

Introduction	3
<i>What is AspeQt?</i>	3
<i>How to obtain AspeQt?</i>	3
<i>What does exactly AspeQt do?</i>	3
<i>What exactly is a SIO2PC?</i>	3
<i>What is a Virtual COM Port?</i>	3
<i>What is hardware handshaking?</i>	4
<i>Can AspeQt work with homemade or over the shelf Serial-to-USB converters which do not provide a hardware handshake signal?</i>	4
<i>Can AspeQt work with wireless SIO2PC type of interface devices?</i>	4
<i>Is it easy to use AspeQt?</i>	5
<i>What are the advantages of running peripheral emulator software on a PC over using real Atari peripherals?</i>	5
<i>How good is AspeQt as a peripheral emulator?</i>	5
Installing AspeQt	5
<i>Supported Platforms</i>	5
<i>Structure of the installation folder</i>	6
<i>System Requirements</i>	6
Configuring AspeQt	7
<i>Storing global application settings</i>	7
<i>Storing session settings</i>	7
<i>Minimum configuration required to start using AspeQt</i>	7
<i>Configuring the Standard Serial Port Backend</i>	7
Serial Port Name	7
Handshake method	8
High speed mode baud rate	8
Use non-standard speeds	8
<i>Configuring AtariSIO Backend</i>	9
<i>Configuring Emulation Mode</i>	9
Use high Speed Executable Loader	9
Use custom baud rate for cassette emulation	9

Filter out special characters from file names	9
Associate file types with AspeQt	10
<i>Configuring the UI (User Interface)</i>	<i>10</i>
Language	10
Minimize to system tray	10
Save window positions and sizes	11
Save D8-DN drive visibility status	11
Change font size for drive slots	11
USAGE	12
<i>Disk Images</i>	<i>12</i>
Supported Image Types	12
Mounting Disk Images	12
Image Explorer	13
Image Context menu	13
Disk slot action buttons	14
<i>Folder Images</i>	<i>14</i>
Atari Dos compatible folder image	14
Emulation details	14
Booting from folder images	15
MyDos compatible folder image	17
SpartaDos X compatible folder image	17
<i>Running Atari executable</i>	<i>18</i>
<i>Cassette Image Playback</i>	<i>18</i>
<i>Printer Output</i>	<i>19</i>
<i>Sessions and running multiple instances of AspeQt</i>	<i>19</i>
<i>AspeCl - AspeQt remote client module for the Atari</i>	<i>21</i>
AspeCl Usage	22
AspeCl compatibility	24

INTRODUCTION

In this section we will try to cover the basics of the Atari Peripheral Emulation on a PC and the hardware and software involved in making that happen. In later sections we will get into the nitty-gritty details of AspeQt's operation, and peripheral emulation features.

What is AspeQt? AspeQt is a cross-platform Atari 8-bit serial peripheral emulator. The name is an acronym for **A**tari **S**erial **P**eripheral **E**mulator for **Q**t, Qt being the cross-platform application development framework used in developing AspeQt.

How to obtain AspeQt? There are two ways to obtain AspeQt. You can either download it from Atari8Warez [website](#), or you could buy an SIO2PC device from [atari8warez](#) and the software will be included in the accompanying DVD. The DVD also contains drivers, driver installation guides, a copy of this user guide, a troubleshooting guide, plus thousands of Atari 8 bit software, electronic copies of old Atari magazines, e-books, hardware mods, and many other Atari related media.

What does exactly AspeQt do? AspeQt runs on a PC and emulates various Atari 8-bit peripherals like disk drives, cassette recorders/players and printers via a SIO2PC cable connected to the Atari computer and a PC.

What exactly is a SIO2PC? A SIO2PC is simply a serial interface device which converts Atari computer's proprietary serial port/serial protocol (*SIO*) signals to a RS232 compatible serial signal, and vice versa. This allows a PC with a serial port (*also called a COM port*) to connect to an Atari 8 bit computer with an SIO port, and through the use of AspeQt or similar applications, provide Atari peripheral emulation services to the Atari computer. There are a few vendors that make and sell SIO2PC devices like myself (Canada), Lotharek (Poland), and AtariMax (USA). If you need to purchase a SIO2PC device [I have a feature comparison chart](#) that may help you in your decision. Note that "AtariMax SIO2PC Universal Interface USB Edition" is not supported by AspeQt as it uses its own proprietary USB drivers that are not designed to provide Virtual COM ports. AtariMax SIO2PC RS232 Edition on the other hand is supported.

What is a Virtual COM Port? As the PC technology moved away from the old RS232 type serial ports to USB type serial connections, SIO2PC devices followed suit. Modern SIO2PCs now employ USB technology that emulates old style serial ports on PCs (*called virtual COM ports*) so that software running on them, and expecting to communicate through classic RS232 COM ports could still function. As far as AspeQt is concerned there is no difference between a "real" or "virtual" COM port, the differences in hardware is handled by the COM port drivers and are completely transparent to AspeQt and the

PC user. One advantage of virtual COM ports over the RS232 COM ports is the former can support arbitrary baud rates, thus providing more speed options in emulating Atari disk drives. The maximum possible baud rate with Atari's SIO port however is about 127,000 bps or 6 times the regular SIO speed.

What is hardware handshaking? Hardware handshaking is a method employed to coordinate the data transfer between two hardware devices like a SIO2PC and a PC. The devices communicating with each other would need to know whether the other device is ready to send or receive data. This can be accomplished by using either a software protocol or a hardware signal. Serial ports on PCs can use either a software protocol or a hardware signal. Atari computers on the other hand only use a hardware signal.

