QPC2

Manual

Revision 5.00

INTRODUCTION

This manual was written on the release of QPC2 version 5.00.

QPC2 is a moving target, therefore the version you have might already incorporate some features that are not mentioned in this manual. Have a look at the supplied version2.txt file for the list of changes.

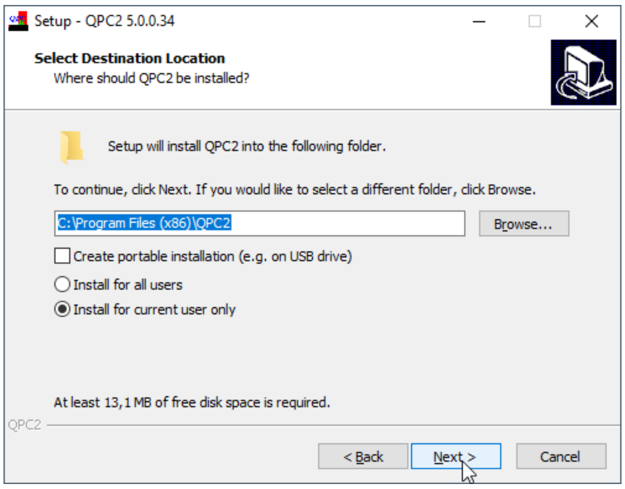
Up-to-date information regarding QPC (current version numbers, new manual revision etc.) can be obtained first hand from my web site at http://www.kilgus.net/

REQUIREMENTS

QPC2 should be run on a PC with at least Windows XP installed, but due to security concerns you should always use a supported version of Windows, which currently means that you should at least use Windows 10. If you have any problems with the graphics it might be prudent to update the DirectX components and the graphics drivers.

QUICK START GUIDE

Start the QPC installation program called something like "QPC2-setup 5.x.x.x.exe". Simply accept the defaults in the following screens, especially in these two:



Setup - QPC2 5.0.0.34

Select Destination Location
Where should QPC2 be installed?

Setup will install QPC2 into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

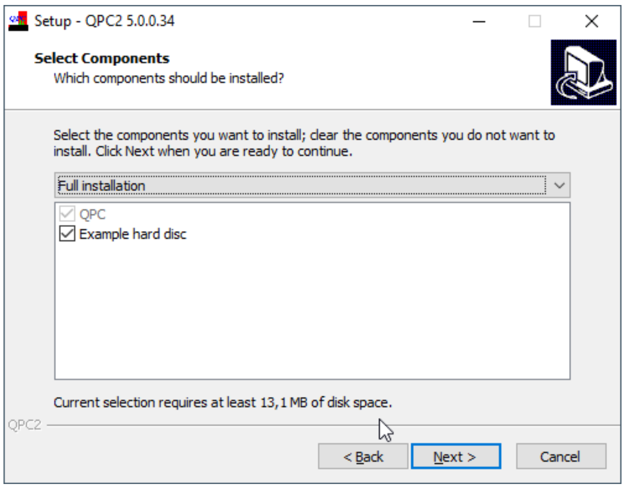
C:\Program Files (x86)\QPC2 — Browse...

☐ Create portable installation (e.g. on USB drive)
○ Install for all users
● Install for current user only

At least 13,1 MB of free disk space is required.

QPC2

< Back | Next > | Cancel



Setup - QPC2 5.0.0.34

Select Components
Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

Full installation

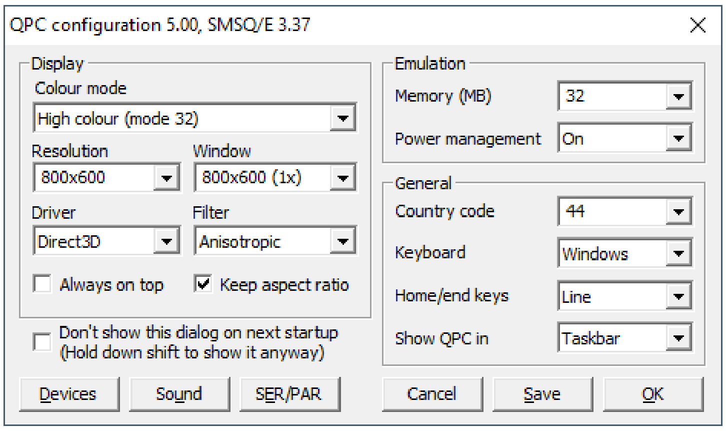
☑ QPC
☑ Example hard disc

Current selection requires at least 13,1 MB of disk space.

QPC2

< Back | Next > | Cancel

QPC already comes with some sensible defaults which you can now tweak a little bit more:



QPC configuration 5.00, SMSQ/E 3.37

Display
Colour mode: High colour (mode 32)
Resolution: 800x600
Window: 800x600 (1x)
Driver: Direct3D
Filter: Anisotropic
☐ Always on top
☑ Keep aspect ratio
☐ Don't show this dialog on next startup (Hold down shift to show it anyway)

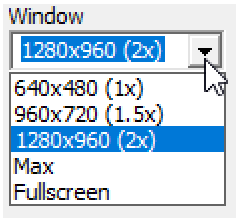
Emulation
Memory (MB): 32
Power management: On

General
Country code: 44
Keyboard: Windows
Home/end keys: Line
Show QPC in: Taskbar

Devices | Sound | SER/PAR | Cancel | Save | OK

Change the display resolution to something suitable for your taste. Starting with QPC2 v4.05 you can also select "max" here, which will always correspond to the resolution of the current monitor.

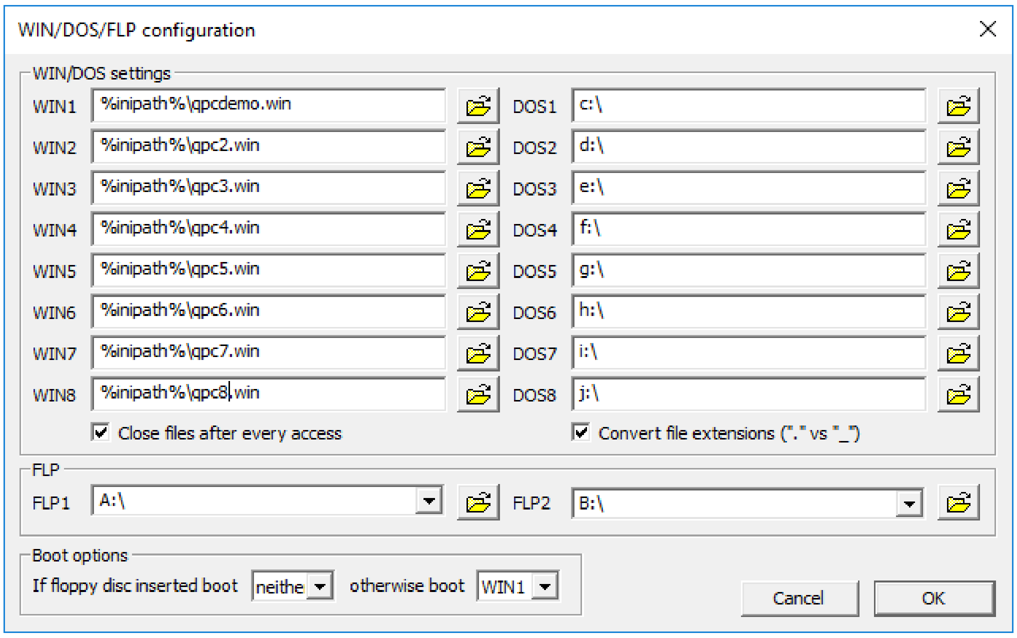
The "window size" is independent of the resolution, so if you increase it the resulting screen will appear larger, this is especially useful in the time of high-resolution screens. The setting dropdown automatically offers some settings with sensible factors, like 1x, 2x etc., providing they fit the monitor:



Window
1280x960 (2x)

640x480 (1x)
960x720 (1.5x)
1280x960 (2x)
Max
Fullscreen

You can also change the country code to the international dialing prefix for your country. If you want to try the supplied qpcdemo.win file, hit the "Devices" button and change the setting for WIN1 to "%INIPATH%\qpcdemo.win":


