, MDV, etc. This setting will remain after reboots (unlike the WIN_USE keyword). For this setting to take effect, SMSQmulator needs to be either reset or restarted.

Screen Size

SMSQmulator supports a number of screen sizes, the largest depending on the screen resolution of the underlying computer system. The default and minimum screen size is 512x256. The screen size is horizontal size and then the vertical size with an “x” in between. Example screen sizes are 1024x512 or 800x600. If the screen size is 512x256 and the color mode is set to QL colors, then SMSQmulator starts the screen at the default QL location of \$20000.

Color Mode

SMSQmulator has two color modes, QL and 16 bit colors. In QL mode, the normal Mode 4 and Mode 8 are supported (but no FLASH in Mode 8). 16 bit colors is the same as the GD2 high resolution screen. Screen redraws are normally faster in 16 bit color mode. Unless testing some older QL programs, it is recommended to use 16 bit color mode.

Memory Size

SMSQmulator can be set with a range of memory size, from 1 MB to 32 MB. If using 16 bit color mode, the minimum is 8 MB, which is also the default setting.

Monitor Mode

Monitor Mode means that there is a monitor where you can halt emulation, trace it step by step, execute a subroutine, examine memory, etc.

In Monitor Mode, there is a menu bar and 4 windows, which are:

1. Main emulator screen
2. Window #1 below the emulator screen
3. Window #2 below the emulator screen
4. Window #3, under the other two screens; the monitor command line.

This mode is not used by the normal user, but it is defined here for those with a more adventure-some spirit.

The monitor command line is used to enter commands to the Monitor. Type h, for help, to get a summary of all of the available commands. Turning on or off Monitor Mode is done via the Config menu.

Fast Mode

In Fast Mode, the emulator is not “listening” for the monitor, so the emulation runs about 25-50% faster. Fast Mode is the default mode for the emulator. Only if you are doing any debugging of the emulator do you need to be in Monitor Mode.

Cursor Blink Pause

Cursor Blink Pause is a way to configure SMSQmulator to take less CPU cycles. When the cursor is blinking, the emulator is waiting for user input. By pausing the emulator right before a cursor is to blink, the emulator takes up less CPU.

The value entered is the number of milliseconds that the emulator will pause right before the cursor blink. A suggested starting value is 100. On a test system, the CPU went from 50% utilization to 39% when the cursor blink pause was set to 100.

Time Correction

SMSQmulator should be in sync with the system clock of the underlying operating system. If it is not, then this option is set to adjust the SMSQmulator clock by a number of seconds, either + or -. Remember that an hour is 3600 seconds.

Double Window Size

This will increase the window size, but also increase the font size making the text in the window larger and easier to see. Functionally, this option increases the size of each pixel.

Hide the Menu Bar

This option will hide the menu bar for SMSQmulator. To get the menu bar back, run the command JVAMBAR.

Warnings

There are three warnings that can be turned off.

Warn if a WIN drive doesn't comply with the standard

Some older QL utilities can create QXL.WIN files that do not fully comply with the QXL.WIN standard. SMSQmulator will try to use the files, but this warning can be used to notify the user when one of these non-standard QXL.WN files is being used. It is suggested the user should migrate data from these type of QXL.WIN files to standard QXL.WIN files.

Warn if a WIN drive can't be opened because the native file does not exist

When configuring SMSQmulator to access QXL.WIN files, it will check to make sure the file exists. This warning can be turned off if so desired.

Warn when a WIN drive is full

When a WIN drive is full, a pop up warning message will appear, warning of the full drive. In the command line of the emulator, the "drive full" message will also appear. Some users may not want the pop up warning.

10.0 Using SMSQmulator

Making SMSQmulator Portable

Everything in the SMSQmulator directory is completely portable across system. Only the configuration file is not portable. It can be moved to different systems, but it will not be usable between a Windows and a Linux system because the file system layout is different.

It is possible to put all of the SMSQmulator files on a USB stick and run it from the USB stick on many systems. The smsqulator.ini file will need to reside on each system in the users home directory. With some operations systems, when plugging in a USB stick, the drive letter is automatically assigned and there is no guarantee that the drive letter will be the same each time. If the drive letter changes, then the smsqulator.ini file will need to be edited to reflect the drive letter change.

It is also possible to keep the SMSQmulator files on a network share (CIFS for Windows and NFS/CIFS for Linux) and mount the share to the local system to use. CIFS and NFS have file locking mechanisms, so that multiple users are not able to use the same files at the same time. A file server can be as simple as a desktop system with drive or directory shared out to the local systems.