The signal line used by the Atari is called the *Command Line (SIO port pin 7)* the status of this line (*High or Low*) is used to indicate when the Atari is about to send a command to its peripherals, and all Atari peripherals "listen" (*or monitor*) this signal before sending or expecting data. This is the only handshake signal used by Atari computers. The PC on the other hand can use one of the several different handshake signal lines. RI (*Ring Indicator*), CTS (*Clear to Send*) and DSR (*Data Set Ready*) are among the ones that can be used by AspeQt to establish a link with the Atari's SIO7 pin.

Can AspeQt work with homemade or over the shelf Serial-to-USB converters which do not provide a hardware handshake signal? Yes, it can. AspeQt also has a 'data monitoring' mode of operation which monitors the arrival of data bytes to the serial port. This is possible thanks to the strict method employed by the Atari's SIO protocol and to the fact that only the Atari computer can initiate serial port activity on the data bus. This method can even be used in an environment where real Atari drives are mixed with virtual drives provided by AspeQt. It is not a 100% foolproof method but very close to it, with chances of failure in the 1 in a million range, so it can practically be ignored and considered safe. Nevertheless it helps to know that it is not completely foolproof and you should weight the benefits against the risks (*depending on your application*) before you decide to use it.

Can AspeQt work with wireless SIO2PC type of interface devices? Yes, AspeQt can also work with wireless SIO2PC devices. Currently there are some Bluetooth modules which are capable to emulate COM port functionality on the PC. These devices provide virtual COM port services just like the wired counterparts. The only difference is that they establish a wireless connection to the PC instead of using wires. One caveat with those devices however is that they do not usually provide a handshake signal, so if those devices are used as an SIO2PC, AspeQt must use the 'data monitoring' mode. Another drawback of these devices is, even though they can work at much higher speeds than

19,200bps, current SIO2PC implementations are limited to 19,200bps. They also require a modified Atari OS, or special loader for proper timing of the SIO bus.

Is it easy to use AspeQt? If you are familiar with software like the original SIO2PC, APE, Atari810, AtariSIO etc., you probably won't have any problems getting used to AspeQt. Otherwise, just like any new software you need to get familiar with some new concepts but the learning curve is quite short for basic disk drive emulation.

What are the advantages of running peripheral emulator software on a PC over using real Atari peripherals? The advantages are many; speed (*up to 6 times faster than a real Atari drive*), storage capacity (*many times over compared to real disk drives*), convenience, lower cost of hardware acquisition, reliability afforded by a lack of moving mechanical parts, no hardware noise, savings on wear and tear to real peripherals just to name a few. The only disadvantage is the lack of authenticity and the 'cool factor' of using real vintage hardware.

How good is AspeQt as a peripheral emulator? AspeQt has now reached a certain level of maturity. It does not however provide all possible emulation functions compared to some other software like APE (*for instance it does not have modem emulation yet, and printer emulation is text only*), however whatever it provides is at least at par or in some cases better than other emulators. It also offers some unique features not found on other emulators. Having said all that, the only way to find out whether AspeQt is the right choice for your needs, is by trying it yourself. So, in the upcoming sections we will explore AspeQt's features in detail.

INSTALLING ASPEQT

Supported Platforms

AspeQt currently runs on MS Windows (*XP/Vista/7/8/10*) and Ubuntu (*14.04 and up are the tested versions*), other Linux distros are not tested. The application does not come with an installer. The main executable and supporting Qt libraries (*.dll files*) are included in the Windows version of the archive. Ubuntu version on the other hand only includes the main executable but not Qt system libraries as those are normally distributed and available in all Ubuntu installations. Windows version is 32 bit and can run on both 32 and 64 bit versions of Windows. This is however not true for Ubuntu, so separate archives of 32 and 64 bit versions are provided. Also note that depending on which version of Ubuntu you are running, you will need to use the appropriate 32 or 64 bit version of AspeQt. For example you cannot run AspeQt compiled with Qt5.2 (*compatible*

with Ubuntu 14.04) on Ubuntu 15.04, you need to use the version compiled with Qt5.4 under that OS.

Structure of the installation folder

Installation is as simple as extracting the contents of the archive to a folder on your PC. You may manually create shortcuts to the executable (*aspeqt.exe*) on your desktop, start menu and quick launch bar if you want. After you extracted the files, you will notice some new folders inside the installation folder. These folders have names starting with '\$boot' and contain special files used to boot your Atari from a PC folder (See section related to Folder Images). Those special files are (depending on the DOS system used and where applicable):

- \$boot.bin Copy of the 3 boot sectors from the DOS disk
- dos.sys Atari/MyDos Disk Operating System
- dup.sys Atari/MyDos DOS Menu
- ramdisk.com Atari Dos ramdisk handler
- x32.dos SpartaDos Disk Operating System
- picodos.sys MyPicoDos Disk Operating System
- menu.com DosXL Dos Menu

Note: All these DOS system files are distributed under the assumption that they all are either public domain or abandonware (*abandoned by their authors/publishers*). If and when I receive a notice from the copyright holders I will most likely remove these files from the distribution archives, you will have to then substitute these files from the original DOS disks you personally own.

Again, these files and folders are only required to boot your Atari from a PC folder; you can always boot your Atari using an .ATR image of your preferred DOS system.