WIN/DOS/FLP configuration

WIN/DOS settings

WIN1	%inipath%\qpcdemo.win	DOS1	c:\
WIN2	%inipath%\qpc2.win	DOS2	d:\
WIN3	%inipath%\qpc3.win	DOS3	e:\
WIN4	%inipath%\qpc4.win	DOS4	f:\
WIN5	%inipath%\qpc5.win	DOS5	g:\
WIN6	%inipath%\qpc6.win	DOS6	h:\
WIN7	%inipath%\qpc7.win	DOS7	i:\
WIN8	%inipath%\qpc8.win	DOS8	j:\

☑ Close files after every access ☑ Convert file extensions ("." vs "_")

FLP

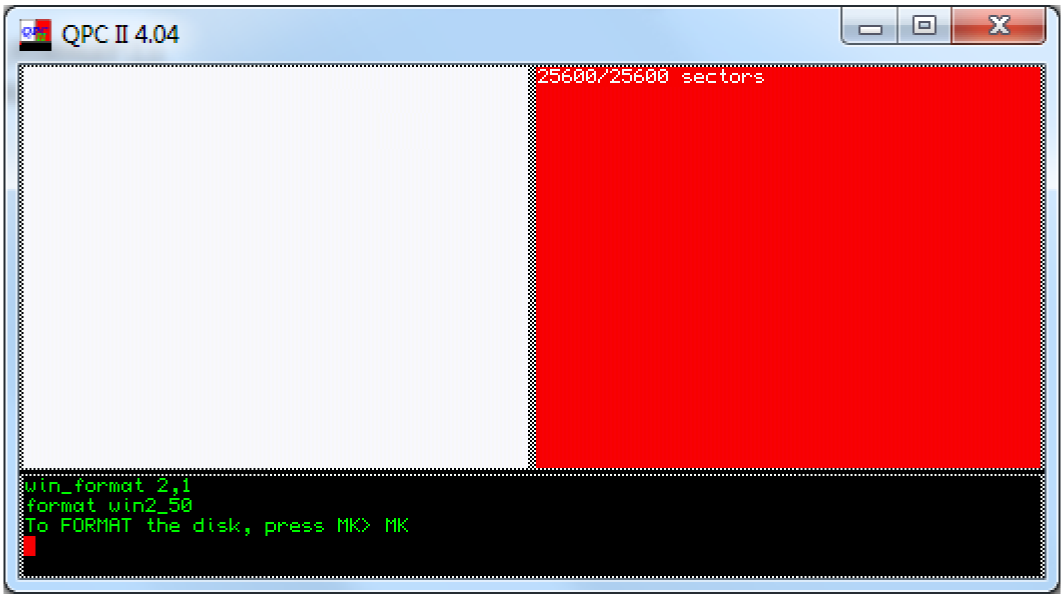
FLP1 A:\ FLP2 B:\

Boot options

If floppy disc inserted boot neithe otherwise boot WIN1

Exit the dialog and hit "Save" to create your first QPC.INI file. The file will be saved in "%APPDATA%\QPC", which will expand to something like "C:\Users\<name>\AppData\Roaming\QPC", depending on your system. **By default all .WIN files are configured to be created there, too!**

If you want to format your own virtual hard disc, for example WIN2 with 50MB, enter "WIN_FORMAT 2,1↩FORMAT WIN2_50↩" and press the keys given in the prompt:



```
25600/25600 sectors
```

```
win_format 2,1
format win2_50
To FORMAT the disk, press MK> MK
```

Congratulations, you've mastered the basic configuration of QPC! But now the real journey of setting up your very own QL system has just begun and this manual cannot really help you in this regard. You can however use the supplied qpcdemo.win file as a starting point, of course.

CONFIGURATION

When running QPC for the first time (and, unless you disable it, also later) you are presented with a configuration dialog from where you can adjust almost all the settings. The dialogs are supplied in three languages: English, German and French. Depending on the system language, your Windows system will automatically decide which one to show.

SETTINGS FILE QPC.INI

Traditionally the settings used to be saved in a Config Level 2 block within the SMSQE.BIN file. This stems from the way all SMSQ/E variants for native hardware are configured. Starting with Windows Vista this posed a problem as the "Program Files" directory was not writable by ordinary users anymore and usually the SMSQE.BIN resided there. It also made updating SMSQ/E a bit hard as you either needed to configure QPC again or use the MenuConfig application to update the new file with your old settings.

Starting with QPC2 v4 the configuration scheme works a bit differently: the default for each setting is still loaded from the SMSQE.BIN file. After applying the defaults however QPC2 starts looking for a file called QPC.INI in these locations and in this order:

1. In the directory the QPC2.EXE file resides
2. In the application data directory that is shared by all users:
    - XP: C:\Document and Settings\All Users\Application Data\QPC
    - Vista/7/10: C:\ProgramData\QPC
3. In the application data directory for the logged in user:
    - XP: C:\Document and Settings\<user>\Application Data\QPC
    - Vista/7/10: C:\Users\<user>\AppData\Roaming\QPC

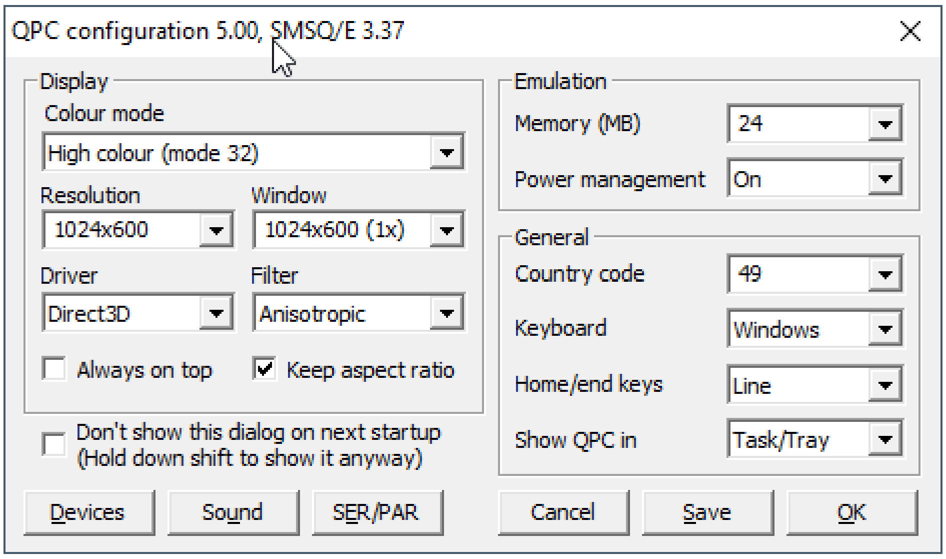
Usually the installer will create a QPC.INI file in one of these locations depending on the type of installation requested ("Portable/USB", "For all users", "Only for this user"). If however no QPC.INI file is found then one will be created in the user specific directory (3.). This also makes it possible to change the configuration even when QPC2 itself is on a read-only medium like a DVD or a shared network drive.

It is recommended to place your .WIN virtual hard drive files in the same directory as QPC.INI! You can then reference them in the configuration with a path like "%INIPATH%\QPC.WIN" and they will always be found.

The QPC.INI file is a simple text file in standard .ini format and can also be edited with any text editor as one pleases.

MAIN DIALOG

The main dialog is divided into 3 sections: display, emulation and general.