If you are using SMSQmulator on multiple systems and not keeping the files in a single location (such as a USB stick or network share), then consideration must be made of syncing the files between the systems so that each system has the most up-to-date version of the files.

Backing up SMSQmulator

Backing up SMSQmulator can be done with a number of native operating system tools and can be as simple as making a copy of the SMSQmulator directories (if the NFA, SFA and QXL directories and files are in the same directory) to a new directory. The SMSQmulator directory can also be stored on

CD- or DVD-rom for a more lasting backup. A copy of the smsqulator.ini file from the user home directory should be copied into the SMSQulator directory for a more complete backup.

11.0 Upgrade

When bug fixes are made, a new SMSQulator.zip file will be released on the website. Download the new zip file to your system and unzip it into the same directory where SMSQulator currently resides. Let the unzip program overwrite the existing files. Since the unzipping process only overwrites files in the zip file, any files that you might have created in the same directory will not be affected.

Restart SMSQulator, from the toolbar, select “?” -> About and verify that the version number has changed and is the same as the upgrade that was downloaded.

12.0 Support

Tim Swenson has volunteered to be the main contact for questions and issues with SMSQulator. Tim was a beta tester of SMSQulator and is familiar with the emulator. Please send all questions, comments, or problems to Tim at swenson_t@sbcglobal.net. For problems, please provide a clear description of what the problem is, what behavior you are seeing or not seeing, etc. Tim will review the issue, see if it is reproducible and forward it on to Wolfgang.

Appendix I - Keywords

Starting on the next page is a list of all keywords used in the SMSQ/E version for SMSQmulator. Not all of these command are implemented.

Changes in Keywords

DISP_TYPE

This command will return a 0 for QL color mode or a 32 for 16 bit color mode. All other DISP_XXXX commands are not implemented and have no effect.

KBD_TABLE

This keyword has no affect on SMSQmulator. The keyboard used is configured in Java.

MACHINE

This command will return a value based on the type of system that SMSQ/E is running on. Here is the return values:

SMSQmulator	20
QPC	30
GoldCard	10
SuperGoldCard	12
Q60	17

JVAMBAR

Makes the SMSQmulator menu bar visable. A Config item can be set to get rid of the menu bar. The way to bring it back is to use this command.

ABS	CURSOR	ERR_EX
ACOS	CURSPRLOAD	ERR_FE
ACOT	CURSPROFF	ERR_FF
ADATE	CURSPRON	ERR_IU
AJOB	DATAD\$	ERR_NC
ALARM	DATA_USE	ERR_NF
ALCHP	DATE	ERR_NI
ALTKEY	DATE\$	ERR_NJ
ARC	DAY\$	ERR_NO
ARC_R	DAY%	ERR_OM
ASIN	DDOWN	ERR_OR
AT	DEG	ERR_OV
ATAN	DELETE	ERR_RO
AUTO	DEL_DEFB	ERR_TE
BAUD	DESTD\$	ERR_XP
BEEP	DEST_USE	ERT
BEEPING	DEVTYPE	ET
BGCOLOUR_24	DEV_LIST	EW
BGCOLOUR_QL	DEV_NEXT	EX
BGET	DEV_USE	EXEC
BGIMAGE	DEV_USE\$	EXEC_W
BIN	DEV_USEN	EXEP
BIN\$	DIMN	EXF
BLOCK	DIR	EXP
BORDER	DISP_BLANK	EXTRAS
BPUT	DISP_COLOUR	EX_M
CACHE_OFF	DISP_INVERSE	FBKDT
CACHE_ON	DISP_RATE	FDAT
CALL	DISP_SIZE	FDEC\$
CDEC\$	DISP_TYPE	FEP
CHAR_DEF	DLINE	FET
CHAR_INC	DLIST	FEW
CHAR_USE	DMEDIUM_DENSITY	FEX
CHK_HEAP	DMEDIUM_DRIVE\$	FEXP\$
CHR\$	DMEDIUM_FORMAT	FILL
CIRCLE	DMEDIUM_FREE	FILL\$
CIRCLE_R	DMEDIUM_NAME\$	FLASH
CKEYOFF	DMEDIUM_RDONLY	FLEN
CKEYON	DMEDIUM_REMOVE	FLUSH
CLCHP	DMEDIUM_TOTAL	FMAKE_DIR
CLEAR	DMEDIUM_TYPE	FNAME\$
CLOCK	DNEXT	FOPEN
CLOSE	DO	FOP_DIR
CLS	DUP	FOP_IN
CODE	ED	FOP_NEW
COLOUR_24	EDIT	FOP_OVER
COLOUR_NATIVE	ELLIPSE	FORMAT
COLOUR_PAL	ELLIPSE_R	FPOS
COLOUR_QL	EOF	FREE_MEM
CONTINUE	EOFW	FTEST
COPY	EPROM_LOAD	FTYP
COPY_H	ERLIN	FUPDT
COPY_N	ERNUM	FVERS
COPY_O	ERR_BL	FXTRA
COS	ERR_BN	GET
COT	ERR_BO	HEX
CSIZE	ERR_BP	HEX\$
CURDIS	ERR_DF	HGET
CURSEN	ERR_EF	HOME_CSET