System Requirements

AspeQt has been tested and works with hardware systems ranging from a Pentium 4 PC running at 2.4 GHz, with 1 GB of RAM all the way up to an Intel i3 PC running at 3.4 GHz, with 8 GB of RAM. Obviously it will run just fine with more powerful systems, and it is quite possible that it will also run fine with less powerful systems at the other end of the spectrum. AspeQt currently needs about 35 MB of RAM when started, and the memory requirements may increase during operation but not by a big margin. So trial and error on older and less powerful systems is the only way to tell if it will run properly. You also obviously need to have an available serial port on your computer and a SIO2PC device. You don't need any real Atari disk drives or printers to use AspeQt.

CONFIGURING ASPEQT

Storing global application settings

AspeQt stores its global settings in platform specific ways. This means in Windows the settings are stored in the Windows Registry at a user level and in Ubuntu inside the .config folder under the home directory of the user.

Storing session settings

You can actually run more than one instance of AspeQt at any given time. Each instance, called a 'session', can connect to a different COM port and can serve a different Atari computer (*so long as you have more than one SIO2PC device and more than one serial port on your PC*). Each session can serve different disk/folder images and run independently from each other. This also means AspeQt can save session specific settings in session files (*files with the extension .aspeqt*) so that they can be remembered next time that particular session is invoked. Session files can be created in any folder on the PC. The session file is basically a simple text file composed of 'keywords' and their values, similar to .ini files used by some applications on the Windows platform and can be edited using Notepad/gedit.

Minimum configuration required to start using AspeQt

When AspeQt is started for the first time, it will display a dialog box asking you whether you'd like to configure it. When you answer YES the Options dialog will be presented.

On Ubuntu, you have to choose a serial I/O backend first. If you have installed the [AtariSIO package](#) and you have a RS232 port, the recommended way is to use the AtariSIO backend. If you have a USB adapter or you don't want to use AtariSIO for any other reason, choose the standard serial port backend. This is also the only available backend on Windows.

Configuring the Standard Serial Port Backend

Serial Port Name

First, you should enter a serial port name:

On Windows, it should be something like **COM1**, **COM2** etc. Check the device manager to see which one of them is assigned to your SIO2PC device (*wired or wireless*).

On Ubuntu, it should be **/dev/ttyS0**, **/dev/ttyS1** etc. for RS232 ports or **/dev/ttyUSB0**, **/dev/ttyUSB1** etc. for USB based virtual COM ports. If you are using a wireless (*Bluetooth*) SIO2PC then the device name would be **/dev/rfcomm0**, **/dev/rfcomm1** etc.

Handshake method

Next, you will have to select a handshake method. The handshaking method tells AspeQt which one of the RS232 pins is connected to Atari's command line. There are 3 supported methods: *RI, DSR, and CTS*, select the one that is supported by your SIO2PC device.

If you are using a homemade SIO2PC or an off the shelf Serial-to-USB converter modified as an SIO2PC which does not provide a hardware handshake line, choose *Wired SIO2PC (No handshake signal)* option, in this mode AspeQt will ignore Atari's command line and will monitor the data line instead.

If you are using a wireless SIO2PC, choose the *Wireless SIO2PC (Bluetooth/Wifi)* option. This mode is limited to 19,200 bps due to the design of the current wireless SIO2PC devices (*i.e. SIO2BT*) which are nothing more than generic Bluetooth modules with Serial-to-USB converters. To achieve higher speeds with these modules a new hardware must be designed with the inclusion of a processing unit to handle baud rate changes.

High speed mode baud rate

The combo box selects the transfer speed to be used for the high speed mode. Some OS /DOS for the Atari supports higher transfer speeds than the standard 19,200 bits per second. Not all DOS/OS are capable of using 3x speed and some treat 2x as a special case for XF551 drives, so you may need some experimentation to find the best setting. 1x option is provided for unreliable connections, like when using a cheap USB adapter or when running AspeQt under virtualization software and/or on a slow CPU computer.

Use non-standard speeds

If your serial port supports arbitrary bit rates you may check the check box and select a POKEY divisor to be used in high speed mode, the lower the divisor, higher the speed. Usually real COM ports and some VCP (*Virtual COM Ports*) don't support arbitrary baud rates, FTDI chip based serial-to-USB cables do support arbitrary baud rates, and are thus the recommended type of chip for use with AspeQt. Atari8Warez SIO2PC devices always use FTDI chips in their design.

The exact formula for the nominal speed is $baud\ Rate = clock / (2*(divisor + 7))$

Where clock is *~1,773,447 for PAL*, and *~1,789,772 for NTSC*, however, it's not always possible to reach nominal speeds. So, 1x is divisor 40, 2x is divisor 16 and 3x is divisor 8. Divisor 0 is ~126 kbps and that means approximately 6x.

Please note that very few OS/DOS will be able to function with such high speeds. Currently the only tested software that can reach divisor 0 with AspeQt is the *Hiassoft's OS patch*. In short, speeds beyond 3x are not guaranteed to work in every case.

Configuring AtariSIO Backend

Note: AtariSIO is available under Linux only!

After installing and running the AtariSIO module and making sure that you have the required permissions, you just need to enter a device name which should be `/dev/atarisio0` under normal circumstances and select a handshaking method as described above for the standard serial port backend. Please refer to AtariSIO documentation on how to build and install AtariSIO.

Configuring Emulation Mode

Use high Speed Executable Loader

This checkbox allows you to run the executable loader module of AspeQt at higher speeds than the default 19,200 bps. Note however not all executable are compatible with this option, so you will have to try and see if it works. Some executable may rely on the default SIO speed and the loader may freeze or stop loading at a certain point. When this happens, switching to default mode will fix the problem. If you are using a high-speed OS/DOS this option should be left otherwise it will conflict with high-speed OS. This option is more useful when a high-speed OS/DOS is not used on the Atari.