QPC configuration 5.00, SMSQ/E 3.37

Display
Colour mode: High colour (mode 32)
Resolution: 1024x600
Window: 1024x600 (1x)
Driver: Direct3D
Filter: Anisotropic
☐ Always on top
☑ Keep aspect ratio
☐ Don't show this dialog on next startup (Hold down shift to show it anyway)

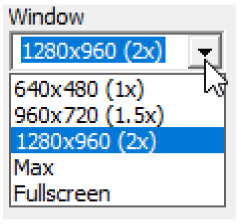
Emulation
Memory (MB): 24
Power management: On

General
Country code: 49
Keyboard: Windows
Home/end keys: Line
Show QPC in: Task/Tray

DISPLAY

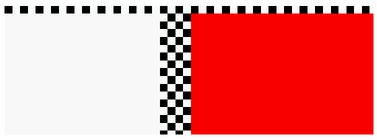
The screen part adjusts anything that affects the QPC window and the SMSQ/E screen emulation. The resolution offered to the left is a mix between common resolutions and the ones your graphics driver supports. Starting with QPC2 v4.05 you can also select “max” here, which will always correspond to the resolution of the current monitor. The X resolution is enforced to be divisible by 8.

The “Window” size is independent of the resolution, so if you increase it the resulting screen will appear larger, this is especially useful in the time of high-resolution screens. The setting dropdown automatically offers some settings with sensible factors, like 1x, 2x etc., providing they fit the monitor:

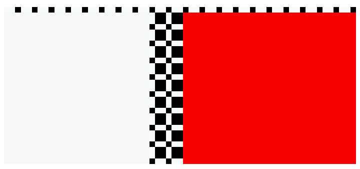


The “Driver” setting decides which display driver will be used. “Direct3D” is based on Direct3D 11 and is the best and most compatible option available. “DirectDraw” is based on the old DirectX3 driver (we’re speaking Windows 95 era here). It’s mostly left in there for emulation layers like WINE that might have better DirectDraw than Direct3D support. For Windows “Direct3D” is the way to go.

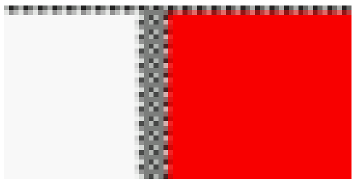
The “Filter” setting is only available for the Direct3D driver. Setting it to “no” will give a blocky look that looks fine when the window size is an exact multiple of the resolution:



but not so much if not:



The same resolution filtered using “Anisotropic” looks a bit washed out but has a much better look in general (especially if not magnified like this example):



The “Keep aspect ratio” option makes sure that the X/Y ratio of the QPC window equals the X/Y ratio of the SMSQ/E resolution. Starting with QPC2 v4.04 this is also enforced when entering full screen mode, so you can use this option to avert stretching if your native display resolution is not divisible by 8 (e.g. 1366x768). This option can also be altered later by using the system menu (left click on the QPC logo in the upper left hand corner, or right click on the QPC button on the task bar). The same applies to the “Always on top” option.

Emulation

In the emulation part you can change how much memory SMSQ/E is allowed to use. Please note that although the drop-down list only offers some selected values, you can manually enter anything up to 255 (MB) into the box.

The second option determines how nice the emulation plays with other applications when it comes to CPU consumption. If “Power management” is turned off, the emulation runs as fast as it can, consuming all CPU power it can get. This setting should usually be avoided. If “Power management” is turned on, the speed of the emulation is reduced if you’re not actively working with QPC and SMSQ/E is not doing anything. The “Suspend” setting takes this one step further and completely halts the emulation when the QPC window is minimized. This might be of use to conserve the last bit of battery power in a laptop, but usually the “on” setting is good enough.

General

In the general section you can set the country code for the default language. This determines the language for error messages and, when the “Keyboard driver” option is set to “SMSQ/E”, the keyboard layout. Possible values, based on the international telephone country codes, are:

1	U.S.A.
33	French
34	Spanish
39	Italian
44	English
49	German

The “Keyboard driver” setting determines which side of the emulation does the translation between key presses and the resulting characters. The default option is “Windows”, which means that the Windows keyboard layout will be used. Loading SMSQ/E keyboard tables does not have any effect in this mode. The “SMSQ/E” option is the old DirectX based driver, which is mostly provided for compatibility reasons. It might not work well anymore with modern PCs as DirectX support in this area is in decline.

“Home/end keys” determine if the keys go to the end of the line (ALT + LEFT/RIGHT) or end of the document (SHIFT + ALT + UP/DOWN).

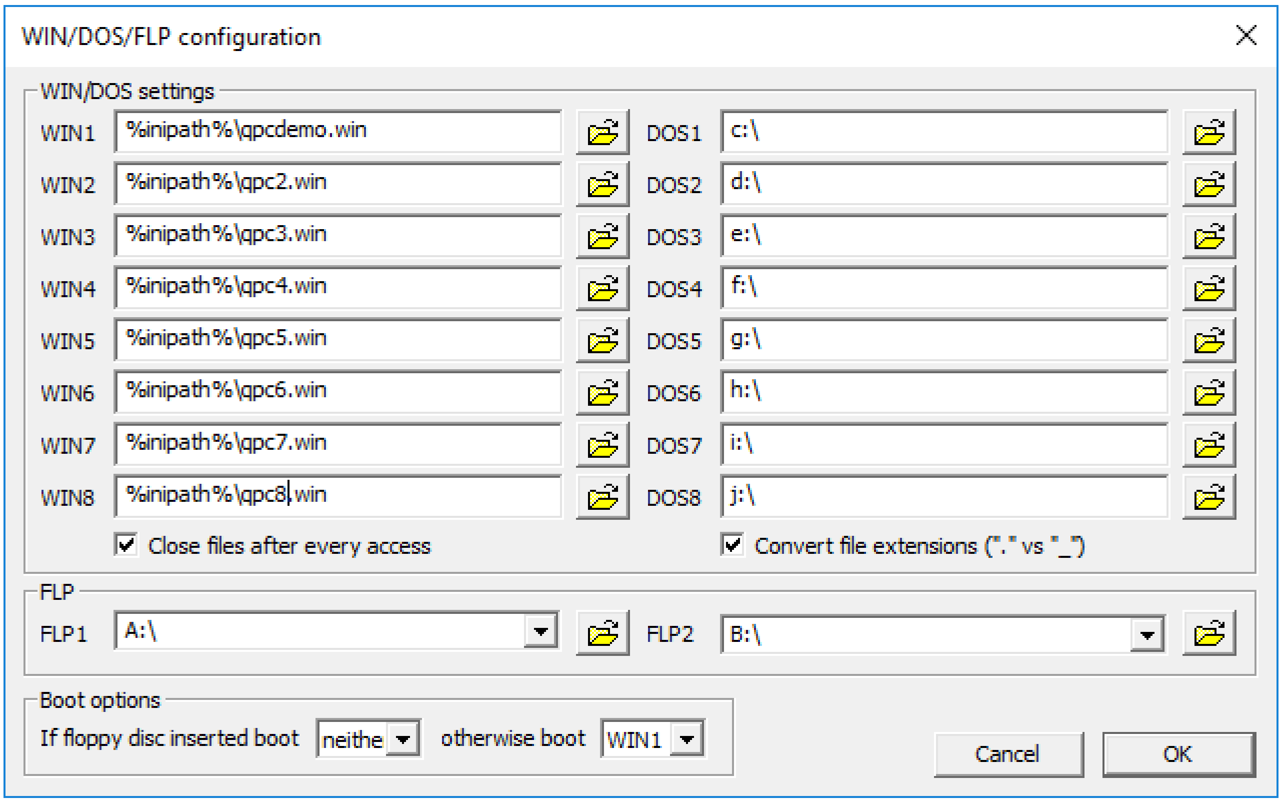
Below that is an option to tell QPC where its button is supposed to show up. “Taskbar” means it is shown along all other application in the Windows taskbar. “System tray” means it shows up in the notification area next to the tasks and “Task/tray” determines that it will show up in the taskbar when visible and in the tray when minimized.

At the bottom of the dialog, you can tell QPC to not show the dialog again on the next start-up unless you’re holding down the shift key. This is the only option that is saved without hitting the “Save” button. If you made any changes other than this option, and you want to retain them, always hit the “Save” button before continuing using “OK”. (For experts trying to put together a QPC distribution: holding CTRL+SHIFT while clicking the “Save” button will write the settings to the SMSQE.BIN file in addition to the QPC.INI file, so that these become the new defaults).