HOME_CURR\$	LRESPR	POKE_F
HOME_DEF	LRUN	POKE_L
HOME_DIR\$	MACHINE	POKE_W
HOME_FILE\$	MAKE_DIR	PRINT
HOME_SET	MERGE	PRINT_USING
HOME_VER\$	MODE	PROCESSOR
HOT_CHP	MONTH%	PROGD\$
HOT_CHP1	MOUSE_SPEED	PROG_USE
HOT_CMD	MOUSE_STUFF	PROT_DATE
HOT_DO	MOVE	PROT_MEM
HOT_GETSTUFF\$	MRUN	PUT
HOT_GO	NET	QLOAD
HOT_KEY	NEW	QLRUN
HOT_LIST	NFA_DRIVE	QMERGE
HOT_LOAD	NFA_DRIVE\$	QMRUN
HOT_LOAD1	NFA_USE	QSAVE
HOT_NAME\$	NXJOB	QSAVE_O
HOT_OFF	OJOB	QUERY_JAVA
HOT_PICK	OPEN	QUIT
HOT_REMV	OPEN_DIR	RAD
HOT_RES	OPEN_IN	RAM_USE
HOT_RES1	OPEN_NEW	RANDOMISE
HOT_SET	OPEN_OVER	READ
HOT_STOP	OUTLN	RECHP
HOT_STUFF	OVER	RECOL
HOT_THING	PALETTE_8	RENAME
HOT_THING1	PALETTE_QL	RENUM
HOT_TYPE	PAN	REPORT
HOT_WAKE	PAPER	RESET
HPUT	PARNAM\$	RESPR
IDEC\$	PARSTR\$	RETRY
INK	PARTYP	RJOB
INKEY\$	PARUSE	RND
INPUT	PAUSE	RUN
INSTR_CASE	PEEK	SAVE
INT	PEEK\$	SAVE_O
IO_PRIORITY	PEEKS	SBASIC
JOB\$	PEEKS\$	SBYTES
JOBID	PEEKS_F	SBYTES_O
JOBS	PEEKS_L	SCALE
JOB_NAME	PEEKS_W	SCROLL
JVASGET	PEEK_F	SCR_BASE
JVASPUT	PEEK_L	SCR_LLEN
KBD_TABLE	PEEK_W	SCR_XLIM
KEYROW	PENDOWN	SCR_YLIM
KILLSOUND	PENUP	SDATE
LANGUAGE	PE_BG OFF	SEND_EVENT
LANGUAGE\$	PE_BG ON	SETRATE10
LANG_USE	PI	SETRATE20
LBYTES	PJOB	SET_FBKDT
LEN	POINT	SET_FUPDT
LGET	POINT_R	SET_FVERS
LINE	POKE	SEXEC
LINE_R	POKE\$	SEXEC_O
LIST	POKES	SFA_USE
LN	POKES\$	SFA_USE\$
LOAD	POKES_F	SIN
LOG10	POKES_L	SLUG
LPUT	POKES_W	SOUNDFILE

SOUNDFILE2
SOUNDFILE3
SOUNDSAMPLE
SOUNDSAMPLE2
SOUNDSAMPLE3
SPJOB
SPL
SPLF
SPL_USE
SP_GET
SP_GETCOUNT
SP_JOBOWNPAL
SP_JOBPAL
SP_RESET
SP_SET
SQRT
STAT
STOP

STRIP
SYSSPRLOAD
TAN
TH_FIX
TK2_EXT
TRA
TRUNCATE
TURN
TURNTO
UNDER
UPUT
VER\$
VIEW
WAIT_EVENT
WCOPY
WDEL
WDIR
WEEKDAY%

WGET
WIDTH
WIN_USE
WIN_USE\$
WINDOW
WMON
WM_BLOCK
WM_BORDER
WM_INK
WM_MOVEMODE
WM_PAPER
WM_STRIP
WPUT
WREN
WSTAT
WTV
YEAR%