Use custom baud rate for cassette emulation

Atari tape recorders (*or program recorders as Atari used to call them*) run at a nominal speed of 600 bps, but they are actually capable of running from 425 to 875 bps.

This option allows you to experiment with the higher speed of 875 bps. Again, like in the executable loader case above, some programs may fail to load properly at this higher speed. You will need to experiment and decide when and if you want to use it.

Filter out special characters from file names

Some DOS like MyDos allows you to use certain special characters (*like @ and _*) in filenames, for other DOS like Atari Dos those characters are illegal. If this option is checked any/all special characters in filenames will be filtered-out when a folder image directory is built by AspeQt. The actual file names on the folder won't be changed but the Atari will see the filename without special characters in it when a disk directory is displayed. If you always use MyDos you can uncheck this option to allow special characters in filenames. Note that the filter is of type '*all or nothing*', meaning all special characters are either filtered or allowed, so it is your responsibility to make sure that if this option is unchecked, file names are compatible with the DOS used on the Atari.

Associate file types with AspeQt

This option allows you to associate certain file extensions with AspeQt. This means when you double-click on a file with one of these extensions, the OS will launch AspeQt and AspeQt in turn will either mount the file on the first disk drive slot (D1:) if the file is a disk image, or launch the appropriate module of AspeQt to execute the file if the file is a cassette image or an executable.

Note that the extension *.aspeqt* is always associated with AspeQt, and is used to launch an AspeQt session. Other extensions can be enabled or disabled at will.

File associations are at the user level in both MS Windows and Ubuntu, and they won't affect other users of the computer.

Configuring the UI (User Interface)

Language

This drop-down list allows you to select the language for the UI which matches your locale. Currently supported languages are:

- English
- German
- Spanish
- Polish
- Russian
- Slovak
- Turkish

Note that the language support in AspeQt is not perfect, or even anything close to perfect as every new feature added requires updated language translation, and in turn this requires involvement of somebody who is familiar with that particular language. In the past, there were contributors to the project who dedicated some of their time in updating the translation files – thank you all - these days however I do not have access to those generous individuals. This means AspeQt may drop some of the language support in the upcoming releases. If you like AspeQt, and want to make it more accessible to a greater number of Atari users, please let me know by [*contacting me*](#), and I will get back to you to make an arrangement.

Minimize to system tray

This option allows you to minimize AspeQt's main window into the system tray. A small balloon will be displayed for a few seconds to indicate where on the system tray AspeQt is minimized and a small icon will be shown. The system tray is located in the Windows taskbar (*usually at the bottom next to the clock*) and contains miniature icons for easy

access to system functions such as fax, printer, modem, volume, and more. Clicking on the icon will restore the AspeQt main window; right clicking on the icon will display a short menu with the commonly used menu functions. This way AspeQt does not occupy any screen space yet able to perform some useful actions.

Note that minimize to system tray currently does not work as expected on Ubuntu and its usage under that OS should be avoided. Should you by mistake check this option and then later minimize AspeQt, an icon will be placed to the system tray, but it will not be possible to display any menu options or to restore the main window. In this case AspeQt must be killed using the system monitor and restarted, and the option must be unchecked from the options menu.

Save window positions and sizes

Checking this option will save screen positions for the full size main window, mini main window, and the user guide window.

Save D8-DN drive visibility status

AspeQt can display either all 14 drive slots or only the first 7 depending on your choice. Hidden drive slots are still active and can be used if disk or folder images are mounted on them, they will just be invisible on the screen. This makes the main window smaller and uses less screen real estate. If you hide drives D8 through DN and want to make this permanent so that AspeQt will always starts showing only the first 7 disk slots, check this option. You can always switch between half-screen and full-screen by using the hot key *Ctrl-H* or by using menu option *Disk/Hide Drives D8-DN*. If Save D8-DN drive visibility option is also checked, AspeQt will save new visibility status for subsequent launches.

Change font size for drive slots

You can change the font size for drive slot descriptions and image file names if the default size is too small for your comfortable viewing. When you check this option a size selection box is enabled and you can select a suitable size by clicking up or down buttons. You'll also have the option of applying the same font size to the Log window of AspeQt.

USAGE

After making sure that you installed and configured AspeQt correctly, you can start using it with your Atari. If you used similar software before, it should be fairly straight forward.

Disk Images

Supported Image Types

AspeQt uses disk images instead of real disks. These are regular files that contain an image of an Atari disk. There are several formats, the most common being the .atr format, currently AspeQt only supports .atr, .xfd, and .pro formats. Future releases may provide support for .scp, .dcm, .di and gzipped (.atz, atr.gz, .xfz and .xfd.gz) formats.

AspeQt emulates 14 disk drives. You have one slot for each of them, labeled D1 through DN. D8 through DN are only supported by SpartaDos X and compatible DOS (see *SDX manual for details on supported disk drive identifiers*). You can make AspeQt to show only the first slot, the first 7 slots, or all 14 slots using options in DISK and WINDOW menu items.

- To hide/show drives D8 through DN, use menu item Disk/Hide drives D8-DN (*CTRL+H*),
- To toggle AspeQt mini mode use menu item Window/Toggle Mini Mode (*CTRL+M*), in mini mode AspeQt shows only the first drive slot,
- To toggle shade mode in mini mode use menu item Window/Toggle shade mode (*CTRL+S*), in shade mode AspeQt shows a semi-transparent window which does not completely block the view of underlying UI objects.

Note that all drives are actually available for use with your Atari in all AspeQt display modes whether they are visible or not.