Attention: the “Save” button and the “Don’t show on next start-up” option are only available if QPC can write to the QPC.INI file. Please make sure it is not set to read-only.

Devices dialog

The “devices” dialog unifies all the settings that somehow affect the storage devices WIN, DOS and FLP.



WIN/DOS/FLP configuration

WIN/DOS settings

WIN1	%inipath%\qpcdemo.win	DOS1	c:\
WIN2	%inipath%\qpc2.win	DOS2	d:\
WIN3	%inipath%\qpc3.win	DOS3	e:\
WIN4	%inipath%\qpc4.win	DOS4	f:\
WIN5	%inipath%\qpc5.win	DOS5	g:\
WIN6	%inipath%\qpc6.win	DOS6	h:\
WIN7	%inipath%\qpc7.win	DOS7	i:\
WIN8	%inipath%\qpc8.win	DOS8	j:\

[✓] Close files after every access

[✓] Convert file extensions ("." vs "_")

FLP

FLP1 A:\ FLP2 B:\

Boot options

If floppy disc inserted boot neithe otherwise boot WIN1

WIN

The WIN device represents the native hard drive format for SMSQ/E. You do not need to partition your PC hard drive to use the WIN device, for the PC it is only one big file on one of the ordinary PC drives. In this dialog you can set where this file is located for each WIN drive (WIN1-WIN8). The filename can contain environment variables, with some being defined specifically by QPC:

%INIPATH%	The path to the QPC.INI file
%QPCPATH%	The path to the QPC.EXE file
%SMSQPATH%	The path to the SMSQE.BIN file

Example: %INIPATH%\QPC.WIN

"Close files after every access" will close the WIN file when no SMSQ/E file is open on them. This makes it easy to share a file between multiple QPC instances.

DOS

Next you can alter the settings of the DOS device. The DOS device grants SMSQ/E direct access to all the drives Windows can see. One limitation however is that the maximum file name is still restricted to 36 characters. This means that some deeper directory levels may not be accessible. To rectify this situation every DOS drive has a configurable base directory, so a "DIR dos1_" might actually access "C:\Documents\Important\Long\". You can connect a DOS drive to any directory in the system and even to shares on remote PCs, like for example

C:\WINDOWS\
\\REMOTEPC\SOMESHARE\

Enabling "Convert file extensions" will convert a file called "test_asm" on the SMSQ/E side into "test.asm" on the PC side and vice versa.

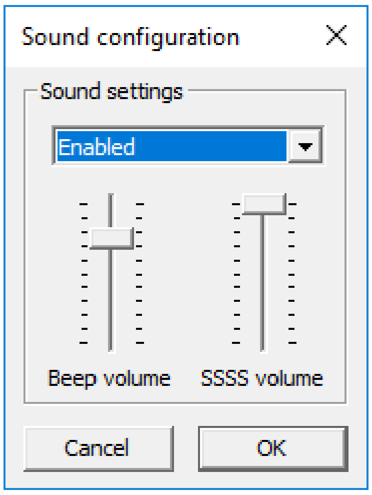
FLP

Starting with v4 QPC2 supports virtual floppy images. If the box is set to "A:\" or "B:\" then the physical drive will be used, otherwise QPC2 tries to read/write to an image file of a floppy. You can change and read back these settings at runtime using the FLP_DRIVE and FLP_DRIVE$ functions.

BOOT OPTIONS

Lastly you can choose on which device SMSQ should look for the boot program on start up. Here only WIN and FLP is available as an option.

SOUND DIALOG

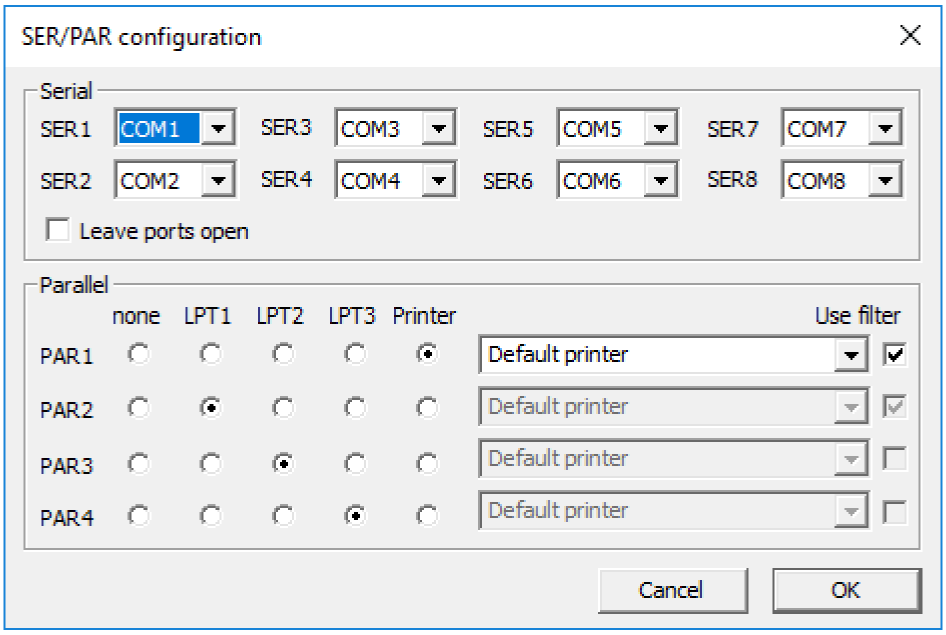

Sound configuration

Sound settings: Enabled

In this dialog you can adjust the volume of the beeper emulation and the sampled sound system ("SSSS"). Also you can set whether sound is enabled, disabled or in automatic mode. See the "beeper" chapter below for more details on this, but starting with Windows XP you can safely keep the sound turned on all the time.

SER/PAR DIALOG

This dialog determines the emulation's SER -> COM binding. When enabling the "Leave ports open" option, COM ports, once opened, are never closed again until QPC is exited. This was only included by special request, most people won't need it.



SER/PAR configuration

Serial

SER1	COM1	SER3	COM3	SER5	COM5	SER7	COM7
SER2	COM2	SER4	COM4	SER6	COM6	SER8	COM8

☐ Leave ports open

Parallel

	none	LPT1	LPT2	LPT3	Printer		Use filter
PAR1	○	○	○	○	●	Default printer	✓
PAR2	○	●	○	○	○	Default printer	✓
PAR3	○	○	●	○	○	Default printer	☐
PAR4	○	○	○	●	○	Default printer	☐

In the parallel section you can set the destination for each PAR port. For traditional parallel printers the LPT1-LPT3 options might be sufficient, for all others (including USB and network printers) choose the “printer” option and select the printer in the drop-down list. One specialty in the list is the “Default printer” option which always corresponds to Windows's default printer.

The “user filter” option is only available if QPC2 was able to find the QPCPrint printer emulator. In this case all print output is put through the filter before being sent to the printer.

COMMAND LINE

QPC has a few options that can already be supplied on startup. Those are:

-? or -h	Show a short summary of the options available.
-cmdline "Text”	The contents of "Text" can be read using the QPC_CMDLINE$ function. Other than that, it has no effect on QPC.
-ini “file”	Specifies a different configuration INI file that will be used in place of the normal QPC.INI
-os “file”	Tell QPC where to look for the operating system. Usually this should not be needed anymore as the configuration can now be specified by a custom INI file
-stretch	Force enabling that QPC can stretch the SMSQ/E display content to a different window size. This is only needed if the graphics driver is lying about its ability to perform a stretch operation.

SOME TECHNICAL EXPLANATIONS

PROCESSOR EMULATION

- QPC emulates the complete 68020 instruction set, except the CALLM/RETM instructions.
- All illegal instructions and all addressing modes are trapped.
- The F-line emulator works as usual, the A-line emulator is however only used for internal communication between SMSQ/E and QPC2.
- Unlike earlier QPC versions, instructions on odd addresses do cause an address fault exception.

SCREEN EMULATION

- The screen is not updated periodically. QPC2 checks any write access for screen operations and acts accordingly. This is not true for instructions that access the stack explicitly (JSR, BSR, PEA...).
- The hardware register at address $18063 is kept up-to-date by SMSQ/E, but changes to it have no effect.
- As one is likely to be using higher resolutions than the original 512x256 most of the time, the base of screen memory will differ from the original $20000. This might cause problems for old programs (although not as often as before, see also next point). However, if you set the resolution to exactly 512x256, the base of system memory will again be at $20000.
- Version 3 introduced a special compatibility feature: QPC2 can map the original video memory into the first 512x256 pixels of a higher resolution screen, and even into the 16bit colour mode. See the description of the QPC_QLSCREMU command for details.

KEYBOARD

There are two keyboard drivers integrated into QPC called "Windows" and "SMSQ/E". The "Windows" driver is the newer one and was written because support for the old "SMSQ/E" method of keyboard access is rapidly declining within Windows.

The Home/End keys of the "Windows" driver work as under Windows, they go to the first respectively last character in a line. Many SMSQ/E style keyboard drivers did this differently, where they could be used to jump to the top/bottom of the complete document. The behavior can be changed in the configuration dialog.

REAL TIME CLOCK

The SMSQ/E clock is linked to the PC clock so that a change in one also affects the other. There is no PROT_DATE functionality like in some native SMSQ/E variants. Due to the way the clock is emulated it can be wrong for a few seconds after resuming the PC from standby or suspend mode.

BEEPER

Since version 3.02, the beeper is no longer emulated through the PC speaker (this was only possible under Win9x/ME anyway) but through the soundcard by using the DirectSound interface. This way all Windows systems can benefit from the emulation. The emulation has been completely rewritten for this purpose and the sound is now quite similar to the original beep.

Sound emulation can be configured to be enabled (default), disabled, or enabled automatically. The reason this item is optional is that under some Windows versions DirectSound applications can work simultaneously with other DirectSound programs but not with old waveform applications. This means that after starting QPC, some applications can't emit any more sounds, even when QPC is not beeping. Now the emulation can be completely switched off or, if you still want the occasional beep and have other applications work correctly, you can set it to "automatic" mode. In this mode, DirectSound is enabled before every beep and released again immediately. The drawback is that this procedure needs some time (there's some "latency") thus short beeps may simply get lost (i.e. the sound is switched off again before the soundcard has time react).

Furthermore there is an option to set the default volume of the beeper emulation. This is useful because an emulated beep can be much louder than normal sounds.

SMSQ/E Samples Sound System (SSSS)

QPC can play sampled audio data. This has been implemented compatibly to the SMSQ/E Sampled Sound System (sometimes also called QLSSS – QL Sampled Sound System), which had been developed for the Q40. In principle every program written to support the SSSS should also work with QPC. I'd especially like to point out Simon Goodwin's SOUND device, it's worth a try.

Fast memory

One problem with earlier SMSQ/E implementations is the so-called "slaving" mechanism. "Slaving" means that the system uses unoccupied memory to cache accesses to slow disk drives, such as microdrives. On the original QL this was a good and clever idea. However, as memory sizes increased, the method employed became increasingly inefficient as it tried to scan through the slave blocks in search of the required information. With QPC it gets even worse, here access to a virtual hard disk is always faster than the slaving cache, as Windows already caches such accesses. This resulted in disk operations becoming slower and slower the more memory QPC was set up to use.

It is not easy to disable the slaving mechanism, or to limit it to a certain maximum memory size. Therefore a different approach was chosen based on so-called called "fast memory". Fast memory originates from the Atari systems, which could have a region of memory faster than the ordinary system memory, but which was at the same time unsuitable for slaving (because it couldn't be directly read/written by the disk drives). Furthermore, there was a gap between the slow and fast memory areas.

What was done in the end was to let the whole SMSQ/E memory map (including the "common heap", the area where the common memory allocations take place) remain in the slow memory area, while the fast memory was included with the TPA (transient program area, originally the area where programs executed by "EX" etc. are loaded). If a program now allocates memory, this is served from the TPA first (instead of the common heap as in traditional systems) and the common heap is only used when the TPA area becomes full. This means that slaving is limited to the area set aside as slow memory.

With the release of QPC2 v3.03 a very similar approach is used. Essentially QPC just declares the first megabyte of RAM as "slow" memory and the rest as "fast". This way slaving only takes place in a small memory area and is thus speeded up considerably. All other memory allocations are served out of the huge "fast" pool. One immediate effect is that programs that try to determine how much free memory the system has will only return the free space within the "slow" memory, i.e. only about 800 kilobytes although there are some more megabytes waiting in the "fast" area. The FREE_MEM function has already been modified to return the right value, other applications might still have to be adapted (for programmers: sms.frtp returns the value you're looking for, don't try to calculate the free memory yourself).

Furthermore, you might experience some strange behaviour from the Sysmon program, and the QPAC2 “Jobs” window can show “Memory corrupt”. These programs have been updated to work correctly again.

MOUSE

QPC supports the mouse wheel transparently to the applications. For every tick of the mouse wheel several Alt + up/down key combinations are stuffed into the keyboard buffer. The amount depends on the global Windows setting. As Alt + up/down are the standard key-combination for scrolling a window, many applications can immediately profit from the wheel.

MOUSE_SPEED

Syntax: MOUSE_SPEED [#ch,] acceleration, wakeup

This function adjusts the mouse acceleration and wake up factor. The acceleration factor is of no consequence to QPC2. The wakeup values, however, may still be set. They range from 1 to 9, with 1 being the most sensitive.

MOUSE_STUFF

Syntax: MOUSE_STUFF [#ch,] hot$

This function adjusts the string that is stuffed into the keyboard queue when the middle mouse button is pressed (or both left and right buttons are pressed simultaneously). The string cannot be longer than two characters, but this is enough to trigger any hotkey, which in turn, can do almost anything.

E.g.:	**MOUSE_STUFF ‘.’**	**Generates a dot if middle mouse button is pressed**
	MOUSE_STUFF CHR$(255)&’.’	**Generates hotkey Alt + .**

MACHINE TYPE

There are two standard functions to determine the machine type

MACHINE

Syntax: mach% = MACHINE

Returns the machine type SMSQ/E is running on; 30 for QPC.

PROCESSOR

Syntax: proc% = PROCESSOR

Returns the 680x0 family type; 20 for QPC (10 for versions below 3.33).

E.g.: **IF MACHINE=30 AND PROCESSOR<>20:PRINT 'This can hardly be QPC!'**

QPC CONTROL

QPC2 reacts to two special key combinations:

SHIFT + CTRL + ScrLock: This key combination terminates the emulation. Unlike the QXL, which has its own memory and can resume a session later on, QPC is really terminated and the memory is given up. Therefore you should make sure that any data in your SMSQ/E applications is saved, that all files are closed etc. For safety, you might want to wait one or two seconds in case something is still in the drive caches.

In windowed mode, you can also use the standard ways of quitting a Windows application, e.g. the “X” button in the title bar, or go via the system menu.

SHIFT + CTRL + F12: This combination is used to switch between window- and full screen mode.

QPC-SPECIFIC COMMANDS

QPC_CMDLINE$
Syntax: cmd$ = QPC_CMDLINE$

This returns the argument that was supplied to QPC after the "-cmdline" command line argument. This can be used to do different actions depending on the way QPC was started.

QPC_EXEC
Syntax: QPC_EXEC command$[, parameter$]

This command can be used to call an external DOS or Windows program. The name of the executable file is given in the first parameter. Optionally, you can also supply a second parameter, which is then passed to the executed program as its command line arguments.

Furthermore, you can supply a data file as the first parameter. In this case, the associated application for this file type is executed.

E.g.:	**QPC_EXEC 'notepad','c:\text.txt'**	**Start notepad and load the c:\text file**
	QPC_EXEC 'c:\text.txt'	**Start the default viewer for .txt files**

QPC_EXIT
Syntax: QPC_EXIT

This simply quits QPC.