Mounting Disk Images

You can mount a disk image to an empty drive slot by:


- Dragging and dropping a file to the desired slot or,
- Using the "Disk/Mount disk image" menu item (*the first available slot will be used*)
- Using the "Mount disk image" tool button available in each slot
- Right clicking on a slot and selecting the "Mount disk image" menu item
- Selecting a recently used disk image from the "Disk" menu
- By pressing Alt+1 through Alt+0 (*for disks greater than 10 use Alt+Shift+1 etc.*)




- Using AspeCL remote module on the Atari computer (*See AspeCL section of the this user manual for more details*)

You can see the result of your operation in the log display which is below the disk slots, if the operation is completed successfully, your Atari should be able to see the mounted disk image just like a real disk in a real drive.

You can use the tool buttons and context menu items to perform other operations like saving the disk image, enabling write protection for the image, reverting the image to its last saved state, ejecting (*un-mounting*) the image, creating a new image and so on. You can also swap images using drag and drop.

Image Explorer

You can click the  button on each disk slot to invoke the Image Explorer. Image Explorer window will show you the contents of the disk (*or folder*) image, and will allow you to manipulate files within the image. If your disk image contains a folder structure (*i.e. MyDos/SpartaDos disk image with folders*), you can also explore those folders within the image. The dropdown box at the bottom right of the explorer window is used to tell AspeQt what kind of disk structure is on the disk image, should AspeQt fails to determine it properly. When AspeQt cannot determine the disk structure it usually displays a blank explorer window. Selecting the right disk structure option from the drop-down box will restore missing content.

You can add or extract files from your disk image by dragging-and-dropping files from/to Explorer window to/from your Desktop or any other folder on your PC. You can also use add  or extract  buttons on the Explorer menu bar to perform the same actions. Clicking  button will delete a file or folder from the disk image.



When you extract text files from a disk image you have the option to convert the Atari End of Line character (*ATASCII 155*) to PC compatible CR/LF characters. This allows you to view/edit text documents created on the Atari on a PC. The conversion also works the other way around when you add a text file to the disk image. To activate text conversion click on the  icon. You can print a directory listing by clicking on the  button.

Image Context menu

Right clicking on an image file name will reveal a pop-up menu where you can perform various actions on the image:

- Save: Saves the changes to the disk image.
- Save As: Saves the changes to the disk image by assigning a new name to the image file.
- Revert to Original: Discards all the changes made to the disk image (*This option will not be available if Auto-Commit was turned ON for the disk image*).

- Mount Disk Image: Mounts another disk image to this drive slot.
- Mount Folder Image: Mounts a folder image to this drive slot.
- Unmount: Unmounts the disk/folder image.
- Write Protect: Disallows changes to the disk image by protecting it.
- Explore: Opens the Image Explorer window.

Disk slot action buttons

There are action buttons to the left and to the right of all disk slots:



Mounts a disk image to drive slot.



Mounts a folder image to drive slot.



Saves changes made to disk image.



Turns auto-commit ON/OFF. Turning auto-commit ON will save every change to the disk image in 'real time'. Images under auto-commit cannot be reverted to the original state using the Revert context menu item. You must have a separate backup of the image file to restore from if you want to revert to original state. Note that Auto-Commit must be turned ON for the disk image in order to remotely unmount or swap a disk image (*See Remote Control – AspeCL - section for details*)



Ejects / unmounts disk / folder image.

Folder Images

Folder images provide a similar functionality to disk images, instead of using a disk image file however (*.atr, .xfd etc.*) files are stored on a PC folder. Files from folder images can be read and written to just like disk images. AspeQt currently implements two different kind of folder images as explained in the following sections.

Atari Dos compatible folder image

Emulation details

This is basically a simulated Atari Dos 2.5 disk. The mounted folder will be seen by the Atari as a standard DOS 2.5 disk and is sequential access only. Although Folder Images are simulated and behave like Atari Dos 2.5, there are some differences. The most important difference is in the way 'read/write sector' commands are handled. AspeQt uses a pre-determined block of sector numbers when it reads and serves data to the Atari, and when this block of sectors are all used up, it then re-uses them in a circular fashion (*i.e. 433-434-435.....1023, 433-434-435.....1023*). Only the first sector number of each file is unique so that a file-number can be calculated, and those sectors are chosen from another pre-determined block of 64 numbers (*369-432*). This is possible because Atari Dos and compatibles don't care about the sector numbers when

they read a file from a disk drive, all DOS cares about is which sector to read next, and that sector number is chosen by AspeQt and handed to the Atari by the previous read from the same file. Atari uses that sector number to request the next sector from AspeQt. The circular sector number logic allows files much larger than standard Atari files to be served because sector numbers can be re-used, each file can be as large as 8MB in size and file sizes are shown as number of bytes as opposed to number of sectors. Maximum file size when writing a new file however is still limited to standard Atari file sizes as DOS won't use sector numbers greater than 1023, and does not recycle sector numbers like AspeQt does. Maximum number of files in the directory is still 64 and subdirectories are not supported due to DOS limitations.

AspeQt does not keep a VTOC for Atari Dos compatible folder images, and does not keep track of which sector numbers are used for files (*other than the first sector of each file*). This means files can only be read / written sequentially and that random access is not possible (*Basic NOTE / POINT commands will not work as expected*). Atari Dos compatible folder images are therefore not suitable as a storage option in applications where random access of files is required.

Changes to files within a folder image take effect immediately, for example when you copy a new file to a folder image, the file is written to the PC disk at 'real time', similarly when you delete a file from a folder image the deletion takes effect immediately. This means there's no '*Revert to Original*' option unlike disk images.

You can of course manipulate folder images through your PC's file manager. From the PC's point of view folder images are just regular PC folders.

Most DOS menu functions can be performed on folder images. The only exception is the FORMAT command (*DOS menu options 'I' and 'P'*). Atari Dos compatible folder images can also be used with other Atari Dos compatible DOS such as MyDos and others, albeit without their more advanced features such as subdirectories, double density, 256 byte sectors etc.

Folder images are also bootable, but since the folder image is a simulation of a standard DOS 2.5 disk, you can only boot into a DOS that is compatible with Atari Dos disk structure. There are two exceptions to this, first one is SpartaDos, AspeQt will allow you to boot into a 3.2G version of SpartaDos but there are some limitations. AspeQt will also allow you to boot into the standard version of MyPicoDos 4.05.

Booting from folder images

To boot your Atari from a Folder Image, first mount an empty PC folder to disk slot 1. Once mounted, right-click on the Folder name and select Folder Boot Options from the menu, select the DOS you would like the boot and click Apply. AspeQt will copy the necessary boot and DOS files to the folder.

Note: Another way to make an Atari compatible folder image bootable is to boot your Atari from your favorite Atari compatible DOS (*either via a real drive, or a disk image*), then mount an empty PC folder, and then choose 'Write DOS Files' option from the DOS menu.

Your folder is now bootable, so make sure AspeQt is ready to receive commands from your Atari, and finally turn the Atari ON and the selected DOS will be booted. You normally have to perform this procedure once for each Folder Image. The Folder will remain bootable with the same DOS between AspeQt sessions (*except SpartaDos – see below for details*)

The following DOS can be booted from a Folder Image:

Atari Dos 2.5 You can boot the DOS and load drivers and binary files (autorun.sys) during the boot process. AspeQt supplies the dos files (dos.sys, dup.sys) and the ramdisk driver (ramdisk.com). You can add your own autorun.sys file by copying the file into the folder \$bootata which is located in AspeQt application directory. You can also use DOS 2.0 instead of 2.5 by simply replacing the dos.sys and dup.sys in \$bootata folder with the ones from a DOS 2.0 disk.

MyDOS 4.5 This DOS is disk structure compatible with Atari Dos, so everything that's said for Atari Dos 2.5 above is also valid for MyDOS. Boot files folder for MyDOS is \$bootmyd. Just remember that you can't use My Dos specific advanced features.

DosXL 2.x OSS DosXL. Boot files folder for DosXL is \$bootdxi.

SmartDos 6.1D Rana Systems SmartDos. Boot files folder for SmartDos is \$bootsma.

MyPicoDos 4.05 This game DOS is provided to support Folder Images which hold Atari games. You can quickly boot and start your games conveniently from a PC Folder. Only standard version of MyPicoDos is provided and supported. When selecting MyPicoDos as your boot DOS there is an extra option on the menu to disable high speed SIO code built into this DOS. If checked, this option will instruct MyPicoDos to run in normal speed. This may be necessary under certain configurations. One example is if you are using an Ultimate 1MB board or a PBI device like IDE Plus 2 on your Atari with high-speed OS enabled. MyPicoDos used in high-speed mode will conflict with the high-speed OS on the Atari, so checking this option and booting and running MyPicoDos in normal speed will solve this problem.

AspeQt will also automatically generate a piconame.txt file during the Folder Image boot process. So if you have game files with long file names they will be displayed with their full names when MyPicoDos menu appears. The piconame.txt file will only be created / updated when you boot, so if you add more game files to your PC folder during your MyPicoDos session they will not show with long names until you reboot.

Boot files folder for MyPicoDos is \$bootpic. It is not recommended to modify this folder, unlike Atari Dos and MyDos there are no customizations you can make for MyPicoDos.

SpartaDos 3.2G SpartaDos is not compatible with Atari Dos. Its disk / file structure and boot scheme is completely different. So this DOS is only partly supported. You will be able to boot SpartaDos version 3.2G from a Folder Image with the following limitations:

To boot SpartaDos from an Atari Dos structured Folder Image is an impossible task. So AspeQt uses some (*not so elegant*) tricks to coerce SpartaDos into booting from an Atari Dos compatible Folder Image by giving it the impression that it's booting from a SpartaDos formatted disk. Once the boot is completed AspeQt forces SpartaDos to re-detect the Folder Image as an Atari Dos formatted disk so that it can display and manipulate the files within the folder. The mechanics of this scheme necessarily limits the boot process in the following ways:

- You cannot load drivers or run an autorun.bat file during the boot process
- Once booted from, a Folder Image won't be bootable again until you right-click on the Folder Image name, select Folder Boot Options, select SpartaDos 3.2G and click Apply. This will reset the Folder Image boot files and will make the folder bootable once again.

The restrictions and limitations may be lifted in the upcoming versions of AspeQt.

MyDos compatible folder image

MyDos compatible Folder Images are different from Atari compatible ones as the former is more of a simulation whereas the latter is more like an emulation. MyDos compatible folder images are implemented to act exactly like a disk drive under MyDos control. Formatting the disk is supported, VTOC and directory/subdirectory entries are maintained the same way they are maintained in MyDos and random access of files are thus possible. This feature is currently under development, so more information will be added as the implementation progresses.

SpartaDos X compatible folder image

SpartaDos X has a completely different file system than Atari Dos and compatibles. It is more advanced and also more complex to emulate. AspeQt will eventually add SDX compatible folder images to its bag of features. The development will start after MyDos folder images are fully implemented, tested and debugged.