QPC_HOSTOS
Syntax: os% = QPC_HOSTOS

This function returns the host operating system under which QPC was started. Possible return codes are:

0 = DOS	(QPC1)
1 = Win9x/ME	(QPC2)
2 = WinNT/2000/XP	(QPC2)

QPC_MAXIMIZE, QPC_MINIMIZE, QPC_RESTORE
Syntax: QPC_MAXIMIZE, QPC_MINIMIZE, QPC_RESTORE

Maximizes, minimizes or restores the QPC window.

QPC_MSPEED
Syntax: QPC_MSPEED x_accel, y_accel

This command has no effect on QPC2.

QPC_NETNAME$
Syntax: name$ = QPC_NETNAME$

This function returns the current network name of your PC (the one you supplied upon installation of Windows). The result can be used to distinguish between different PCs (e.g. in a BOOT program).

QPC_QLSCREMU
Syntax: QPC_QLSCREMU value

Enables or disables the original QL screen emulation. When emulating the original screen, all memory write accesses to the area $20000-$207FFF are intercepted and translated into writes to the first 512x256 pixels of the big screen area. If the screen is in high colour mode, additional colour conversion is done.

Possible values are:

-1: automatic mode
 0: disabled (default)
 4: force to 4-colour mode
 8: force to 8-colour mode

When in QL colour mode, the emulation just transfers the written bytes to the larger screen memory, i.e. when the big mode is in 4-colour mode, the original screen area is also treated as 4-colour mode. In high colour mode however, the colour conversion can do both modes. In this case, you can pre-select the emulated mode (parameter = 4 or 8) or let the last issued MODE call decide (automatic mode). Please note that that automatic mode does not work on a per-job basis, so any job that issues a MODE command changes the behaviour globally.

Please also note that this transition is one-way only, i.e. bytes written legally to the first 512x256 pixels are not transferred back to the original QL screen (in the case of a high colour screens this would hardly be possible anyway). Unfortunately, this also means that not all old programs will run perfectly with this type of emulation. If you experience problems, start the misbehaving application in 512x256 mode.

QPC_SYNCSCRAP
Syntax: QPC_SYNCSCRAP

In order to rapidly exchange text passages between Windows and SMSQ/E the Syncscrap functionality has been introduced. The equivalent of the Windows clipboard is the scrap extension of the menu extensions. After loading the menu extensions you can call this command, which creates a job that periodically checks for changes in either the scrap or the Windows clipboard, and synchronizes their contents if necessary. Please note that only text data is supported. The character conversion between the QL character set and the Windows ANSI set is done automatically. The line terminators (LF or LF+CR) are converted too.

QPC_VER$
Syntax: v$ = QPC_VER$

This returns the current QPC version.

E.g. PRINT QPC_VER$ will print 4.00 or higher.

QPC_WINDOWSIZE
Syntax: QPC_WINDOWSIZE x, y

This sets the size of the client area (the part that displays SMSQ/E) of the QPC window. It does NOT alter the resolution SMSQ/E runs with, so the pixels are effectively zoomed. It is equivalent to the "window size" option in the main configuration window. If QPC is currently in full screen mode it will switch to windowed mode. Window size cannot be set smaller than the SMSQ/E resolution or bigger than the desktop resolution.

E.g. DISP_SIZE 512,256

QPC_WINDOWSIZE 1024,512 do a 200% zoom of the QPC window

QPC_WINDOWTITLE
Syntax: QPC_WINDOWTITLE title$

Sets the string that can be seen when QPC runs in windowed mode. This can be used to easily distinguish between several QPC instances.

E.g. QPC_WINDOWTITLE “Accounting” sets the title to “Accounting”

Serial (COM) Ports

Unlike the basic SMSQ serial port drivers, the SMSQ/E serial port drivers are dynamically buffered. There is therefore no need to use the PRT device.

The Baud rates supported by SMSQ/E on QPC are

115200
57600
38400
19200
9600
4800
2400
1200
600
300

The BAUD command works as in SMSQ/E on the Atari: If the port number is omitted, only SER1 is affected:

BAUD 19200	**set SER1 to 19200**
BAUD 3,115200	**set SER3 to 115200 baud**

SER_GETPORT$

Syntax: com$ = SER_GETPORT$(port%)

Returns the device the SER port is connected to, for example "COM1".

SER_SETPORT

Syntax: SER_SETPORT port%, com$

Sets the COM port a SER port should be connected with. The change will take effect on the next open of the specified serial port.

SER_SETPORT 4,"COM32"	**Associate SER4 with COM32**

Printer support (PAR)

The PAR device can be linked to a printer port or to a Windows printer queue. By default, output is dynamically buffered: the PRT device is not required.

Please note that no translation of the printer data is done, the data sent to the device is directly piped into the printer itself. The only exception is the “filter” option in the configuration dialog. A filter gets the raw data and can process it however it wants. QPCPrint, which is sold separately, is just such a filter that emulates the popular ESC P/2 printer language.

PAR_DEFAULTPRINTER$
Syntax: name$ = PAR_DEFAULTPRINTER$

This returns the name of Windows' default printer. The name can later be used with PAR_SETPRINTER for example.

PAR_GETPRINTER$
Syntax: name$ = PAR_GETPRINTER$(port%)

This returns the PAR port setting: "LPT1", "LPT2" or "LPT3" if it isn't linked to a printer but directly to a printer port or the name of the printer otherwise. An empty string designates the default printer.

PAR_SETPRINTER
Syntax: PAR_SETPRINTER port%, name$

Connects the PAR port either to a hardware port (e.g. name$ is "LPT1") or to the printer spooler (name$ is one of the names returned by PAR_PRINTERNAME$).

PAR_GETFILTER
Syntax: state% = PAR_GETFILTER(port%)

This returns whether the printer filter is enabled for the specified port.

PAR_SETFILTER
Syntax: PAR_SETFILTER port%, state%

Enables (state% = 1) or disables (state% = 0) the printer filter for the specified port. If the printer should be enabled although none is available a “not found” error is returned.

PAR_PRINTERCOUNT
Syntax: n% = PAR_PRINTERCOUNT

This returns the number of printers available on this system.

PAR_PRINTERNAME$
Syntax: name$ = PAR_PRINTERNAME$(n)

This returns the name of printer number n (counted from 1 to PAR_PRINTERCOUNT).

TCP/IP

QPC possesses TCP/IP capabilities in form of the TCP, UDP and SCK devices. For this it relies on the Windows TCP/IP stack, so if applications like the Internet Explorer can access the net and you're not behind a proxy server then QPC can do it, too.

This manual won't go into the details of the interface, however, as the interface is mostly compatible to the uQLx implementation all documentation for that is also valid for QPC.

Attention WIN95 users: QPC2 now needs a Winsock2 capable system to work. If you don't have it already please download and install the Winsock2 update for Win95 from the Microsoft homepage.

PC Floppy Disks

QPC2 supports both native floppy access and access to floppy images.

Native floppy support

Access to native PC floppy disks is done via low-level Windows calls. Read and write accesses are buffered internally by QPC2 and should be quite fast. As of version 4 QPC does not physically format floppy disks anymore. The disk must already have been formatted by Windows or any other means. Formatting the disk in QPC only writes the SMSQ/E file system to it.

Floppy image support

QPC2 can accept any standard floppy disk image. You can create your own images by formatting an image and filling it with contents:

FLP_DRIVE 2,"C:\TEMP\FLOPPY.IMG"	**change location of image file**
FORMAT FLP2_	**format HD floppy image**
WCOPY FLP1_,FLP2_	**copy contents from physical floppy**

Floppy Disk Driver Name

The default name of the floppy disk driver is FLP.

FLP Control Commands

FLP_USE

FLP_USE may be used to set the name of the FLP device. The name should be three characters long, in upper or lower case.

FLP_USE mdv	**The FLP device is renamed MDV**
FLP_USE FLP	**The FLP device is restored to FLP**
FLP_USE	**The FLP device is restored to FLP**

FLP_DRIVE

Syntax: FLP_DRIVE drive%, drive$

This changes the drive/image the floppy device is connected to.

FLP_DRIVE 2,"C:\FLOPPY.IMG"	**now FLP2_ is assigned to the floppy image FLOPPY.IMG**
FLP_DRIVE 2,"B:\"	**now FLP2_ is assigned to the physical B:\ floppy**

FLP_DRIVE$

Syntax: drive$ = FLP_DRIVE$(drive%)

This reads back the current connection of the floppy device.

PRINT FLP_DRIVE$(2)	**will tell you the current setting**

FLP_DENSITY

Syntax: FLP_DENSITY code

The SMSQ/E format routines will usually attempt to format a disk to the highest density the medium supports. The FLP_DENSITY (code) is used to specify a particular density during format. The density codes are "S" for single sided (double density), "D" for double density and "H" for high density.

FLP_DENSITY S	**Set the default format to single sided**
FLP_DENSITY H	**Set the default format to high density**
FLP_DENSITY	**Reset to automatic density selection**

The same code letters may be added (after a *) to the end of the medium name to force a particular density format. (For compatibility with older drivers, if the code letter is omitted after the *, single sided format is assumed).

FORMAT 'FLP1_Disk23'	**Format at highest density or as specified by FLP_DENSITY**
FORMAT 'FLP1_Disk24*'	**Format single sided**
FORMAT 'FLP1_Disk25*S'	**Format single sided**
FORMAT 'FLP1_Disk25*D'	**Format double sided, double density**

FLP_SEC, FLP_START and FLP_STEP

QPC has no influence over how the Windows disk driver works, therefore these commands are ignored.

WIN DISKS

SMSQ hard disks for QPC are just large files on the host operating system's file system. The files usually have the suffix ".WIN" but anything else is fine, too. Name and directory can be configured separately for all drives. (See also the configuration section). SMSQ/E's FORMAT command creates the file.

HARD DISK DRIVER NAME

The default name of the hard disk driver is WIN.

WIN_USE

WIN_USE may be used to set the name of the WIN device. The name should be three characters long, in upper or lower case.

WIN_USE mdv	**The WIN device is renamed MDV**
WIN_USE WIN	**The WIN device is restored to WIN**
WIN_USE	**The WIN device is restored to WIN**

FORMATTING

Formatting a WIN drive simply creates a large file on the PC's hard disk. The "name" of the WIN device should be the size required in megabytes.

Before you issue the FORMAT command, you have to allow the drive to be formatted. SMSQ/E has a two-level protection scheme to make sure you (or somebody else) cannot format your hard disk accidentally. All drives are protected by default so you have to enable them to be formatted first.

Please note that the FORMAT command for a WIN drive should only be used from the console of job 0, i.e. the first SBASIC.

FORMAT will fail if there is not sufficient space left on the specified drive, if the medium is write-protected, or if the file *.WIN already exists and contains invalid information (e.g. a DOS-subdirectory).

WIN_FORMAT 1	**Allow WIN1_ to be formatted**
FORMAT WIN1_10	**Create a 10 Megabyte WIN device.**
	... You have to echo the two characters displayed...
WIN_FORMAT 1,0	**protect WIN1_ again against unwanted formatting**

DRIVE/FILENAME ASSIGNMENT

Every WIN drive number is assigned to a file on your hard disk, which contains the complete contents of your WINx_. It is possible to change the filename of the file that is assigned to a WIN drive number while QPC2 is running:

WIN_DRIVE 2,"D:\QPC.WIN"	**now WIN2_ is assigned to the WIN file QPC.WIN**
PRINT WIN_DRIVE$(2)	**will tell you the current filename.**

Removable drives, such as ZIP or SyQuest, are now supported. If auto-detection fails use the WIN_REMV command:

WIN_REMV 2	**declares WIN2_ to be a removable drive.**
WIN_REMV 2,1	**does the same for WIN2_.**
WIN_REMV 2,0	**magic - WIN2_ is no longer removable!**

When a drive is declared removable, the .WIN file is closed after all SMSQ/E files on it are closed. This can also be used to allow a single .WIN file to be shared over a network. (Files on a remote computer are automatically set to be removable). As long as one instance of QPC has open files on the drive, no other instance can access it.

THE DOS DEVICE

The DOS device has been created to transfer data between the Windows and SMSQ/E environments. Using this device you can directly browse your PC hard disks (network drives, CD-ROMs or whatever), as well as read and write files.

Since v5.00 the DOS device can be used as a full replacement for native drives like the WIN device. It stores SMSQ file header information if needed by prepending QemuLator compatible headers to the files. I have thought long and hard about which mechanism to use and while all have certain drawbacks this seems to be the best compromise.

And to better play along with the host operating system file extensions are automatically converted to PC standard and vice versa, so a file "test_asm" on the QL side will become the file "test.asm" on the PC side. This option can be disabled in the configuration.

DRIVE/DIRECTORY ASSIGNMENT

By default, DOS1_ corresponds to C:\, DOS2_ to D:\ and so on, but the base can be freely chosen in the configuration dialog or even at runtime:

DOS_DRIVE 2,"C:\WINDOWS"	**assign DOS2_ to the windows directory**
PRINT DOS_DRIVE$(2)	**would now return "C:\WINDOWS"**

RESTRICTIONS AND SOME BACKGROUND INFORMATION ON THE DOS DEVICE

You can use this device in the same way as any other QL directory device to access and exchange files between Windows and SMSQ/E. It is easier than ever before. The usual restrictions imposed by the general QDOS file naming convention apply, i.e. the length of the directory + filename is limited to 36 characters. Names longer than that won't show up in the directory lists! Therefore, it is a good idea to place files that you want to access from both SMSQ/E and Windows only one or two directory levels deep, or change the base of a DOS drive to one directly above the desired directories.

Many filenames that are valid under SMSQ are not valid under Windows. The offending characters (e.g. *, /, ? etc. or filenames with spaces at their end) are translated into other, valid ANSI characters. This conversion works quite well, but you are advised to only use valid filenames wherever possible.

One problem with the SMSQ way of accessing files is that the "_" separator can be a valid part of a name or a directory separator. Therefore, the relation SMSQ filename -> Windows filename is ambiguous. This can cause some problems:

Let's say you have two directories named C:\QL\STUFF\ and C:\QL\STUFF_NEW\ and you want to create a file called dos1_QL_STUFF_NEW_BRANDNEW.TXT. Where does that file belong? It could mean any of the following:

```
C:\QL_STUFF_NEW_BRANDNEW.TXT
C:\QL\STUFF_NEW_BRANDNEW.TXT
C:\QL\STUFF\NEW_BRANDNEW.TXT
C:\QL\STUFF_NEW\BRANDNEW.TXT
```

Your intention was probably the last one, but how should QPC now? The easy solution is not to use underscores in directory names. However, if you can't help it, it becomes essential to know how the DOS device works. The current algorithm is based on the simple assumption that if you have a directory called "QL_STUFF" you won't also want to create "QL\STUFF".

The exact working of the algorithm is not easy to describe, but I'll try nonetheless. The basic principle is that the algorithm always searches for the longest consecutive part of the name. In the example above, QPC would begin by searching for any directory starting with "C:\QL". If none was found, the process completes and the result is simply "C:\QL_STUFF_NEW_BRANDNEW.TXT". Otherwise, it will look for any directory starting with "C:\QL_STUFF". If found QPC will try "C:\QL_STUFF_NEW" and so on. If not found, however, it will test whether the last successful part ("C:\QL_STUFF") is itself a directory. If it is, it is considered part of the filename and all future searches use this as the base (i.e. the next step would be "C:\QL_STUFF\NEW"). If not, the search terminates with the result again being "C:\QL_STUFF_NEW_BRANDNEW.TXT".

If this sounds too confusing or too badly explained (probably both) just remember one thing: never use "_" within directory names.

DOS Control Commands

DOS_USE

DOS_USE may be used to set the name of the DOS device. The name should be three characters long, in upper or lower case.

DOS_USE mdv	**The DOS device is renamed MDV**
DOS_USE DOS	**The DOS device is restored to DOS**
DOS_USE	**The DOS device is restored to DOS**

DOS_DRIVE

Syntax: DOS_DRIVE drive%, directory$

This changes the directory the DOS device is connected to.