Running Atari executable

Most Atari programs floating around on the internet come as Atari DOS executable. These files may have .exe, .com, .xex or any other extension. Instead of dealing with disk image software and DOS, you may directly run these files in your Atari using AspeQt.

You can either drag-and-drop a file with .exe, .com or .xex extension into any slot or you can use the menu item ***File/Boot Atari executable*** to access this feature. A dialog with the necessary instructions will pop up and the file will be loaded and run. You can leave the dialog open to boot from the same executable more than once. A reload button is provided which will reload the executable into memory. This button is intended for Atari software developers who may be developing on the PC and testing their software after making changes to it. Reload button will ensure the most recent executable is loaded from the PC, and as such is mainly a convenience feature for such developers. It may also be used by gamers who may want to reload the software after the 'game over'.

The executable booter has an optional high speed code which will allow you to load programs a lot faster. You can enable/disable it with the ***Tools/Options/Emulation/Use high speed executable loader*** check box. The high speed code is not able to cope with higher speeds than divisor 3 so check your configuration before attempting to load a file in this way.

Please note that this feature is not compatible with every executable and, in practice, it's not even possible to implement such a loader. High speed loader has even more issues. So there will always be some programs that you won't be able to run with the executable booter but hopefully the number of the compatibility problems will decrease with future versions of AspeQt.

Cassette Image Playback

AspeQt can playback cassette images in .cas format. These are PC files that contain data extracted from an Atari cassette. You can either drag-and-drop a file with .cas extension into any slot or you can use the menu item ***File/Playback cassette image*** to access this feature. A dialog with the necessary instructions will pop up and the file will be played back.










The cassette emulator can be configured to ignore the baud rate that is embedded in the image file in favor of a custom one. This may speed up the load times but can cause compatibility problems. You can enable/disable it with the ***Tools/Options/Emulation/Use custom baud rate for cassette emulation*** check box. When enabled, you can use the spin box below to set the custom baud rate. The available values range from 425 through 875 bps. These values are the lowest and highest speeds that the Atari OS can process. The normal speed is 600 bps.

The cassette emulation does not support rewinding or seeking in the images. This may change in the future versions.

Printer Output

AspeQt emulates a generic text-only Atari printer. It only emulates the first printer device, that is "P1." You can view, save or print the emulated printer output using the ***File/View printer text output***. Support for ASCII and ATASCII is provided. The text printer output window is divided into two panels. Left panel shows the text in ATASCII, and the right panel is for ASCII display.

Action buttons on the menu bar of the window have the following functions:

-  Saves output to a text file. Atari EOL characters (155) are converted to CR/LF.
-  Prints output to a PC Printer.
-  Cycles through 3 different Atari fonts (*Atari Classic Chunky / Smooth / Extra Smooth*)
-  Cycles through 3 different font sizes (*6, 9, and 12 points*). Size changes on both panels.
-  Toggles ATASCII panel ON/OFF. This also expands and contracts ASCII panel accordingly.
-  Toggles ASCII panel ON/OFF. This also expands and contracts ATASCII panel accordingly
-  Strips line numbers from a program listing.
-  Toggles word wrap ON/OFF.
-  Clears panel contents.

Sessions and running multiple instances of AspeQt

AspeQt allows you to save and load your disk sessions, that is, the order and names of the mounted images and their settings, the communication port used, various speed settings etc. You can access this feature from the ***File/Open session*** and ***File/Save session*** menu items. This allows you to launch multiple instances of AspeQt using different session files and have different configurations for each session. This makes possible serving more than one Atari computer from one PC so long as you have more than one COM port and SIO2PC cable available. Simply create different sessions with different settings and save them to a permanent session file (*a file with .aspeqt file extension*).

If you plan on serving more than one Atari computer simultaneously, make sure that the PC is fast enough to handle simultaneous SIO requests as Atari SIO is very time critical and can fail if the requests are not serviced in a timely fashion. Experiment with the multi-session capability and verify that it can be run reliably before you put it on serious use.

The following parameters can be configured for individual sessions (*stored in each session file*)

- Backend
- Atari SIO driver name
- Atari SIO handshaking method
- Serial port name
- Serial port handshaking method
- Serial port speed
- Serial port Pokey divisor
- Use of High speed exe loader
- Custom cassette baud rate
- Main window screen geometry
- Printer window screen geometry
- D8-DN Disk visibility
- Filter special characters
- Drive slot/log window font size
- Image Explorer on top flag
- Enable shade by default flag
- Preferred Language (if you need to run different language sessions)
- Mounted disk image/folder image file list

The following configuration parameters are global and apply to all sessions (*stored in Windows registry*)

- First Time flag (indicates it's the first time ever AspeQt was launched on that computer)
- Last Disk image directory
- Last Folder image directory

- Last Cassette image directory
- Last Executable file directory
- Last Extract directory
- Last Saved Printer file directory
- Last Session file directory
- Minimize to tray option
- Recently mounted disk image file list

The following configuration parameters apply when AspeQt is launched without a session file (*stored in Windows registry*)

- Mounted disk image file list

To launch a session, create a shortcut (*Windows*) or a link (*Unix/Linux*) in a folder with the session file name as a command line argument:

An example of a shortcut for Windows would be:

"C:\Program Files\AspeQt\aspeqt.exe" C:\Program Files\AspeQt\session.aspeqt

AspeCl - AspeQt remote client module for the Atari

AspeQt also implements a client server device (*\$46*) which communicates with the client module AspeCl that runs on the Atari computer. AspeCl is a general purpose client module and incorporates the following functionalities:

- Downloading Date/Time from the PC and setting it on the Atari computer (*SpartaDos v2.5 and up only*)
- Turning Date/Time line ON or OFF on the Atari computer (*SpartaDos v2.5 and up only*)
- Remotely Swapping AspeQt mounted disk images.
- Remotely un-mounting AspeQt mounted disk images. A single image or ALL images at once.
- Remotely mounting an existing disk image. Image must reside in the *Last Folder Image Directory* of AspeQt
- Remotely creating and mounting a new disk image in AspeQt in Single, Enhanced, DD, DSDD and Hard disk formats. The new disk image will reside in the *Last Folder Image Directory*

- Remotely toggling auto-commit feature of a mounted disk image. Auto-commit, when activated, saves the changes to the disk image in real time (*as opposed to manually saving the changes by clicking on the Save button*). Note that AspeCI can only swap/un-mount disk images if auto-commit is active for that disk image.

AspeCI keeps you updated with a plenty of on-screen status/info/error messages. When and If this verbosity is a concern (*i.e. the screen must be left undisturbed*), you can run it with I/O redirection, sending the screen output to the NULL device (*or to another actual device like a printer or a disk file*), effectively eliminating screen interaction but still being able to keep a log of AspeCI activities.

For example: `ASPECL DMGAMES01.ATR >>NULL`

Will mount the image file GAMES01.ATR to the first available disk slot without any info/error messages displayed on the screen. The disk slot number used for mounting the image file will not be displayed back either.

Similarly: `ASPECL DMGAMES01.ATR >>PRN:`

Will mount the image file GAMES01.ATR to the first available disk slot and will log all info/error messages to the printer.

AspeCI Usage

Command line switches, at least one must be specified:

AspeCI **TS** | **TO** | **TF** | **DA** | **DS** | **DU** | **DM** | **DN**

TS Set Date/Time on Atari

TO Set Date/Time on Atari and turn TD Line ON

TF Set Date/Time on Atari and turn TD Line OFF

DA[d/*] Toggle Auto-Commit ON or OFF on a given disk (*where d is 1 to 9 and J to N, following SDX naming conventions*) or on all (*) disks. Toggling auto-commit switch to ON will commit all pending changes to the affected disk image and all future changes will be committed automatically. If the toggle is set ON with a newly created and mounted disk image (*mounted locally by AspeQt*), operator intervention will be required from the AspeQt side to assign a permanent file name to the image file before it's saved to PC hard disk.

DS[dd] Swap disks, where dd represents the drive numbers to be swapped For example DS1J will swap disks 1 and 10.

DU[d/*] Unmount disk(s). Will either un-mount a single disk [d], or all disks [*]. Unmount will not un-mount disks that have changes until the changes are committed. Disks set to "auto-commit" in AspeQt will be un-mounted immediately as the changes to these images are committed automatically as they occur. So if there are pending changes for a disk when you want to un-mount it

remotely, first issue a DA remote command to the same disk to toggle auto-commit. Once auto-commit is toggled ON any pending changes will be committed to disk. You can then issue the DU command to un-mount that disk remotely.

DM[fname.ext] Mount a disk image command will mount a supported disk image file into the first available disk slot. Command will return the used disk slot number to the Atari. It is important to note that AspeQt needs to know where the disk image file resides on the PC. AspeQt will look for the disk image file in the *Last Disk Image Directory*. If no disk images were ever mounted with AspeQt, or if the file is not found, AspeQt will display an error message indicating that it does not know where to find the remotely requested disk image file.

Future releases of AspeCl will add directory name passing to the DN command and/or implement a new working directory command (DW)

DN[fname.ext].[1/2/3/4/5/6] => Create and mount a new disk image. Will create a new disk image with the given name, and mount it. The disk image will be created in the *Last Disk Image Directory* used by AspeQt and be mounted on the first available disk slot. Command will return the used slot number to the Atari. It is important to note that AspeQt needs to know where the disk image file will reside on the PC. AspeQt will save the disk image file in the *Last Disk Image Directory*. If no disk images were ever mounted with AspeQt, *Last Disk Image Directory* will be undefined, and AspeQt will display an error message indicating that it does not know where to save the remotely created disk image file.

Future releases of AspeCl will add directory name passing to the DN command and/or implement a new working directory command (DW)

The Disk Image can be created in the following capacities:

1. Standard Single Density (90KB)
2. Standard Enhanced density (130KB)
3. Standard Double Density (180KB)
4. Double sided, double density (360KB)
5. Double density Hard disk (256 bytes/sec, 65535 sectors, 16MB)
6. Quad density Hard disk (512 bytes/sec, 65535 sectors, 32MB)

Example: *ASPECL DNmyhd.atr.6* will create and mount a quad density hard disk .atr image with the name 'myhd'

And here's an example of invoking AspeCl with multiple command line switches:

ASPECL TF DS18 DMBASICXE.ATR will set the date/time and turn TD line OFF, will swap disks 1-8, and will mount '*basicxe.atr*' disk image to the first available disk slot, returning back the slot number to Atari.

AspeCl compatibility

AspeCl is currently a command line based client which is fully compatible with and runs only under SpartaDos (v2.5 and up) and SpartaDos X versions. A separate menu driven client for menu based DOS (like AtariDOS and MyDos versions) will be available in the future. Some of the functionality in AspeCl also depends on the underlying DOS capabilities. For example setting the Date/Time from the PC is a DOS dependent feature as it requires specific Date/Time drivers from the DOS. Other functionality like mounting/un-mounting/swapping disk or folder images are DOS independent and therefore are available under different DOS.