DOS_DRIVE 2,"C:\WINDOWS" now DOS2_ points to C:\WINDOWS

Environment variables are supported, too:

DOS_DRIVE 2,"%USERPROFILE%"	**now DOS2_ points to C:\Users\<your-user>**
DOS_DRIVE 2,"%INI_PATH%"	**now DOS2_ points to the path QPC.ini lies in**

DOS_DRIVE$

Syntax: directory$ = DOS_DRIVE$(drive%)

This reads back the currently connected directory of the DOS device.

THE QPC CD-AUDIO MODULE

As a little extra bonus, QPC contains a module to play Audio CDs. There are 23 new BASIC commands for the complete control of all audio functions of a CD-ROM drive. A tiny CD-player comes as an example. The functions of the player are described at the end of this chapter.

First some terms of CD programming:

Track:	one title
Frame:	one sector of a CD. The sector length on Audio CDs is 2352 Bytes

REDBOOK-Format: a standard format for direct sector addressing. Sectors are addressed through a time index in the form of a longword formatted as $00MMSSFF. MM is the minute, SS the second and FF is the frame. One second has 44100(Hz)*2(Stereo)*2(16 Bit)/2352 (sector length) = 75 frames.

HSG-Format: another format to address a sector. Here they are only addressed sequentially. HSG=(minute*60+second)*75+frame

NEW BASIC COMMANDS

As usual, all parameters in square brackets are optional.

Unless specified, all sectors are addressed in Redbook-Format.

CD_INIT
Syntax: CD_INIT ['name']

This command must be used before any other in order to initialize the CD drive for SMSQ. After the first call, the command is ignored on all subsequent calls. The string parameter is ignored on QPC2.

CD_PLAY
Syntax: CD_PLAY [start[,end]]

This is the most important command. Without parameters the whole CD is played. An optional start and end track can be given. The command returns as soon as the CD starts playing. The parameters are given in tracks (bit 31 clear) or in sector units (bit 31 set).

E.g.: CD_PLAY 3 or with the same effect
CD_PLAY CD_TRACKSTART(3) + $80000000

CD_STOP
Pauses playing. If the driver was already in pause mode, a complete stop is performed (as if a new CD was inserted; restart from track 1 and so on)

CD_RESUME
Resumes playing from where it stopped.

CD_EJECT, CD_CLOSE
Opens/closes the drive tray.

CD_ISPLAYING, CD_ISCLOSED, CD_ISINSERTED, CD_ISPAUSED
Syntax: x% = CD_xxx

These functions return the current status according to the keyword. Please note that Windows cannot tell whether the tray is closed or not, therefore CD_ISCLOSED always returns the same result as CD_ISINSERTED when used on QPC2. An empty tray was obviously something the Microsoft geniuses could not imagine.

CD_TRACK
Syntax: track% = CD_TRACK

Returns the number of the track currently being played.

CD_TRACKTIME
Syntax: x = CD_TRACKTIME

Returns the elapsed time within the current track.

CD_ALLTIME
Syntax: x = CD_ALLTIME

Returns the total elapsed time of the CD.

CD_HSG2RED, CD_RED2HSG
Syntax: red=CD_HSG2RED hsg
hsg=CD_RED2HSG red

Converts an HSG address to Redbook and vice versa.

CD_TRACKSTART
Syntax: x = CD_TRACKSTART track

Returns the start sector of a track.

CD_TRACKLENGTH
Syntax: x = CD_TRACKLENGTH track

Returns the length of a track.

Attention: This is the only function that returns an HSG-number.

CD_FIRSTTRACK, CD_LASTTRACK
Syntax x% = CD_xxx

Returns the number of the first/last track.

CD_LENGTH
Syntax: x = CD_LENGTH

Returns the total length of the CD.

CD_HOUR, CD_MINUTE, CD_SECOND
Syntax: x% = CD_xxx Redbook

Returns the hour, minute or second of a Redbook address.

TROUBLESHOOTING

In only get strange noise from the speaker when I try to write to a WIN device.

The noise usually means that the disc cannot be written to. In later Windows versions disc locations that were perfectly acceptable to contain .WIN files, like "C:\" or "C:\Program Files\" are not writable anymore for ordinary applications for security reasons. Please chose a writable directory, like your own "My documents" folder or the default "%AP-PDATA%\QPC" path, to contain you .WIN files and make sure the files are not write protected. This can happen if they were copied from a CD-ROM for example.

The display is distorted

First, make sure that it is not a QL program incompatibility issue, by starting QPC2 in the QL mode, not in high colour mode. If the problem persists, then it is likely that you need to update your computer's graphics card display driver and/or DirectX.

BASIC is always in the left hand area of the screen

This can easily be changed. For example, select a higher resolution (e.g. DISP_SIZE 640,480) and type WMON ,50,50 - you see? Procedures like WINDOW with larger parameters than usual can use the higher resolutions!

Sometimes the display looks strange, especially with games

Some programs and games assume that the screen base is located at address $20000 (131072) and that the display size is 512x256. If you are using a different resolution this is no longer the case and the output no longer goes to the real display. Either switch back to 512x256 QL mode or try the QPC_QLSCREMU command.

I have problems accessing floppy disks.

If you are using aggressive antivirus software, try disabling it. But in general floppy disc support is in a steep decline in current machines and operating systems. You will probably be better of using disc images.

How can I switch between QPC2 and other Windows applications?

Use one of the standard Windows keystroke functions: hold the ALT key and press TAB once or several times until the application you want is highlighted. Alternatively, hold ALT and press ESC to cycle through the various Windows applications.

The "Save" button on the QPC2 configuration dialog is disabled.

The button is automatically disabled when QPC detects that it can't write to the QPC.INI file. Please make sure the file is not read-only and it's not used by any other application.

The middle mouse button doesn't do anything, though pressing both buttons does the mouse hotkey.

QPC can only recognize the middle mouse button if the mouse driver is set to use it. Go to the Windows mouse setup in Control Panel and set the middle button to "middle button".

Sysmon just makes noises after startup.

Update Sysmon to version 2.06 or higher.

QPAC2 "Jobs" just shows "corrupt memory"

Update QPAC2 to version 1.40 or higher.

PAR_SETFILTER returns “Not found”/PAR_GETFILTER always returns 0

No printer filter was found, install QPCPrint.exe into the QPC directory.

MANUAL REVISION HISTORY

Revision 5.00

- Updated for QPC2 5.00

Revision 4.02

- Updated for QPC2 4.04 (rename support for DOS device)
- Added quick-start guide

Revision 4.01

- Added back FLP_DENSITY, DOS_USE, DOS_DRIVE, DOS_DRIVE$

Revision 4.00

- Complete overhaul for QPC2 v4

Revision 2.10

- CPU chapter reworked for new 68020 core.

Revision 2.09

- Explained PAR_GETFILTER and PAR_SETFILTER.
- Explained QPC_WINDOWSIZE
- Small chapter about TCP/IP added

Revision 2.08

- New SER functions explained.
- Explained PAR_DEFAULTPRINTER$.
- New command QPC_WINDOWTITLE.
- New configuration dialogs explained.
- Small chapter about SSSS added.

Revision 2.07

- Revised language and grammar (thanks to Per Witte)
- Removed out-of-date references (e.g. availability of certain software updates).

Revision 2.06

- New PAR functions explained.
- New command line explained, including new QPC_CMDLINE$ function.
- Separated syntax definition from titles. Makes editing this manual much easier.

Revision 2.05

- Deleted obsolete paragraph messing with “much memory slows down disk access”.
- Mentioned “left+right mouse button together” substitute for middle mouse button in MOUSE_STUFF.

- Added paragraph that WIN devices should only be formatted from within the console of job 0.
- Some new troubleshooting items

Revision 2.04

- Added new configuration options to beeper chapter
- Added chapter about fast memory
- Added DOS_DRIVE and DOS_DRIVE$ commands and updated the DOS device text
- Added manual revision history :-)